Univerzita Palackého
Filozofická fakulta
Katedra obecné lingvistiky

Aplikace kvantitativní lingvistiky na analýzu sekvencí

Disertační práce
Mgr. Vladimír Matlach

Školitel:
Mgr. Dan Faltýnek, Ph. D.

Prohlašuji, že jsem disertační práci zpracoval samostatně a uvedl veškeré prameny a použitou literaturu.

……

Olomouc, 25. listopadu 2018

Poděkování

Děkuji především svému školiteli Danu Faltýnkovi a Katedře obecné lingvistiky za možnost věnovat se skutečně zajímavým tématům, za množství inspirace a za četné diskuze. Stejně tak bych rád poděkoval Jiřímu Miličkovi, Diegovi Krivochenovi, Lukášovi Zámečníkovi, Josefu Šlerkovi, Marianu Novotnému, Alexandru Bolshoyovi a Edwardu Trifonovovi za veškeré rady a pragmatický pohled.

Obsah

Úvod

Smyslem této práce je představit možnosti využití metod, nástrojů a empirických zákonů kvantitativní lingvistiky v kontextu mezioborových aplikací, a to s cílem přinést nové poznatky i způsoby analýz textů a sekvencí. V této práci se pokusíme nalézt a ověřit způsoby, jak využít klasických nástrojů kvantitativní lingvistiky tak, aby byly přínosné pro ostatní obory pracující s rozmanitými daty ve formě lineárního zápisu symbolů, tj. se sekvencemi. Cílem této práce je tedy poskytnout nové možnosti analýzy sekvencí vycházející z analýzy přirozeného jazyka, a tím poskytnout potenciální oporu při interpretaci výsledků. Právě analýza sekvencí je disciplínou se širokým polem aplikací a je využívána řadou rozmanitých oborů. Sekvencemi zde proto chápeme texty v obecném smyslu slova, tvořené konečnou abecedou a lineárním zápisem, u nichž nemusíme předpokládat znalost jejich hierarchické stavby. Takové předpoklady by měly zaručit univerzálnost poznatků předložených v této práci a jejich užitečnost pro více oborů.

Lingvistika, věda tradičně se zaměřující na analýzu textu, je sama v některých případech odkázána na metody analýzy sekvencí, kdy například u zkoumaných textů neznámých či domnělých jazyků neexistuje *apriorní* způsob segmentace slov a vyšších jednotek. Metody, které často korespondují s těmi lingvistickými, extenzivně využívá například bioinformatika a molekulární biologie při analýze sekvencí DNA. Genetické texty, jak tyto sekvence můžeme bez větší nadsázky vnímat, sdílí s přirozeným jazykem více atributů než jen lineární zápis *písmen* A, C, T a G. Z tohoto a dalších důvodů můžeme nalézt množství metod využívaných v molekulární biologii, které pochází právě z jazykovědy a z disciplíny matematického modelování přirozeného jazyka, od využití markovovských procesů, frekvenčních analýz, editačních vzdáleností, až po nezávislý vývoj shodných metod jako Latentní Dirichletovy alokace představené v Blei, Ng a Jordan 2003 pro modelování témat textů a o tři roky dříve představené v Pritchard a Donnelly 2000 pro aplikaci v genetice. V případě přirozeného jazyka a jeho psaného textu i v případě sekvencí DNA se jedná o sekvence symbolů sloužící rozmanitým činnostem.

Biologie samozřejmě není jediným oborem, který čerpá z lingvistických metod při analýze sekvencí. Tyto metody mají své specifické postavení i v sociologii, ve které se za pomoci analýz markovovských procesů, *n*-gramů, editačních vzdáleností (stejně jako u analýzy genů) a dalších metod analyzují mezilidské i další interakce. Ty jsou převáděny na lineární symbolický zápis, na základě kterého je umožněno hledání vzorů, podobností a shluků, tj. kvantitativních výsledků, od kterých se odvíjí jejich následná interpretace a inference. Zde jsou prakticky využity nejen poznatky kvantitativní lingvistiky nebo zpracování přirozeného jazyka, ale i poznatky z teorie komunikace. Více o těchto metodách a jejich aplikacích nalezneme v Cornwell 2015.

Překvapivě může působit důležitost analýzy sekvencí v oblasti počítačové bezpečnosti, která je v dnešní době závislá především na jednom konkrétním typu sekvencí, a to těch náhodných. Dokonale náhodné sekvence jsou používány při generování kryptografických klíčů a tzv. klíčů sezení, které slouží k ochraně soukromí a k identifikaci uživatelů přihlášených do *online* služeb, od emailu až po internetové bankovnictví. Náhodnost těchto sekvencí má za cíl bránit uživatele proti uhádnutí těchto klíčů, a tedy proti ztrátě soukromí a převzetím identity útočníkem. U moderního šifrování slouží náhodné sekvence a klíče k začlenění variability do zašifrovaných zpráv a jejich případná nenáhodnost může vést k odvození ukrytého obsahu a ztrátě soukromí. Analýza sekvencí je v tomto kontextu zaměřena právě na odhalení předvídatelnosti generovaných náhodných sekvencí a na ověření relevance jejich zdroje. V této práci se k tématu aplikace analýzy sekvencí na genetiku i počítačovou bezpečnost vrátíme a budeme se jimi zabývat detailněji, neboť se jedná o krajně důležitá témata.

První kapitola této práce se bude odvíjet od toho nejobecnějšího problému, a to od způsobu analýzy a způsobu charakterizace anonymních sekvencí, tedy sekvencí tvořených řetězem rozlišitelných symbolů neimplikujících jakékoliv další znalosti jejich segmentace na vyšší celky. V této kapitole navrhneme a rozpracujeme novou metodu, jak takové sekvence analyzovat a jak z nich vyčíst charakteristiky, které nám prozradí detaily o nich samotných i o zdroji, ze kterého pochází. K vytvoření takové metody využijeme klasických metod kvantitativní lingvistiky, které nám umožní především snadnou interpretaci výsledků analýz textů, a to vzhledem k jejich transparentnosti a relativně snadné enumeraci artefaktů, které jejich kombinací mohou vznikat. Cílená jednoduchost metody nám poskytne možnost jednoduše formalizovat řadu jejích vlastností a definovat statistické testy využitelné při analýze náhodných sekvencí. Kromě předeslané analýzy náhodných sekvencí pocházejících z různých zdrojů se zde setkáme s analýzou přirozených jazyků, od beletrie až ke graficky strukturovaným textům (např. poezie), dostaneme se k analýze neznámých jazyků, dat šifrovaných vojensky silnou šifrou a dalších typů dat, včetně těch pseudonáhodných vzniklých nahodilým psaním na klávesnici nebo za využití specializovaných generátorů. Cílem první kapitoly je vyvinout robustní nástroj, který nám umožní nahlížet na libovolné sekvence a napomoci nám charakterizovat jejich zdroje. Tato charakterizace by nám měla poskytnout vodítka k tomu, abychom u sekvencí identifikovali jejich blízkost k přirozenému jazyku, strukturovanému textu, textu produkovanému gramatikou vytvářející komplexní repetice nebo zda se jedná o sekvence dokonale náhodné nebo o sekvence s pečlivým plánováním využití slov poukazující na přítomnost jejich systematického generátoru. Od ryze praktické problematiky analýzy sekvencí představené první kapitolou plynule přejdeme k návrhu teoretické metody mající za cíl odhadnout způsob segmentace sekvencí, a to na základě jejich vnitřní kombinatoriky.

Ve druhé kapitole se zaměříme na teoretickou metodu využívající poznatky z kvantitativní lingvistiky, a to především jednoho z jejích nejznámějších empirických zákonů, Zipfova zákona. Tento empirický zákon využijeme k tvorbě heuristické metody umožňující odhad jednotek blízkých slovům přirozeného jazyka uvnitř libovolných sekvencí, a to na základě jejich vnitřní kombinatoriky. V konkrétní a pro předložený přístup zároveň ilustrativní aplikaci se zaměříme na jednotky genetických textů, na tzv. nukleotidové báze DNA. Zaměříme se na jejich vnímání jakožto analogie *písmen* přirozeného jazyka a následně přehodnotíme jejich roli. I přesto, že jsou poznatky získané v této kapitole ryze teoretické, přinese tato aplikace změnu náhledu na způsob segmentace DNA a tím umožní posun ve vnímání rovin tak, že umožní nově a s vyšší mírou racionality aplikovat dosud opomíjené nástroje kvantitativní lingvistiky *a priori* vyžadující jasnou segmentaci jednotek a jejich rolí v textu. Tento nový náhled nás přivede k překvapivým aplikacím kvantitativně lingvistických poznatků ve třetí kapitole, která se s praktickými i teoretickými implikacemi věnuje sekvencím proteinů.

Třetí kapitola této práce je věnována zcela nové a ryze praktické aplikaci kvantitativně lingvistických zákonitostí, především aplikaci Menzerath-Altmannova zákona na sekvence proteinů. V této kapitole vycházíme ze změny v porozumění segmentaci DNA nastíněné v předchozí kapitole a namísto studia vztahu nukleotidových bází genetického textu budeme studovat jejich vyšší celky vytvářející fyzické nástroje, tj. proteiny. Od prvotní myšlenky, zda je specifický vztah Menzerath-Altmannova zákona na sekvencích proteinů přítomný, přechází kapitola k formálnímu i empirickému důkazu jeho netriviálnosti, načrtnutí možností jeho reálných aplikací při identifikaci nevalidních anotací proteinů či nekódujících sekvencí až k jejich simulacím a statistickému vyhodnocení jejich úspěšnosti. Závěry této kapitoly představují řadu teoretických i praktických poznatků týkajících se kódujících sekvencí DNA.

V rámci této práce bylo nutné vytvořit řadu specifických nástrojů, skriptů a implementací jednotlivých metod, které jsou k dispozici v datové příloze. Jedná se především o nástroje pro zpracování dat z genetických bank, nástroj na analýzu sekvencí metodou představenou v první kapitole, generátory sekvencí, nástroje pro hromadné stahování dat a další. Každý matematický model, který je v práci formálně zaveden, byl ověřován na základě simulací vytvořených v jazycích R či Python, každou simulaci lze najít v datové příloze. V datové příloze jsou rovněž obsaženy veškeré užité sekvence a původní nezpracovaná data. Na použité nástroje, data a jejich umístění je u jednotlivých aplikací odkázáno v poznámce pod čarou.

Analýza sekvencí

V úvodu této práce jsme definovali první oblast zájmu, kterou je analýza sekvencí. O sekvencích zde budeme uvažovat jako o zcela obecných lineárních řadách symbolů, které se nemusí vázat pouze k přirozenému jazyku a mohou pocházet z libovolných jiných, nejazykových zdrojů, jakými jsou například zápisy interakcí komunikantů, časové sekvence obsahující informace o rozpadu radioaktivních částic, rádiové signály a cokoliv dalšího, co lze zapsat v lineární formě a s konečnou abecedou. U takových sekvencí dále z hlediska jejich obecnosti předpokládáme i jejich anonymitu, tj. nevyžadujeme o nich žádné *apriorní* informace o jejich původu, segmentaci vyšších rovin (jako například způsob rozlišení slov, vět, odstavců atd.) nebo směru čtení. Jediné, co u těchto sekvencí předpokládáme, je možnost jejich segmentace pomocí jednotlivých prvků abecedy. Je zřejmé, že analýza sekvencí, o kterých nevíme prakticky nic kromě abecedy, délky a možnosti je segmentovat na jednotlivé prvky abecedy, bude poměrně komplikovaná. Takto nastavené chápání oblasti našeho zájmu nás však připravuje na reálné situace plánovaných aplikací, pro které se v následujících kapitolách pokusíme odvodit metodu analýzy.

Abychom ilustrovali problematiku analýzy sekvencí, můžeme se podívat na příklady jejích tří typů, se kterými se později setkáme a o kterých bychom chtěli, a později i dokážeme říci, jaký typ informace pravděpodobně ukrývají. Například, sekvence 1 je sekvencí vzniklou pomocí vojensky silné šifry, sekvence 2 je pseudonáhodný řetězec blízký přirozenému jazyku a sekvence 3 je zřejmě přirozený jazyk využívající silnou strukturaci textu (například báseň; zde je nutné podotknout, že uvedené sekvence jsou skutečné a spolu se stovkami dalších budou sloužit k testování a demonstraci odvozené metody):

Sekvence 1:
10110010000001001010000010111001110101000000011000001011101000110010100010010001010000110011100000101101010111001011100001011010100111010111101100000010101011111110101011010100000010111000011001101001101010000111001010…

Sekvence 2:
00101100100010001001000110100010000001011000001100011010001000001000100010010001100000110100010110001100000101100010000001101000100100011000001011000110000010010001101000100000010110000011001110000001110000100000101 10…

Sekvence 3:
11001111010000110000110100101110010001000011000011110011010000101100110100010000000011000100000000110000101100000110001000000010100000011000010110011000001000100101100010100000100010010100000001100010000000110001000110010…

Cílem této práce je tedy navrhnout metodu, která k takovým obecným sekvencím dokáže za pomoci pozorované kombinatoriky či jiných vlastností určit dostatečnou charakteristiku na to, abychom dokázali s určitou pravděpodobností stanovit jejich nejbližší známý typ. Především pak nezávisle na využité abecedě, její velikosti, kódování, směru čtení nebo segmentaci.

Metody zabývající se analýzou sekvencí s cílem určit jejich základní charakteristiky, jako například náhodnost nebo podobnost s jinými sekvencemi, samozřejmě existují a s těmi nejpodobnějšími k námi navrhované metodě, ve smyslu jejich cíle, se dále setkáme. Jak ukážeme při algoritmickém porovnání a porovnání výsledků, věnují tyto již existující metody pozornost buď pouze jediné z uvažovaných charakteristik (jako například testu náhodnosti sekvence) nebo, jak uvidíme, je jejich použití problematizováno některou z prerekvizit, jako například velikostí abecedy atd. V tomto smyslu obecná, formálně ukotvená a stabilní metoda analýzy sekvencí tak, jak ji zde představíme, prozatím nebyla uvedena. Předtím než se odvození takové metody začneme věnovat, nahlédneme na nástroje kvantitativní lingvistiky umožňující reflektovat vlastnosti textu a na jejich využitelnost při analýze sekvencí.

Kvantitativní lingvistika

Kvantitativní lingvistika nabízí řadu dobře popsaných způsobů, jak kvantifikovat snad libovolné vlastnosti přirozeného textu. Při analýze sekvencí a kvantifikaci jejich charakteristik jde především o nalézání vzorů, které dokáží poukázat na komplexitu zdroje sekvence. Obsažené a opakující se vzory v sekvencích a textech můžeme chápat jako výsledek určitého plánu, který udává, jak jednotlivé symboly či slova opakovat. Stejně tak text obsahující stovky slov bez toho, aby se jediné z nich zopakovalo, nám poskytuje důležité vodítko o typu generátoru a jeho možném plánu neopakovat žádné ze slov a uvážit i jeho hypotetické vlastnosti, například obsažené paměti. Stejně tak text, který tisíckrát zopakuje jediné slovo, nám opět podává nápovědu o jeho generátoru. Kvantifikace velikosti textu a počtu různých slov v něm obsažených nám už z tohoto pohledu nabízí jednoduchý náhled na podstatu jeho zdroje.

Právě kvantifikace opakování slov je jedním z témat bohatě reflektovaných kvantitativní lingvistikou. Pro řadu jejich formalizací, které jsou označovány jako *indexy*, je na opakování slov nahlíženo jako na specifický vztah mezi počtem slov textu a počtem jejich různých realizací. Pojem realizovaných slov je zobecněn na tzv. *tokeny*, realizované ze slovníku tvořeného *typy*. Slovník v tomto případě, oproti běžnému smyslu tohoto slova, představuje pouze množinu různých slov (tokenů) realizovaných v daném konkrétním textu – nejedná se tedy o slovník celého jazyka nebo jeho reprezentativního korpusu. Označením *slovník* nebo *typy* tak v této práci myslíme množinu

různých slov (tokenů) daného textu. O velikosti textu pak mluvíme jako o *počtu tokenů* a o počtu různých slov – velikosti slovníku – jako o počtu *typů* (blíže viz Čech, Popescu a Altmann 2014 a Popescu *et al*. 2009). Vztah tokenů a typů daného textu je studován různými indexy s vlastním způsobem náhledu, charakterizace i interpretace. Na některé z nich se nyní podíváme.

Nejjednodušším způsobem kvantifikace opakování slov textu je výpočet poměru velikosti slovníku a počtu realizovaných slov neboli poměr počtu typů a tokenů označovaný jako TTR (*type-to-token ratio*). Index TTR pro realizovanou velikost slovníku V a realizovaný počet tokenů N vypočítáme jako $TTR = V/N$. Tento jednoduchý princip může být, jak jsme si již ukázali předběžně výše, relativně efektivním způsobem sledování vzorů uvnitř textu (reflektujícím například i stylistiku autorů textu, a jak demonstruje Juola *et al.* 2018) a zároveň jsou jeho výsledky snadno interpretovatelné. Specifické hodnoty, kterých může tento index nabývat, o textu prozrazují zajímavé informace reflektující ty nejzákladnější požadavky na tvorbu a fungování metody, kterou jsme explikovali výše. Index TTR může nabývat hodnot od nuly do jedné včetně. Hodnoty TTR blízké nule indikují, že je v textu realizován větší počet tokenů, než kolik obsahuje různých typů. To jiným slovy znamená, že text slova ze slovníku vyčerpal všechna a následně je opakuje. Čím je hodnota TTR blíže nule, tím se slova opakují více. V tomto ohledu je extrémním případem nekonečně dlouhý text opakující jediné slovo. V takovém případě by hodnota TTR byla limitně blízká nule a pro charakterizaci textu by to znamenalo jediné – zřejmě se jedná o triviální repetici. Druhým extrémem je hodnota TTR rovna jedné, která indikuje, že počet tokenů je v textu stejný jako počet typů (neboli velikost slovníku). V takovém případě text ani jedno ze slov nezopakoval. V extrémním případě nekonečně dlouhého textu neopakujícího žádné ze slov by to mohlo znamenat, že generátor takového textu obsahuje paměť a určitý plán neopakovat slova. Stačí však zopakování jediného slova a hodnota TTR bude vždy nižší než jedna. V případě nekonečně dlouhého textu, který neopakuje žádné ze slov až na jediné, bude jeho hodnota limitně blízká hodnotě 1. Je zřejmé, že i takto jednoduchý index nám dokáže o textu a jeho generátoru poskytnout zajímavé informace.

Komplexnější náhled na opakování slov dále přináší například tzv. *Giniho koeficient* pocházející z ekonomie, primárně sloužící ke studiu ekonomické rovnosti rozdělení bohatství (blíže viz Gastwirth 1972). V kontextu studia opakování slov lze myšlenku této metody analogizovat tak, že některá slova (typy) *vlastní* či realizují více textu než jiná slova. Kvantifikace takové nerovnosti je založena na poměru dvou ploch vznikajících pod diagonálou jednotkového čtverce, který je pomyslně tvořen v grafu osami x a y. Ose x odpovídá kumulativní počet různých slov v %, ose y odpovídá kumulativní velikost textu v %. Zanesením pozorování do takového grafu vzniká tzv. *Lo-*

renzova křivka. Velikost plochy mezi diagonálou a Lorenzovou křivkou v poměru k celkové velikosti plochy pod diagonálou; udává Giniho koeficient (blíže na přirozeném jazyce viz např. Čech *et al.* 2014, 41). V kontextu kvantitativní lingvistiky lze Giniho koeficient vypočítat dle odvození v Popescu *et al.* 2009 (57) jako index *G* následovně:

$$G = \frac{1}{V}\left(V + 1 - \frac{2}{N}\sum_{r=1}^{V} rf(r)\right) \qquad ,$$

kde *r* je rank typu (tj. číslo odpovídající pořadí typu ve slovníku seřazeném od nejvyšší frekvence po nejnižší) a $f(r)$ je frekvence typu s rankem *r.* Giniho koeficient může nabývat hodnot od nuly do jedné včetně, obdobně jako TTR, jak ale uvidíme, význam obou hodnot se značně liší. Nulový výsledek Giniho koeficientu odpovídá situaci, kdy každý typ realizuje stejnou proporci textu. Jinými slovy, nulový výsledek získáme tehdy, když se žádné ze slov neopakuje vícekrát než ostatní. Již z tohoto je patrné, že Giniho koeficient registruje zcela odlišné vlastnosti opakování slov než TTR, u kterého jde pouze o kvantitu velikosti slovníku a počtu slov textu, a nikoliv o jejich rozdělení. Hodnoty limitně blízké nebo rovno jedné jsou pak v případě kontextu kvantitativní lingvistiky a odvozeného indexu *G*, oproti původní aplikaci v ekonomii, paradoxně těžko dosažitelné. V případě ekonomie vychází Giniho koeficient roven jedné v případě, kdy jediný subjekt z celé populace *vlastní* 100 % prostředků. Ovšem v případě kvantitativní lingvistiky a indexu *G* nastává konflikt právě v definici populace, kterou je pro tento index slovník zkoumaného textu. Text realizovaný jediným slovem, abychom dostáli použité analogie, obsahuje ve své populaci jediný subjekt, který zároveň realizuje 100 % textu. Index *G* proto v tomto případě vyjde roven nule, neboť je rozdělení zcela rovnoměrné (aby vyšel roven jedné, bylo by nutné uvažovat i nerealizovaná slova s nulovou frekvencí). Hodnota rovna nule tak u indexu *G* může nastat ve dvou zcela odlišných případech, a to takových, které nejsou pro analýzu sekvencí příliš pozitivní. I přes pozoruhodný koncept Giniho koeficientu je pro nás tedy z tohoto pohledu těžko využitelný.

Třetím a historicky důležitým způsobem nahlížení na *opakování slov* je Shannonova entropie, která dala za vznik modernímu přenosu informací (viz Shannon 1949). Shannonova entropie kvantifikuje množství informace, kterou získáme pozorováním konkrétních tokenů pocházejících ze známé distribuce. Pro intuitivní pochopení můžeme na entropii nahlédnout následovně. Entropii lze vnímat jako průměrný počet chytře položených otázek s možnou odpovědí ano/ne, které musíme položit, abychom dokázali uhádnout náhodně zvolené slovo ze zkoumaného textu (blíže viz Krippendorff 1986, 13-20). Zmíněný důvtip při kladení otázek je založen na znalosti distribuce jednotlivých typů (tj. na základě jejich frekvencí), kdy se nejprve zeptáme na ten nejfrekventovanější type, dále na druhý nejfrekventovanější atd. Pokud je v textu opako-

váno jediné slovo, nepotřebujeme ani jednu otázku, abychom toto slovo uhádli – entropie je tedy v takovém případě nulová. Pokud jsou naopak jednotlivé typy distribuovány rovnoměrně a není možné využít žádnou nápovědu vycházející z předem pozorovaných pravděpodobností, nutně využijeme počet otázek, který se bude s počtem typů textu zvyšovat. Nahlédnuto z původního úhlu je entropie vyjádřením celkového množství informace obsažené v daném textu. Výpočet entropie pro slovník o velikosti V, délku textu N, absolutní frekvenci typu na ranku r, tj. $f(r)$, vypočítáme jako index H následovně:

$$H = -\sum_{r=1}^{V} \frac{f_r}{N} \log_2 \frac{f_r}{N} \quad .$$

Minimální možnou hodnotou entropie je tedy nula, která nastává v situaci, kdy pozorované tokeny nepřináší žádnou informaci a predikce následujícího tokenu je jistá. Tato situace odpovídá textu s jediným, neustále se opakujícím typem, a tedy hodnotou TTR blížící se nule s rostoucí délkou textu. Maximální hodnota entropie není shora omezená tak, jako u TTR nebo u Giniho koeficientu principem vlastního výpočtu, ale je omezena počtem typů, které se v textu objeví, a to konkrétně hodnotou $\log_2 V$, kde V je počet typů. Původní výpočet entropie proto můžeme na základě této znalosti normalizovat na interval $\langle 0; 1 \rangle \in \mathbb{R}$ jako H_1 (viz např. Kumar 1986, 57):

$$H_1 = -\frac{1}{\log_2 V} \sum_{r=1}^{V} \frac{f_r}{N} \log_2 \frac{f_r}{N} = -\sum_{r=1}^{V} \frac{f_r}{N} \log_V \frac{f_r}{N} \quad .$$

Zde si také můžeme všimnout, že ekvivalentní normalizace dosáhneme i prostou záměnou základu logaritmu za počet typů V (tento poznatek pro nás bude později důležitý). Hodnota normalizované entropie rovna jedné pak nastává v případě, kdy je nejistota (nepředvídatelnost či informační zisk) maximální, tj. v případě, kdy jsou jednotlivé typy v distribuci užity rovnoměrně. Naopak minimální, nulovou hodnotu, získáme v případě, kdy nejistota neexistuje a v textu se opakuje jediné slovo. V tomto ohledu je normalizovaná entropie blízká Giniho koeficientu, neboť oba indexy sledují svým specifickým způsobem rozdělení pravděpodobností typů namísto prosté kvantifikace opakování jejich tokenů. Normalizovaná entropie tak může být spolu s TTR vhodným kandidátem kvantifikace vlastností sekvencí.

Dalších způsobů měření opakování slov je celá řada, jmenujme například index *RR* neboli *Repeat-Rate* (Popescu *et al*. 2009, 165) definovaný jako kvadrát pravděpodobností každého typu, tj. $RR = \sum_{r=1}^{V} p_r^2$, kde V je počet typů a p_r je pravděpodobnost typu r a další.

V úvaze nad realizací metody schopné charakterizovat anonymní a obecné sekvence je však důležité si uvědomit, že i přes jistou popisnou komplexnost Giniho koeficientu, Shannonovy entropie a jednoduchost TTR, reflektují tyto indexy pouze kvantitu či rozdělení užití slovníku, ovšem neregistrují způsob, jakým se tak děje. Takové pozorování můžeme jednoduše ilustrovat na příkladu tří textů A, B a C se shodnou délkou, shodným slovníkem a shodnou frekvencí jednotlivých typů.

Text A:
a b a b a b a b a b a b a b a b a b a b a b a b a b a b a b a b a b a b a b a b a b a b a b a b a b

Text B:
a a a a a a a a a a a a a a a a a a a a a a a a a a b b b b b b b b b b b b b b b b b b b b b b b b

Text C:
a a a b a b b a b a b a b b a b a a a b b b a b b b a a b a b a a a b b a b a b a b a a b a b a b b a b

Tyto tři texty mají všechny délku $N = 52$ tokenů a stejnou velikost slovníku $V = 2$ typy. Pravděpodobnost výskytu typu *a* je ve všech případech ½, tedy 50 %. Stejně tak i pravděpodobnost výskytu typu *b* je 50 %. Na základě definic TTR, Ginho koeficientu a entropie výše, vyjdou hodnoty těchto indexů pro všechny tři texty A, B a C stejně, neboť hodnoty V, N a jednotlivé frekvence/pravděpodobnosti jsou totožné. Z hlediska kvality je však mezi všemi třemi texty nezanedbatelný rozdíl, který ale pomocí uvedených indexů nelze registrovat. Z takového pohledu se jeví aplikace představených indexů jako nepraktická, neboť nám o sekvencích obdobného charakteru neposkytne žádné informace.

Další a zásadnější problém aplikace indexů na analýzu sekvencí vyvstává téměř okamžitě. U sekvencí nepočítáme, na rozdíl od přirozených textů, s jakoukoliv *apriorní* delimitací tokenů, kromě té na jednotlivé elementy abecedy. V takovém případě ani není zřejmé, na jakých jednotkách bychom měli celou analýzu postavit bez ohledu na využitý index. Zohledníme-li navíc i výše uvedený poznatek, kdy indexy neregistrují rozdíl mezi texty A, B a C, tj. v případě, kdy mají texty stejnou abecedu, délku a distribuci slov, docházíme nutně k závěru, že indexy sami o sobě nemají dostatečnou sílu na analýzu anonymních a obecných sekvencí tak, jak pro plánovanou metodu uvažujeme. Kvantitativní lingvistika však disponuje dalšími velmi obecnými nástroji, se kterými můžeme tyto problémy relativně snadno řešit a které, společně s celou metodou, představíme v následující kapitole.

Metoda MKM

V předešlé podkapitole jsme se seznámili se základními možnostmi charakterizace množství informace v textech pomocí Shannonovy entropie, s kvantifikací opakování slov pomocí indexu TTR nebo jejich kombinaci Giniho koeficientem. Tyto metody užívané v kvantitativní lingvistice k charakterizaci stylů textů, autorství, žánrů a dalších (viz již zmíněný Čech, Popescu a Altmann 2014), lze za určitých okolností využít i k charakterizaci sekvencí. Jak ale víme, jejich bezprostřední využití má především dvě kritické překážky. První překážka je spojená přímo s podstatou samotných sekvencí, a to neexistující *apriorní* znalosti delimitace tokenů implikující neznalost, a tedy i nejistotu v tom, jaké jednotky pro analýzu indexů zvolit. Druhá překážka je spojena s podstatou představených indexů, a jejich absentující schopností registrovat komplexnějších vztahy tokenů, než které vyplývají z analýzy jejich množin. V této kapitole si však ukážeme, že tyto indexy je možné použít k analýze sekvencí, a to za využití klasických nástrojů kvantitativní lingvistiky. V této kapitole se tedy budeme věnovat odvození metody analýzy sekvencí za použití nástrojů kvantitativní lingvistiky s cíli vytyčenými v úvodu této práce, tj. se schopností charakterizovat obecné a anonymní sekvence takovým způsobem, abychom dokázali podat informace o jejich potenciálním zdroji a možném obsahu, se zvláštním zřetelem k náhodnosti či nahodilosti sekvence. Metodu nejprve odvodíme teoreticky, následně ji konkretizujeme a doplníme formálním popisem. Ten porovnáme s některými již existujícími způsoby analýzy sekvencí s obdobnými cíli a dále metodu ilustrativně aplikujeme na základní vzorek dat, u kterého postupně prozkoumáme charakteristiky jednotlivých kategorií tak, abychom ozřejmili důvody pozorovaných výsledků a zmapovali tak vlastnosti metody.

Prvním problémem, který musíme při vytváření metody schopné analyzovat obecné sekvence řešit, je problém delimitace tokenů.[1] Jak už víme, u sekvencí dopředu neznáme nic než jednotlivé elementy či symboly abecedy. Zároveň z předešlé podkapitoly víme, že uvedené indexy nejsou schopny registrovat rozdíly sekvencí v případech, kdy je mohutnost slovníku, počtu tokenů a jejich distribuce stejná. Ačkoliv se potenciální shoda v takové parametrizaci zdá jako marginální, je nutné zvážit její potenciální důsledky ve formě naprosté invariance různých zdrojů sekvencí a narozeninovému paradoxu. (Jak navíc dále uvidíme, jedním z problémů, se kterým se budeme nutně muset vypořádat, bude velikost abecedy, kterou budeme normalizovat na binární, což velikost abecedy standardizuje pro všechny sekvence.) Z těchto důvodů musíme nutně registrovat i způsob, se kterým se jednotlivé tokeny, symboly abecedy v sekvenci objevují a vnímat je jako uspořádaný řetěz, a nikoliv jako neuspořádanou množinu.

[1] Problém delimitace slov je ostatně problém i u přirozeného jazyk a, např. v barmštině a korejštině nejsou mezery použity k oddělení slov vůbec.

Způsobem, jak snad nejjednodušeji registrovat vztahy mezi elementy sekvence, je využít markovovských modelů (řetězů), které na základě podmíněných pravděpodobností registrují *k* sousedních prvků. To jinými slovy znamená, že pro každou *k*-tici symbolů sekvence evidujeme její pravděpodobnost či frekvenci. Výhodou tohoto způsobu je, že již neregistrujeme pouze jednotlivé elementy sekvence, ale jejich kombinace či kontext, u kterých následně můžeme vybranými indexy sledovat jejich opakování, a tedy kvantifikovat charakter nalezených vzorů. Druhou výhodou této metody je způsob její implementace mechanismem notoricky známým v kvantitativní lingvistice, kdy je v takovém případě řeč o tzv. *n*-gramech, které pro danou sekvenci či text realizují registraci všech *n*-tic bezprostředně následujících tokenů. Třetí výhodou této registrace je všeobecné využití *n*-gramů právě při analýze sekvencí, u kterých není známá delimitace vyšších celků nebo skutečný začátek sekvence. S analýzou sekvencí pomocí *n*-gramů se proto můžeme setkat v mnoha oborech. V bioinformatice a na sekvencích DNA jsou *n*-gramy využívány například vzhledem k nejistotě v umístění tzv. čtecího rámce, kterým sekvence skutečně začíná a požadované trojkombinace nukleotidových bází je tak nutné registrovat všechny (viz např. Bolshoy *et al*. 2010, 61; Aguilar *et al*. 2007). Dále se s *n*-gramy setkáme v kryptografii, konkrétně v postupech kryptoanalýzy (Lasry 2018, 20, Jain a Chaudhari 2015) a dále i při analýze anomálií (Hamid 2005). Analýza *n*-gramů je všestranným nástrojem registrace nejjednodušších vztahů sousedících tokenů v textu a sekvencích obecně. Otázkou, která však užití markovovských procesů či jen pouhé *n*-gramové analýzy doprovází, je správný výběr čísla *n*, tedy čísla určující počet po sobě jdoucích prvků k registraci. Například uvedená aplikace v bioinformatice využívají velikost $n = 3$, neboť v tomto konkrétním případě existuje *apriorní* znalost kombinací bází DNA (tedy *písmen* A, C, T a G) do vyšších celků, tzv. tripletů, kódujících jednu z dvaceti aminokyselin. Snížením i zvýšením tohoto čísla by došlo ke ztrátě či zkreslení informací způsobených chybným způsobem čtení. Je tedy zřejmé, že nastavení velikosti *n*-gramů opět ideálně závisí na určité *apriorní* znalosti, která nám napovídá její správné nastavení. Takové *apriorní* znalosti však pro všechny typy sekvencí nemáme a nemůžeme s nimi počítat. Jedinou jistou odpovědí, jak nevynechat správné nastavení *n*-gramů, je provést analýzu pro všechna *n* od 1 až do určité hraniční délky, pro kterou tato analýza kombinatoricky dává smysl. Tímto způsobem obejdeme celý problém chybějících delimitací a nutnosti, jakkoliv odhadovat relevantní velikost *n*-gramů. Využití *n*-gramů nyní ilustrujme na jednoduchém příkladu sekvence „Tato_věta_je_příklad“:

1-gramy = { t, a, t, o, _, v, ě, t, a, _, j, e, _, p, ř, í, k, l, a, d }
2-gramy = {ta, at, to, o_, _v, vě, ět, ta, a_, _j, je, e_, _p, př, ří, ík, kl, la, ad }
3-gramy = {tat, ato, to_, o_v, _vě, vět, ěta, ta_, a_j, _je, je_, e_p, _př, pří, řík, ikl, kla }
atd.

Pokud budeme analyzovat v tomto případě všechny velikosti n od 1 do délky ukázkové sekvence, s jistotou budeme registrovat rovinu hlásek (grafémů), neostře slabik a dále slov. Je zřejmé, že tato hrubá metoda nebude dokonale registrovat detailně jednotlivé roviny, nicméně již tímto způsobem získáme pohled na užitou kombinatoriku uvnitř sekvence a tendence vytvářet vzory. Výhodou n-gramů je tedy možnost užití na libovolné sekvence a užitím hrubé síly i schopnost registrovat (i přes jistou neostrost) obsažené roviny. Způsob analýzy sekvencí hrubou silou, společně s jedním z představených indexů využijeme, jak uvidíme dále konkrétněji, pro realizaci celé metody.

Problémem, který však nově vyvstává, je problém efektu různě velkých abeced sekvencí, které mohou kombinatoriku, a tedy i potenciál vzniku vzorů, velmi snadno ovlivnit. Sekvence využívající pouze dva symboly ve své abecedě, budou mít nesrovnatelnou kombinatoriku se sekvencemi užívající abecedu o stovce znaků. Tento problém je mnohem větší, jakmile si uvědomíme, že do něj dále vstupuje i nejistota, zda jsou jednotlivé prvky abecedy kompozity nebo se jedná již o atomické, dále nedělitelné symboly. Ilustrací může být korejský hangul, ve kterém je slovo *dům* zapsáno jediným slabičným grafémem 집 a při analýze sekvence by proto vystupoval jako jeden samostatný symbol abecedy. Tento znak je však složen ze tří separátních znaků abecedy jamo: ㅈ | ㅂ. Namísto testování kombinatoriky od těch teoreticky nejnižších komponentů, by analýza n-gramy začínala rovinou vyšší, tj. slabik nebo slov. V případě porovnání přirozených textů v korejštině a češtině budou 1-gramy digitálně identifikovaných symbolů odpovídat jiným jazykovým jednotkám. Obdobně tomu je pak i v případě čínštiny nebo japonštiny. Problém charakteru abecedy a její interpretace je však bez *apriorních* znalostí prakticky neřešitelný a samotná metoda využívající n-gramy tak musí pracovat s domnělými či přisouzenými grafémy. Zároveň také víme, že už pouhá velikost abecedy je faktorem, který dokáže ovlivnit výsledky a je nutné se s ní alespoň nějak vypořádat. Řešením takového problému může být představená normalizace uvnitř výpočtu entropie nebo normalizace samotné abecedy sekvence, ideálně na počet minimalizující její mohutnost, tj. na binární abecedu.

Abecedu sekvence S (tj. všechny symboly sekvence, včetně mezer, speciálních znaků, včetně rozlišení malých a velkých písmen atd. v případě přirozených textů), kterou zde budeme reprezentovat uspořádanou množinu Σ, můžeme normalizovat tak, že každý obsažený symbol $\alpha \in \Sigma$ převedeme do binárního zápisu pomocí funkce $\phi(\alpha, \Sigma)$ zobrazující pořadí i znaku α_i do binární podoby. Číselný výsledek tato funkce ϕ dále převede na textový řetězec, který je zleva doplněn znaky „0“ tak, aby jeho výsledná délka vždy odpovídala počtu binárních číslic největšího binárního čísla vzhledem k abecedě Σ, tj. čísla $|\Sigma|_2$. Výsledkem jsou pro zadanou sekvenci vždy unikátní řetězce jedniček a nul o stejné délce, které arbitrárně kódují jeden ze symbolů původní abecedy. Každý znak α je následně v sekvenci S substituován výsledkem $\phi(\alpha, \Sigma)$, čímž

vzniká nová sekvence S' s normalizovanou abecedou $\Sigma' = \{0,1\}$. Tento postup jinými slovy znamená, že každému znaku nalezenému v sekvenci jednoznačně přiřadíme nejkratší možný řetězec tvořený znaky 0, 1 a zleva je doplněný symboly 0 tak, aby byly všechny řetězce stejně dlouhé. Původní znaky v sekvenci následně nahradíme novými binárními a arbitrárně kódujícími reprezentacemi. Tím je abeceda sekvence normalizována na znaky 0 a 1. Po normalizaci sekvence na abecedu 0, 1 můžeme provést její analýzu pomocí *n*-gramů. Je nicméně zřejmé, že tato normalizace nedokáže řešit problémy rovin či kompozicionality symbolů abecedy, nicméně nám poskytne alespoň stejnou konfiguraci na začátku analýzy pro nízké hodnoty n. Doposud představená myšlenka fungování metody je tak prozatím založena na dvou krocích. Prvním krokem je normalizace abecedy (pokud bude dle výběru indexu třeba) a druhým krokem je registrace frekvencí (či pravděpodobností) jednotlivých typů pomocí n-gramů pro n od 1 do konkrétně zvolené meze. Nalezená slova (typy, *n*-gramy) nám umožní vypočítat kterýkoliv z indexů uvedených v úvodní podkapitole výše.

Posledním krokem před kompletací celé metody je výběr indexu, který využijeme ke kvantifikaci opakování tokenů a který nám poskytne náhled na to, zda se *n*-gramy opakují a v jaké míře. Právě tato informace by pro každou délku n měla ve finále přinést charakterizaci zadané sekvence tak, aby umožnila její porovnání. Z této kvantifikace navíc budeme chtít dokázat a formálně odvodit, jak je analyzovaná sekvence blízká triviálním sekvencím nebo sekvencím vzniklým dokonalou náhodou. Triviální repetice lze snadno odhalit tím, že budou oproti své délce obsahovat jen malý počet typů a s narůstající délkou sekvence se zachováním počtu typů bude jistota triviality repetic stoupat. Namísto zkoumání rovnoměrnosti distribuce jednotlivých slov je v tomto ohledu jednodušší sledovat poměr velikosti slovníku a sekvence. Druhý zmíněný typ, tedy dokonale náhodné sekvence, jsou však obtížnější k uchopení. Intuitivní řešení, například za pomoci testování uniformnosti distribuce však vedou k problému dostatečně dlouhých sekvencí, které by měly být v tomto ohledu ideálně nekonečně dlouhé. Na tento problém lze nahlédnout i z jednoduššího hlediska, z hlediska opakování slov. U dokonale náhodných sekvencí s konkrétní velikostí slovníku očekáváme zopakování slov v určité, předem definovatelné míře, tj. kvantitě registrovatelné i prostým indexem TTR. Z tohoto důvodu, a především pak z důvodu udržovat celou metodu co nejjednodušeji interpretovatelnou, využijeme v této práci pro kvantifikaci opakování právě tu nejjednodušší ze zmíněných metod, kterou je právě TTR. Tento výběr nám svou jednoduchostí navíc, jak dále uvidíme, pomůže při stanovování modelů a testů náhodnosti, v rychlosti samotných výpočtů a interpretaci. Celou metodu nyní shrňme.

První věcí, kterou metoda analýzy sekvencí vyžaduje, je testovaná sekvence s jedinou *apriorní* znalostní prerekvizitu, kterou je možnost identifikace jednotlivých symbolů abecedy. Jednotlivé symboly abecedy budou dále chápány jako typy, které budou normalizovány na binární řetězce pomocí metody popsané výše. V prvním kroku metody, který nazvěme jako inicializační, je proto nutné sekvenci normalizovat do binární podoby, která nám, alespoň částečně, napomůže ve srovnatelnosti výsledků s rozdílnými abecedami. Druhým krokem je výpočet počtu různých n-gramů v sekvenci S (tj. počet typů) pro každou délku n od 1 až do maximální délky $Z \in \mathbb{N}$: $1 \leq Z \leq |S|$ (kterou stanovíme relativně k délce sekvence dále). Pro každou délku n-gramů n známe počet tokenů $N_n = |S| - n + 1$ a zjištěný počet typů V_n. Pro každou tuto délku n vypočítáme hodnotu $TTR_n = V_n/N_n$. Tímto způsobem získáváme vektor hodnot $s = (TTR_1, TTR_2, \dots, TTR_Z)$ kterým můžeme sekvenci S charakterizovat a využít jej dále například při shlukovací analýze, klasifikaci atd. (viz dále). Jak dále uvidíme, výsledky TTR jsou pro jednotlivé délky n-gramů a zdroje těchto sekvencí velmi specifické a sledování vývoje hodnot TTR v závislosti na velikost n-gramů umožňuje sekvence a jejich zdroj relativně snadno charakterizovat. Taková vizualizace má zároveň, oproti např. vícerozměrovým vizualizačním technikám (kterým se budeme věnovat později) kritickou výhodu v kontrole každé jednotlivé hodnoty TTR. Ze začátku proto budeme tuto vizualizaci využívat nejen pro náhled na podobnost sekvencí, ale i pro objasnění fungování celé metody a artefaktů, které může pro jednotlivé n-gramy vytvářet. Dále z těchto grafů odvodíme i další zajímavé vlastnosti vyplývající ze samotné metody a identifikujeme i prototypické průběhy křivek, které nám nabídnou další možnosti interpretací.

Představená metoda analýzy sekvencí pomocí n-gramů, TTR a normalizací abecedy sekvence s cílem umožnit jejich shlukování a klasifikace na základě typu zdroje, byla prezentována v Matlach a Krivochen 2015 a 2016 a dále rozvedena a doplněna o některé matematické modely v Matlach, Krivochen a Milička 2018. Prozatím však byla tato metoda prezentována pouze v aplikacích bez formálního ukotvení a analýzy formálních aspektů včetně vysvětlení jejích vlastností či porovnání s konkurenčními metodami. Zde tuto metodu, kterou pracovně dle jmen autorů konceptu označíme jako metodu MKM, formálně zavedeme a popíšeme.

Algoritmus metody MKM aplikovaný na sekvenci S a maximální velikost n-gramů Z formalizujeme následovně:

Algoritmus metody MKM

Vstupy: Sekvence S reprezentovaná jako vektor symbolů $\alpha_1, \ldots, \alpha_{|S|}$, kde $|S|$ je délka sekvence.
Maximální rozsah n-gramů Z, *kde* $Z \in \mathbb{N}: 1 \leq Z \leq |S|$.

Výstupy: Vektor $\mathbf{r}$ obsahující výsledné hodnoty $TTR_1, \ldots, TTR_Z$

Pomocné funkce:
$\Phi(\Sigma)$ … konvertuje symboly abecedy Σ na binární řetězce zleva zarovnané znaky 0 tak, aby počet symbolů každého řetězce byl roven $|\Phi(\Sigma)_{|\Sigma|}|$.
$subst(S, \Sigma, \Sigma')$ … provádějící substituci původní abecedy Σ za normalizovanou verzi Σ_2 v sekvenci S a vytvářející tak novou sekvenci S'.
n-gram(S, n) … získá vektor n-gramů sekvence S o délce n.

Inicializační krok:
1) $\Sigma \leftarrow (\alpha \in S)$
2) $\Sigma' \leftarrow \Phi(\Sigma)$
3) $S \leftarrow subst(S, \Sigma, \Sigma')$

Pro každé $n \in \mathbb{N} \wedge 1 \leq n \leq Z$ proveď:
1) $t_n \leftarrow$n-gram(S, n)
2) $N_n \leftarrow |t_n|$
3) $V_n \leftarrow |\{t_n\}|$
4) $\boldsymbol{r}[n] \leftarrow TTR_n = V_n/N_n$

Výsledkem algoritmu je vektor $\boldsymbol{r}$ obsahující Z prvků odpovídajících jednotlivým hodnotám TTR pro danou délku n-gramů. Vhodnou implementací tohoto algoritmu je možné dosáhnout lineární asymptotické složitosti závislé pouze na délce sekvence $O(|S|)$.

Maximální velikost n-gramů Z byla původně v Matlach a Krivochen 2015 stanovena na ½ délky sekvence S. Dosavadní empirie (viz pak dále) naznačuje, že tato maximální délka n-gramů, která je v proporci s délkou sekvence, může být i dramaticky menší. Jak uvidíme, pro normalizované binární sekvence o 6 000 bitech lze využít $Z \approx 30$ k jejich charakterizaci a shlukování. Určení vhodné hodnoty Z pro zadanou délku sekvence S však prozatím nebylo vyřešeno.

Na metodu MKM se nyní zaměřme z hlediska principu jejího fungování. Metoda pro zadanou sekvenci kvantifikuje opakování binárních n-tic o délkách ze zadaného intervalu. Důsledkem je, že nachází bezprostředně se vyskytující vzory a jejichž zjištěné opakování vede k náhledu na komplexitu celé sekvence. Srovnáme-li koncept celé metody s těmi již existujícími, nalezneme několik principiálně blízkých metod,

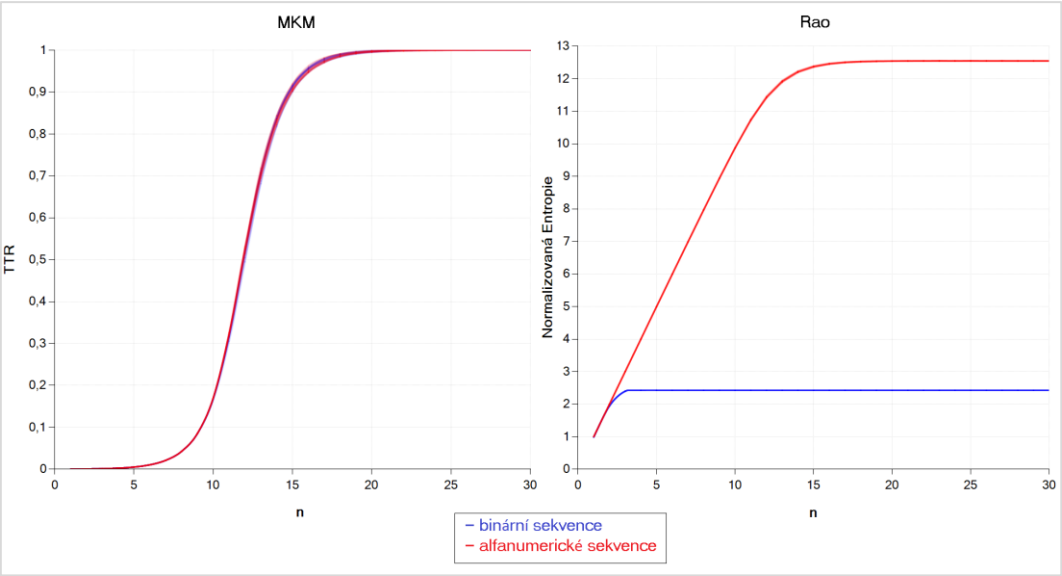
především pak z oblasti komprese dat a detekce náhodnosti sekvencí. Příkladem kompresního algoritmu zkoumajícího komplexitu sekvence na základě enumerace bezprostředních podřetězců, je např. Lempel-Ziv (Ziv a Lempel 1978), který tímto způsobem hledá nejefektivnější způsoby překódování původní sekvence na kratší, bez ztráty informací. Cílem, avšak principiálně poněkud vzdálená je pak metoda z Hamano a Yamamoto (2010) představující metodu *T-komplexity* s cílem detekovat náhodnost sekvencí, což je i jedním z plánovaných cílů metody MKM. Hamano a Yamamoto zde nicméně pro svou metodu neuvažují další možné aplikace. Principiálně nejbližší metodu k metodě MKM je metoda uvedená v Rao 2010, která rozšiřuje předchozí verzi z Rao *et al.* 2009 (diskuze a námitky k původní metodě viz Sproat 2010). Metoda Rao, jak ji zde pracovně pojmenujeme, je metodě MKM blízká principiálně i cílem. Je použita primárně pro analýzu a porovnání textů či sekvencí napsaných v harappském jazyce. U těchto textů není známé, zda se jedná o písmo a přirozený jazyk, nebo se jedná o nejazykové značení (více o problematice např. Mahadevan 1977). Cílem metody Rao a jejího apriorního odvození tedy je charakterizovat sekvence tak, aby bylo umožněno kvantitativně porovnat harappské texty s dodanými vzorky jazykových i nejazykových sekvencí. Porovnáním algoritmu obou metod nalezneme shodu v základním principu založeném na analýze *n*-gramů a kvantifikaci jejich vlastností. Zároveň však nalezneme i řadu odlišností, které obě metody kvalitativně oddělují. (V tomto ohledu je nutné říci, že metoda v Rao *et al*. 2009 a následně v Rao 2010 byla publikována s cílem analýzy harappských textů, *ad hoc*, a ne samostatně jako obecná metoda analýzy sekvencí či textů. Z tohoto důvodu byla metoda Rao při odvozování metody MKM pro její autory neznámá a shoda v některých principech je tak dána pouze jejich triviální dostupností.) Vzhledem k uvážené blízkosti tyto dvě metody dále porovnáme. Porovnání obou metod je však komplikováno tím, že Rao *et al*. 2009 a ani Rao 2010 neuvádí žádnou formalizaci své metody, její implementaci a pro některé kroky nevysvětluje užité parametry (tento problém, spolu s dalšími výtkami, je řešen v následné diskuzi ve Sproat 2014, 461). Porozumění metodě Rao je proto převážně založeno na jejím slovním popisu a interpretaci ze Sproat 2014.

Metoda Rao, shodně jako metoda MKM, využívá analýzu *n*-gramů, pro jejichž velikosti ze zadaného intervalu kvantifikuje, oproti metodě MKM, entropii (viz dále). Výsledné hodnoty entropie následně pro každou velikost *n* zobrazuje v grafu závislosti *n* a získané entropie. Z tohoto pohledu jsou obě metody téměř shodné. Metoda Rao i metoda MKM však musí nutně zohlednit rozdílné velikosti abeced různých textů. Metoda MKM tento problém řeší předzpracováním textů a normalizací jejich abecedy do binární podoby tak, že vždy pracuje s texty tvořenými symboly 0 a 1. Metoda Rao (2010, 79) tento problém řeší normalizací uvnitř výpočtu entropie, a to úpravou základu logaritmu na velikost abecedy zkoumaného textu. Je však poměrně zajímavou otázkou, zda je normalizace abecedy úpravou základu skutečně efektivní – jak jsme uvedli v úvodu této kapitoly, entropii lze normalizovat změnou základu logaritmu,

avšak počtem typů, a nikoliv velikostí abecedy. Z tohoto důvodu lze předvídat, že taková normalizace abecedy nepovede k univerzálním výsledkům, které navíc nebudou pro srovnatelnost normalizovány na interval 0-1. Entropie je v metodě Rao dále odhadována pomocí metody Nemenman-Shafee-Biale (zkráceně NSB; Rao 2010, 79), jejímž cílem je poskytnout odhad v případě nedostatečné velikosti vzorku dat. Tento krok vychází zřejmě z pragmatiky nedostupnosti větších vzorků harappských textů a je zřejmě také důvodem k využití poměrně netransparentního kroku (jak uvádí Sproat 2014, 461) aplikace vyhlazování pravděpodobností *n*-gramů pomocí metody Kneser-Ney, jejímž cílem je interpolace nepozorovaných *vztahů* mezi jednotlivými tokeny. V metodě MKM k žádné interpolaci či využití heuristik nedochází, metoda pracuje pouze s pozorovanými daty. Vlastnosti a postup metody Rao tedy zřejmě odráží její aplikaci na porovnání podvzorkovaného korpusu harappského jazyka s ostatními typy sekvencí, ať už jazykových nebo nejazykových. Taková dispozice je nicméně protichůdná k obecnosti metody a zároveň tím ztěžuje interpretaci výsledků a porozumění metodě samotné. Využití entropie, která je počítána i z interpolovaných vztahů metodou Kneser-Ney a u které je měněn základ logaritmu na základě velikosti abecedy a následný přepočet metodou NSB s sebou přináší řadu potenciálních artefaktů a nejistot. Zřetězení heuristických výpočtů má potenciál do výsledků vnášet řadu artefaktů, které emergentně vznikají jen ze samotné podstaty výpočtu a potenciálně tak mohou vést k chybným interpretacím výsledků, ale i ztížení porozumění samotné metodě. Takové problémy však Rao 2010 nediskutuje a ponechává je k pozdější kritice (Sproat 2014). Smyslem metody MKM je proto držet co nejjednodušší princip fungování, který umožňuje přímočaré zmapování artefaktů vznikajících ze samotného principu fungování metody a umožní tak jednodušší interpretaci jejích výsledků. Jak uvidíme dále, využití TTR nám, oproti entropii, v tomto ohledu usnadní formální odvození takto uměle vznikajících vlastností. Kromě uvedených nedostatků metody Rao, které jsou prozatím pouze teoretického rázu, budeme níže u první demonstrace metody MKM ilustrovat i její konkrétní, empiricky zjištěný kritický nedostatek, který povede především k otázce diskutované relevance způsobu užité normalizace abecedy. Zde je nutné podotknout, že vzhledem k chybějící formalizaci a neexplikované parametrizaci je metoda Rao implementována bez užití vyhlazování Kneser-Ney a odhadu entropie pomocí NSB. Vynechání těchto bodů by však nemělo dramatickým způsobem ovlivnit hlavní vlastnosti metody a její srovnání s metodou MKM.

První aplikací metody MKM a metody Rao ilustrujme výše uvedený teoretický rozdíl vycházející z odlišného typu normalizace abeced. Obě metody aplikujeme na dvacet náhodných sekvencí pocházejících ze služby RANDOM.ORG, využívající jako zdroj entropie elektromagnetický šum z atmosféry (Haahr 2018; náhodným sekvencím, zdrojům entropie a jejich testování se budeme detailně věnovat později). Abychom problematiku rozdílných abeced mohli ilustrovat, je prvních deset náhodných sekvencí binárních (tj. tvořených abecedou se symboly 0 a 1), zatímco druhá desítka

sekvencí je tvořena alfanumerickými znaky vycházející z anglické abecedy (tj. celkem 62 symbolů a-z, A-Z, 0-9). Sekvence jsou pro obě metody shodné a dlouhé 6 000 symbolů (v případě metody MKM jde o 6 000 bitů binární normalizované sekvence, v případě Rao o 6 000 původních symbolů). Z ilustrativních důvodu zvolíme testované velikosti *n*-gramů na hodnoty 1 až 30. Aplikací pro jednotlivé délky *n* získáváme metodou MKM hodnoty TTR a metodou Rao hodnoty entropie, které vykreslíme do vlastních grafů (včetně proložení pozorování *spline* křivkou) s osou *x* odpovídající *n* a osou *y* vypočítané hodnotě TTR a entropie. Výsledky obou metod vidíme v grafu 1, výsledky metody MKM vidíme v levé části grafu a výsledky metody Rao v pravé části grafu. V grafu metody MKM si můžeme jako první všimnout specifického tvaru křivek připomínající logaritmickou funkci. Tento průběh křivek bude u metody MKM důležitý a dále pro něj odvodíme řadu specifických intervalů a vlastností. Důležitější je však pozorování, že se oba typy náhodných sekvencí, tj. sekvencí s binární i alfanumerickou abecedou, téměř dokonale překrývají – metoda MKM mezi oběma typy kódování neregistruje žádný mimořádný rozdíl, což je výsledek, který bychom ideálně očekávali od metody charakterizující obsah sekvence nezávisle na využité abecedě. Je rovněž zajímavé, že překryv není dokonalý, což naznačuje, že metoda MKM registruje pravděpodobně systematičnost užitého ASCII kódování. Důležité je také to, že výsledky TTR jsou z principu výpočtu omezeny na interval 0-1. Podíváme-li se na výsledky metody Rao (vpravo), všimneme si několika odlišností. Prvním, snadno pozorovatelným rozdílem, je odlišný typ výsledných křivek, který je dán využitím entropie namísto TTR. Druhým a klíčovým rozdílem je separace obou typů kódování náhodných sekvencí, což je v protikladu s tím, co bychom od charakterizace náhodných sekvencí očekávali, neboť jde o sekvence se stejným zdrojem a typem lišící se pouze v abecedě. Tento výsledek poukazuje buď na rozdílné pojetí identifikace typu sekvencí obou metod nebo na neefektivitu normalizace výpočtu entropie pomocí velikosti abecedy. Tento způsob normalizace pak zapříčiňuje i třetí pozorovatelný rozdíl, kterým je odlišný a shora neomezený obor hodnot. Z této ukázky plyne, že využití metody Rao je v případě sekvencí s odlišnou abecedou problematické, neboť nezaručuje podobnost sekvencí se shodným či podobným zdrojem a typem obsahu. Metodu Rao z tohoto důvodu dále nebudeme uvažovat.



***Graf 1**: Porovnání výsledků metody MKM a metody Rao na náhodných sekvencích s rozdílnou abecedou.*

Metodu MKM dále ilustrujme na obsáhlejším datasetu čítajícím 374 textů pěti různých kategorií či typů. První kategorií je 50 náhodných textů pocházejících ze služby RANDOM.ORG. Druhou kategorií je 150 českých beletristických knih. Třetí kategorií je 150 Biblí v odlišných jazycích (kompletní přehled jazyků viz Příloha 1). Čtvrtou kategorií je 21 tzv. *monkey-typed* textů, tj. textů psaných nahodilými úhozy od 18 různých autorů a poslední kategorií jsou 3 sekvence tvořené triviálními repeticemi 2 až 4 znaků. Tyto kategorie textů byly vybrány vzhledem k jejich prototypické povaze s cílem poskytnout co nejjednodušší náhled na to, zda je možné jednotlivé typy vizuálně odlišit a identifikovat. Shrnutí základních vlastností jednotlivých kategorií nalezneme v tabulce 1.

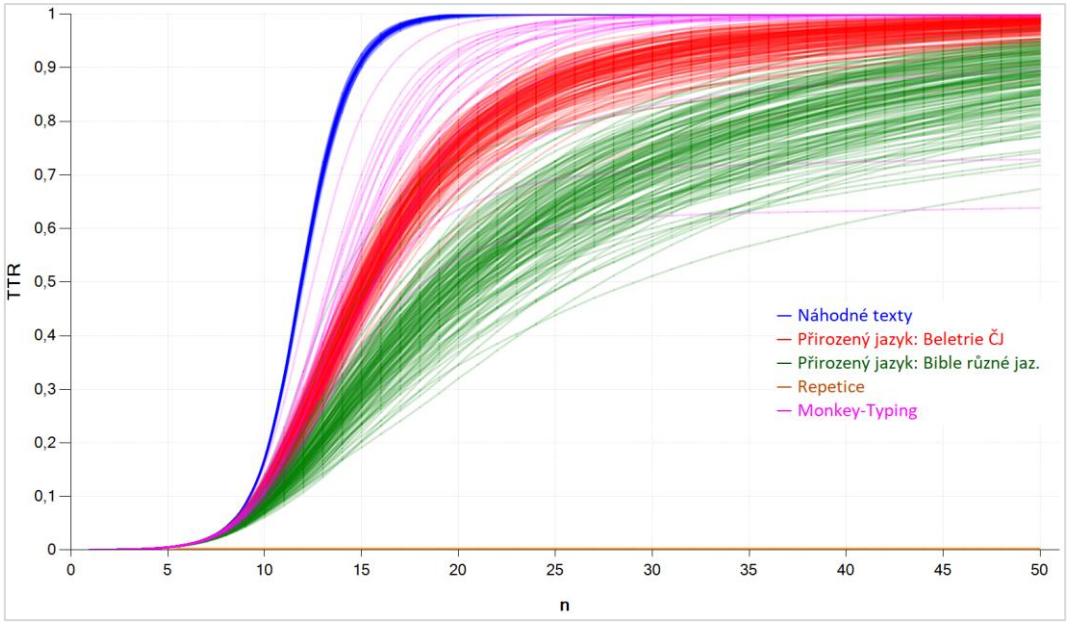
Typ	**Počet**	**Abeceda***		
		Průměrná velikost	***Min***	***Max***
Náhodné texty (uniformní, RANDOM.ORG)	50	62	62	62
Přirozený jazyk: Beletrie ČJ	150	93	61	120
Přirozený jazyk: Bible 150 jazyků	150	117,8	74	3726
Monkey-Typed	21	75,1	40	128
Triviální repetice	3	3	2	4

***Tabulka 1**: Přehled základního datasetu užitého k testování metody MKM. *Abecedou jsou myšleny veškeré symboly obsažené v původní sekvenci, tj. malá/velká písmena, čísla, interpunkce, nové řádky, mezery atd.*[2]

Z každé uvedené sekvence pro metodu MKM náhodně vybereme podřetězec dlouhý 6 000 bitů (tato délka je vybrána čistě experimentálně). Tímto krokem zajistíme jednak porovnatelnost sekvencí vzhledem k možnému vlivu délky sekvence na metodu MKM a dále nahodilý výběr umožní reálnější náhled na celou sekvenci, např. oproti

[2] Testované sekvence nalezneme v datové příloze: Sekvence/.

využití prvních k bitů (které u beletrií mohou odpovídat obsahům namísto uceleného textu atd.). Výsledky jednotlivých sekvencí společně vykreslíme do grafu, včetně jejich barevného rozlišení dle kategorie a nastavení průhlednosti na 50 % tak, aby byla alespoň částečně zachována informace o množství překryvů produkujících sytější barvu. Graf je pro přehlednost omezíme na rozpětí délek n od 1 do 50. Výsledek pak vidíme v grafu 2, ve kterém si můžeme okamžitě všimnout pěti výrazných pásem odpovídajících jednotlivým kategoriím či typům sekvencí z tabulky 1. Každou kategorii si nyní zvlášť projdeme a popíšeme.



***Graf 2**: Aplikace metody MKM na 374 sekvencí 5 různých kategorií.*

Náhodné sekvence pocházející z atmosférického šumu (tmavě modrá barva) jsou v grafu 2 umístěny maximálně vlevo s téměř dokonalým překrytím kromě jisté viditelné distorze v rozmezí 15 až 20-gramů. Průběh křivky je strmý a v porovnání s ostatními dosahuje hodnot TTR=1 nejrychleji, přibližně u 20-gramů. Hodnot TTR=1 nabývá sekvence tehdy, když je počet typů roven počtu tokenů, jinými slovy, počet různých slov je roven celkovému počtu užitých slov – žádné ze slov se tedy v takové sekvenci neopakuje. Hodnoty TTR blížící se 0 naopak znamenají, že počet typů je oproti počtu tokenů násobně nižší, což jinými slovy znamená maximálně se opakující slovník. U uvedených náhodných sekvencí to znamená, že až do velikosti přibližně $n = 20$, stále dokážeme najít zopakování některého z n-gramů. Od velikosti n-gramů 20 a výše již takto velké tokeny zopakovány nenalezneme. Oba pozorované jevy, kdy je TTR téměř nulové a kdy je TTR rovno jedné, jsou velmi specifické konfigurace, které vysvětlíme a formalizujeme dále.

Sekvence přirozeného jazyka, konkrétně sekvence pocházející z českých beletrií (červená barva; dále zaměnitelné s *českými texty*), jsou oproti náhodným sekvencím umístěny vpravo a mají mnohem vyšší variabilitu. Křivky českých textů mají nižší sklon a k hodnotě TTR=1 konvergují velmi pomalu – většina z testovaných českých přirozených textů konverguje, dle výpočtu, k hodnotě TTR=1 až při $n \geq 140$. I zde tak pozorujeme zmíněný jev, kdy pro nízké velikosti *n*-gramů jsou hodnoty TTR velmi nízké a následně s jejich rostoucí velikostí konvergují k hodnotě TTR=1. Jak dále uvidíme, jedná se skutečně o artefakt metody MKM vyplývající z několika formálních vlastností.

Monkey-typed, nebo také pseudonahodilé texty (fialová barva), jsou umístěny mezi pomyslnými pásmy sekvencí české beletrie a pásmem náhodných sekvencí vytvořených atmosférickým šumem. Takový výsledek rozhodně odpovídá intuitivnímu umístění sekvencí mezi přirozený jazyk a náhodné sekvence, neboť text je sice psán nahodile a dle libosti, ovšem je stále produkován lidmi a jeho tvorba je ovlivněna různými pragmatikami fyzického psaní na klávesnici, ať už jejím rozložením, fyziologií ruky nebo jinými faktory. V důsledku tak tyto sekvence nemusí být náhodné tak, jako sekvence atmosférického šumu, ale budou zřejmě obsahovat vzory přibližující se přirozenému jazyku. Jev přibližování *monkey-typed* textů k přirozenému jazyku si dále prohlédneme podrobněji, abychom získali větší jistotu v logice interpretace výsledků metody MKM. Z vykreslených křivek sekvencí *monkey-typed* textů můžeme dále pozorovat, že oproti přirozenému jazyku křivka stoupá strměji a dále rychleji konverguje k hodně TTR=1 obdobně – avšak pomaleji – než náhodné sekvence. Konvergence křivek k hodnotě TTR=1 dochází pro většinu nahlížených sekvencí v blízkosti 60-gramů (kromě dvou testovaných sekvencí, kterým byly shodou okolností vybrány podřetězce obsahující triviální repetice). Konvergence *monkey-typed* sekvencí k hodnotě TTR=1 (tj. $n \approx 60$) se tak nachází opět mezi konvergencí náhodných sekvencí ($n \approx 20$) a přirozeným jazykem českých beletrií ($n \approx 140$).

Sekvence triviálních repetic (oranžová barva), tj. řetězce „a b a b a b a…“, „a b c a b c a b c …“ aj. splývají s osou *x*, tj. s hodnotami TTR blízkými nule. Hodnoty TTR blízké nule jsou u triviálních repetic dány normalizovanou velikostí abecedy na binární, která omezuje počet typů pro nízké hodnoty *n* a samotná podstata triviálních repetic omezuje počet typů vůči počtu tokenů ze své definice. Jinými slovy, ať už vybereme libovolnou délku *n*-gramů *n*, vždy u triviálních repetic získáme méně typů než počet tokenů. Sekvence triviálních repetic proto budeme v grafu očekávat vždy ve spodní části grafu a můžeme odhadnout, že sekvence obsahující časté opakující se vzory budou právě umísťovány blíže k ose *x*, a to na základě nižších hodnot TTR.

Bible ve 150 různých jazycích (zelená barva) jsou v grafu zobrazeny vpravo pod českými texty beletrie, tj. směrem k opakujícím se vzorům než směrem k nahodilým sekvencím. Tento výsledek není překvapivý, pokud vezmeme v úvahu, že Bible vykazují několik typů vzorů vyplývajících především z veršované formy textu, prediktabilního formátování textu a číslování veršů. Můžeme tak sledovat, že přidáním pevné strukturace do textů docílíme jejich logického umístění mezi (téměř) neformátovaný přirozený text a mezi triviální repetice a dále dosáhneme i jejich pomalejší konvergence k hodnotám TTR=1, která je dána tím, že i přes stále rostoucí velikosti *n*-gramů stále dokážeme nalézt nějaké jejich zopakování. Všimnout si také můžeme opět opakujícího se jevu splývajících průběhů křivek všech typů sekvencí (kromě těch s triviálními repeticemi) pro *n*-gramy s délkami od 1 do 6. Jak dále při formalizaci uvidíme, jedná se o specifický artefakt metody MKM vytvářený testovanou kombinatorikou binárních sekvencí.

Pohled na graf 2 s prvními výsledky aplikace metody MKM na sekvence různých typů nám prozradil několik důležitých poznatků. Prvním poznatkem (a v kontextu vývoje metody snad tím nejdůležitějším) je viditelná separovatelnost jednotlivých kategorií sekvencí, a to až na zmíněnou shodu pro prvních několik *n*-gramů. Pozorovaná separovatelnost znamená, že metoda MKM je, alespoň na základě tohoto prvotního testu, aplikovatelná k charakterizaci obecných a anonymních sekvencí s různě velkou abecedou, neboť už jen na základě té nejjednodušší analýzy se jednotlivé kategorie shlukují společně do viditelných pásem, které zároveň odpovídají i možnému logickému řazení od těch náhodných až po ty triviálně opakující se. Druhým důležitým poznatkem je, že jsme jen na základě průběhu křivky zřejmě schopni popsat základní charakteristiku sekvencí a na základě blízkosti k jednotlivým pásmům předvídat i její typ. Třetí poznatek je v důsledku triviální, nicméně je stále důležitý, a to, že analýza výsledných vektorů nebo také *embeddingů* je proveditelná i takto jednoduchou vizualizací, která umožňuje vhled do výsledků i do metody samotné (oproti např. okamžité aplikaci vícerozměrových nebo shlukovacích metod).

Pseudonáhodné sekvence

Jedním až z překvapivě pozitivních dosavadních výsledků aplikace metody MKM je zařazení tzv. *monkey-typed* sekvencí přímo mezi dokonale náhodné sekvence (v grafu 2 modře) a sekvence vzniklé z českých beletrií (v grafu 2 červeně). *Monkey-typed* sekvencím se nyní, jak jsme výše předeslali, budeme detailněji věnovat, především abychom nahlédli na to, zda je jejich umístění konzistentní s předběžně nabídnutou logikou, tj. řazením sekvencí dle určitého pravo-levého trendu od triviálních repetic, přes strukturovaná data, přirozený jazyk, pseudonáhodná data, až k náhodným sekvencím. Tento trend však musíme lépe ověřit a vysvětlit. Dalším důvodem analýzy *monkey-typed* sekvencí je předběžný úvod do problematiky náhodných a pseudonáhodných sekvencí. Zároveň se problematikou pseudonáhodných sekvencí či textů psaných na klávesnici zabývá jen velmi málo publikací (viz např. Ferrer-i-Cancho, del Prado Martín 2011 a NSA 1975). Přitom se jedná o oblast, která má své aplikace v počítačové bezpečnosti, a to na několika kritických místech, kterým se budeme, vzhledem k propojení s možnou aplikací metody MKM a zde využitelnosti kvantitativně lingvistických postupů, krátce věnovat. Bližší pohled na *monkey-typed* texty nám tedy poskytne mnohem více než jen kontrolu konzistence intuitivního chápání výsledků trendů a řazení z grafu 2. Příkladem nahodile napsané *monkey-typed* sekvence může být sekvence MonkeyTyped_Martin:

LKdjalkfjaijfcalkndkjwdkj jdaodwpjdalkjfaslkj kljdfjkrpoiwurj ncymnycknjmqpofj dfja…

Tato sekvence, ačkoliv se může zdát na první pohled jako náhodná, ve skutečnosti obsahuje řadu vzorů. Prvních 12 písmen (L, K, D, J, A, L, K, F, J, A, J, F) jsou všechna z prostřední řady klasicky užívané QWERTY klávesnice. Písmena I a W jsou z její horní řady a písmeno C ze spodní řady. Z této sekvence lze předpokládat, že se její autor drží klasického stylu psaní na počítačové klávesnici, kdy jsou prsty položeny právě na prostřední řadě, s malíkem položeným na klávese *A*, prsteníkem na *S*, prostředníkem na *D*, ukazovákem na *F* atd. a oběma palci na mezerníku. K napsání písmen v horní nebo spodní řadě se pak přesouvá nejbližší prst. Pohledem na frekvenční tabulku jednotlivých znaků (kláves) v tabulce 2 (vč. podbarvení kláves ze stejné řady) získáme detailnější náhled na systém, se kterým byla sekvence napsána. To, že autor sekvence používá klasický styl prstokladu, napovídá i pozorování, že nejméně frekventovaná písmena jsou z tohoto pohledu na těch nejvzdálenějších místech klávesnice (např. klávesy *ž*, *ř*, *ý*, *é*) nebo klávesy ve spodní řadě (klávesy *m*, *x*, *y*) atd., tj. klávesy buď vyžadující delší, nebo méně přirozený pohyb rukou. Nejfrekventovanější klávesy jsou naopak ty snadno dosažitelné ze základního prstokladu a nalezneme je nejčastěji v horní řadě klávesnice, případně se jedná o klávesy typické pro český jazyk. Z této sekvence a uvedené tabulky tedy lze, alespoň teoreticky, rekonstruovat fyzické rozvržení užité klávesnice i fyziologické vlastnosti rukou.

Rank	Type	Frekvence	Rank	Type	Frekvence
1	*[mezera]*	993	**22**	*ě*	77
2	*o*	895	**23**	*í*	76
3	*i*	864	**24**	*š*	75
4	*w*	806	**25**	*č*	53
5	*f*	721	**26**	*b*	52
6	*h*	610	**27**	*á*	47
7	*u*	495	**28**	*g*	43
8	*j*	345	**29**	*t*	35
9	*q*	321	**30**	*y*	30
10	*s*	241	**31**	*x*	29
11	*p*	223	**32**	*m*	20
12	*z*	203	**33**	*é*	18
13	*k*	192	**34**	*ů*	18
14	*e*	186	**35**	*ý*	16
15	*a*	182	**36**	*+*	5
16	*r*	166	**37**	*ú*	5
17	*d*	156	**38**	*ř*	5
18	*v*	149	**39**	*[tab]*	3
19	*n*	139	**40**	*ž*	3
20	*c*	120	**41**	*.*	1
21	*l*	114			

Legenda řádků *[diakritika]* *[horní]* *[prostřední]* *[spodní]* *[ostatní]*

***Tabulka 2**: Frekvence znaků textu MonkeyTypedMartin. Barevně jsou vyznačeny řady: okrová --diakritické, zelená = horní řada klávesnice, modrá = prostřední řada, červená = spodní řada*

Jediná *monkey-typed* sekvence nám umožnila nahlédnout na to, jaké vzory můžeme v takto vytvářených sekvencích očekávat a čím se budou tyto sekvence zřejmě lišit od těch zcela náhodných, tj. přítomností vzorů vycházejících z pravidel ovlivňujících pravděpodobnosti užití konkrétních symbolů vedoucích k omezení potenciální možné kombinatoriky následujících *n*-tic, což je právě jev zachytitelný metodou MKM pomocí *n*-gramů a TTR. Abychom myšlenku fyzicko-fyziologického trendu omezování kombinatoriky ověřili, celou analýzu zopakujme a dále rozvedeme na celém korpusu 21 *monkey-typed* textů s přibližně stejnou délkou dvou normostran. Analýzou frekvencí získáváme tabulku 3, ve které nalezneme prvních 42 nejfrekventovanějších znaků (kláves; pro lepší přehlednost). I zde můžeme pozorovat obdobný trend, který jsme zachytili u jediné sekvence výše: Nejfrekventovanější klávesou je opět mezerník, obecně nejfrekventovanější řadou je řada prostřední, následovaná horní řadou, dále spodní a nejméně frekventovanou je řada s českou diakritikou umístěná nejvýše. Opět tak pozorujeme trend využívat klávesy blízké prostřední řadě, což naznačuje využití klasického prstokladu. Nejfrekventovanější klávesy jsou z prostřední či horní řady, na které jsou prsty standardně položeny nebo je stačí posunout o jednu pozici dopředu. V protikladu pak stojí méně frekventované klávesy spodní řady, na které je nutné posunout prsty směrem k sobě a dále řada s diakritikou, kdy je naopak prsty nutné zcela natáhnout či přesunout celou ruku.

Rank	Type	Frekvence	Rank	Type	Frekvence
1	*[mezera]*	17571	**22**	*t*	2825
2	*j*	9601	**23**	*z*	2778
3	*i*	8863	**24**	*m*	2570
4	*f*	8378	**25**	*y*	2395
5	*d*	8179	**26**	*q*	2122
6	*o*	7889	**27**	*ů*	2023
7	*h*	7406	**28**	*x*	1645
8	*s*	7242	**29**	*í*	1404
9	*u*	6883	**30**	*á*	1319
10	*e*	6883	**31**	*č*	1168
11	*k*	6614	**32**	*š*	938
12	*a*	6580	**33**	*é*	919
13	*n*	5278	**34**	*,*	877
14	*l*	5001	**35**	*ř*	806
15	*w*	4669	**36**	*ž*	766
16	*v*	4563	**37**	*ú*	674
17	*g*	4526	**38**	*ý*	621
18	*p*	4326	**39**	*ě*	621
19	*r*	4233	**40**	*.*	580
20	*c*	3956	**41**	*;*	460
21	*b*	3888	**42**	*§*	419

Legenda řádků	*[diakritika]*	*[horní]*	*[prostřední]*	*[spodní]*	*[ostatní]*

***Tabulka 3**: Frekvenční analýza znaků korpusu monkey-typed sekvencí.*

Zjišťujeme tak, že nahodilý způsob psaní je ovlivněný ergonomií klávesnice a fyziologickou pragmatikou ruky. Takové zjištění vede k otázce, zda, kromě již odhalených tendencí užívat některé klávesy více než jiné, existují i jejich fyzicky preferované kombinace. Z korpusu sekvencí proto získáme pomocí *n*-gramů všechny 2 až 5-kombinace znaků, které následně seřadíme dle jejich frekvence. Nejfrekventovanějších 80 kombinací vidíme v tabulce 4. Zde si můžeme všimnout, že i přes možnost tisknout klávesy zcela libovolně, je jen pouhých 16 % z nich tvořeno kombinací kláves pocházejících z různých řad. To znamená, že v 83 % případů autoři nejčastěji kombinují klávesy nacházející se právě ve stejném řádku. Z tohoto pohledu již není překvapivé, že z uvedených 16 % kombinací řádků je 40 % z nich tvořeno kombinací kláves, které jsou umístěny bezprostředně vedle sebe. Takové pozorování odpovídá myšlence ergonomické pragmatiky a ekonomizačního principu nejmenšího úsilí – při zcela libovolném výběru mezi možnostmi bez jakéhokoliv možného zisku či ztráty může být jediným kritériem takového výběru jen nutná energie k rozhodnutí a vykonání. vložená do procesu výběru a vykonání.

Rank	Type	f	Rank	Type	f	Rank	Type	f	Rank	Type	f
1	sd	1331	21	js	616	41	oj	466	61	if	400
2	oi	1321	22	dj	604	42	gh	463	62	ow	397
3	io	1257	23	lj	602	43	da	460	63	hi	396
4	df	1165	24	hj	600	44	jj	456	64	fff	394
5	jk	1042	25	fa	568	45	ji	454	65	iw	390
6	kj	1020	26	sj	563	46	ee	453	66	oa	386
7	as	992	27	ad	558	47	ei	447	67	ffff	385
8	iu	942	28	fh	552	48	uh	447	68	fs	383
9	kl	885	29	dk	548	49	dd	444	69	fffff	382
10	ui	862	30	fd	545	50	ie	436	70	kh	380
11	er	806	31	jh	545	51	aj	427	71	dh	377
12	we	772	32	ff	537	52	ks	426	72	pa	374
13	jd	694	33	ih	536	53	fi	420	73	hd	373
14	hf	672	34	ja	526	54	wo	419	74	dc	372
15	fj	659	35	se	509	55	fk	417	75	ha	372
16	lk	652	36	op	498	56	sk	411	76	re	372
17	po	648	37	kd	480	57	vn	410	77	kf	371
18	ds	647	38	ou	473	58	sf	407	78	wi	369
19	jf	644	39	ij	471	59	jo	403	79	is	367
20	ew	636	40	ef	470	60	hs	402	80	oe	367

Legenda řádků	[horní]	[prostřední]	[spodní]	[kombinace]

***Tabulka 4**: Nejfrekventovanější 2-5-gramy.*

Rank	Type	f	Rank	Type	f	Rank	Type	f	Rank	Type	f
1	e_	3265	21	_d	1413	41	ej	973	61	_je	785
2	_	3140	22	u_	1375	42	en	958	62	ře	779
3	_p	3053	23	_m	1346	43	ou	940	63	la	768
4	_s	2756	24	na	1304	44	ní	932	64	t_	734
5	a_	2635	25	ho	1262	45	_b	920	65	ed	728
6	_n	2586	26	al	1256	46	ak	904	66	il	726
7	o_	2406	27	m_	1245	47	ta	902	67	ka	724
8	_v	2337	28	_z	1201	48	ve	902	68	ra	724
9	_t	1750	29	je	1200	49	le	901	69	že	717
10	_j	1684	30	ch	1174	50	_na	901	70	el	710
11	l_	1669	31	te	1159	51	_ne	883	71	by	696
12	_k	1602	32	ov	1158	52	se_	876	72	ho_	685
13	po	1545	33	_po	1131	53	ně	870	73	na_	677
14	_a	1531	34	to	1084	54	k_	869	74	_h	663
15	st	1526	35	_se	1066	55	ro	868	75	_ž	648
16	_	1509	36	_a_	1052	56	y_	862	76	an	647
17	i_	1491	37	li	1045	57	_se_	861	77	no	647
18	._	1435	38	í_	1025	58	do	858	78	os	642
19	ne	1432	39	ko	996	59	va	816	79	vo	641
20	se	1430	40	_o	973	60	em	808	80	_je	785

***Tabulka 5**: Nejfrekventovanější 2-5-gramy znaků knihy Osudy dobrého vojáka Švejka za světové války od Jaroslava Haška. Zeleně jsou zvýrazněny kombinace znaků obsahující mezeru, modře kombinace obsahující vokál a konsonant.*

Monkey-typed sekvence založené na libovolných úhozech do klávesnice tak můžeme shrnout jako pseudonáhodné. Tyto sekvence obsahují řadu vzorů, které jsou dány fyziologickým a ekonomickým omezením obdobně, jako je tomu u přirozeného jazyka. Na druhou stranu jsou tyto vzory zároveň více rozmanité než ty v přirozeném jazyce, jak můžeme pozorovat v tabulce 5 mapující 2 až 5-gramy znaků sekvence české beletrie o 180 000 znacích (téměř shodně s délkou korpusu *monkey-typed* textů). V této tabulce si můžeme všimnout, že se polovina nejfrekventovanějších písmen pojí s mezerou (zvýrazněno zeleně), se kterou se stejně tak pojí i všechny nejdelší – tříznakové kombinace. V případě, že se v kombinaci znaků nevyskytuje mezera, je tato kombinace až na jediný případ tvořena jedním konsonantem a jedním vokálem (zvýrazněno modře). Oba typy textů tak vykazují společný rys v omezení kombinatoriky některých symbolů abecedy, a to na základě určitých pragmatických důvodů. Pro *monkey-typed* sekvence se může jednat o ekonomizaci pohybu při psaní a v případě přirozených textů o fonotaktiku. Jak už bylo řečeno výše, kombinatorika *monkey-typed* textů je rozmanitější, tj. umožňuje kombinovat větší sady symbolů abecedy, než můžeme pozorovat v příkladu přirozeného jazyka. Z tohoto náhledu je umístění *monkey-typed* sekvencí mezi náhodné sekvence kombinující libovolné znaky bez pravidel a sekvence přirozeného jazyka logické. Výsledky metody MKM jsou v tomto ohledu konzistentní s nalezenou vnitřní logikou obou typů textů.

Detekce a analýza pseudonáhodných či *monkey-typed* sekvencí je přitom zajímavým problémem právě z hlediska možných aplikací. Jak se dozvíme detailněji dále v textu, generování skutečně náhodných sekvencí není triviální úkol, který v kontextu počítačů vyžaduje i externí zdroj entropie. V tomto ohledu je zajímavé, že takovým zdrojem entropie může být právě člověk a jeho klávesnice s explicitním požadavkem softwaru o náhodné tisknutí kláves (tedy *monkey-typing*, popřípadě o náhodný pohyb myši). Jak již víme, sekvence vzniklé tímto způsobem nejsou skutečně náhodné, což se dále promítá do snížení míry zabezpečení informací a dat v kyberprostoru. Shodný problém nastává v případě tvorby šifračních klíčů a všednější verzí téhož, tj. při tvorbě hesel. Prolamování (hádání) šifračních klíčů a hesel počítačovými útočníky je založeno především na poznatcích těžících z empirie a pragmatiky jejich volby odvíjejících se z klasických lingvistických analýz korpusů textů a uniklých hesel (viz např. Dürmuth *et al*. 2015 nebo Narayanan a Shmatikov 2005). Zatímco jsou hesla volena intuitivně tak, aby nebyla uhodnutelná jiným člověkem, jsou tato hesla prolamována slovníky tvořenými existujícími, různě kombinovanými slovy, včetně jejich používaných[3] mutací, velkými-malými písmeny, čísly atd. V případě absence hesla ve slovníku je volena forma útoku hrubou silou testující všechny možné kombinace písmen a znaků,

[3] Zajímavé je, že se tvorbě šifrovacích klíčů náhodným psaním na klávesnici/stroji obdobným způsobem věnovala i Národní bezpečnostní agentura Spojených států, zkráceně NSA, viz odtajěné číslo interního časopisu *Cryptolog* (NSA 1975, 12).

kdy jsou nejprve testovány takové kombinace, které jsou z uvedených pragmatik nejčastější – např. na základě jednoduchosti zapamatování nebo nově i na základě poznatků z analýzy korpusu *monkey-typed* sekvencí v případě, že je heslo naivně tvořeno jako *náhodné*. Smysl náhodných hesel či klíčů tkví v tom, že absence jakýchkoliv vzorů vede útočníka k nutnosti procházet celý prostor kombinací k jeho jistému uhádnutí. Tento prostor může být od konkrétní délky hesla natolik velký, že jej nebude možné v reálném čase projít a heslo uhádnout. Zběžná lingvistická analýza *monkey-typed* textů výše nám tak odhalila především existenci řady vysvětlitelných vzorů, které by bylo možné využít jako heuristiku při hádání šifračních klíčů či hesel.

Důležitým poznatkem, ke kterému se z náhledu na aplikace tohoto partikulárního výsledku vraťme, je, že metoda MKM dokáže registrovat rozdíly mezi pseudonáhodnými sekvencemi, přirozeným jazykem, a navíc i skutečně náhodnými sekvencemi. Naše pozornost věnovaná *monkey-typed* sekvencím nám navíc poskytla náhled na to, jak vypadají vzory uvnitř pseudonáhodných sekvencí ovlivněných řadou faktorů. Porozumění pseudonáhodným sekvencím nás následně posouvá k náhodným sekvencím, kterým se budeme věnovat v následující podkapitole tak, abychom vysvětlili vlastnosti pozorované v grafu 2.

Náhodné sekvence

Obecným cílem metody MKM je charakterizace obecných, anonymních sekvencí, u kterých známe pouze užitou abecedu. Jedním z důležitých cílů této charakterizace je i schopnost určit, zda jsou zkoumané sekvence náhodné nebo ne. Předběžně víme, že metoda MKM dokáže testované náhodné sekvence separovat od všech ostatních natolik, že v grafu 2 vytváří vlastní, homogenní pásmo. Je však otázkou, zda metoda MKM dokáže rozlišit mezi různými stupni náhodnosti, jejich zdroji a jejich různými typy. Jak v této podkapitole uvidíme, takto jemná identifikace náhodných sekvencí je komplexní problém, který poukáže na další důležité vlastnosti metody MKM s důsledky pro interpretaci výsledků a principu jejího fungování. Schopnost detekce náhodnosti sekvencí je pak mimořádně zajímavá z mnoha praktických důvodů, neboť skutečná nebo pouze domnělá náhodnost sekvencí je zásadní otázkou pro řadu aplikací mimo vědní disciplíny a praxi. Náhodnosti sekvencí se zde proto budeme věnovat o něco hlouběji.

Z předešlých analýz a grafu 2 víme, že metoda MKM dokáže na základě vizuálního porovnání křivek odlišit náhodné sekvence od pseudonáhodných sekvencí, sekvencí přirozeného jazyka nebo triviálních repetic. Takové pozorování je pro metodu MKM a její teoretickou schopnost formálně rozlišovat náhodné sekvence zcela zá-

kladní. Důležitým pozorováním je rovněž i to, že se jednotlivé křivky náhodných sekvencí v tomto grafu jeví – v porovnání s ostatními kategoriemi sekvencí – jako téměř dokonale překrývající se. To znamená, že metoda MKM registruje podobnost mezi samotnými náhodnými sekvencemi, která je vyšší než u sekvencí přirozeného jazyka. Taková míra shody se může zdát pro *náhodné* sekvence poněkud paradoxní, vzhledem k tomu, že jsou náhodné. Jak ale dále odvodíme, princip metody MKM vede ke specifickým kombinatorickým modelům, které tuto podobnost náhodných sekvencí vysvětlují a které dále využijeme k testování náhodnosti sekvencí. Tyto poznatky navíc dále poskytnou porozumění samotné metodě a již pozorovaným a zmíněným artefaktům. Předtím náhodnost a náhodné sekvence nejprve definujeme a následně nahlédneme na metodu MKM znovu, s novým kontextem.

S definicí náhodných sekvencí a jejich vlastnostmi jsme se zběžně setkali v předchozí podkapitole věnované *monkey-typed* sekvencím vytvořených nahodilými úhozy do klávesnice. U těchto sekvencí jsme pozorovali různě frekventované vzory reflektující způsob psaní i povahu fyzického zařízení, na kterém byly napsány. Seřazením nalezených vzorů podle jejich frekvence jsme odhalili alespoň jedno pravidlo ekonomické dosažitelnosti kláves tvořící specifický skrytý systém odporující zcela náhodnému výběru. Není proto překvapivé, že definice náhody plyne právě z absence systému, který by upřednostňoval kterýkoliv vzor či kombinaci. Definice náhodnosti je dána rovnoměrnou pravděpodobností výskytu každého symbolu abecedy a zároveň absencí jakýchkoliv pravidel či mechanismů jejich výběru. Náhodnou sekvenci můžeme definovat jako sekvenci znaků abecedy Σ, u které nelze předvídat následující nebo předcházející symbol či symboly na základě znalosti libovolného podřetězu symbolů ani čehokoliv dalšího (viz např. Kneusel 2018, 1). Predikce následujícího symbolu skutečně náhodné sekvence tak nemůže být nikdy lepší než hod dokonale férové kostky o počtu stěn shodným s velikostí abecedy. Jak ovšem dále uvidíme, náhodnost sekvence neznamená neobsažnost vzorů. Uvidíme, že vzory mohou vznikat i prostou shodou okolností, tj. náhodou a právě hranice, kdy jsou vzory využívány příliš mnoho, nebo na náhodu dokonce příliš málo, bude základním principem aplikace metody MKM na identifikaci a rozřazování náhodných sekvencí.

Jak uvidíme dále, užití domnělých náhodných sekvencí namísto těch skutečně náhodných, není jen partikulární problematikou hesel, ale jde o obecný problém zahrnující zabezpečení informací, bezpečnost komunikačních protokolů a dále například způsoby ověřování identit. Z tohoto důvodu existuje řada metod, jejichž cílem je testování náhodnosti sekvencí a ověřování jejich kvality. S jednou takovou metodou jsme se zde již dříve setkali, a to s metodou T-komplexity od Hamano a Yamamoto (2010), na kterou se zaměříme později. Pro účely testování kvality generátorů náhodných sekvencí ovšem byla *Národním institutem standardů a technologií* ve Spojených státech

(zkráceně NIST; Rukhin *et al.* 2001)[4] publikována i sada patnácti standardizovaných testů. Na některé z testů nejbližších k metodě MKM proto nahlédneme tak, abychom si vytvořili představu o užívaných metodách a možné relaci k MKM (dále dle Rukhin *et al.* 2001). Prvním testem sady NIST je ten nejjednodušší, kterým musí analyzovaná sekvence projít vždy, tj. test *Monobit*, který kontroluje frekvence bitů 0 a 1, které by měly být u náhodných sekvencí přibližně stejné (respektive totožné v případě nekonečně dlouhých sekvencí). Druhým testem je aplikace *Monobit* na bloky definované délky, u kterých opět testuje poměr symbolů 0 a 1. Doporučená délka bloků je alespoň 20 bitů. Třetím je tzv. *Runs test*, který testuje pravděpodobnosti výskytu po sobě jdoucích („*runs*") stejných bitů. Další z testů kontroluje přítomnost uživatelem vybraného vzoru v blocích o vybrané velikosti n (tj. na oknech a případně v následujícím testu na n-gramech) sekvence. Za principiálně blízké testy ze sady NIST k metodě MKM pak můžeme považovat např. *Maurerův univerzální statistický test*, který sekvenci bezeztrátově komprimuje a v případě zjištění signifikantní změny velikosti sekvence po kompresi je tato sekvence označena za nenáhodnou, a to vzhledem k přítomnosti opakujících se vzorů. Nejblíže metodě MKM je *Serial Test* (a respektive i *Approximate Entropy Test*), který mapuje frekvence n-gramů pro zadanou velikost n a dvě nižší délky, tj. n-1 a n-2 a pro které následně *chí*-kvadrátem testuje jejich teoretickou frekvenci. Metoda MKM je pak rozdílná především v neomezení se na pouhé tři vybrané délky n-gramů a dále nabízí i alternativní způsob provedení statistického testu bez využití *chí*-kvadrát testu. Princip metody MKM tedy předběžně můžeme, na základě porovnání s metodami sady NIST, považovat za relevantní. Také můžeme říci, že aktuální návrh metody MKM rozšiřuje některé z uvedených testů, a některé z nich navíc automaticky a implicitně zahrnuje principem fungování, tj. například zahrnutí úvodního testu *Monobit*, který bude v důsledku testován při analýze 1-gramů. Kromě testů z NIST existují i další metody. Jmenujme především testy z baterie *Die Hard* vyvinuté Georgem Marsgliou (2008), ze kterých čerpá alespoň jedna z metod NIST. Všechny z uvedených metody jsou však určeny výhradně a pouze k testování nebo charakterizaci náhodných sekvencí, což je jen jedním z cílů metody MKM. Testování náhodných sekvencí, vzhledem k množství existujících testů, očividně není triviální úlohou. Přitom stále můžeme nalézt sekvence, které testy náhodnosti projdou a které jsou nenáhodné anebo obsahují nenáhodné a uměle dodané vzory schopné ovlivnit cílové aplikace (viz např. Govindan *et al.* 2018 nebo Hamano a Yamamoto, 2010).

[4] Dokumentace NIST k testování náhodných sekvencí je dostupná online: https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-22r1a.pdf, cit. 14. 8. 2018.

Před odvozováním modelů náhodných sekvencí v kontextu metody MKM nejprve připomeňme, že metoda MKM funguje na jednoduchém principu kvantifikace opakování *n*-gramů pro každou zkoumanou délku *n*. I přes principiální jednoduchost této metody je však nesnadno představitelná a neintuitivně uchopitelná pro odvozování modelů vysvětlujících chování náhodných sekvencí a jejich výsledků. Na celou metodu MKM proto nahlédneme jiným způsobem, ze kterého potřebné modely odvodíme jednodušeji, za pomoci již dobře známých poznatků. V intuitivním chápání metody je nutné zopakovat, že metoda pouze opakuje výpočet TTR pro jednotlivé délky *n*-gramů. Získávání *n*-gramů ze sekvence můžeme vnímat jako proces, při kterém postupně vytváříme nový text obsahující pouze délky slov *n*. To znamená, že aplikací 3-gramů na binární sekvenci *S* získáváme nový text tvořený *slovy* o délce 3 symboly jedniček anebo nul. Na takto vzniklém textu následně kvantifikujeme opakování těchto slov pomocí TTR. Pokud dále tímto způsobem nahlédneme na tvorbu náhodného textu tvořeného daným počtem binárních slov o délce *n*, bude úvaha nad tvorbou modelů prakticky zřejmá. Tvorbu takového náhodného textu můžeme uchopit jako postupné zaplňování *N* volných políček na listu papíru pomocí herní kostky s tolika stěnami, kolik je dostupných slov ve slovníku, tj. *V*, které je dáno všemi variacemi s opakováním *n* symbolů abecedy {0, 1}. Pomocí *N* hodů kostkou postupně náhodně vybereme slova pro celý text. Je proto již intuitivně zřejmé, že se v textu některá slova zopakují, některá budou realizována pouze jednou a některé slova z dostupného slovníku variací nebudou realizována vůbec. Stejně, jako se při hodu hracími kostkami stává, že několikrát za sebou padne stejné číslo, i u tvorby náhodných textů se může zopakovat některé slovo několikrát za sebou. Opakování či jednoduché vzory jsou tedy běžné, ovšem i intuicí vnímáme, že taková opakování musí být v určitých mezích. Všechny z popsaných jevů realizace slov, jejich opakování, nevybrání, vybrání pouze jednou atd. budou závislé na počtu slov textu (počet hodů, *N*) a velikost slovníku, ze kterého vybíráme (počet stran kostky, *V*). Takové chápání, a především i pouhé využití kvantifikace opakování slov pomocí TTR, nám poskytne velmi cennou formální oporu, neboť budeme moci těžit z existujících formalizmů de Moivra a Stirlinga (De Moivre 1967; DasGupta 2010; Bellhouse 2011).

První model, který odvodíme, je model mapující chování dokonale náhodných sekvencí v kontextu metody MKM. Cílem tohoto modelu je pro zadanou binární sekvenci *S* o délce *k* bitů a jednotlivé délky *n*-gramů *n* od 1 do Z vypočítat konkrétní teoretické hodnoty TTR. Již dopředu je zřejmé, že hodnoty TTR skutečně náhodných sekvencí mohou být od takového modelu odchýleny, a to právě vlivem náhody, buď vyšším či nižším zopakováním některých z možných slov. Model dokonale náhodné sekvence, který zde odvodíme, tak nutně musíme vnímat jako průměr nekonečného počtu takových dokonale náhodných sekvencí, namísto modelu jediné náhodné sekvence. V odvození nám následně pomůže předeslaný způsob uvažování nad metodou MKM, a to tak, že na jednotlivé kroky výpočtů TTR jednotlivých délek *n*-gramů (slov)

nahlédneme jako na tvorbu Z pomyslných separátních texů. V tomto pohledu stačí odvodit průměrný počet různých binárních slov V_{avg} o délce n, který se v textu o délce k bitů (neboli D binárních slov) vyskytne. Získaný počet různých slov V_{avg} následně vydělíme délkou sekvence ve slovech D a tím získáme hodnotu $TTR_{avg}(n,k)$, neboli průměrné TTR sekvence o délce k bitů a délce slov n. Využitím pouze binární abecedy pro slova o velikosti n pomocí (5) snadno vypočítáme velikost možného (nebo také tzv. potenciálního) slovníku V, ze kterého vybíráme slova do náhodného textu:

$$V(n) = 2^n \quad . \tag{5}$$

Počet slov D o délce n, kterými je tvořena sekvence o délce k, vypočítáme dle (6):

$$D(n,k) = k - n + 1 \quad . \tag{6}$$

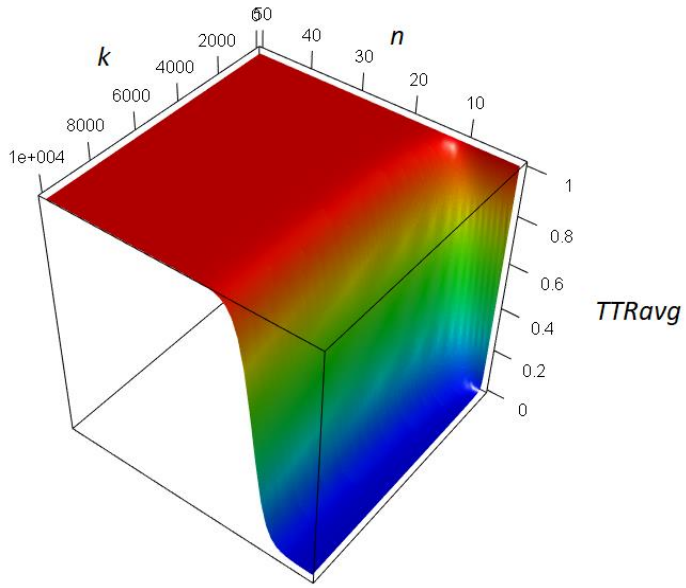
Pro zadanou binární sekvenci S o bitové délce k, velikosti slova n získáme průměrnou velikost slovníku pomocí (7; dle Matlach *et al*. 2018, blíže viz např. Conroy 2018, 21-23):

$$V_{avg}(n,k) = V(n)\left(1 - \left(1 - \frac{1}{V(n)}\right)^{D(n,k)}\right) \quad . \tag{7}$$

Jakmile známe průměrnou velikost užitého slovníku V_{avg} pro zadanou konfiguraci (tj. známe průměrný počet typů), snadno vypočítáme průměrné TTR pomocí (8):

$$TTR_{avg}(n,k) = \frac{V_{avg}(n,k)}{D(n,k)} \quad . \tag{8}$$

Pomocí modelu (8) nyní dokážeme pro zadanou délku textu (sekvence v bitech) a velikost binárních slov (velikost n-gramů) určit hodnotu TTR, které dosáhne zprůměrování nekonečného počtu dokonale náhodných sekvencí. Z tohoto modelu následně vyplývají pozoruhodné vlastnosti inherentní principu fungování metody MKM a jí sledované kombinatoriky, především delikátní vztah mezi délkou textu k, velikostí slova n a výsledným TTR. Počet podob, kterých může nabývat binární slovo o délce n bitů, vypočítáme dle (5) jako $V(n) = 2^n$. Právě tato exponenciální podstata velikosti potenciálního slovníku dává vzniknout třem komplementárním situacím, které jsme pozorovali v úvodním grafu 2 a které následně ozřejmíme. Nejprve se podívejme na vztah délky slov n a délky textu k, který pro hodnoty $n = \langle 1{,}50 \rangle$ a $k = \langle 1{,}10^4 \rangle$ vykreslíme pomocí modelu (8). Na výsledek se podívejme do grafu 3, ze kterého jsou patrná tři důležitá pozorování.



***Graf 3**: Modelový vztah velikosti binární sekvence v bitech k, velikosti slov n a vypočítané TTR, pro průměr dokonale náhodných sekvencí.*

Prvním a patrně nejvýraznějším pozorováním je, že dosažení hodnot TTR = 1 (červená barva) je v případě náhodných sekvencí velmi rychlé. Výsledek TTR = 1 nastává v případě, kdy je velikost využitého slovníku shodná s počtem slov textu. V grafu vidíme, že už pro binární slova o délce n=20 bitů je i délka sekvence 10^4 natolik malá, že náhodným výběrem slov nedojde ke zopakování jediného slova. Konfigurace velkého potenciálního slovníku a krátké sekvence tak stojí za rychlou konvergencí k hodnotám TTR = 1. Druhým výrazným pozorováním v grafu 3 je pravý opak předchozí situace, tj. konfigurace velmi malého potenciálního slovníku oproti dlouhé sekvenci. Například pro slova o délce 1 bit s potenciálním slovníkem {0, 1} je od specifické a relativně krátké délky sekvence již prakticky téměř jisté, že se náhodným výběrem využije celý slovník a jeho slova se budou s narůstající délkou sekvence opakovat, což vede k hodnotám TTR konvergujícím k nule (modrá barva). Mezi oběma extrémy je následně pásmo (zelená), ve kterém dochází k vyrovnání obou velikostí, tj. velikosti potenciálního slovníku a velikosti sekvence. Tyto tři jevy či konfigurace jsou pro nás důležité, neboť se týkají vysvětlení hodnot TTR a důvodů jejich konvergence. Přitom se jedná o vlastnosti vyplývající přímo ze samotné metody MKM a vysvětlující některé vlastnosti pozorovaných křivek v úvodním grafu. Uvedené tři jevy, které pracovně nazveme jako jevy kombinatorického vyčerpání, kombinatorického nasycení a harmonie, dále rozpracujeme a formalizujeme.

Jev kombinatorického vyčerpání

Jev kombinatorického vyčerpání vychází z výše popsané konfigurace krátkých slov (*n*-gramů) a dlouhých textů (sekvencí). Krátké *n*-gramy mají velmi omezenou kombinatoriku. Například počet slov potenciálního slovníku, který je dán binárním 3-gramem, tj. pomocí 3 bitů, je pouze $2^3 = 8$ různých slov. Pokud text o cílové délce 50 slov náhodně zaplňujeme slovy z takto malého slovníku, intuitivně víme, že některé ze slov brzy zopakujeme. Čím kratší *n*-gramy jsou, tím menší je potenciální slovník jejich variací a tím dříve v textu využijeme všechna jeho slova, která posléze začneme opakovat. Z hlediska TTR takový vztah znamená, že s narůstající délkou sekvence *k* a snižující se délkou *n*-gramů *n* bude platit:

$$\lim_{\substack{n\to 1 \\ k\to\infty}} \frac{V(n)}{D(n,k)} = 0$$

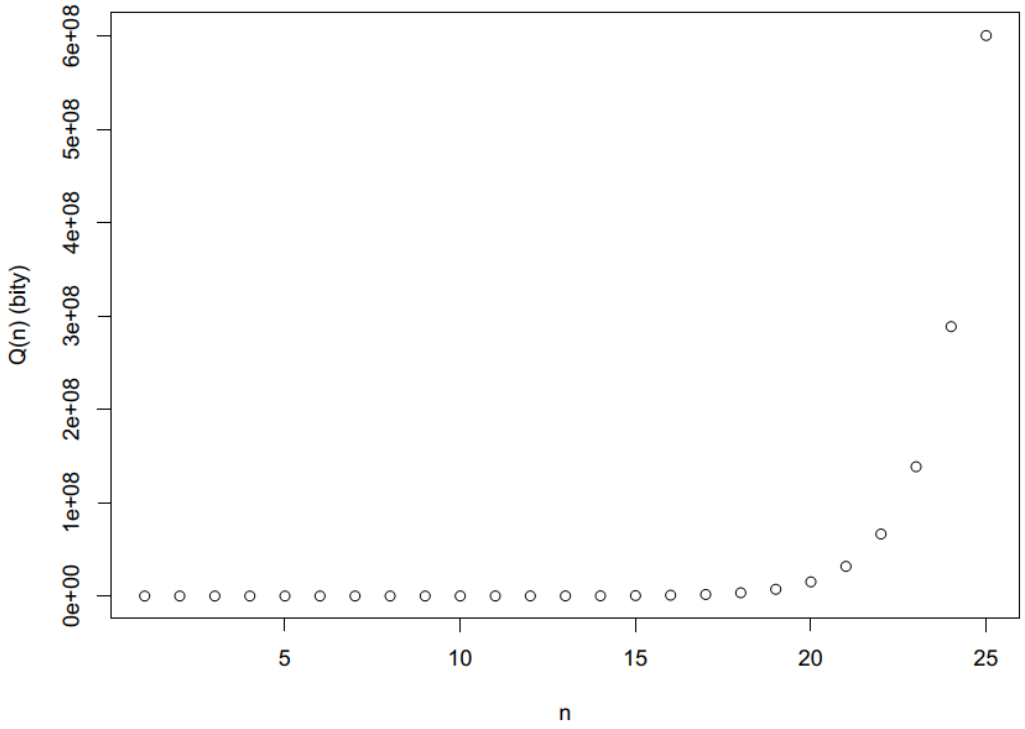
Takový vztah je zřejmý, ale pro nás je dále zajímavé odvodit, pro jaké délky řetězců *k* a zadanou velikost slova *n* už bude v průměru náhodných sekvencí docházet k využití celého slovníku, tj. k jeho vyčerpání. Z intuitivního chápání popsaného výše víme, že pravděpodobnost vyčerpání slovníku stoupá tím, čím je kratší délka slov a čím je delší sekvence. V případě binárních slov o délce $n = 1$ slovník obsahuje pouze slova {0,1}. Náhodně vytvořená sekvence o délce $k = 100$ bitů z tohoto slovníku intuitivně a s jistotou využije všechna slova. Obdobně pak i pro $n = 2$ (4 slova), $n = 3$ (8 slov) atd. až do konkrétní kritické hodnoty, po které už průměrná náhodná sekvence celý slovník s jistotou nevyužije. Otázku takové kritické hranice můžeme nahlédnout i z druhé strany: Pro jakou minimální bitovou délku *k* vyčerpá sekvence celý potenciální slovník binárních slov o délce *n*? Výslednou hodnotu *k* můžeme pro zadanou délku binárních slov *n* nazvat jako kritická hranice vyčerpání, značenou jako *Q(n)*. Tato hranice je pro nás důležitá, neboť nám poskytne porozumění, za jakých okolností je TTR blízké 0 věcí náhody a kdy jde o artefakt samotné metody. Abychom ji vypočítali, odvodíme nejprve průměrnou délku sekvence *Q'(n)* ve slovech *T* délky *n* tak, aby se v ní objevila všechna slova z potenciálního slovníku alespoň jednou, tj. tak, aby vyčerpala celý potenciální slovník (v důsledku tak jde o analogii problému sběratele kupónů, viz Conroy 2018, 16 a Mitzenmacher a Upfal 2005, 33):

$$Q'(n) = E(T) = V(n) \sum_{i=1}^{V(n)} \frac{1}{i} \quad .$$

Odsud již snadno získáme **průměrnou bitovou délku sekvence vyčerpávající celý slovník**, tj. **kritickou hranici vyčerpání Q**, pomocí (9):

$$Q(n) = n * Q'(n) \qquad . \qquad (9)$$

Důsledky (9) jsou zajímavé a vhodné k nahlédnutí pomocí grafu jednotlivých délek slov od 1 do 25. Výsledek vidíme v grafu 4. Zde můžeme pozorovat délku binárních slov n a nutné délky sekvence v bitech vypočítané pomocí (9) tak, aby se v ní v průměru vyskytlo každé slovo z potenciálního slovníku alespoň jednou. Exponenciální růst slovníku vyžaduje pro takovou realizaci slov i exponenciálně se zvyšující délku sekvence. Sekvence, ve které například v průměru náhodnými výběry využijeme neboli *vyčerpáme* celý slovník tvořený všemi možnými binárními slovy o délce $n = 25$, jsou v průměru dlouhé 600 822 143 bitů, tj. přibližně 75 MB. V grafu 4 si dále můžeme všimnout, že se hodnoty v rozmezí $n = \langle 1; 15 \rangle$ jeví jako konstantní, pohledem do tabulky 6 s rozepsanými hodnotami zjistíme, že tomu tak není a jedná se jen o zkreslení dané rozsahem exponenciálně se zvyšujících hodnoty na ose y.



***Graf 4**: Vztah velikosti binárního slova o délce n a očekávané délky sekvence Q(n) (v bitech), ve které bude použita každá variace s alespoň jedním opakováním.*

n	⌈Q(n)⌉	n	⌈Q(n)⌉	n	⌈Q(n)⌉	n	⌈Q(n)⌉
1	3	**8**	12 543	**15**	5 394 157	**22**	1 460 381 083
2	17	**9**	31 411	**16**	12 234 343	**23**	3 187 258 490
3	66	**10**	76 894	**17**	27 542 457	**24**	6 930 767 799
4	217	**11**	184 777	**18**	61 595 872	**25**	15 020 553 567
5	650	**12**	437 213	**19**	136 940 486		
6	1 822	**13**	1 021 104	**20**	302 832 100		
7	4 869	**14**	2 358 286	**21**	666 473 733		

***Tabulka 6**: Rozepsané výsledky (9) pro jednotlivé velikosti binárních slov n.*

V úvodním grafu metody MKM, tj. grafu 2, jsme si pro sekvence dlouhé 6 000 bitů a krátké délky slov n od 1 přibližně do 7 mohli všimnout prakticky shodného průběhu křivek (indikujících shodné nebo podobné hodnoty TTR) různých kategorií dat (kromě triviálních repetic). Tento jev můžeme nyní vysvětlit pomocí popsaného jevu vyčerpání. Z tabulky 6 zjistíme, že pro náhodné sekvence o 6 000 bitech dojde v průměru k jevu vyčerpání slovníku pro délky slov 1 až 7. Náhodné sekvence tak pro tyto délky budou mít v průměru stejně velký realizovaný slovník a (dle definice) i shodný počet tokenů, tj. shodné hodnoty TTR. Naopak příklady sekvencí triviálních repetic všechny dostupné typy nevyčerpávají a s hodnotami TTR zůstávají blízko nule. Z tohoto pohledu jsou však zajímavé sekvece přirozeného jazyka, které ty náhodné v uvedených velikostech n-gramů kopírují. Nyní víme, že u sekvencí přirozeného jazyka v tomto rozsahu dochází k jevu vyčerpání, přitom z příkladu triviálních repetic víme, že tak činit nemusí. Důvod vyčerpání slovníku je v případě přirozeného jazyka dán přímo metodou MKM, a to konkrétně způsobem normalizace abecedy. Například, pokud má abeceda textu celkem 63 různých symbolů (písmena, číslovky, mezery, interpunkce atd.), je minimální velikost binárního slova kódující tuto abecedu 6 bitů ($2^6 = 64$ možných kódů symbolů). To znamená, že až na 1 nevyužitý kód ze 64 budou tato slova pouze dle princopu jejich překódování pro n od 1 do 6 vyčerpána. Shodu průběhů náhodných sekvencí a těch *přirozených* označíme za artefakt metody, který o sekvencích neposkytuje žádné informace kromě možné přítomnosti rozsáhlé abecedy nebo náhodnosti sekvence.

Od kritické hranice bitové délky sekvence Q(n), od které se průměrně zopakuje každé ze slov potenciálního slovníku alespoň jednou a dojde k jevu pojmenovaném jako vyčerpání, můžeme odvodit i další důležité informace. Již víme, že k jevu vyčerpání dochází tehdy, když je sekvence dostatečně dlouhá na to, aby v ní byla na základě náhodného výběru slov ze slovníku vybrána všechna možná binární slova o délce n. Na tento problém můžeme nahlédnout i z jiné strany: Pro konkrétní zadanou sekvenci o konkrétní délce se můžeme ptát, pro jaké délky slov n dojde k vyčerpání slovníku. Na základě tabulky 6 už můžeme takovou otázku snadno zodpovědět. Například právě pro sekvence dlouhé 6 000 bitů vyčerpáme slovník délek binárních slov n od 1 do 7, a to proto, že potřebná průměrná délka sekvencí k vyčerpání slovníku těchto délek slov je stále nižší než délka studované sekvence, tj. 6 000 bitů. Délka slov 7 bitů je tedy pro studovanou délku sekvencí hraniční. Využitím (9) proto můžeme pro libovolnou zadanou sekvenci S o délce k bitů určit **kritickou délku slova n, při které naposledy dojde k jevu vyčerpání, tato velikost je dána jako $q(k)$ vzorcem (10):**

$$q(k) = \arg\max_{n \in \mathbb{N}} Q(n) < k \qquad (10)$$

Pomocí (10) následně odvodíme interval velikostí slov n, u kterých dojde v průměru dokonale náhodných sekvencí o délce k k jevu vyčerpání slovníku, tento interval pracovně označíme jako $\mathbb{Q}$:

$$\mathbb{Q}_k = \langle 1;\ q(k) \rangle \subset \mathbb{N}$$

Interval vyčerpání $\mathbb{Q}_k$ má dva důležité důsledky. Prvním důsledkem je, že pro náhodné sekvence budeme pro délky $n \in \mathbb{Q}_k$ očekávat využití celého slovníku o velikosti $V(n)$, což pro konkrétní testovanou sekvenci o délce k vede k očekávání hodnot $TTR_n = V(n)/D(k)$. Druhým důsledkem je i to, že v takovém případě očekáváme shodné výsledky vypočítaných hodnot TTR_n s modelem dokonale náhodných sekvencí TTR_{avg}. Jev kombinatorického vyčerpání a vypočítaný interval délek slov v úvodním grafu 2 tak vysvětlují shodné průběhy náhodných křivek na intervalu $n \in \langle 1{,}7 \rangle$. Shodné průběhy sekvencí přirozených textů jsou však artefaktem metody zapříčiněným způsobem normalizace abecedy, jak bylo vysvětleno výše, jedná se tedy o praktickou shodu okolností.

Nyní máme k dispozici funkci (9), pomocí které dokážeme určit průměrnou délku náhodné sekvence nutné k vyčerpání či využití celého potenciálního slovníku slov o délce *n*. Rovněž máme k dispozici funkci (10), s jejíž pomocí dokážeme odvodit maximální délku slova, pro kterou k vyčerpání v dané sekvenci ještě dojde. Zajímavé je, že následující velikost slova, tj. $q(k) + 1$, už je velikostí, od které jev vyčerpání neplatí a která pro nás bude rovněž zajímavá. Abychom takový nově nastíněný interval dokázali uchopit, odvodíme nejprve druhý extrémní interval vytvářený jevem, který pracovně nazveme jako *jev nasycení*.

Jev kombinatorického nasycení

Oproti krátkým *n*-gramům, které stojí za jevem vyčerpání, generují dlouhé *n*-gramy (např. $n = 40$) velmi mohutné potenciální slovníky. Například počet různých slov, které můžeme vytvořit pomocí $n = 40$ bitů je $2^{40} = 1\,099\,511\,627\,776$. Je tak zřejmé, že například pro text o padesáti slovech je realizace 1 bilionu slov principiálně nemožná, což je triviální poznatek. Méně triviální, avšak stále intuitivní poznatek pak je, že pravděpodobnost výběru kteréhokoliv z bilionu slov dvakrát za použití padesáti pokusů, je téměř nulová (formálně viz později). Sekvence (či text) není, jednoduše řečeno, vůči velikosti potenciálního slovníku tak velká, aby umožnila náhodně vybrat alespoň dvakrát některé ze slov. To znamená, že pro zadanou bitovou délku sekvence existuje kritická délka *n*-gramů (slov), která bude produkovat tak rozsáhlý potenciální slovník, že u skutečně náhodných sekvencí nebudeme v průměru očekávat zopakování kteréhokoliv z realizovaných slov. Naopak budeme pro takovou konfiguraci délky *n*-gramů a sekvence očekávat, že každá pozorovaná variace *n*-gramu bude obsažena pouze jednou. Takový jev, kdy je potenciální slovník vůči délce testované sekvence natolik velký, že pravděpodobnost náhodného výběru kteréhokoliv ze slov je

blízká nule, nazveme jako jev kombinatorického nasycení. Zajímavé je, že vzhledem k exponenciální povaze velikosti potenciálního slovníku binárních slov tento jev nastane skutečně rychle, a to navíc v takovém rozsahu, že přestane být (za aktuálních podmínek) reálné vytvořit tak dlouhé sekvence, které by v průměru zopakovaly alespoň jediné z již realizovaných slov. Tento jev tedy vede od své kritické hodnoty (pro konečně dlouhé náhodné sekvence) k hodnotám TTR = 1, neboť každé ze slov je realizováno pouze jednou, což vede ke stejnému počtu tokenů a typů. Obdobně jako u jevu vyčerpání odvodíme kritickou hranici tohoto jevu, a to ve vztahu k délce sekvence i ve vztahu k délce slov.

Hranici jevu nasycení nejjednodušeji odvodíme tak, že pro zadanou délku slov n nalezneme takovou bitovou délku sekvence k, od které se již v průměrné náhodné sekvenci zopakuje alespoň jedno slovo. V tomto ohledu si je nutné dát pozor, neboť k samotnému jevu nasycení proto dochází v případě, kdy je sekvence kratší než vypočítaná hodnota. Pracovně takovou bitovou délku sekvence označíme jako **kritická hranice délky sekvence nasycení** a označíme ji jako $C(n)$. Její odvození je opět jednodušší v případě, kdy na délku sekvence nahlédneme jako na počet slov textu o délce n, čímž vypočítáme kritickou délku textu $C'(n)$. Úloha se tímto kognitivně zjednoduší na problém analogický k hodu kostkou o počtu stran $V(n)$ a otázce, kolik hodů je v průměru nutné provést do zopakování alespoň jedné ze stran. Počet předpokládaných hodů odvodíme rekurentním vztahem $E(r)$:

$$E(r) = 1 + \frac{r}{V(n)} E(r-1)$$
$$E(0) = 1$$

kde r je na začátku rovno $V(n)$. Platí tedy:

$$C'(n) = E(\,V(n)\,)\ .$$

Rekurentní zápis lze převést na následující analytické řešení:

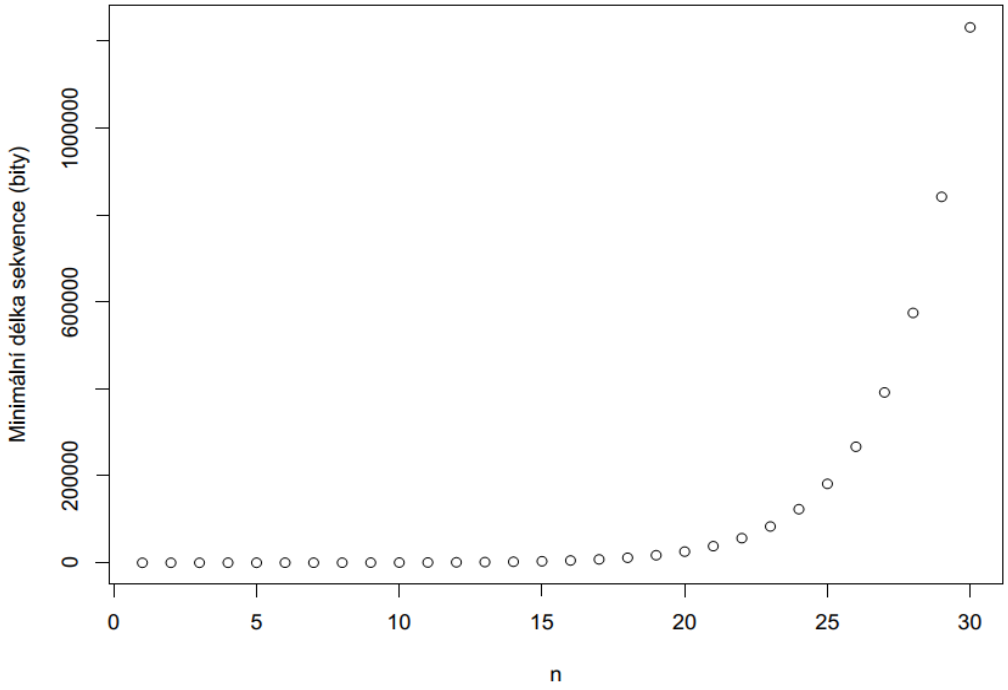
$$C'(n) = e^V\, V^{-V}\, \Gamma(V+1, V) = e^V\, V \int_1^\infty \frac{e^{-Vx}}{x^{-V}}\, dx = \int_0^\infty \left(\frac{V+x}{V}\right)^V e^{-x} dx$$

kde V je pro přehlednost substitucí $V(n)$ a $\Gamma(a,x)$ je nekompletní gama funkce definovaná dle Abramowitz a Stegun (1972, 260) jako $\Gamma(a,x) = \int_x^\infty t^{a-1}\, e^{-t}\, dt$. Při implementaci je nicméně vhodné použít uvedenou rekurentní verzi výpočtu. Z kritické délky textu ve slovech $C'(n)$ nyní můžeme odvodit tuto délku v bitech pomocí (11):

$$C(n) = n * C'(n) \qquad (11)$$

Pomocí (11) můžeme nyní nahlédnout na vztah minimální kritické délky sekvence v bitech, ve které se zopakuje v průměru alespoň jedno slovo a délky slov n, kdy pro takovou sekvenci bude v průměru platit $TTR < 1$. Výsledky aplikace (11)

na délky n-gramů n od 1 do 30 vidíme v grafu 5. Ose x v tomto grafu odpovídá velikosti n-gramu a ose y odpovídá délka sekvence v bitech. Z tohoto vztahu zjišťujeme, že pro zopakování alespoň jediného slova o délce n je opět vyžadována exponenciálně rostoucí délka sekvence. Například pro průměrné zopakování alespoň jediného libovolného 25-gramu, s potenciálním slovníkem $2^{25} = 33\,554\,432$ slov, potřebujeme v průměru délku sekvence minimálně 181 500 bitů neboli 7 260 slov. Pro zopakování alespoň jediného 30-gramu je v průměru nutné mít sekvenci alespoň o 1 232 078 bitů, tj. 41 069 slov atd., na konkrétní hodnoty minimálních délek sekvencí se můžeme podívat do tabulky 7.



***Graf 5**: Průměrná minimální délka sekvencí v bitech tak, aby se v nich zopakoval alespoň jeden n-gram.*

n	⌈C(n)⌉	n	⌈C(n)⌉	n	⌈C(n)⌉
1	3	**11**	632	**21**	38 129
2	7	**12**	971	**22**	56 484
3	13	**13**	1 484	**23**	83 506
4	23	**14**	2 256	**24**	123 222
5	39	**15**	3 414	**25**	181 516
6	65	**16**	5 145	**26**	266 964
7	104	**17**	7 726	**27**	392 057
8	166	**18**	11 563	**28**	574 980
9	262	**19**	17 256	**29**	842 176
10	408	**20**	25 682	**30**	1 232 078

***Tabulka 7**: Vypočítané průměrné délky sekvencí, ve kterých se zopakuje alespoň jeden z n-gramů.*

Z odvozené průměrné délky náhodných sekvencí $C(n)$, ve kterých se zopakuje alespoň jediný n-gram o délce n pak vyplývá jednoduché pravidlo: Pokud je bitová délka k sekvence S kratší, než je kritická délka sekvence pro zopakování alespoň jediného n-gramu, **tj. platí $k < C(n)$, pak pro sekvenci S dojde v průměru náhodných sekvencí k jevu nasycení.** Sekvence S má v takovém případě oproti velikosti potenciálního slovníku nedostatek kapacity, aby se v průměru náhodných výběrů zopakovalo libovolné ze slov. To v důsledku znamená, že pro takové případy budeme očekávat hodnoty TTR = 1.

Nyní máme k dispozici vše, abychom mohli – analogicky k intervalu vyčerpání – definovat i interval nasycení, tj. definovat interval délek slov n, u kterých pro zadanou délku náhodných sekvencí k dojde k jevu nasycení neboli nepravděpodobnému zopakování jediného slova, a tedy k předpokládané hodnotě TTR rovno jedné, neboť všechna slova budou nejpravděpodobněji unikátní. K definici takového intervalu stačí identifikovat nejmenší délku slov n, pro kterou platí pravidlo jevu nasycení, tj. $k < C(n)$. Jakmile takové pravidlo platí, víme, že délka slov n vyžaduje pro zopakování alespoň jediného slova delší sekvenci, než která je právě k dispozici. Například, pokud máme sekvenci o bitové délce $k = 6\,000$, pak k zopakování alespoň jediného slova o délce $n = 26$ je potřeba 5 145 bitů. V takovém případě očekáváme, že hodnoty TTR budou menší než jedna. Naopak pro náhodné zopakování alespoň jediného slova o délce $n = 27$ už je v průměru vyžadována délka sekvence alespoň 7 726 bitů. Hodnota $n = 27$ je v tomto případě začátkem intervalu nasycení, který pokračuje až do nekonečna, od této hodnoty očekáváme v průměru TTR = 1. Pro zadanou délku sekvence k proto odvodíme počátek intervalu nasycení $c(k)$ jako takovou délku slov n, pro které jako první začne platit pravidlo nasycení, tj. bude platit $k < C(n)$, dle (12):

$$c(k) = \arg\min_{n \in \mathbb{N}} \; k < C(n) \quad . \tag{12}$$

Finálně můžeme **definovat interval jevu nasycení** $\mathbb{C}$ jako:

$$\mathbb{C}_k = \langle c(k); +\infty)$$

U tohoto intervalu můžeme pro náhodné sekvence očekávat hodnoty TTR = 1 a nulový rozdíl od modelu TTR_{avg}. Je však nutné podotknout, že tyto hodnoty jsou v pravém smyslu očekávány a ve skutečnosti budou k těmto hodnotám postupně konvergovat. Tento jev a interval v úvodním grafu 2 vysvětlují shodné průběhy křivek náhodných sekvencí na intervalu $n \geq 17$.

Nyní máme k dispozici vzorec (12) pro výpočet kritické velikosti slova $n = c(k)$, od které již dochází pro sekvenci o délce k bitů k jevu nasycení a hodnotám $E(TTR) = 1$. Dále disponujeme vzorcem (10) pro výpočet maximální délky slov $n = q(k)$, pro kterou jako maximální dojde k jevu vyčerpání. Přitom platí $q(k) < c(k)$. Zbývá tedy definovat interval, který leží právě mezi těmito dvěma hraničními délkami slov a kterého jsme si všimli už při popisu grafu 3.

Jev kombinatorické harmonie

Na základě obou výše uvedených jevů nyní pro sekvenci S o délce k bitů dokážeme pro náhodné sekvence stanovit dva velmi důležité intervaly: interval vyčerpání a interval nasycení. Na základě jevu vyčerpání dokážeme stanovit interval délek slov $n = \langle 1;\ q(k)\rangle$, ve kterém v průměrné náhodné sekvenci dochází k využití a opakování slov celého slovníku. Důvodem takového jevu je nízká mohutnost potenciálního slovníku dané délky n-gramů oproti velikosti sekvence. Z tohoto důvodu neočekáváme, že by měly být mezi náhodnými sekvencemi a modelem TTR_{avg} rozdíly. Na základě jevu nasycení dále dokážeme stanovit interval $\langle c(k); +\infty)$, ve kterém rovněž neočekáváme pro náhodné sekvence přílišné rozdíly od modelu TTR_{avg}, a to vzhledem k disproporčně velkému potenciálnímu slovníku, který v omezeně dlouhých sekvencích v průměru neumožní náhodně zopakovat kterékoliv ze slov. Mezi oběma extrémními intervaly je ovšem třetí interval $\langle q(k)+1; c(k)-1\rangle$, za předpokladu, že $q(k)+1 < c(k)-1$, který popisuje jisté ekvilibrium mezi velikostí potenciálního slovníku a délkou sekvence, či **pracovně řečeno, vystihuje určitý jev *harmonie***. Slovník tvořený délkami slov z tohoto intervalu není natolik velký, aby z něj nebylo *možné* znovu náhodně vybrat kterékoliv z už vybraných slov, a zároveň není ani dost malý na to, aby z něj s jistotou byla použita všechna slova. Právě zde v tomto intervalu může docházet k největším výkyvům naměřených hodnot TTR. Pro délky slov z tohoto intervalu proto u náhodných sekvencí očekáváme nejvyšší varianci rozdílů změřených hodnot TTR od modelu TTR_{avg} (8). Samotný interval harmonie $\mathbb{H}$ pak definujeme jako:

$$\mathbb{H}_k = \langle q(k)+1; c(k)-1\rangle$$

Tři uvedené modelové intervaly, které zde **pracovně pojmenujme jako model QHC** (dle zvolených jmen funkcí a intervalů) můžeme pro sekvenci S o délce k bitů, maximální testovanou délku n-gramů $Z \in \mathbb{N}$ a pozorované hodnoty TTR náhodných sekvencí pro délku slov n, tj. $TTR(n)$, shrnout do tabulky 8.

Jméno intervalu	Označení	Interval	$\mathbf{E}(\boldsymbol{TTR(n) - TTR_{avg}(n)})$
Vyčerpání	$\mathbb{Q}$	$\langle 1; q(k)\rangle$	0
Harmonie	$\mathbb{H}$	$\langle q(k)+1; c(k)-1\rangle$	$0 \le x \le 1, x \in \mathbb{R}$
Nasycení	$\mathbb{C}$	$\langle c(k); Z\rangle$	$\lim_{n\to+\infty} E = 0$

***Tabulka 8**: Shrnutí modelových intervalů délek n-gramů QHC v závislosti na bitové délce sekvence*

Pro přehlednost uveďme i tabulku 9 s využitými vzorci, jejich referencemi a popisem.

Jev vyčerpání		
Reference	Značení	Výsledek
-	Q'(n)	*Velikost náhodné sekvence ve slovech o délce n, u které dojde k využití všech slov z potenciálního slovníku.*
(9)	Q(n)	*Délka náhodné sekvence v bitech, u které dojde k využití všech slov z potenciálního slovníku.*
(10)	q(k)	*Maximální délka n-gramu (slova) pro sekvence o bitové délce k, u které jako maximální ještě dojde k jevu vyčerpání.*
Jev nasycení		
Reference	Značení	Výsledek
-	C'(n)	*Minimální velikost náhodné sekvence ve slovech o délce n, pro kterou nedojde k jevu nasycení (je zde v průměrné náhodné sekvenci zopakováno alespoň jedno ze slov).*
(11)	C(n)	*Minimální velikost náhodné sekvence v bitech, pro kterou nedojde k jevu nasycení n-gramy o délce n.*
(12)	c(k)	*Minimální délka slova pro náhodnou sekvenci o délce k bitů, od které dojde k jevu nasycení (tj. k postupné snižování pravděpodobnosti zopakování slov).*

***Tabulka 9**: Shrnutí formálních modelů QHC včetně referencí*

Prozkoumání a odvození intervalů QHC pro náhodné sekvence nám umožnilo nahlédnout na fungování metody MKM, odhalit konkrétní vlastnosti náhodných sekvencí a určit jejich prvotní charakteristiky, které dále rozvedeme do statistického testu náhodnosti sekvencí. Náhled na tyto intervaly, především na interval vyčerpání, nám poskytl i další důležitý poznatek o samotné metodě MKM, a to o existujícím artefaktu vyvstávajícím z principu normalizace abecedy, kdy shodné průběhy sekvencí přirozeného jazyka a náhodných sekvencí pro nízké délky *n*-gramů nemůžeme bez další analýzy považovat za projev náhodnosti. Dále tedy nalezené intervaly a poznatky získané z jejich analýzy použijeme k definici statistického testu, jehož cílem bude otestovat a zjistit, zda je zadaná sekvence náhodná či nikoliv.

Statistický test nahodilosti pomocí konfidenčních intervalů

Prozatím jsme mohli náhodnost sekvencí hodnotit kvalitativně na základě grafu nebo vyhodnocením výsledků vektoru TTR na základě očekávaných charakteristik z tabulky 8. Tato hodnocení však nejsou exaktně uchopitelná. Odvodíme zde proto statistický test odvíjející se od modelu náhodných sekvencí TTR_{avg} (8) a z něj odvozených konfidenčních intervalů pro každou testovanou délku slov *n* tak, aby bylo možné říci, s jakou pravděpodobností je každý partikulární výsledek TTR jednotlivých délek *n*-gramů testované sekvence náhodný. Každé hodnotě TTR tedy pro konkrétní délku slova *n* a délku sekvence stanovíme interval, ve které se ještě může tato hodnota pohybovat s arbitrárně zvolenou jistotou (typicky 95 % a 99 %; např. NIST v Rukhin *et al*. 2001, 1-4 uvádí 99 % až 99,9 %). Pro každou délku slova zadané sekvence jinými slovy

odvodíme konfidenční interval o zvolené šířce, pomocí kterého rozhodneme, zda je výběr opakování slov v sekvenci dílem náhody nebo jde o projev systematičnosti. Abychom tyto konfidenční intervaly mohli odvodit a následně na nich vystavět test, potřebujeme nejprve definovat několik základních výpočtů.

Prvním z takových výpočtů je odvození pravděpodobnosti výskytu daného počtu typů *V* (neboli počtu různých slov) pro náhodnou sekvenci dlouhou *k* bitů, čítající $K = D(n,k)$ slov o délce *n* a velikostí potenciálního slovníku $N = V(n)$. Na výpočet takové pravděpodobnosti můžeme nahlédnout méně náročným způsobem jako na výpočet pravděpodobnosti hodu právě *V* různých stran kostky o *N* stranách po *K* hodech, kterou můžeme vypočítat jako (13a; Riedel 2018, upraveno):

$$\begin{aligned} p(K,N,V) &= \frac{1}{N^K}\binom{N}{V} V! \left\{ {K \atop V} \right\} \\ &= \frac{1}{N^K} \left\{ {K \atop V} \right\} \prod_{z=N-V+1}^{N} z \\ &= \frac{1}{N^K}\left(\prod_{z=N-V+1}^{N} z \right) \sum_{i=0}^{V} (-1)^i \frac{(V-i)^K}{i!\,(V-i)!} \end{aligned} \tag{13a}$$

kde *K* je počet slov, *N* velikost potenciálního slovníku, *V* požadovaný počet typů, $\binom{N}{V}$ kombinační číslo a $\left\{ {K \atop V} \right\}$ Stirlingovo číslo druhého řádu. Aplikací (13a) získáme pro každou délku slov *n* a délku sekvencí *k* pravděpodobnost, s jakou by se zadaný počet různých slov (typů) objevil v dokonale náhodné sekvenci se všemi zmíněnými specifiky. Aplikace (13a) má nicméně pro reálné využití kritickou nevýhodu, a to vzhledem k využití faktoriálu proměnné *N* odpovídající exponenciálně se zvětšující velikosti potenciálního slovníku: Už jen uvažovaná délka *n*-gramu $n = 7$ vytváří potenciální slovník o velikosti $N = 128$ slov, přitom ale faktoriál této hodnoty, tj. $N! = 128! = 3{,}8 \times 10^{215}$, je tak velké číslo, že znemožňuje další výpočty. Vzhledem k uvažovanému testování *n*-gramů o délce 30 i více je výpočet pravděpodobnosti (13a), i přes snahu směřující k eliminaci faktoriálů, neprůchozí. V tomto ohledu sice lze uvážit za předem daných hodnot *V* a *K* aproximaci sumy, ta však v ohledu tvorby statistického testu není příliš výhodná. Alternativou (13a) je rekurentní tvar (13b) odvozený Jiřím Miličkou v Matlach *et al*. 2018, který problém faktoriálů pro shodné proměnné z implementačního hlediska obchází:

$$\begin{aligned} &p(K,N,V) = p(K-1,N,V)\frac{V}{N} + p(K-1,N,V-1)\left(1 - \frac{V-1}{N}\right) \\ &p(1,N,1) \;= 1 \\ &p(K,N,0) \;= 0 \quad kde\ K > 0 \\ &p(0,N,V) \;= 0 \end{aligned} \tag{13b}$$

Využití (13b) tak bezpečně vede k získání pravděpodobnosti toho, že zadaná sekvence o K slovech bude obsahovat právě V typů z N možných. To v důsledku znamená, že (13a, b) je funkcí diskrétního pravděpodobnostního rozdělení počtu typů V pro konkrétní konfiguraci délky slova a délky sekvence.[5] Na základě principu odvození tohoto výpočtu v Matlach *et al*. 2018 můžeme takové rozdělení považovat za unimodální, beta-binomiální rozdělení, což nám umožňuje využít (13a,b) k definování konfidenčních intervalů pro jednotlivé velikosti slov n a délky sekvencí k. Konfidenční interval počtu typů náhodné sekvence pro **délku slov n, délku sekvence v bitech k** a **hladinou významnosti α** (typicky 5 % nebo 1 %) definujeme za využití (5), (6) a (13a,b) jako (13 CI):

$$CI_{Vavg}(n,k,\alpha) = \langle L;\ R\rangle \qquad \text{(13 CI)}$$

$$K = D(n,k)$$
$$N = V(n)$$
$$\mathrm{N_{max}} = \min(\mathrm{K},\mathrm{N})$$

kde pro $L, R \in \mathbb{N}$ platí $L \leq R \leq N_{max}$ a zároveň platí:

$$\sum_{v=1}^{L} p(K,N,v) \approx \alpha/2$$

$$\sum_{L\leq\, v\, \leq N_{max}}^{R} p(K,N,v) \approx \alpha/2$$

Jinými slovy (13 CI) odvozuje $100(1-\alpha)\%$ konfidenční interval na základě identifikace levé a pravé strany rozdělení tak, že jejich (zleva a zprava) kumulované pravděpodobnosti aproximují hodnotu $\alpha/2$. Aproximace hodnoty $\alpha/2$ je vzhledem k diskrétnímu pravděpodobnostnímu rozložení (13a,b) nutná. To v důsledku znamená, že nalezený interval nemusí být vždy přesně roven $1-\alpha$. Výpočet konfidenčního intervalu (13 CI) využijeme k vytvoření statistického testu náhodnosti neznámé sekvence. Předtím však připomeňme nejprve, co nám konfidenční interval vypočítaný dle (13 CI) vlastně říká.

[5] Zde je zajímavé poznamenat, že zobecněním užitého faktorialu na funkci gama a převedením sumy na integraci v (13a) získáme spojitou funkci, tedy funkci hustoty pravděpodobnosti p pro $K, N, V \in \mathbb{R}^+$:

$$p = \frac{1}{N^K}\frac{\Gamma(N+1)}{\Gamma(N-V+1)}\int_0^V (-1)^i \frac{(V-i)^K}{\Gamma(i+1)\Gamma(V-i+1)}\,di$$

Interval odvozený pomocí (13 CI) nám říká rozmezí, ve kterém se bude v případě skutečně náhodných sekvencí pohybovat právě $100(1-\alpha)$ % realizovaných různých slov délky n. To znamená, že pro zadanou délku sekvence a každou délku slova n dokážeme odvodit interval, do kterého bude svým počtem typů spadat, například pro $\alpha = 0{,}05$ právě 95 % všech náhodných sekvencí. Zda je neznámá sekvence S náhodná tak můžeme otestovat tak, že pro zadanou hladinu významnosti α a všechny délky n-gramů n od 1 do zvoleného maxima $Z \in \mathbb{N}$ ověříme, zda zjištěný počet typů o délce n spadá do vypočítaného intervalu, do kterého by mělo spadat $100(1-\alpha)$% všech náhodných sekvencí. Abychom tedy o sekvenci S o délce k bitů mohli říci, že je náhodná, musí pro každou testovanou délku slov n od 1 do $Z \in \mathbb{N}$ platit, že počet jejich typů v_n náleží do vypočítaného konfidenčního intervalu dané délky slov dle (13 CI):

$$v_n \in CI_{Vavg}(n, k, \alpha)$$

Takový test má však závažný teoretický problém. Hladina α určuje šířku konfidenčního intervalu, do kterého by mělo spadat právě $100(1-\alpha)$ % všech skutečně náhodných sekvencí. V případě volby $\alpha = 0{,}05$ to znamená, že do vypočítaného konfidenčního intervalu bude spadat právě 95 % všech skutečně náhodných sekvencí a 5 % bude mimo tento interval, což dle testu výše znamená, že budou automaticky považovány za nenáhodné. V konkrétních číslech to znamená, že ze 100 skutečně náhodných sekvencí bude v průměru 5 z nich chybně označeno za nenáhodné. Tato situace je mnohem horší, jakmile si uvědomíme, že test s pravděpodobností chybného odmítnutí α provádíme v rámci jediné sekvence tolikrát, kolik testujeme délek slov n, tj. Z. To znamená, že u jediné sekvence S, u které testujeme délky slov n od *1* do *100* s $\alpha = 0{,}05$, dojde v průměru k chybnému odmítnutí 5 partikulárních testů ze 100. Jak bylo definováno výše, nesplnění jediného z těchto partikulárních testů znamená odmítnutí sekvence jako nenáhodné. Pokud dopředu víme, že při testu jediné skutečně náhodné sekvence délkami slov od 1 do 100 neprojde přibližně 5 ze 100 testů, nemůže takovým testem projít téměř žádná sekvence, nezávisle na jejich náhodnosti. Pravděpodobnost, že by skutečně náhodná sekvence sérií těchto testů prošla, je pro maximální velikost testovaných slov $Z \in \mathbb{N}$ rovna (za předpokladu, že jsou tyto testy na sobě nezávislé) hodnotě p_{accept}:

$$p_{accept}(\alpha, Z) = \prod_{i=1}^{Z} 1 - \alpha = (1-\alpha)^Z \quad .$$

Pro uvedený příklad je tato pravděpodobnost rovna $p_{accept}(0{,}05;\ 100) = \prod_{i=1}^{100} = 0{,}592$ %, což paradoxně znamená, že s tímto nastavením projde z 1 000 skutečně náhodných sekvencí testem náhodnosti přibližně jen 6 z nich. Je však nutné si uvědomit, že hladina významnosti α platí pro partikulární testy, a ne pro jejich sérii. V takovém

případě můžeme provést korekci hladiny α jejím snížením, a to v závislosti na počtu těchto partikulárních testů. Snížení hladiny α ovšem znamená rozšíření konfidenčního intervalu, kterým se vymezujeme vůči nenáhodným sekvencím. Do širšího konfidenčního intervalu mohou s vyšší pravděpodobností spadat i nenáhodné sekvence, které bychom následně chybně klasifikovali za náhodné. Vhodné je proto hodnotu α snížit o co nejmenší hodnotu tak, aby umožnila v lepším poměru procházet skutečně náhodné sekvence a aby zároveň byla stále dostatečně striktní proti nenáhodným sekvencím.

První korekcí, kterou pro hladinu α definujeme, je její proporční snížení vůči počtu jednotlivých testů daných počtem testovaných délek slov Z. Novou hodnotu α' můžeme vypočítat pomocí Bonferroniho korekce jako:

$$\alpha_{Bonferroni} = \frac{\alpha}{Z}$$

Abychom nicméně hladinu α snížili Bonferroniho korekcí co nejméně, můžeme využít popsaného jevu vyčerpání (viz tabulka 8). U sekvencí o bitové délce k a délce slov n spadajících do intervalu vyčerpání $\mathbb{Q}_k = \langle 1; q(k) \rangle$ neočekáváme, skrze popsanou kombinatoriku, jakoukoliv odlišnost od modelu TTR_{avg}. To znamená, že pro $n \in \mathbb{Q}_k$ očekáváme $p\big(D(n,k), V(n), V(n)\big) \approx 1$, neboli, že jistým počtem typů je v takových případech velikostí slov sama velikost potenciálního slovníku $V(n)$. Důsledkem je, že pro tyto délky slov očekáváme *konfidenční intervaly* s jediným prvkem, a to prvkem $V(n)$, což znamená, že v těchto případech nejde o testování konfidenčních intervalů, ale pouze o test rovnosti pozorovaného počtu typů s počtem získaných modelem dokonale náhodné sekvence TTR_{avg}. Z tohoto důvodu můžeme délky slov $n \in \mathbb{Q}_k$ při úpravě hladiny α Bonferroniho korekcí vynechat a nesnižovat ji více, než je skutečně nutné. Namísto hodnoty Z proto při Bonferroniho korekci využijeme pouze počet testů, které skutečně využívají konfidenční intervaly, tj. počet M:

$$M = Z - q(k) \quad .$$

Finálně můžeme hodnotu α pro sekvence o bitové délce k a maximální délce testovaných slov Z přepočítat na novou hodnotu α' následovně:

$$\alpha' = \frac{\alpha}{Z - q(k)}$$

Například, pro sekvenci o délce $k = 1\,000$ bitů, maximální délku n-gramů $Z = 100$ a hladinu významnosti $\alpha = 0{,}05$, vypočítáme upravenou hladinu α' následovně:

$$\alpha' = \frac{0{,}05}{100 - q(1000)} = \frac{0{,}05}{100 - 4} = 0{,}00052 \quad .$$

Pravděpodobnost přijetí skutečně náhodné sekvence testem složeným ze 100 partikulárních testů, ve kterých $q(1000) = 4$ z nich odpovídají 100% konfidenčním intervalům a zbylých 96 odpovídá $1 - \alpha' = 1 - 0{,}00052 = 99{,}948\%$ konfidenčním intervalům, je ve výsledku rovna $\prod_{i=1}^{96} 1 - 0{,}00052 = 95{,}12$ %, tj. téměř takové pravděpodobnosti, kterou bychom dle definice hladiny α pro test očekávali. Abychom se k takové pravděpodobnosti dostali, bylo nutné rozšířit konfidenční intervaly pro jednotlivé délky testovaných slov. Díky poznatkům z jevu vyčerpání však toto rozšíření nebylo provedeno v plné naivní výši, ale bylo částečně omezeno.

Výše navržený statistický test náhodnosti sekvencí založený na metodě MKM zde pracovně pojmenujme jako **test konfidenčními intervaly.** Hlavní výhodou tohoto testu je existence kritéria specifikující náhodnost pro každou testovanou délku slov n. Tato výhoda je však vykoupena nutnou korekcí hladiny významnosti α, která vede ke zvýšené pravděpodobnosti označení nenáhodné sekvence jako náhodné. Další nevýhodou tohoto testu je náročnost samotných výpočtů. Pro každou testovanou délku slov je třeba vypočítat konfidenční interval, přitom výpočet každého z nich pomocí (13) je relativně náročný, především pro vysoké hodnoty počtu slov K a testované velikosti slovníku V. Pro praktické užití této metody je proto vhodné uvážit následující tři pragmatické kroky. Prvním a druhým krokem je možnost využít obecnosti vypočítaných konfidenčních intervalů pro bitovou velikost sekvence a velikost slov a vypočítané intervaly pro další použití ukládat do tabulky. Tento krok je výhodný i v případě výpočtu zcela nových intervalů, neboť rekurentní zápis (13b) může využít již předpočítaných hodnot nižších počtů slov a délek sekvencí.[6] Třetím pragmatickým krokem implementace testu je heuristické upřednostnění těch partikulárních testů, které mají nejvyšší pravděpodobnost vyřazení sekvence jako nenáhodné, tj. testovat nejprve velikosti n-gramů z intervalu harmonie $\mathbb{H}$, ve kterém očekáváme nejvyšší varianci od modelu TTR_{avg}. Tímto krokem maximalizujeme pravděpodobnost brzkého vyřazení sekvence využitím co nejmenšího počtu kroků.

Test náhodnosti sekvencí nyní vyzkoušíme, včetně náhledu na již dosažené poznatky, na vzorcích náhodných sekvencí získaných z různých zdrojů a na kontrolních vzorcích sekvencí přirozeného jazyka. Jak dále uvidíme, z této experimentální aplikace a testování vyvstanou další poznatky o metodě MKM.

[6] Konkrétní implementace viz datová příloha: Skripty/MKM/KRITICKE_HRANICE.R.

Experimentální analýza náhodných sekvencí

Metoda MKM umožňuje analyzovat libovolné druhy sekvencí s cílem je charakterizovat a pomoci tak určit typ zdroje, ze kterého mohou pocházet. V předchozí podkapitole jsme se setkali s analýzou metody MKM v kontextu náhodných sekvencí a vlastností, které od takových sekvencí můžeme očekávat. Získané poznatky jsme následně formalizovali a ve výsledku použili pro tvorbu statistického testu náhodnosti i k popisu artefaktů vznikajících samotnou metodou. V této podkapitole se budeme věnovat experimentální aplikaci metody MKM na náhodné sekvence pocházející z různých zdrojů entropie, které následně otestujeme výše odvozeným testem. Testování těchto zdrojů nás dovede i k důvodům jejich použití, a tedy i náhledu na jejich skutečnou důležitost v praxi.

Zajímavost náhodných sekvencí vyvstává především ze dvou důvodů. Prvním je nesnadnost generování skutečně náhodných sekvencí a druhým je pak důležitost jejich využití v praxi. Generování náhodných čísel se může jevit jako prostý úkon i pro člověka, nicméně výzkumy uvádí, že lidé jsou při vědomém vytváření náhodných sekvencí dobře předvídatelní (Kneusel 2018, 11) a jak dále ukázala analýza *monkey-typed* textů výše, i texty psané libovolnými úhozy na klávesnici jsou předvídatelné s řadou vzorů reflektujících fyzické podmínky procesu psaní. Generování náhodných sekvencí počítačem je ještě problematičtější úlohou. Od elektroniky obecně vyžadujeme logické a deterministické chování, a jak trefně uvádí Haahr (2018), počítač, chovající se náhodně, považujeme za rozbitý. To v důsledku znamená, že tvorba náhodných sekvencí v počítači musí být založena na jasných logických krocích, tj. algoritmech, které sekvenci generují. Prvním takovým algoritmem, který přetrval dodnes, je lineární kongruentní generátor (LCG) a jeho varianta multiplikativní kongruentní generátor (MCG; detailně viz Law 2013, 397-409). Metoda LCG například spočívá v aplikaci rekurentního vzorce:

$$x_{i+1} = (ax_i + c) \bmod \mathrm{m}\,,$$

pomocí kterého je postupně generována sekvence čísel. Hodnoty *a*, *c* a *m* jsou zde zvolené konstanty splňující několik vzájemných pravidel (detailně viz Law 2013, 397). Hodnota x_0, však musí být nastavena na náhodnou hodnotu, od které se následně celá *náhodnost* vzorcem generované sekvence odvíjí. Takovou počáteční hodnotu označujeme jako *seed*. Je zřejmé, že sekvence generované tímto způsobem nejsou skutečně náhodné, ale pouze pseudonáhodné, protože aplikací algoritmu na stejné vstupní hodnoty získáme vždy stejný výsledek. To v důsledku znamená, že algoritmy pseudonáhodných generátorů čísel či sekvencí pro jejich fungování nutně vyžadují externí vstup hodnoty *seed*, a to ideálně ze skutečně náhodného zdroje entropie. Kvalita náhodných sekvencí generovaných pseudonáhodnými generátory (zkráceně PRNG) tak v důsledku závisí na kvalitě algoritmu a na kvalitě zdroje hodnoty *seed*. Jak ale uvidíme,

kvalitní zdroje entropie nejsou triviální a nejsou jednoduše dostupné. Jako snadno dostupné hodnoty *seed* se proto používají různé externí údaje jako je například čas, časové rozestupy mezi stisky kláves apod., což už předesílá jisté problémy se skutečnou náhodností výsledků.

Důležitost náhodných sekvencí tkví v jejich aplikacích. Klasickým příkladem je statistika a modelování jevů, kde lze některé komplexní jevy pro zjednodušení redukovat na zcela náhodné nebo je třeba zajistit skutečně náhodný výběr vzorku ad. Kritickou oblastí využití náhodných čísel a sekvencí je v dnešní době kryptografie, kterou denně a implicitně využíváme k ochraně soukromí, dat a identity v počítačích, mobilních telefonech a dalších zařízeních. S důležitostí skutečné nahodilosti jsme se již setkali v předešlé podkapitole u diskuze nad designem počítačových hesel, kdy využití skutečně náhodného hesla vede k irelevanci pragmatických heuristik jeho uhodnutí a k nutnosti využít časově náročný či ideálně přímo neudržitelný útok hrubou silou hádající všechny kombinace, neboť všechny možnosti jsou stejně pravděpodobné. Hesla jsou v tomto ohledu nicméně pouhou špičkou ledovce. Například autorizovaný přístup k webové stránce, tj. i přístup k online bankovnictví nebo emailu, je primárně založen na jednoznačné identifikaci uživatele. Uživatel se do služby přihlašuje svým přihlašovacím jménem a heslem, ale činí tak pro dané sezení typicky pouze jednou. Po přihlášení server (či služba) pro uživatele vygeneruje náhodný a „neuhodnutelný" klíč, označovaný jako *session token*, který předá (ideálně) v šifrované podobě zpět internetovému prohlížeči uživatele. Uživatel (respektive jeho prohlížeč) následně prokazuje serveru svou identitu pouhým předložením tohoto náhodného *session tokenu*, který by měl být neuhodnutelný. To znamená, že identita přihlášeného uživatele je dána právě náhodným klíčem a jeho neuhodnutelnost je zárukou autenticity uživatele. Kdokoliv pak může získat přístup do libovolného účtu libovolné služby jen tím, že uhádne vygenerovaný *session token*. Právě zde je zřejmé, že jakákoliv předvídatelnost při tvorbě *session tokenů* vede úměrně ke zvýšení pravděpodobnosti jejich uhádnutí, a tedy k potenciálnímu neautorizovanému přístupu k účtu. Uhodnutí *session tokenu* na základě nedostatečné náhodnosti generátoru je známý jev, viz např. Stuttard a Pinto 2011 (218-219), kdy prediktabilnost zdroje entropie (zdroje hodnoty *seed*) a znalost algoritmu generátoru vedly k predikci budoucích *session tokenů.* Stejně jako náhodná hesla slouží dokonale náhodné sekvence *session tokenů* ke zirelevantnění útoku využívající heuristiky založené na vzorech a vynucují tak neudržitelné prohledávání celého prostoru kombinací k jejich uhádnutí. Obdobný problém dále vzniká například při generování certifikátů a digitálních podpisů ověřujících jednotlivé strany komunikace nebo při užívání moderních kryptografických systémů pro šifrování dat, zpráv, emailů a dalších. Příkladem může být hybridní šifrování PGP vyvinuté Philipem Zimmermannem a jeho varianta Open GPG (blíže viz např. Kościelny *et al*. 2013, 147 a Callas *et al*. 2007). Principem PGP je využití kombinace symetrické a asymetrické šifry tak, aby byla zaručena bezpečnost šifrované zprávy a rychlost šifračního procesu.

Proces šifrování si ve zkratce ilustrujme. Zpráva *M* je pomocí PGP zašifrována dostatečně silnou symetrickou šifrou. Jako šifrační klíč slouží vygenerovaná náhodná sekvence o dostatečné fixní délce označená jako *session key*. *Session key* je následně sám zašifrován pomocí asymetrické šifry (veřejným klíčem adresáta) a přidán jako *příloha* k zašifrované zprávě *M'*. Samotná zpráva *M'* je tedy zašifrována pouze symetrickou šifrou na základě *náhodného* klíče, který vygeneroval počítač. V případě nedostatečné náhodnosti tohoto klíče získává útočník výhodu při jeho hádání stejně, jako je tomu u hesel a *session tokenů*. Použití náhodného klíče navíc brání tomu, aby byla zpráva *M* po zašifrování deterministickým algoritmem vždy stejná. V případě nedostatečného zdroje entropie by tak šifrování stejné zprávy *M* vedlo k prediktabilním výsledkům. Nabourání zdroje entropie uvnitř počítače proto může vést nejen k odposlouchávání šifrované komunikace, ale i k únosu účtů. Zájem o takové techniky ilustrují koncepty publikované v akademickém prostředí, viz např. Govindan *et al*. 2018. Zřejmě z těchto důvodů například nástroj Open GPG na systému Microsoft Windows definuje a využívá vlastní zdroj entropie těžící z řady systémových ukazatelů[7], např. množství volné paměti, velikosti fronty systémových zpráv, času spuštění samotné aplikace, statistiky síťového provozu, výkonu disku, stavu baterie a dalších. Tato data jsou postupně shromažďována a transformována hashovací funkcí. Data šifrovaná pomocí GPG budou dále součástí datasetu, který budeme metodou MKM zkoumat, abychom zjistili, zda se takto šifrovaná data budou jevit jako skutečně náhodná, bez metody viditelných vzorů.

Kromě řady systémových ukazatelů, které ovšem lze jistým způsobem považovat za nenáhodné, existuje řada externích alternativ specializovaných pouze na poskytování náhodných sekvencí založených na měření různých fyzikálních jevů. S jednou z těchto alternativ jsme se již setkali, a to se zdrojem entropie pocházejícím z atmosférického šumu služby RANDOM.ORG (Haahr 2018), který jsme použili jako úvodní ilustraci náhodných sekvencí v grafu 2. Sekvence získané z tohoto zdroje podrobíme analýze, tentokrát však s cílem otestovat jejich skutečnou náhodnost. Další zdroj entropie, který zde budeme testovat, je založen na metodě měření kvantové fluktuace vakua (zkráceně ANU QRN; Haw *et al*. 2015). Čtvrtým zdrojem je patentovaný způsob rngResearch (Koopman 1995), který zaznamenává chaotické proudění vzduchu mikrofonem. Dále existují i jiné typy zdrojů entropie, zejména pak sledování radioaktivního rozpadu (např. Alkassar *et al*. 2005) nebo sledování chaotických systémů jako např. lávových lamp metodou LavaRand (Noll *et al*. 1996) a mnoho dalších. Jednotlivé metody se mnohdy liší i způsobem práce s naměřenými surovými daty, které jsou doupravovány transformacemi, např. hashováním, s cílem vstupy nevratně promíchat, maskovat charakter původního zdroje entropie a dodat chybějící entropii

[7] Na základě zdrojového kódu knihovny libgcrypt, především funkce *_gcry_rndw32_gather_random_fast* a *slow_gatherer*, dostupné online: https://github.com/gpg/libgcrypt/blob/master/random/rndw32.c , cit 19.7.2018

v případě jevů inherentně produkujících např. více nul než jedniček (viz např. využití hashovací funkce MD5 u rngResearch, obecné hashovací funkce u LavaRand, funkce AES u ANU QRN atd.).

Pragmatickým problémem fyzických zdrojů entropie je získání vzorků náhodných sekvencí v dostatečném množství k testování pomocí metody MKM, a to z důvodů komercionalizace těchto zdrojů jako služeb poskytovaných například kasinům apod. Z tohoto důvodu byly vybrány 4 popsané zdroje sekvencí shrnuté v tabulce 10. Počet sekvencí byl z uvedených důvodů omezen na 200 pro každý zdroj a délka omezena na náhodně vybraných 6 000 bitů. Pro porovnání výsledků také otestujeme sekvence z pseudonáhodného multiplikativního kongruentního generátoru (MCG) nastaveného dle specifikace generátoru *RANDU* (tj. $m = 2^{31}$, $a = 65539$, $c = 0$) vytvářející nežádoucí vzory viditelné v případě trojrozměrné interpretace vygenerovaných dat (Law 2013, 400-416). Jako hodnota *seed* je pro tuto metodu zvolena časová známka tvorby sekvence. Jako kontrolní, nenáhodný vzorek použijeme sekvence pocházející ze 200 různých českých beletrií. Na každou uvedenou sekvenci aplikujeme metodu MKM a ověříme, zda a jak jsou tyto sekvence blízko modelu náhodných sekvencí TTR_{avg} (8) a zda budou odpovídat popsaným modelům intervalů vyčerpání, nasycení a harmonie. Model dokonale náhodných sekvencí TTR_{avg} nám bude sloužit jako základ, ke kterému budeme měřit rozdíl pozorované hodnoty *TTR(n)* jednotlivých sekvencí. Pro náhodné sekvence tedy očekáváme rozdíly blízké nule.

Typ	Zdroj	Zdroj entropie	Počet sekvencí
Náhodná data	RANDOM.ORG[8]	Atmosferický šum	200
Náhodná data	rngResearch[9]	Turbulence vzduchu, komprese	200
Náhodná data	ANU Quantum Random Numbers[10]	Kvantová fluktuace vakua	200
Šifrovaná data GNU GPG (OS Windows)	Jediný soubor[11]	Ukazatele systému Windows, hashování	200
Pseudonáhodná data	MCG	Aktuální čas	200
Přirozený jazyk	Beletrie CZ	-	200

***Tabulka 10**: Zdroje dat určených k testování metody MKM a odvozeného testu náhodnosti.[12]*

První charakteristikou, na kterou se u jednotlivých sekvencí zaměříme, jsou intervaly vyčerpání, nasycení a z nich vyplývající interval harmonie. Jednotlivé intervaly můžeme stanovit dle tabulek 6 a 7 pro délky sekvencí $k = 6\,000$:

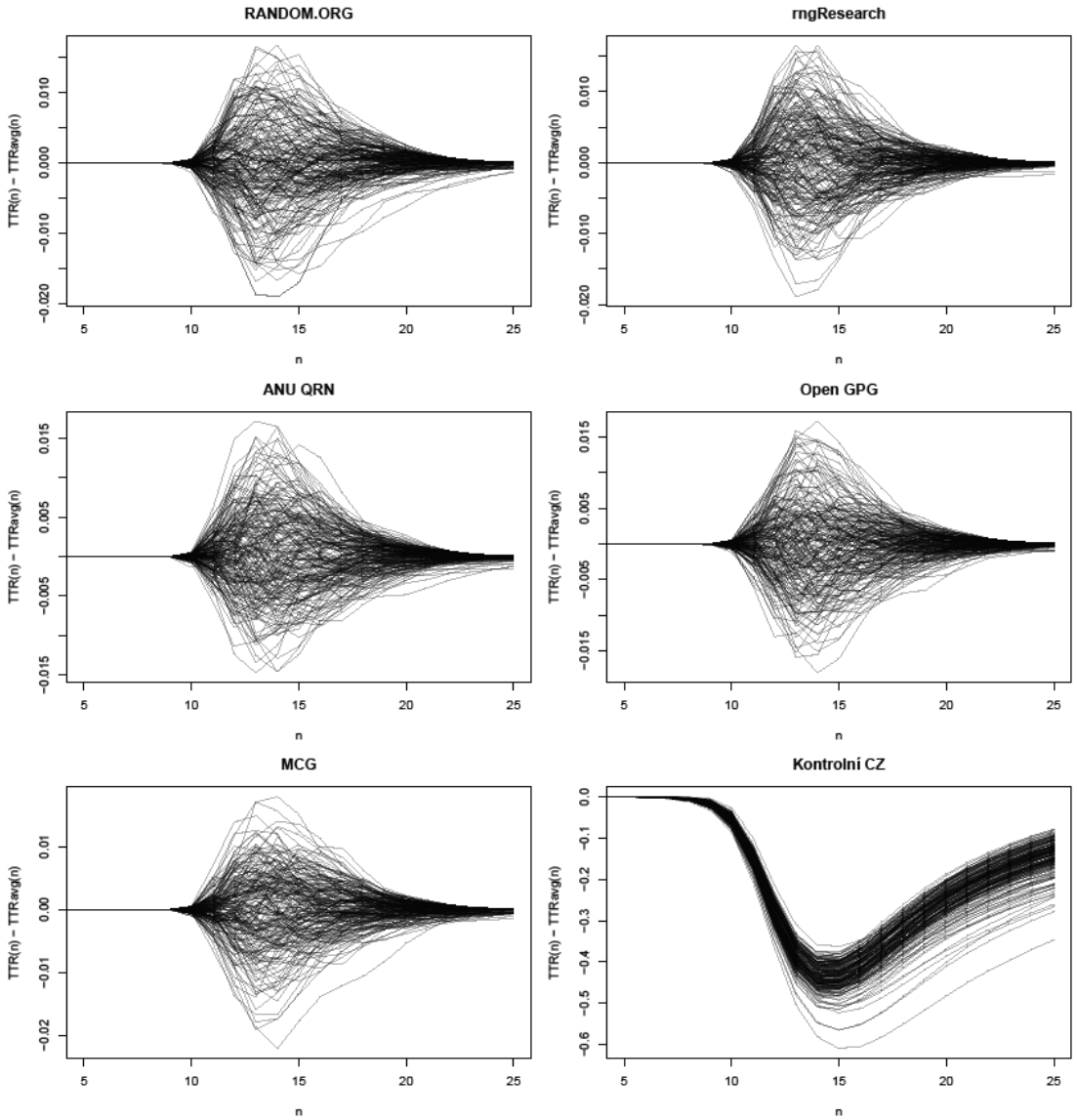
$$q(k) = q(6000) = 7 \quad ,$$
$$c(k) = c(6000) = 17 \ ,$$
$$\mathbb{Q}_{6000} = \langle 1; 7\rangle, \quad \mathbb{H}_{6000} = \langle 8; 16\rangle, \quad \mathbb{C}_{6000} = \langle 17; +\infty\rangle \quad .$$

[8] Data dostupná online archive.random.org/binary, cit. 12. 7.2018
[9] Data dostupná online rngresearch.com/download, cit. 12. 7. 2018
[10] Data dostupná online qrng.anu.edu.au, cit 12.7. 2018
[11] Tento jediný soubor byl šifrován hromadně.
[12] Veškeré testované sekvence jsou dostupné v datové příloze: Sekvence.

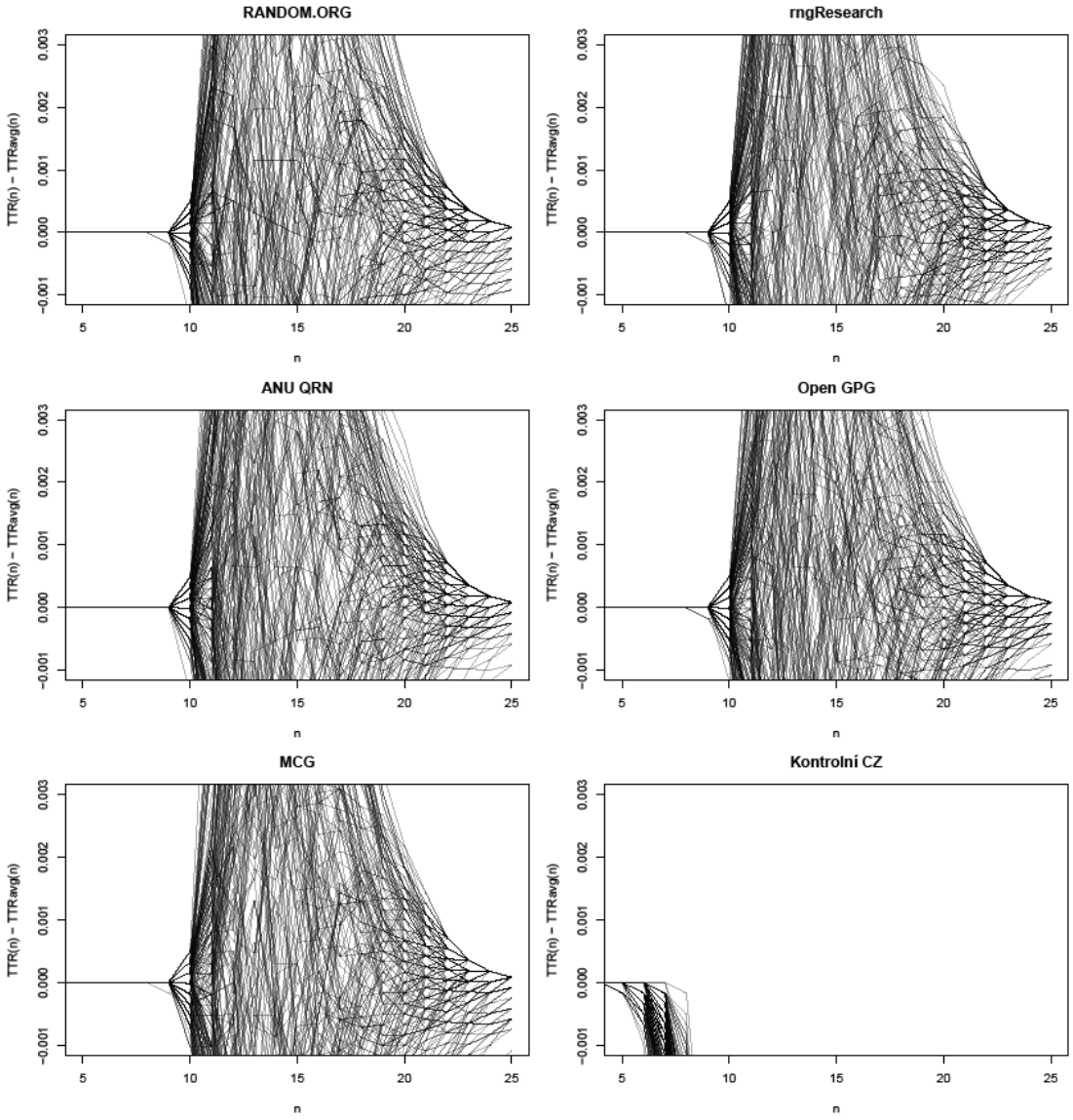
Na základě charakteristik jednotlivých intervalů popsaných výše můžeme odvodit následující: Pro velikosti n-gramů z intervalu vyčerpání, tj. $n \in \mathbb{Q}_{6000}$, očekáváme vyčerpání slovníku vůči délce sekvence – pro tento interval by neměl existovat rozdíl mezi modelem TTR_{avg} a testovanými náhodnými sekvencemi. Na intervalu harmonie, tj. $n \in \mathbb{H}_{6000}$, uvažujeme největší varianci rozdílů naměřených hodnot TTR od modelu, neboť zde je možné plně využít náhodný výběr a diverzifikovat sekvenci. Na intervalu nasycení, tj. $n \in \mathbb{C}_{6000}$ následně pro náhodné sekvence uvažujeme postupné snižování variance rozdílů – v tomto intervalu očekávanou postupnou konvergenci k modelu.

Na výsledky rozdílů jednotlivých typů sekvencí z tabulky 10 od modelu TTR_{avg} nyní nahlédneme v grafu 6. Osa x zde odpovídá velikosti slov n (tj. velikosti n-gramů) a osa y odpovídá rozdílu mezi pozorovaným TTR a modelem TTR_{avg}. V případě, že by sekvence byla dokonale totožná s modelem, byla by zobrazena pouze jako úsečka s hodnotou $y = 0$. Linky sekvencí jsou v grafu vykresleny s 1/3 průhledností a jejich překryv tak vytváří tmavší barvu indikující častěji realizovaný průběh. V samotném obsahu grafu si můžeme všimnout zejména obdobného tvaru shluku náhodných a pseudonáhodných sekvencí, které se jím zcela odlišují od kontrolního vzorku sekvencí přirozeného jazyka, který se od modelu rychle vzdálí. Pro všechny typy sekvencí je však zřetelná jejich počáteční blízkost k modelu TTR_{avg}, což je dáno jevem vyčerpání. Následující délky slov pak způsobují očekávanou maximální varianci od modelu s pozorovatelným vrcholem přibližně pro $n = 13$. Dále můžeme u náhodných a pseudonáhodných sekvencí pozorovat rychlou konvergenci k nulovým rozdílům očekávaným intervalem nasycení. Přirozené texty se pak chovají oproti těm náhodným logicky divergencí pouze na jedinou stranu, tj. bez oscilace kolem modelu skutečné náhodnosti. Zda můžeme pozorované průběhy sekvencí považovat za náhodné nám později potvrdí či vyvrátí odvozený statistický test. Zajímavější pohled na průběhy sekvencí se nám nicméně naskytne, jakmile si jednotlivé grafy přiblížíme. Výsledky tohoto přiblížení vidíme v grafu 7. Právě zde si můžeme všimnout několika zajímavých detailů. Zřetelně zde můžeme pozorovat interval vyčerpání s nulovými rozdíly od modelu a bod, který bychom mohli označit jako bod bifurkace, od kterého dochází k rozvětvení celé pozorovatelné struktury postupně do místa s nejvyšší variancí. Tento bod bifurkace by měl být shodný s odvozeným počátkem intervalu harmonie a tedy stavu, kdy už délka slov vytváří dostatek variací na to, aby některé ze slov potenciálního slovníku nemuselo být do sekvence vybráno. S rostoucí délkou slov je pak pravděpodobnost opakovaného výběru stejného slova stále menší a následně vede k opětovné konvergenci k modelu TTR_{avg}. Relativně překvapivá je viditelná tendence u $n \approx 9$ mít spíše nižší TTR než model a zároveň u velikostí $n \approx 23$ mít spíše vyšší TTR než model (na základě tmavší barvy indikující překryvy). Důležitým pozorováním je následně i to, že grafické struktury vytvářené křivkami u každého zdroje (kromě kontrolních sekvencí přirozeného jazyka) jsou vizuálně téměř symetrické podle $y = 0$, tj. modelu TTR_{avg} odpovídajícímu průměru nekonečného počtu

náhodných sekvencí. Pozorovaná symetrie může v důsledku znamenat, že průměry těchto sekvencí mohou být modelu ještě mnohem blíže, tj. ve shodě s logikou modelu TTR_{avg} průměrujícího *nekonečné množství* náhodných sekvencí. Zda průměry testovaných náhodných sekvencí odpovídají logice modelu TTR_{avg} proto před testem všech sekvencí nejprve v krátkosti nahlédneme.



***Graf 6**: Rozdíly pozorovaných hodnot TTR(n) jednotlivých sekvencí od modelu TTRavg.*



***Graf 7**: Přiblížení pohledu na rozdíly pozorovaných hodnot TTR sekvencí a modelu TTRavg.*

Jednotlivé sekvence každého zdroje zprůměrujeme, tím pro každý zdroj sekvencí vznikne jediný průměrný vektor reprezentující všech 200 sekvencí. Na výsledky rozdílů těchto průměrů od modelu TTR_{avg} se podívejme do tabulky 11. V této tabulce navíc nalezneme i vyznačené intervaly QHC, tj. intervaly vyčerpání, harmonie a nasycení. Výslednou tabulku 11 tak můžeme interpretovat z pohledu modelu náhodných sekvencí i modelů intervalů QHC. První věcí, které si zřejmě v tabulce povšimneme, jsou výrazné nulové rozdíly náhodných a pseudonáhodných sekvencí od modelu v intervalu vyčerpání, sahající dále do intervalu harmonie. Tyto sekvence tak pro prvních 9 až 12 délek slov prakticky kopírují model TTR_{avg}. U kontrolního vzorku přirozeného jazyka je interval vyčerpání porušen už při délce $n = 5$, což v důsledku říká, že zkoumané sekvence přirozeného jazyka v průměru pro sekvenci o délce 6 000 bitů a užitém kódování abecedy metodou MKM nevyužijí vždy všech $2^5 = 32$ dostupných slov potenciálního slovníku. Zároveň můžeme pozorovat, že všechny generátory využívající

fyzikální zdroj entropie (tj. generátory *náhodných sekvencí*) mají maximální odlišnost od modelu na *n*-gramech 13 a 14 (v tabulce zvýrazněno tučně), tj. v intervalu harmonie, u kterého očekáváme nejvyšší odlišnosti od modelu. Průměr sekvencí českých knih má rozdíl nejvyšší na *n*-gramech 14 a 15. Generátor pseudonáhodných sekvencí MCG má tento rozdíl nejvyšší až na *n*-gramech 16 a 17, tj. na hraně a v intervalu nasycení a navíc prakticky 5x vyšší hodnotou, než je nejvyšší odchylka u náhodných sekvencí. Takový posun je zajímavý, protože značí nepravděpodobné opakování slov. Zda je taková hodnota stále v rámci testu náhodnosti přípustná, se dozvíme dále. U náhodných sekvencí dále můžeme v intervalu nasycení pozorovat rychlou konvergenci rozdílů k nule. Zatímco u pseudonáhodného generátoru MCG dochází k nasycení s pozorovanými anomáliemi na délkách 22 a 25 (podtrženo), u sekvencí přirozeného jazyka pak na tomto intervalu dochází k nasycení velmi pozvolna. Nenulové hodnoty v intervalu nasycení můžeme interpretovat tak, že i přes rozsáhlou velikost potenciálního slovníku generátor (ať už jakýkoliv) stále vybírá již použitá slova a opakuje je, a to v rozporu s rychle klesající pravděpodobností jejich náhodného výběru. Z této tabulky tak lze uvážit, že zdroj MCG je od ostatních zdrojů odlišný a pozorované anomálie v opakování slov nám dávají tušit, že generátor není skutečně náhodný.

n	Iv.	RAND.ORG-TTR_{avg}	rngRes.-TTR_{avg}	ANUQRN-TTR_{avg}	OpenGPG-TTR_{avg}	MCG-TTR_{avg}	Kontr. CZ - TTR_{avg}
1	ℚ	0	0	0	0	0	0
2	ℚ	0	0	0	0	0	0
3	ℚ	0	0	0	0	0	0
4	ℚ	0	0	0	0	0	0
5	ℚ	0	0	0	0	0	-0,00001
6	ℚ	0	0	0	0	0	-0,00017
7	ℚ	0	0	0	0	0	-0,00106
8	ℍ	0	0	0	0	0	-0,00447
9	ℍ	0	0	0	0	0	-0,01671
10	ℍ	0	0	0	0	-0,00003	-0,05592
11	ℍ	0,00001	0	0	0	-0,00002	-0,14907
12	ℍ	0,00003	0,00003	0,00002	0,00002	0,00008	-0,28001
13	ℍ	**0,00005**	**0,00005**	**0,00004**	**0,00005**	-0,00012	-0,38600
14	ℍ	**0,00005**	**0,00005**	**0,00004**	**0,00005**	0,00014	**-0,43466**
15	ℍ	0,00004	0,00003	0,00003	0,00003	0,0002	**-0,43599**
16	ℍ	0,00002	0,00002	0,00002	0,00002	**0,00027**	-0,41000
17	ℂ	0,00001	0,00001	0,00001	0,00001	**0,00028**	-0,37145
18	ℂ	0,00001	0	0,00001	0,00001	0,00016	-0,32942
19	ℂ	0	0	0	0	0,00004	-0,28981
20	ℂ	0	0	0	0	0,00001	-0,25520
21	ℂ	0	0	0	0	-0,00001	-0,22590
22	ℂ	0	0	0	0	-0.00003	-0,20097
23	ℂ	0	0	0	0	0	-0,17880
24	ℂ	0	0	0	0	0	-0,15919
25	ℂ	0	0	0	0	-0.00001	-0,14252

***Tabulka 11**: Rozdíly průměrných sekvencí jednotlivých zdrojů od modelu TTR_{avg}.*

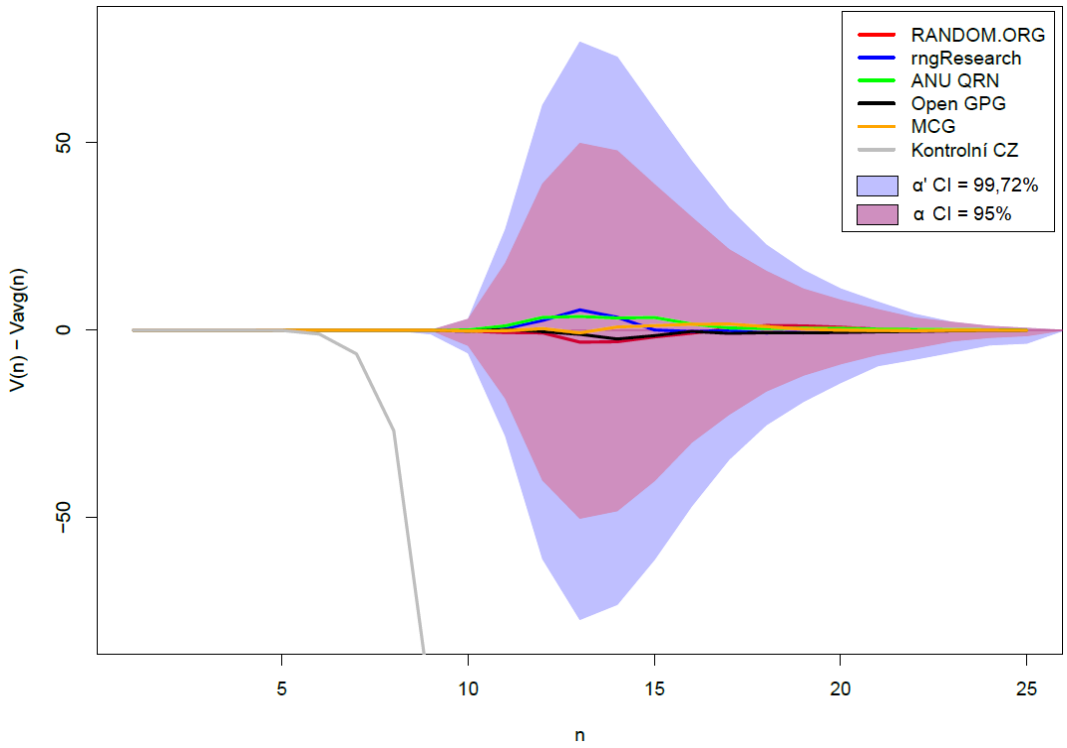
Statistickou významnost rozdílů sekvencí od modelu TTR_{avg}ověříme odvozeným testem. Z výsledků testů očekáváme odmítnutí nenáhodných zdrojů, tj. především kontrolního vzorku přirozeného textu a ideálně i pseudonáhodného generátoru MCG. Pro testované délky slov n od *1* do *25*, délku sekvencí v bitech $k = 6\,000$ a hladinu významnosti $\alpha = 0{,}05$ vypočítáme pomocí (13 CI) konfidenční intervaly počtu typů V_n, které se mohou ve skutečně náhodných sekvencích objevit s $(1 - \alpha) \times 100\%$ pravděpodobností. Aby byla sekvence testem označena jako náhodná, je nutné, aby počty zjištěných typů V_n náležely právě do stanovených intervalů. Nejprve však pro představu nahlédneme na samotné konfidenční intervaly. V tabulce 12 nalezneme vypočítané konfidenční intervaly pro jednotlivé délky slov n odečtené od modelu TTR_{avg}, respektive od jeho přepočtu z poměru počtu typů a tokenů jen na počet typů, tj. na model $V_{avg}(n)$. Tímto získáme představu o *vůli* v počtu typů, kterými se mohou náhodné sekvence od modelu na hladině α lišit. Pro kompletní představu a srovnání jsou uvedeny hodnoty pro hladinu $\alpha = 0{,}05$ i její odvozenou a dále uvažovanou korekci $\alpha' = 0{,}05/(25 - q(6000)) = 0{,}05/(25 - 7) = 0{,}00277$. První věcí, které si můžeme v tabulce 12 všimnout, je dodržení intervalu vyčerpání, kdy z hlediska uvedených konfidenčních intervalů neexistuje žádná vůle pro jakoukoliv odlišnost od modelu. Dále si můžeme všimnout, že maximální vůle je v obou případech právě uprostřed intervalu harmonie, od kterého následně dochází ke konvergenci k téměř nulové vůli intervalu nasycení. To, co je skutečně zajímavé, je vyplývající šikmost rozložení. Můžeme si všimnout, že konfidenční intervaly umožňují realizaci spíše méně než více typů (dle vyšších hodnot ve sloupci s minimem než s maximem). To znamená, že model (13 CI) pro testovanou konfiguraci udává větší pravděpodobnost zopakování již využitých typů než realizaci či výběru nových. Pozorovatelná je i symetrie horní a dolní meze. Dále si můžeme všimnout, že konfidenční interval vytvořený na základě korekce hodnoty α je skutečně větší než bez jejího užití.

		Hladina α vč. korekce		Původní hladina α	
n	lv.	$V_{avg} - \min(CI_{\alpha'})$	$\max(CI_{\alpha'}) - V_{avg}$	$V_{avg} - \min(CI_{\alpha})$	$\max(CI_{\alpha}) - V_{avg}$
1	$\mathbb{Q}$	0	0	0	0
2	$\mathbb{Q}$	0	0	0	0
3	$\mathbb{Q}$	0	0	0	0
4	$\mathbb{Q}$	0	0	0	0
5	$\mathbb{Q}$	0	0	0	0
6	$\mathbb{Q}$	0	0	0	0
7	$\mathbb{Q}$	0	0	0	0
8	$\mathbb{H}$	1	1	1	1
9	$\mathbb{H}$	1	1	1	1
10	$\mathbb{H}$	7	3	5	3
11	$\mathbb{H}$	30	28	19	18
12	$\mathbb{H}$	63	63	40	40
13	$\mathbb{H}$	**80**	**79**	**51**	**50**
14	$\mathbb{H}$	77	75	49	48
15	$\mathbb{H}$	64	61	41	39
16	$\mathbb{H}$	49	47	30	31
17	$\mathbb{C}$	37	34	23	22
18	$\mathbb{C}$	27	24	17	16
19	$\mathbb{C}$	20	16	13	11
20	$\mathbb{C}$	14	12	9	9
21	$\mathbb{C}$	11	8	7	6
22	$\mathbb{C}$	8	5	5	4
23	$\mathbb{C}$	6	3	3	3
24	$\mathbb{C}$	4	2	2	2
25	$\mathbb{C}$	4	1	2	1

***Tabulka 12**: Vypočítané konfidenční intervaly počtu typů pro testovanou konfiguraci sekvencí.*

Na konfidenční intervaly z tabulky 12 se můžeme podívat i grafičtěji – oba typy konfidenčních intervalů (CI), tj. bez korekce hladiny α (α CI) a s korekcí (α' CI), zobrazíme pro testované délky slova n do grafu, včetně vykreslení křivek zprůměrovaných sekvencí jednotlivých zdrojů odečtených od modelu TTR_{avg} (stejně jako u grafů 6 a 7). Náhodné zdroje by měly na $(1-\alpha) \times 100\,\%$ do tohoto intervalu spadat. Zda půjde o volnější interval tvořený korekcí hladiny α', nebo o užší interval tvořený na hladině α, závisí na kvalitě náhodnosti. Výsledek můžeme sledovat v grafu 8. Můžeme se v něm snadno přesvědčit o symetrii konfidenčních intervalů i vlivu korekce hladiny α. Z grafu je patrné, že všechny zdroje náhodných a pseudonáhodných sekvencí jsou zřejmě uvnitř striktnějšího intervalu tvořeného bez užití korekce hladiny α. Zda tyto sekvence skutečně náleží do určených intervalů ověříme vzápětí. Kontrolní vzorek sekvencí přirozeného jazyka není, oproti ostatním, v konfidenčním intervalu zahrnut již od velikosti n-gramů 6 a již na základě vizuální evaluace můžeme o tomto zdroji tvrdit, že je nenáhodný. Zdroj MCG se zde však jeví jako bezchybný. Nyní se podívejme na exaktní hodnocení náhodných a pseudonáhodných zdrojů pomocí tabulky 13. V ní nalezneme konkrétní hodnoty minimálního (L) a maximálního (P) počtu typů pro danou délku slov a hladinu α a dále pozorované počty typů jednotlivých průměrů zdrojů sekvencí. Jakékoliv odchylky od intervalu jsou značeny červeným

podbarvením a indikátorem, zda je pozorovaná hodnota menší či větší než mez intervalu. Pohledem zjišťujeme, že všechny zdroje náhodných i pseudonáhodných sekvencí náleží do vypočítaných konfidenčních intervalů a o těchto zdrojích tak můžeme prohlásit, že se v průměru vzorku dvou set sekvencí chovají dle odvozených konfidenčních intervalů na hladině $\alpha = 0{,}05$, a to bez užití její korekce, tak, jak by se při takové konfiguraci chovaly dokonale náhodné sekvence.



***Graf 8**: Rozdíly zprůměrovaných sekvencí jednotlivých zdrojů od modelu TTRavg, včetně vykreslených konfidenčních intervalů*

n	CI α‘		CI α		RAND.ORG	rng Re-search	ANU QRN	Open GPG	MCG	Kontr. CZ
	L	*P*	*L*	*P*						
1	*2*	*2*	*2*	*2*	2	2	2	2	2	2
2	*4*	*4*	*4*	*4*	4	4	4	4	4	4
3	*8*	*8*	*8*	*8*	8	8	8	8	8	8
4	*16*	*16*	*16*	*16*	16	16	16	16	16	16
5	*32*	*32*	*32*	*32*	32	32	32	32	32	32
6	*64*	*64*	*64*	*64*	64	64	64	64	64	63 <
7	*128*	*128*	*128*	*128*	128	128	128	128	128	122 <
8	*256*	*256*	*256*	*256*	256	256	256	256	256	229 <
9	511	512	512	512	512	512	512	512	512	412 <
10	1015	1024	1017	1024	1021	1021	1021	1021	1021	686 <
11	1909	1966	1920	1956	1938	1939	1939	1938	1938	1045 <
12	3084	3209	3107	3186	3146	3149	3150	3147	3147	1470 <
13	4169	4327	4198	4298	4245	4253	4252	4247	4247	1937 <
14	4939	5090	4967	5063	5012	5018	5018	5013	5016	2413 <
15	5408	5532	5431	5510	5469	5471	5474	5470	5472	2861 <
16	5671	5766	5690	5750	5719	5719	5722	5720	5721	3266 <
17	5813	5883	5827	5871	5850	5849	5850	5849	5851	3627 <
18	5889	5939	5899	5931	5916	5914	5916	5915	5916	3944 <
19	5929	5964	5936	5959	5949	5947	5949	5947	5948	4214 <
20	5950	5975	5955	5972	5965	5964	5964	5963	5964	4438 <
21	5961	5979	5965	5977	5972	5971	5972	5971	5971	4621 <
22	5967	5979	5970	5978	5975	5975	5975	5974	5975	4773 <
23	5970	5978	5973	5978	5976	5976	5976	5976	5976	4907 <
24	5972	5977	5974	5977	5976	5976	5976	5976	5976	5024 <
25	5972	5976	5974	5976	5975	5976	5976	5975	5975	5124 <

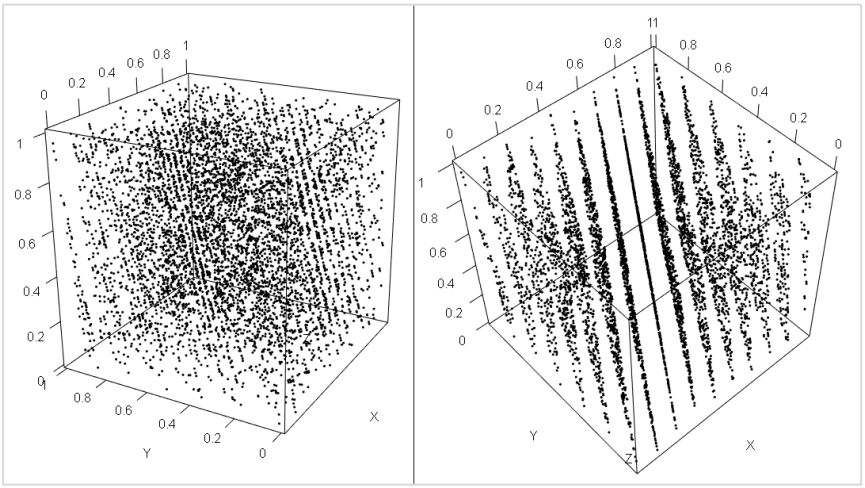
***Tabulka 13**: Vypočítané konfidenční intervaly (CI) jednotlivých délek slov pro hladiny α a α‘ v počtu typů a změřené počty typů jednotlivých zdrojů sekvencí.*

Zdroj	Spadající do 99,72% CI (α‘) [%]	Spadající do 95% CI (α) [%]
RANDOM.ORG	68,5	31,0
rngResearch	78,0	30,5
ANU QRN	76,0	35,5
Open GPG	73,0	34,0
MCG	74,0	33,0
Kontrolní CZ	0	0

***Tabulka 14**: Výsledky testů náhodnosti jednotlivých sekvencí každého ze studovaných zdrojů, včetně vyznačení maximální úspěšnosti (zeleně) a minimální úspěšnosti (červeně; kontrolní vzorek není započítán).*

Jednotlivé zdroje náhodných i pseudonáhodných sekvencí reprezentované jejich průměry se tedy z pohledu metody MKM a odvozených modelů jeví jako náhodné. Otázkou ovšem je, zda a jak budou tyto zdroje úspěšné, pokud budeme testovat každou z nich zvlášť. Všech 200 sekvencí každého zdroje proto podrobíme testu na hladině $\alpha = 0{,}05$ a její korekci. Procentuální úspěšnost splnění testu následně shrneme v tabulce 14, ve které můžeme pozorovat kvalitativní skok. Kromě úspěšného odmítnutí všech sekvencí kontrolního vzorku jsou tu další velmi zajímavé údaje.

Zprůměrované verze sekvencí náhodných a pseudonáhodných zdrojů byly téměř dokonale náhodné. Samotné sekvence však testem prochází v průměru jen na 73,9 %. V případě, když by byly všechny sekvence skutečně náhodné, byla by tato úspěšnost pro hladinu α' rovna přibližně $\left(1-\frac{0{,}05}{25-7}\right)^{25-7} \approx 95{,}11\ \%$. Bez použití korekce by za tohoto předpokladu mělo procházet $(1-0{,}05)^{25} \approx 27{,}73\ \%$ sekvencí, prošlo jich však v průměru 32,8 %. Oba výpočty samozřejmě platí za podmínky, že jsou partikulární testy konfidenčních intervalů jednotlivých délek *n*-gramů na sobě nezávislé a zároveň, že jsou sekvence skutečně náhodné. Pozorovaný výsledek proto vede k několika možným interpretacím. Prvním možným výkladem je, že žádný ze zdrojů není dokonale náhodný. Tato interpretace je však validní pouze tehdy, pokud jsou všechny partikulární testy délek *n* na sobě nezávislé. Pokud je mezi partikulárními testy závislost a sekvence jsou skutečně náhodné, vedla by tato závislost k divergenci od vypočítaných hodnot úspěšnosti. Druhá možnost je zřejmá v tom, že se může jednat o kombinaci obou uvedených problémů, a to, že jsou obsaženy sekvence, které nejsou skutečně náhodné a zároveň jsou partikulární testy délek slov závislé. Ošemetnou situací zde je, že kromě jiných statistických testů prakticky nelze nalézt odpověď, zda jsou konečně dlouhé náhodné sekvence skutečně náhodné. Tento problém se promítá i do problematiky interpretace jednotlivých zdrojů. Jednoznačně můžeme říci, že jsme dokázali identifikovat kontrolní nenáhodné sekvence. Dále ale víme, že zdroj MCG je pseudonáhodný generátor s konfigurací odpovídající generátoru RANDU, který vytváří specifické vzory viditelné v grafu 9 a nelze jej považovat za náhodný i s případným využitím skutečně náhodného zdroje pro *seed*. Test založený na metodě MKM však celý zdroj v průměru označil nejen jako náhodný pro striktnější nastavení hladiny α (viz tabulka 13), ale výsledky jednotlivých sekvencí zároveň nejsou tím nejméně úspěšným zdrojem (viz tabulka 14). Metoda MKM tak selhává v odhalení generátoru pseudonáhodných sekvencí RANDU.



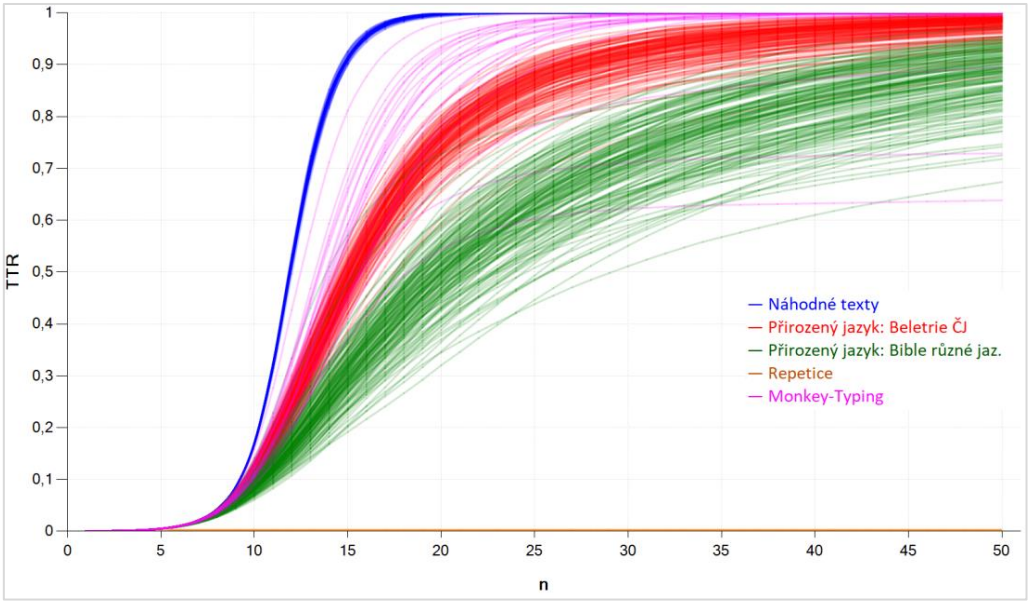
***Graf 9**: Prostorová interpretace hodnot vygenerovaných pomocí MCG RANDU.*

Poněkud zajímavé je ovšem to, že v odhalení generátoru RANDU dle Hamano a Yamamoto 2010 (1351-1352) selhávají na hladině $\alpha = 0{,}01$ i všechny metody NIST, kdy jako náhodné prochází testy v průměru 98,67 % sekvencí (o délce 10^6 bitů, oproti zde testovaným 6×10^3 bitům). Nejméně sekvencí, tj. 96,3 %, pak projde právě u metody *Serial Test* popsané v předchozí podkapitole jako nejpodobnější metodě MKM. Metoda MKM přitom odmítne na stejně hladině α 82,5 % sekvencí s kompletní korekcí, 84,5 % pouze s Bonferroniho korekcí a 53,5 % sekvencí bez užití jakékoliv korekce. Z tohoto lze odvodit, že metoda MKM může lépe registrovat obsažené vzory, může mít lépe definovaný statistický test, nebo se jedná jen o shodu okolností. Naopak metoda t-komplexity (Hamano a Yamamoto 2010, 1351) vedla ke 100% odmítnutí všech testovaných sekvencí pocházejících ze stejné konfigurace generátoru MCG. Dále jsme v tomto ohledu identifikovali, že nejmenší úspěšnost splnění testu náhodnosti má zdroj RANDOM.ORG. Otevřenou otázkou je, zda jsou sekvence z tohoto zdroje skutečně méně náhodné než MCG, nebo zda je testování metodou MKM právě neodmítnutím sekvencí z MCG znerelevantněno. Například porovnání obou zdrojů metodou t-komplexity by vyžadovalo doplnění testů i na ostatních uvedených zdrojích, neboť u uvedené metody absentuje porovnání např. toho, zda testem vůbec nějaký zdroj dokáže projít. Vzhledem k tomu, že je testování náhodnosti sekvencí jen částí celé metody MKM, ponecháme tuto evaluaci na další práci, stejně jako hodnocení jednotlivých zdrojů sekvencí, která by aktuálně byla založena pouze na spekulacích. Celý tento problém ilustrujme nejistotou, která panuje při porovnání MCG, atmosférického šumu ze služby RANDOM.ORG a jediného šifrovaného souboru pomocí Open GPG. Na základě tabulky 14 víme, že atmosférický šum i Open GPG dosáhly horšího počtu úspěšného přijetí testem náhodnosti než sekvence pocházející z MCG, tj. zdroje dokazatelně tvořícího nenáhodné vzory. Můžeme spekulovat, že sekvence z Open GPG i atmosférický šum obsahují větší množství nenáhodných vzorů, než které obsahují sekvence z MCG. Problém je v dokazatelnosti takového tvrzení. Pokud by tato spekulace ale byla pravdivá, znamenalo by to dále, že metoda založená na procesování turbulence vzduchu rngResearch dosahuje vyšší náhodnosti než kvantová fluktuace, což by mohlo být zajímavé pro fyziky.

Výsledkem experimentální analýzy zdrojů náhodných sekvencí metodou MKM jednoznačně je, že metoda i test dokáží rozlišit náhodné a pseudonáhodné sekvence od kontrolního vzorku sekvencí přirozené jazyka. Zároveň je výsledkem poznatek, že je test citlivější vůči pseudonáhodnému generátoru MCG než kterýkoliv z testů náhodnosti obsažených v NIST. Takový výsledek lze považovat za úspěch, neboť cíle metody MKM jsou širší než identifikace náhodných či nenáhodné sekvencí. Velmi důležité také je, že veškerá měření a testování byla prováděna na sekvencích o délce 6 000 bitů, tj. 750 bytů přibližně představitelných jako polovina normostrany textu. Dále s nově nabytými vědomostmi dokážeme vysvětlit i další specifikace křivek, respektive hodnot TTR, které jsme viděli v úvodním grafu metody MKM.

Trendy křivek metody MKM

Prozatím jsme se setkali s modelem dokonale náhodných sekvencí TTR_{avg} (8) a s modely popisujícími intervaly QHC (9, 10, 11, 12) a jejich očekávané hodnoty TTR. Také jsme se na začátku, oproti náhodným sekvencím, setkali se sekvencemi triviálních repetic, pro které libovolná délka slova n získává stále stejný počet typů. Z úvodního grafu 2, který zde pro přehlednost znovu umístíme jako graf 10, studujícího průběhy křivek metody MKM na různých typech sekvencí, a to od přirozeného jazyka až po *monkey-typed* texty, bychom mohli získat jistou představu o rozdělení určitými pomyslnými pásmy. Pro porozumění typům křivek a různých pásem v grafu je nejprve vhodné určit extrémní případy a jejich modely, tj. takové případy, které svými hodnotami ohraničují výsledky, které vůbec mohou v metodě MKM nastat.



***Graf 10**: Analýza sekvencí různých zdrojů metodou MKM pro délky n-gramů 1 až 50.*

Prvním z extrémů metody MKM je hranice tvořená triviálními repeticemi (oranžová křivka kopírující osu x v grafu 10). Triviální repetice jsou tvořeny neustálým opakováním jediného vzoru, který se může u různých sekvencí lišit svou komplexitou. Maximální triviální sekvence je pak taková, která opakuje právě jeden stejný bit, což u n-gramů libovolných délek vede k nalezení vždy a právě jediného typu. Je tak zřejmé, že s narůstající délkou sekvence budou výsledné hodnoty TTR konvergovat k nule, tj. platí $\lim_{|S|\to\infty} 1/D(n,|S|) = 0$, kde $|S|$ je délka sekvence v bitech a $D(|S|)$ je počet tokenů o délce n sekvence o dané bitové délce. Opakovaný vzor může být i komplexní, pokud bude ale stále opakován za sebou, bude počet nalezených typů odpovídat právě délce tohoto vzoru. Pokud budeme například opakovat vzor „a a b c“, pak maximální počet typů, který dokážeme pomocí n-gramů o libovolné délce získat, bude právě 4. Podívejme se na příklad sekvence S:

S = „a a b c a a b c a a b c a a b c a a b c a a b c a a b c “

1-gramy: { a, b, c }
2-gramy: { a a, a b, b c, c a}
3-gramy: { a a b, a b c, b c a, c a a }
4-gramy: { a a b c, a b c a, b c a a, c a a b }
…
26-gramy: { a a b c a a b c a a b c a a b c a a b c a a b c a a,
a b c a a b c a a b c a a b c a a b c a a b c a a b,
b c a a b c a a b c a a b c a a b c a a b c a a b c } .

Na příkladu sekvence S vidíme, že minimální počet typů je dán velikostí abecedy, a to na základě použití délky *n*-gramů 1. Maximální počet typů je pro libovolnou délku *n*-gramů dán velikostí samotného vzoru. Omezení počtu typů shora nás opět s délkou sekvence vede ke konvergenci hodnot TTR k nule, tj. pro libovolné $n \in \mathbb{N} \wedge 1 \leq n \leq |S|$ platí $\lim_{|S|\to\infty} V_n/D(n,|S|) = 0$, kde V_n odpovídá počtu typů o délce n. Z tohoto následně vyplývá, že maximálně triviální repetice opakující jediný bit budou mít pro stejnou délku sekvence vždy nižší TTR než ty komplexnější a lze je tak označit za extrém dosahující nejnižších možných hodnot metodou MKM. Hodnoty TTR maximálně triviální sekvence pro libovolnou délku *n*-gramů *n* a délku sekvence v bitech *k* vypočítáme následovně:

$$TTR_{min}(n,k) = 1/D(n,k)$$

Tímto je vysvětlen a formalizován první extrém minimálních hodnot viditelných v grafu 10 a který můžeme pro určitou orientaci považovat za pravý extrém. Nyní se podíváme na levý extrém, tedy na extrém maximálních hodnot TTR, kterých může metoda MKM dosáhnout.

Vyšší hodnoty TTR, než které mají triviální repetice, mají postupně strukturované přirozené texty, přirozené texty a následně *monkey-typed* texty, které jsou umístěny mezi přirozené jazyky a náhodné sekvence pocházející z RANDOM.ORG. Z tohoto popisu by se proto mohlo zdát, že je zde přítomen určitý pravo-levý trend postupující v grafu 10 zprava od triviálních repetic směrem ke strukturovaným informacím, komplexnosti systému přirozeného jazyka s určitou volností výběru slov, dále k *monkey-typed* textům reflektujícím jen několik málo pravidel až k náhodným sekvencím. Mohlo by se proto jevit, že maximálně vlevo v grafu budou umístěny náhodné sekvence postrádající jakýkoliv systém. Jak ale uvidíme, tímto levým extrémem metody MKM, který bude pro každé *n* zísávat maximální možné hodnoty TTR, nejsou náhodné sekvence, ale velmi specifické sekvence vytvářené specifickým systémem, který označíme jako *maximální model TTR*, zkráceně TTR_{max} nebo pouze *maximální model*.

Maximální model vystihuje sekvence, u kterých je využito maximální množství slov z potenciálního slovníku pro všechny testované délky *n*-gramů. Jinými slovy jde o sekvence, ve kterých je pro každou testovanou délku slov využit maximální možný počet typů daných binární kombinatorikou. Maximální počet typů, které se mohou v sekvenci vyskytnout, je pro sekvenci a délku *n*-gramů dán dvěma triviálními faktory: (1) délkou sekvence, tj. nelze mít více typů, než je počet tokenů a (2) velikostí potenciálního slovníku, kdy nelze realizovat více typů než kolik jich lze kombinatoricky vytvořit. Počet typů (různých slov), který může sekvence maximálně realizovat, vypočítáme pro její délku *k* v bitech a délku *n*-gramů *n* pomocí (14):

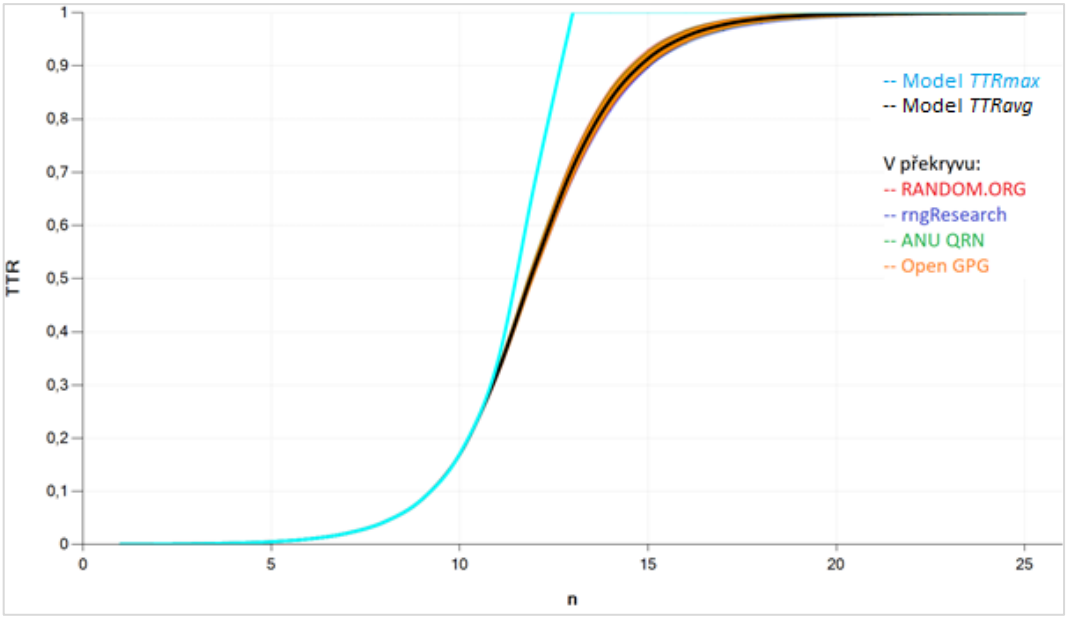
$$V_{max}(n,k) = \min\big(V(n), D(n,k)\big) \quad . \quad (14)$$

Následně můžeme odvodit model TTR_{max}(15) jako poměr mezi maximálním možným slovníkem a délkou sekvence v tokenech:

$$TTR_{max}(n,k) = \frac{V_{max}(n,k)}{D(n,k)} \quad , \quad (15)$$

kde *k* je délka sekvence v bitech, *n* je velikost *n*-gramu a platí $k \geq n$ pro $k, n \in \mathbb{N}$. Jev, kdy sekvence pro zadanou délku n-gramů vyčerpá možný slovník, nazveme jevem maximalizace slovníku.

Na nově odvozený model sekvencí maximalizující využití svého potenciálního slovníku TTR_{max} nahlédneme společně se všemi náhodnými sekvencemi z tabulky 10 a modelem TTR_{avg} v grafu 11, ve kterém se můžeme přesvědčit, že je model TTR_{max} skutečně maximální levou hranici (samozřejmě kromě prvních *n*-gramů, u kterých dochází k jevu vyčerpání a jsou tak ve shodě s náhodnými sekvencemi a modelem TTR_{avg}). Dále je však růst křivky strmý a prakticky lineární, kdy s každým navýšením délky *n*-gramů přibývá slovníku možnost výběru, která umožňuje rychleji vyrovnávat podíl mezi typy a tokeny až k hodnotě 1.

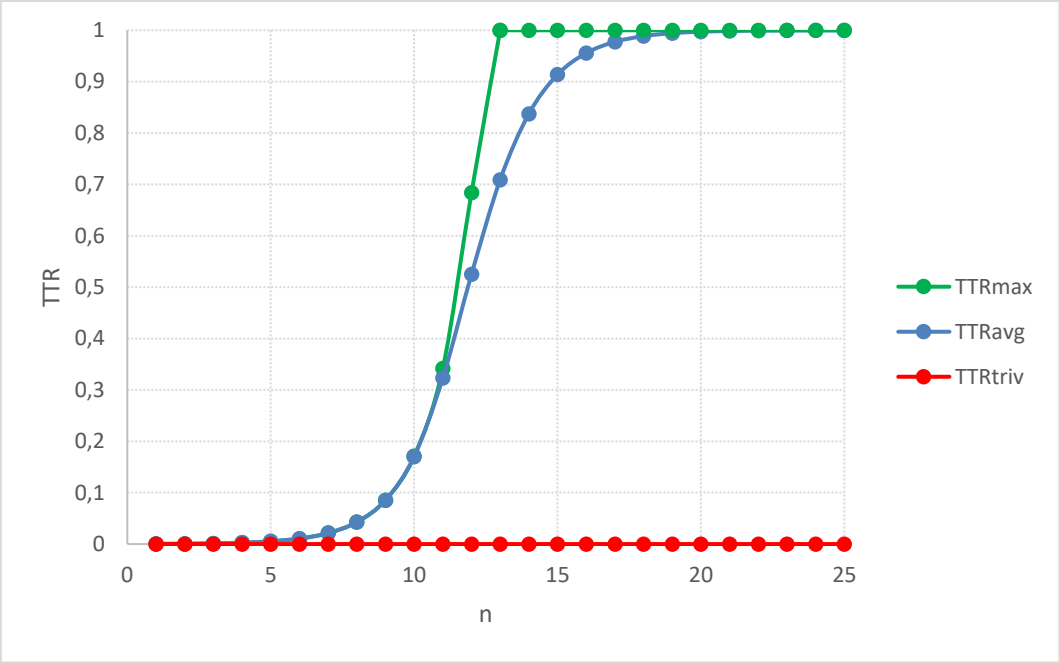


***Graf 11**: Zobrazení náhodných sekvencí a modelů maximálního TTR a dokonale náhodných sekvencí TTRavg.*

Původní intuitivní myšlenku, že rozdělení grafu výsledků MKM je rozčleněno na pásma s pravo-levým trendem od triviálních repetic až po dokonale náhodné sekvence tak musíme upravit. Výše jsme zjistili, že maximální levou hranicí je model maximálního TTR, který však může být náhodný jen stěží (viz dále). *Apriorní* odmítnutí náhodnosti těchto sekvencí vyplývá z principu samotného modelu a pravděpodobnosti vzniku takových sekvencí. Připomeňme, že sekvence odpovídající tomuto modelu nesmí pro maximalizaci využití potenciálního slovníku zopakovat jediné slovo, pokud je to možné (tj. potenciální slovník není na sekvenci příliš malý). Při tvorbě takové sekvence je tedy ideálně vyžadován systém, který *hlídá* využití slovníku tak, aby byl koncem sekvence použit v jeho maximální možné míře. Takový systém nutně vyžaduje paměť a určitý kalkul. Tento princip je přesným opakem náhodnosti, ve kterém jsou prvky vybírány zcela nepředvídatelně a nezávisle na aktuálním a předchozím stavu. Uvedený systém sice může být stochastický, tj. může vybírat slova ze slovníku náhodně, ale s rostoucím počtem už vybraných slov bude postupně klesat nejistota, která slova mohou být z potenciálního slovníku vybrána v následujících krocích. Příkladem může být situace, kdy je velikost potenciálního slovníku shodná s počtem tokenů a do sekvence je vybíráno posledního slovo – v takové situaci je jisté, které slovo bude vybráno. Z tohoto důvodu nelze sekvence odpovídající tomuto modelu považovat za náhodné. Levá, maximální hranice, kterou model TTR_{max} vytváří, proto opět identifikuje specifický systém stojící za tvorbou sekvence.

Nalezené modely TTR_{min} a model TTR_{max} jsou pro metodu MKM důležité, neboť ohraničují možné výsledky metody MKM a napomáhají tak určit typ sekvencí dosahujících minimálních a maximálních hodnot TTR. Společně s modelem TTR_{avg} pak tyto tři modely vytváří náhled na možné průběhy křivek a poskytují představu o jejich zdroji. Všechny tři modely nyní ilustrujme v grafu 12 pro sekvence o délce 6 000 bitů

a délky slov od 1 do 25. Tímto máme definované extrémní typy sekvencí, které nám mohou napovědět, o jaký typ analyzované sekvence se jedná a jaký může být její zdroj. Tyto poznatky nám napomáhají při interpretaci grafu křivek a mohou nám sloužit i jako etalony u dalších metod pro klasifikaci neznámých sekvencí.

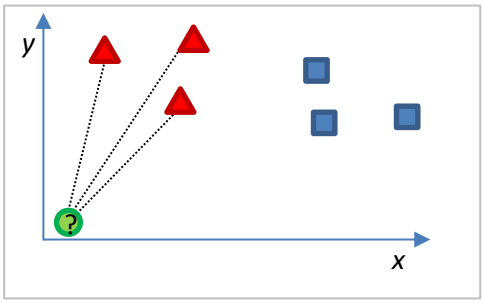


***Graf 12**: Ilustrace modelů TTRavg, TTRmax a TTRmin na sekvencích o délce 6 000 bit pro n-gramy o délkách 1 až 25.*

Klasifikace sekvencí pomocí metody MKM

Určení typu či jinak řečeno klasifikace neznámých sekvencí můžeme realizovat několika způsoby, přičemž každý z nich má určité výhody i nevýhody. Prozatím jsme se setkali s grafickým přístupem pozorování jednotlivých vektorových reprezentací sekvencí v grafu pomocí křivek a odhad typu neznámé sekvence jsme mohli intuitivně pojmout jako nalezení typu sekvencí s nejbližším a nejpodobnějším průběhem. Každá klasifikační metoda však stojí na obecné problematice, a to na dostupnosti vzorků typů sekvencí, které chceme dokázat klasifikovat a které budou sloužit jako referenční etalon k porovnání. Takový dataset musí být dostatečně rozsáhlý a reprezentativní natolik, aby poskytl náhled na distribuci hodnot TTR populace daného zdroje, což nemusí být snadné, nebo v některých případech ani možné. Tento problém nabývá větší dramatičnosti tím, že k analýze metodou MKM můžeme dostat prakticky libovolnou sekvenci, a to například ze zdroje, který v referenčním datasetu není, nebo jej vůbec neznáme. Je nicméně zřejmé, že je tento problém univerzální a lze se s ním vypořádat jen zavedením explikace nejistoty při klasifikaci. Tento problém konkrétně explikujme.

Prvním a nejsnazším způsobem klasifikace výsledků metody MKM je interpretace výsledných vektorů hodnot TTR jako souřadnic bodů v *n*-rozměrném prostoru. Při této interpretaci stačí pro klasifikaci využít metodu *k nejbližších sousedů* (kNN, *k-nearest-neighbours*), kdy pro neznámou sekvenci nalezneme *k* nejbližších bodů s *a priori* známým typem (třídou) a neznámé sekvenci přisoudíme tu nejzastoupenější z nich (viz např. Marsland 2015, 158-160). Míru nejistoty v určení třídy můžeme kvantifikovat např. procentuální shodou tříd jednotlivých sousedů nebo lze využít např. Fleissovu kappu kvantifikující shodu anotátorů (Fleiss a Cohen 1973). Vzdálenost mezi jednotlivými body lze obecně měřit jakoukoliv metrikou, nicméně pro metodu MKM lze uvážit euklidovskou vzdálenost jako principiálně nejsnazší, což vyplývá z potřeby k sobě přiřazovat vektory s nejbližšími hodnotami. Vzhledem k určení třídy sekvence pouze na základě počtu nejbližších sousedů je zřejmé, že v datasetu musí být všechny třídy vyváženy (tj. každá třída musí být zastoupena stejným počtem vzorků) nebo je jejich případnou disproporci nutné zohlednit pomocí vah. Výsledky klasifikační metody kNN jsou snadno interpretovatelné, nicméně metoda samotná může vést ke zcela iluzivním výsledkům vyplývajícím z jejího diskriminačního principu. Například, pokud je klasifikována sekvence, jejíž zdroj je unikátní a není zahrnut v referenčním datasetu, povede aplikace kNN v nejlepším případě k explicitně nejistému výsledku zachytitelnému např. zmíněnou Fleissovou kappou kvůli neshodě tříd sousedů. Naopak v tom horším případě bude této unikátní neznámé sekvenci přiřazena kterákoliv z nejbližších tříd. Tento problém můžeme ilustrovat na schematickém grafu 13, ve kterém chceme pomocí kNN pro $k = 3$ klasifikovat kroužek pomocí datasetu trojuhelníků a čtverců ve dvourozměrném prostoru. Výsledkem klasifikace je jednoznačná shoda, že se jedná o trojúhelník, neboť právě tři nejbližší sousedé jsou trojúhelníky. V případě, že bychom tímto způsobem klasifikovali anonymní sekvenci, a to i přes její jednoznačnou vzdálenost, získali bychom výsledek s vysokou a zároveň iluzivní přesností. Tento problém lze řešit např. dodáním pravděpodobnosti, se kterou je klasifikovaný objekt vůbec známou třídou, nebo zda se jedná o třídu novou, v závislosti na empiricky známých vzdálenostech mezi objekty. Takové heuristické nadstavby však začínají být těžko uchopitelné, přitom lze uvážit jiný model, který jsme již v této práci definovali a je založený přímo na pravděpodobnostech.



***Obrázek 13**: Metoda k-nejbližších-sousedů a problém unikátní sekvence.*

Druhý způsob klasifikace vektorů můžeme namísto jejich umístění do prostoru odvinout od jejich statistických vlastností. Obdobně, jako vnímáme pomyslná pásma jednotlivých typů sekvencí v grafu 10, můžeme tato pásma definovat i čistě formálně pomocí konfidenčních intervalů. Každá třída (zdroj) sekvencí by tak byla reprezentována pravděpodobnostním pásmem tvořeným jednotlivými konfidenčními intervaly délek *n*-gramů, do kterého spadá zvolených *x* % užitého vzorku. Nespornou výhodou tohoto přístupu je způsob klasifikace, která by probíhala stejně, jako u testu náhodných sekvencí, jen rozšířeně o pásma dalších typů. Nespornou výhodou takového přístupu je rovněž fakt, že neznámá testovaná sekvence bude klasifikována jako daný typ jen tehdy, pokud svými hodnotami (a průběhem v kontextu grafu) náleží celému pravděpodobnostnímu pásmu daného typu. Právě tento způsob testování umožní metodě odpovědět, zda je sekvence doposud neznámá a odlišuje se od zdrojů již existujících tříd. Ovšem obdobně jako u metody kNN výše bychom touto metodou dokázali získat i procentuální zastoupení shody s existujícími třídami, se kterými klasifikovaná sekvence sdílí průběh, čímž by nám umožnila říci o sekvenci alespoň určité její vlastnosti. Stanovení jednotlivých konfidenčních intervalů můžeme, vzhledem k rozmanitosti podkladových dat a *a priori* neznámým tendencím v distribuci hodnot TTR, docílit pomocí *bootstrapu* (Efron a Tibshirani 1994). V tomto ohledu mají *konfidenční pásma*, jak můžeme tuto metodu pracovně označit, určitou výhodu oproti klasifikaci metodou kNN. Ta vyžaduje vyvážený počet tříd v datasetu nebo zohlednění jeho nevyvážení pomocí vah. Metoda s konfidenčními pásmy však může operovat i s nevyváženým datasetem, a to právě vhledem k prováděné statistické inferenci. Naopak nejproblematičtější nevýhodou této metody je pak to, že pravděpodobnostní pásmo může být natolik široké, že v jeho rámci mohou být zahrnuty i takové hodnoty, které by ve skutečnosti daným zdrojem realizovány nebyly. Teoreticky lze proto nabídnout samotné testování podobnosti křivek, nicméně takové metody vedou k dalším pravděpodobnostním heuristikám. V důsledku to znamená, že ani tato metoda není dokonalá a vhodná k plně automatizovanému použití, neboť dokážeme nalézt takové případy, které mohou vést k desinterpretaci výsledků.

Obecným problémem evaluace výsledků metody MKM tkví v komplexitě, která kombinuje nejistoty plynoucí z inference ve stanovování pásem, problematiku odhalování nových typů sekvencí a dostatečnosti referenčního datasetu. Nalezení přesné a zároveň transparentní metody pro spolehlivou klasifikaci výsledků je obsáhlejším problémem než který lze zde explikovat. Paradoxně je tedy racionální evaluace výsledků grafu prozatím výhodnější volbou než užití automatických nástrojů, alespoň v podobě, v jaké zde byly navrženy. Zároveň je interpretace doposud používaného grafu křivek sekvencí metodou, která zobrazuje každou jednotlivou hodnotu TRR délek *n*-gramů a umožňuje jejich explicitní zahrnutí do interpretace. Problém této nejjednodušší metody je však v nepřehlednosti grafu ve chvíli, kdy je obsaženo více různých zdrojů (viz dále).

Jako alternativu ke grafu křivek jednotlivých hodnot TTR lze experimentálně nabídnout zobrazení využívající vícerozměrových metod, které interpretují výsledné *embeddingy* metody MKM jako souřadnice bodů nebo jako vektory v *n*-dimenzionálním prostoru. Taková interpretace má opět tu výhodu, že zahrnuje všechny hodnoty TTR jednotlivých *n*-gramů, takže by výsledné vizualizace měly, v ideálním případě, odrážet maximum obsažené informace. Grafické výsledky těchto metod by navíc mohly být přehlednější a jednodušší k interpretaci než grafy křivek, a to vzhledem k redukci celého *embeddingu* na jediný bod v grafu namísto prostor zabírající křivky. Nevýhodou aplikace vícerozměrných metod na zobrazení *n*-dimenzionálních *embeddingů* do klasického dvoj- nebo trojrozměrného grafu je přítomná možnost ztráty určitého množství informací, a to vzhledem k nutnosti redukovat počet původních rozměrů *embeddingu* na dané dva či tři. Ztráta informací je však typicky užívanými metodami cíleně minimalizována.

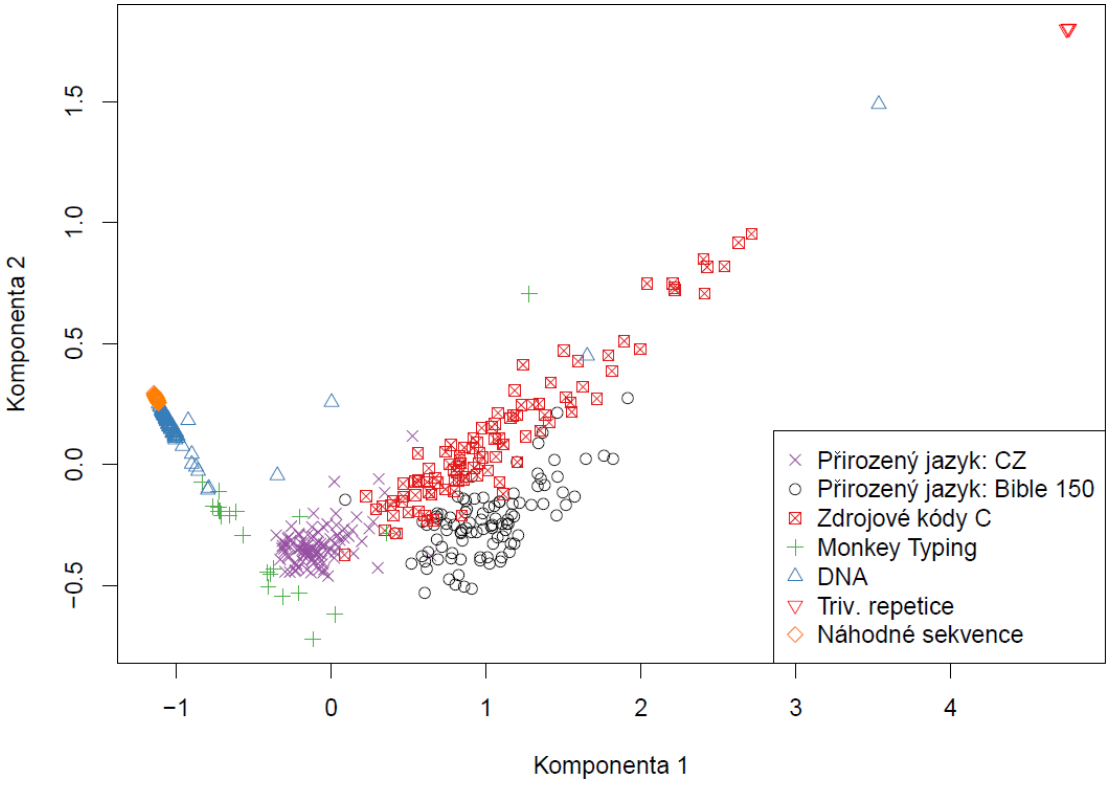
První z metod, kterou za účelem zobrazení vícerozměrných *embeddingů* využijeme, je klasické vícerozměrové škálování (MDS; Torgerson 1958). Pro použití MDS *embeddingy* hodnot TTR každé sekvence interpretujeme jako body v *Z*-rozměrném prostoru (kde *Z* je počet testovaných délek *n*-gramů). MDS se následně pokouší co nejvěrněji rekonstruovat zjištěné vzdálenosti mezi jednotlivými body v *Z*-rozměrném prostoru do *k*-rozměrného prostoru, tj. např. právě do 2-rozměrného nebo do 3-rozměrného *grafu*. MDS tak operuje pouze se vzdálenostmi *bodů* a nevyužívá žádných znalostí o existujících či domnělých třídách či typech sekvencí. Výsledná zobrazení tedy odpovídají pouze blízkosti jednotlivých *embeddingů* mezi sebou a vytváří tak jejich intuitivně interpretovatelnou analogii běžné mapy. Je zřejmé, že budou existovat takové konfigurace bodů, které nebude možné ze Z-rozměrného prostoru rekonstruovat do dvojrozměrného grafu bez ztráty informací. Metoda MDS nám však nabízí i vyčíslení množství zachované informace, která by v ideálním případě odpovídala 100 %. Rekonstruované vzdálenosti pak lze ve formě grafu natočit zcela arbitrárně, nicméně MDS volí takové natočení grafu, aby osa *x* odpovídala největšímu rozptylu pozorovaných vzdáleností a osa *y* druhému největšímu pozorovanému rozptylu atd. Lze tedy říci, že osa *x* je *nejdůležitější* latentní důvod odlišnosti jednotlivých bodů či zkoumaných sekvencí a osa *y* je druhý takový důvod. Interpretace a pojmenování těchto os vyžaduje vhled a porozumění zobrazeným datům.

Abychom možnosti MDS ilustrovali, rozšíříme doposud používaný dataset sekvencí o dva další typy zdrojů, o *zdrojové kódy programovacího jazyka C* (se všemi náležitostmi vč. komentářů) a o *kódující DNA*. Shrnutí sekvencí určených k testování nalezneme v tabulce 16. Na tento dataset sekvencí aplikujeme metodu MKM pro *n*-gramy o délce 1 až 50 a výsledné *embeddingy* následně ve formě matice vzdáleností předáme metodě MDS k rekonstrukci do dvourozměrného grafu. Z vektorů o původ-

ních 50 rozměrech vzniknou body v grafu tak, aby jejich rozmístění co nejlépe odpovídalo jejich původním euklidovským vzdálenostem. Výsledky zobrazení *embeddingů* vidíme v grafu 14 zobrazujícím jednotlivé sekvence všech analyzovaných typů jako body. Zobrazení zde rekonstruuje 99,36 % původních vzdáleností a výsledný graf tak můžeme považovat za velmi přesný. Díky principu MDS můžeme tento graf navíc vnímat jako *geografickou* mapu sekvencí, což nám umožňuje nejen porovnávat jejich vzdálenosti, ale identifikovat specifická místa a regiony.

Typ	Počet
Náhodné texty (uniformní, RANDOM.ORG)	100
Přirozený jazyk: Beletrie ČJ	100
Přirozený jazyk: Bible 100 jazyků	100
Monkey-Typed	21
Triviální repetice	3
Zdrojové kódy jazyka C	100
Kódující DNA sekvence	100

***Tabulka 16**: Rozšířený dataset pro testování metody MKM.*[13]



***Graf 14**: Torgersonovo klasické vícerozměrné škálování (MDS) výsledných vektorů metody MKM sekvencí rozšířeného datasetu s využitím euklidovské vzdálenosti.*

[13] Testované sekvence lze najít v datové příloze: Sekvence.

První, čeho si tak můžeme v grafu všimnout, je, že každý typ zdroje má relativně dobře ohraničené prostorové umístění či vlastní region (kterých by bylo možné využít např. při konfiguraci vícerozměrných pravděpodobnostních klasifikátorů). Dále si můžeme všimnout, že dominantou celého grafu jsou zdrojové kódy jazyka C s největším obsazeným regionem sousedícím s oběma zdroji přirozených jazyků (české beletrie a vzorkem Biblí v různých jazycích) a dále se přibližující k sekvencím triviálních repetic zcela detašovaným v pravém horním rohu. Zajímavý je rovněž volný prostor mezi sekvencemi Biblí a sekvencemi beletrií českého jazyka, který může být způsoben strukturací textu Biblí (verše, řádkování atd.). Také si můžeme všimnout, že české beletrie mají relativně vysokou homogenitu, což je zřejmě dáno tím, že se jedná o jediný jazyk, navíc obdobně strukturovaný. Blízko českým beletriím a směrem k tzv. kódujícím sekvencím DNA se blíží *monkey-typed* sekvence, tedy texty napsané libovolnými úhozy na klávesnici, u kterých jsme v předchozí kapitole sledovali existující a ergonomií dané vzory. DNA pak sousedí právě s těmito *monkey-typed* sekvencemi a náhodnými sekvencemi. Jak ovšem víme, metoda MKM není dostatečně citlivá, aby svou charakterizací dokázala odlišit skutečně náhodné sekvence od těch pseudonáhodných obsahujících komplexní vzory (jako u generátoru MCG výše). Sousedství DNA a náhodných sekvencí tak zajisté neimplikuje jejich náhodnost, pouze nepřítomnost snadno odhalitelných vzorů. Zajímavé však je, že se některé sekvence DNA vyskytují i v blízkosti programovacího jazyka i triviálních repetic. Stejně tak nalezneme jeden zdrojový kód na periferii regionu českých beletrií, což můžeme vysvětlit např. využitím náhodného vzorku zdrojového kódu obsahujícího především komentáře atd.

Charakterizace neznámých sekvencí je vzhledem k principu celé metody jednoduchá. Neznámá sekvence je jako bod (či body, v případě vzorkování) umístěna do prostoru mezi známé *regiony*, které nabídnou referenci k možné identifikaci nebo charakterizaci zdroje testované sekvence. Je však zřejmé, že umístění bodu neznámé sekvence do určitého regionu automaticky neznamená její jasnou klasifikaci, ale jde spíše o nápovědu její charakteristiky, což je cílem metody MKM. Důvod k takové skepsi vyplývá z řady již předeslaných problémů od chybějících referenčních sekvencí v datasetu až k chybějícím informacím ztraceným při redukci dimenzionality. Problematiku evaluace výsledků a nutnou racionální úvahu nad tím, jak je interpretovat, ilustrujeme v následující kapitole na skutečně neznámých sekvencích navzorkovaných z tzv. Vojničova rukopisu, který se pomocí metody MKM pokusíme charakterizovat a odhadnout typ jeho zdroje.

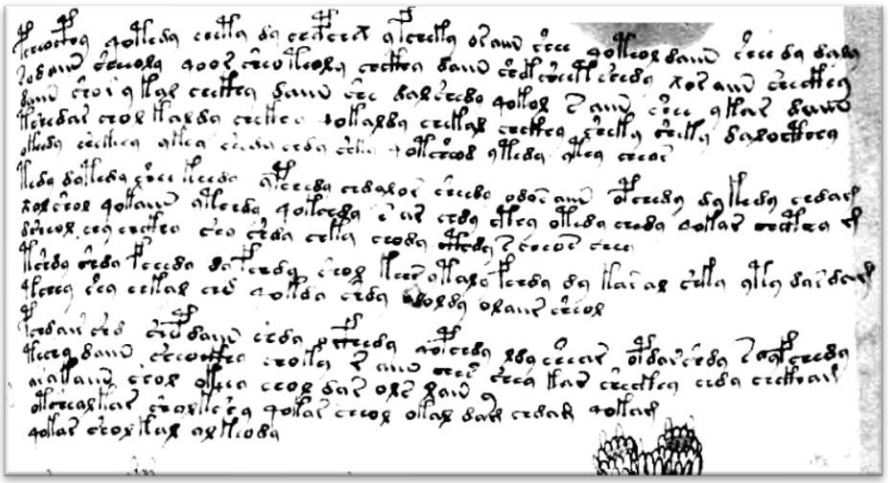
Ilustrativní aplikace metody MKM

V této podkapitole vyzkoušíme metodu MKM na jednom z nejznámějších nerozluštěných textů, a to na tzv. *Vojničově rukopisu*, jehož problematiku a historický kontext stručně popíšeme. Cílem této podkapitoly je především poukázat na problematiku vyhodnocování a interpretace výsledků metody MKM a obecně metod charakterizujících sekvence na základě kvantifikace jejich vlastností. Aplikace na reálný nerozluštěný text nám poskytne skutečný vhled do nejistoty interpretace vedoucí k nutným racionálním úvahám nad alternativními vysvětleními výsledků – tato podkapitola nám tedy bude ilustrovat skutečnou problematiku aplikace metody MKM na anonymní a obecné sekvence pocházející z neznámého zdroje a dále nám poskytne náhled na to, zda nabídnuté nástroje skutečně dokáží přinést jakékoliv informace. Potenciálem metody MKM, který tak zde vyzkoušíme, je schopnost charakterizovat a klasifikovat neznámé sekvence realizované různě velkými abecedami a bez jakékoliv *apriorní* identifikace hranic slov nebo znalosti jejich přítomnosti. Zmíněný Vojničův rukopis (rovněž jako *Voynichův*) má pro tyto účely především kritickou pragmatickou výhodu, kterou je jeho rozsah čítající přibližně 250 stran textu a ilustrací. Takový rozsah nám napomůže vytvořit dostatečný vzorek sekvencí a zlepšit tak celkově interpretabilitu výsledků. Nyní na tento rukopis stručně nahlédněme.

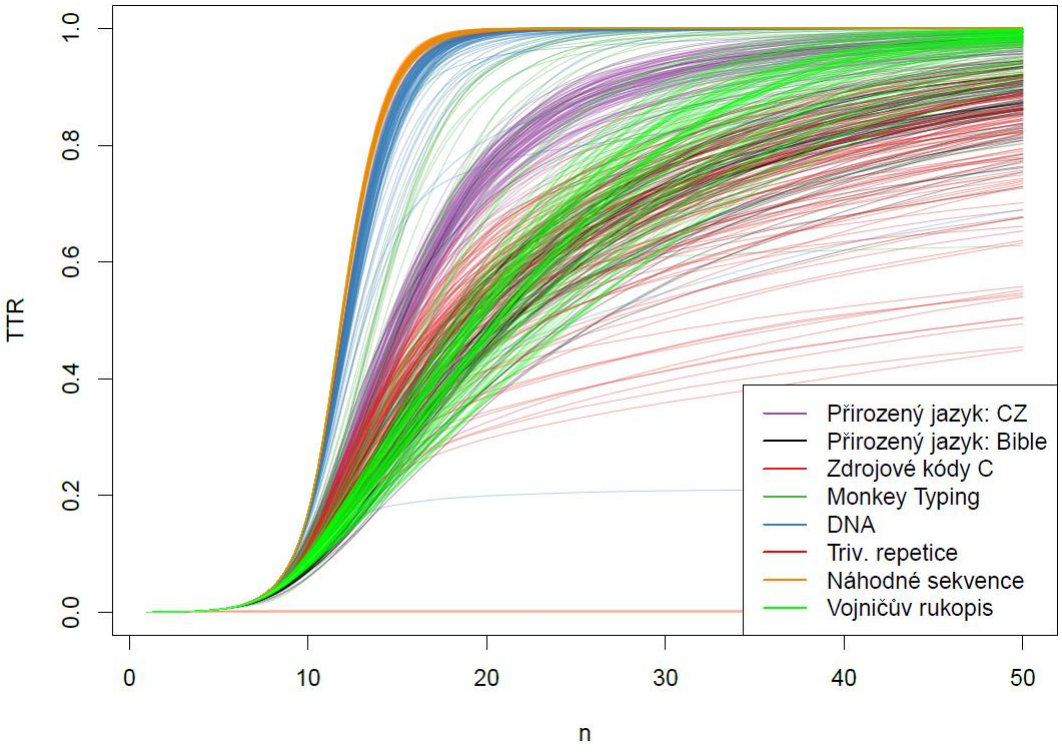
Vojničův rukopis je středověký rukopis napsaný v neznámém jazyce a neznámém písmu (viz náhled v obrázku 15), který prozatím uniká překladu. Autorství rukopisu bylo přisuzováno Rogeru Baconovi a rukopis měl být vlastněn např. i panovníkem Rudolfem II. (Manly 1931). Celý rukopis je rozdělen do kapitol s vlastními ilustracemi, pomocí kterých lze soudit přítomnost témat od lékařství, alchymie, chemie až k botanice ad. Rukopis byl studován kapacitami v oblasti kryptoanalýzy Williamem Friedmanem (americká armáda, později Národní bezpečnostní agentura NSA), Johnem Tiltmanem (1967; britská armáda)[14], Mary D'Imperio (1979; NSA)[15], Doris Miller (NSA; 1975)[16] nebo Prescottem Currierem (1976; NSA). Až překvapivý zájem o rukopis ze strany bezpečnostních služeb lze zřejmě přičítat strategické pragmatice uvažující existenci knihy, která může využívat dosud neznámou a historicky neprolomenou šifru využitelnou nepřítelem. Rukopis byl několikrát transkribován a digitalizován (např. právě Currierem nebo Takeshi Takahashim), což umožnilo následnou řadu statistických analýz (viz např. Landini 2001 nebo Reddy a Knight 2011) vedoucí k poukázání, že se jedná o přirozený jazyk blízký těm semitským. Oproti myšlence, že se jedná o šifrovaný nebo neznámý jazyk, existují i racionální domněnky, že je celý rukopis

[14] Dostupné online https://www.nsa.gov/news-features/declassified-documents/tech-journals/assets/files/voynich-manuscript-mysterious.pdf , cit 13. 7. 2018.
[15] Dostupné online https://www.nsa.gov/about/cryptologic-heritage/historical-figures-publications/publications/misc/assets/files/voynich_manuscript.pdf, cit. 13. 7. 2018.
[16] Dostupné online https://www.nsa.gov/news-features/declassified-documents/cryptologs/assets/files/cryptolog_12.pdf , cit 13. 7. 2018.

pouze propracovaným podvodem a náhodně vytvořeným textem s cílem získat peníze od panovníka Rudolfa II. Na možnost podvodu poukazuje především Schinner (2007) na základě analýzy kontextů slov, korelací vzdáleností slov a dalších vlastností. Je tedy zřejmé, že tento text není triviální a vede k protichůdným závěrům.



***Obrázek 15**: Ukázka písma Vojničova rukopisu (str. 91/46)*



***Graf 16**: Výsledky aplikace metody MKM na vzorky Vojničova rukopisu a rozšířeném datasetu popsaném v tabulce 16.*

K samotné analýze Vojničova rukopisu metodou MKM využijeme digitální transkripci od Takeshi Takahashi[17] ze které využijeme 100 náhodně vybraných podřetězců o délce 6 000 bitů tak, jako doposud, pro možnost porovnání. Jako referenční dataset budou sloužit sekvence uvedené v tabulce 16, tj. 100 náhodných sekvencí, 100 sekvencí českých beletrií, 100 sekvencí Biblí v různých jazycích, 21 sekvencí *monkey-typed* textů, 3 triviální repetice, 100 sekvencí zdrojového kódu jazyka C a 100 sekvencí proteiny kódující DNA. Metodu MKM aplikujeme opět na 1 až 50 gramy pro možnost porovnání s dosavadními výsledky.

Výsledné *embeddingy* TTR nejprve zobrazíme do klasického grafu křivek s osami x a y odpovídající velikosti n-gramu a hodnotě TTR. Na výsledek se podívejme v grafu 16. První, čeho si zde můžeme všimnout, je, že Vojničův rukopis rozhodně není dokonale náhodný a nejedná se ani o triviální repetici. Svým překryvem se vzorky Biblí různých jazyků a sekvencemi zdrojových kódu pak výsledky poukazují na strukturovanost celého rukopisu, což při pohledu na jeho strany snadno ověříme, vzhledem k vizuálně strukturovaným pasážím (viditelným právě i v obrázku 15). Tato strukturace však není vždy dodržena a přechází do volně psaného textu, což pravděpodobně na vyšších velikostech n-gramů zapříčiňuje podobnost s českou beletrií. Z průběhů jednotlivých sekvencí získáváme indicii, že se pravděpodobně jedná o strukturovaný text ne nepodobný Biblím či programovacímu jazyku a při zvážení širšího kontextu i méně strukturovaným přirozeným textům, jakými jsou beletrie. Rozhodně však nelze tvrdit, že by průběh sekvencí odpovídal jakémukoliv typu zdroje, který jsme doposud viděli. Pro velikosti n-gramů 10 až 30 je růst křivek srovnatelný s Biblemi, což lze vysvětlit explikovanou podobnou strukturací vytvářející časté opakující se vzory například prefixů a sufixů slov s mezerami a konci řádků. Právě opakování n-gramů vede k nižším hodnotám TTR. Překvapivá je však následná rychlost konvergence Vojničova rukopisu k hodnotám TTR blízkým jedné, a to překonáním Biblí přibližně na 30-gramech a následnému splynutí s trendem českých beletrií, což je zapříčiněno neopakováním a tedy unikátností 30-gramů a více. Nejsnazším vysvětlením takového průběhu je výrazná strukturace textu a krátkost slov či omezená kombinatorika symbolů, ze kterých se tato slova skládají. Krátké n-gramy budou v takových případech registrovat omezený počet prefixů a postfixů kombinovaných opět s omezeným počtem prvků vizuální strukturace (nové řádky, mezery ad.) a tím převažovat TTR k nízkým hodnotám. Krátká slova či omezená kombinatorika symbolů uvnitř nich budou tento efekt dále umocňovat. Ovšem u dlouhých n-gramů dochází k registraci kombinací těchto slov včetně vizuálních separátorů, což, jak jsme si mohli všimnout, vede k neopakování těchto n-gramů a tedy k hodnotám TTR blízkým jedné.

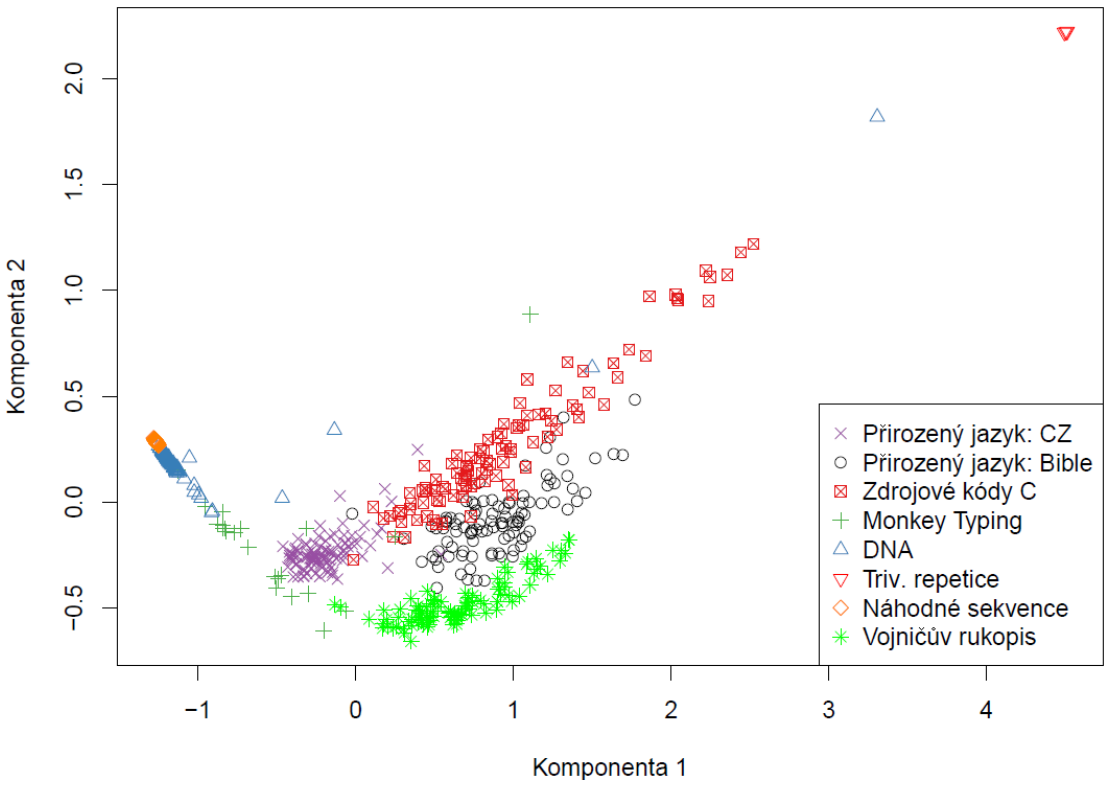
[17] Dostupné online: http://voynich.freie-literatur.de/index.php?show=extractor, cit 13. 7. 2018.

Takové pozorování je ovšem přirozené a očekáváme jej například u slabik tvořících slova atd. Problémem metody MKM je však to, že velikosti *n*-gramů v grafu bezprostředně nekorespondují s počty symbolů v textu, neboť jednotlivé symboly jsou normalizovány do binární podoby. Z grafu tak nelze okamžitě říci, zda uvedené délky *n*-gramů již zasahují uvažovaná slova oddělená mezerami nebo jen kombinace symbolů. Můžeme však zjistit, že abeceda Vojničova rukopisu transkribovaného Takeshi Takahashim čítá právě 24 symbolů (grafémů, vč. bílých znaků atd.), což znamená, že každý symbol byl překódován do řetězce o 5 bitech. To znamená, že 30-gramy zde uvažují šestice symbolů, zatímco 45-gramy devítice symbolů. Průměrná délka grafického celku, který můžeme v rámci rukopisu považovat za slova na základě užití mezer, má 5,18 symbolů (se směrodatnou odchylkou 1,93 symbolů). Vzorek české beletrie má pro srovnání (vzhledem k využití velkých a malých písmen, které zde považujeme za rozdílné a dalších) abecedu čítající 67 symbolů překódovaných pomocí 7 bitů, s průměrnou délku slova 4,93 symbolů (se směrodatnou odchylkou 2,7 symbolů). Což znamená, že u 45-gramů, u kterých oba zdroje sekvencí nabývají obdobných hodnot TTR, registrují velikosti *n*-gramů celky větší, než jsou průměrné délky jejich slov, tj. registrují kombinace slov.

Tyto výsledky poukazující na podobnost Vojničova rukopisu a přirozených textů však nejsou rozhodující. Shodného efektu bychom totiž teoreticky dosáhli i uměle, např. využitím tzv. Cardanovy mřížky (tj. mechanismu výběru náhodných slabik z tabulky pomocí posunující se karty s výřezy), která by vytvářela opakující se vzory uvnitř slov, zatímco jejich náhodné řazení by vedlo k absenci vzorů, a tedy k vysokým hodnotám TTR. K otestování této možnosti bychom však potřebovali dataset rozšířit právě i o texty uměle vytvořené touto metodou.

Na základě dosavadních výsledků tak zatím nemůžeme říci, zda je, či není Vojničův rukopis umělý text vytvořený pravidly s cílem imitovat přirozený jazyk, nebo zda se jedná o přirozený jazyk, byť i šifrovaný. Jak je následně pohledem do grafu 16 zřejmé, doplnění dalších typů sekvencí by pravděpodobně vedlo k horší čitelnosti celého grafu – již teď průběhy některých zdrojů sekvencí splývají a například průběhy Biblí nejsou kvůli překryvu se sekvencemi zdrojových kódů téměř viditelné. Tento problém jsme výše v návrhu metody předpokládali a ke grafu průběhů křivek jsme nabídli alternativu založenou na vícerozměrové interpretaci výsledných *embeddingů* metodou MDS. Tato metoda má tu výhodu, že by ve výsledku měla intuitivně a v jediném bodu shrnovat výsledky všech délek *n*-gramů sekvence a na základě snadno interpretovatelné vzdálenosti podat informaci o podobnosti (blízkosti) k ostatním sekvencím. Jak vidíme na výsledném grafu 17, je aplikace MDS skutečně pozoruhodnou alternativou a doplněním.

Graf 17 s výsledky metody MDS rekonstruuje 99,04 % původních vzdáleností mezi jednotlivými *embeddingy* a můžeme jej považovat za velmi věrné zobrazení vzdáleností z původního 50-rozměrového prostoru. V tomto grafu si můžeme všimnout lépe čitelné lokalizace sekvencí Vojničova rukopisu, než tomu bylo v případě grafu průběhů. Nyní jsou zcela explicitně separovány do vlastního regionu či shluku, který je nejblíže stovce vzorků Biblí a rovněž se setkává s několika vzorky *monkey-typed* textů. V tomto ohledu je nutné říci, že Bible jsou zde zastoupeny ve sto různých jazycích a viditelný rozptyl je u nich předpokládatelný. Také vidíme, že sekvence českého jazyka jsou ve svých vzdálenostech poměrně homogenní. Dále si můžeme všimnout, že sekvence zdrojových kódů, Biblí a Vojničova rukopisu tvoří pomyslná tři pásma, která se částečně překrývají, ale jinak stojí proti sobě. Zároveň můžeme pozorovat chybějící typ sekvencí mezi Vojničovým rukopisem, Biblemi, zdrojovými kódy a sekvencemi české beletrie. Všimnout si také můžeme určitého eliptického tvaru celého grafu připomínajícího podkovu: Tento tvar poukazuje na nedostatečnou rekonstrukci vícerozměrových dat do dvou rozměrů a implikující přesnou rekonstrukci vzdáleností pouze nejbližších bodů (Diaconis *et al*. 2008; k tomuto problému se včetně využití atlernativní vizualizační metody vrátíme později). Důležité je, že se sekvence shlukují dle svého typu do vlastních regionů, včetně sekvencí Vojničova rukopisu, a to bez užití jakýchkoliv metod zohledňujících jejich zdroj. Číselné reprezentace sekvencí, tj. jejich *embeddingy* vytvořené metodou MKM jsou tak pro sekvence ze shodného zdroje podobné, což je velmi dobré zjištění.



***Graf 17**: Výsledky aplikace metody MKM a MDS na vzorky Vojničova rukopisu a rozšířeného datasetu.*

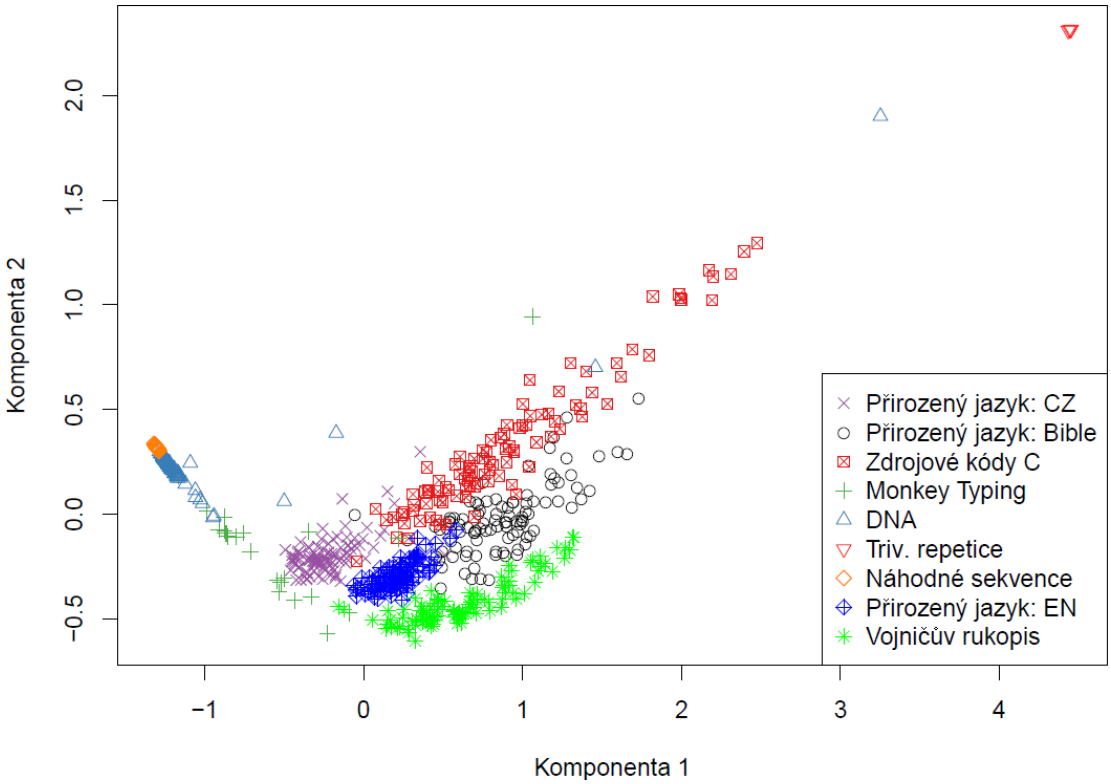
Výsledky z grafů 16 a 17 nám nabízejí indicii, že Vojničův rukopis je blízký přirozeným jazykům, a především vzorkům s vysokou strukturací textu jako Bible. Tato podobnost nás však vede k problematickému bodu: Sekvence Vojničova rukopisu jsou nejblíže právě Biblím a pohledem do grafu s MDS navíc zjistíme, že oba tyto typy sekvencí mají i obdobný rozptyl. Problematickým bodem je, jak už bylo řečeno, že sekvence Biblí jsou zastoupeny stovkou různých jazyků, zatímco u Vojničova rukopisu bychom očekávali jediný jazyk. Pozoruhodné naopak je, že na možnost užití více *jazyků* v rukopisu naráží i Currier (1976, 58 a 61-73), který spekuluje nad užitím alespoň dvou různých *jazyků*. Také musíme uvážit, že jako druhou referenci rozptylu přirozeného jazyka užíváme sekvence českých beletrií – je možné, že beletrie v jiných jazycích budou mít větší rozptyl a ty české tak zkreslují očekávání. Další alternativou vysvětlující poměrně velký rozptyl Vojničova rukopisu je možnost, že jde o jediný, avšak specificky zašifrovaný jazyk. Vzhledem k okolnostem a zájmu uvedených institucí lze tuto možnost snad vyloučit. Alternativním vysvětlením je, že rukopis je psán novým, uměle vytvořeným jazykem, což je možnost zvažovaná samotným Friedmannem (Tiltman 1967, 9) nebo se jedná o důmyslný podvod nebo o umění (např. Miller 1975 a Schinner 2007).

Pro užití umělého jazyka ovšem existuje téměř dobová opora, která se odvíjela od tehdejších znalostí prolomitelnosti substitučních šifer (Wilkins 1641, 88) a nabízela řadu jejích vylepšení. Tato vylepšení spočívala například v rozšíření abecedy, užívání zkratek slov, zanesení informací pouze do určitých částí textu identifikovatelných geometrickým tvarem či mřížkou a dále použitím vlastního umělého jazyka a vlastních znaků (Wilkins 1641, 97-118), tj. vylepšení, o kterých bychom mohli u rukopisu uvažovat. Pozoruhodné je i doporučení střídat různé jazyky (Wilkins 1641, 23). Uvedená Currierova spekulace nad užitím více jazyků i Friedmannova myšlenka užití umělého jazyka tak mohou být reálné, což by vysvětlovalo i relativně velký rozptyl jednotlivých sekvencí v grafu MDS. V tomto ohledu je pak porovnání výsledků Vojničova rukopisu s moderní hybridní šifrou GPG a náhodnými sekvencemi generovanými atmosférickým šumem ne úplně relevantní.

Pokud nahlédneme na Vojničův rukopis jako na možný uměle vytvořený text např. s cílem zisku peněz od tehdejšího panovníka, můžeme snadno spekulovat, že si autoři tohoto rukopisu byli dobře vědomi metod kryptoanalýzy substitučních šifer, a tedy i vlastností, které měl vykazovat zašifrovaný přirozený text. S takovou znalostí by následně mohli vytvořit metodu generování textu, která by vytvářela text z kvantitativního pohledu blízký přirozenému jazyku. Jak ukazuje Rugg (2004), lze za tímto účelem použít už zmíněnou Cardanovu mřížku, která byla zveřejněna v 16. století. Rugg zde dále uvádí, že výsledek generování textu může být natolik komplexní, že se bude podobat přirozenému jazyku.

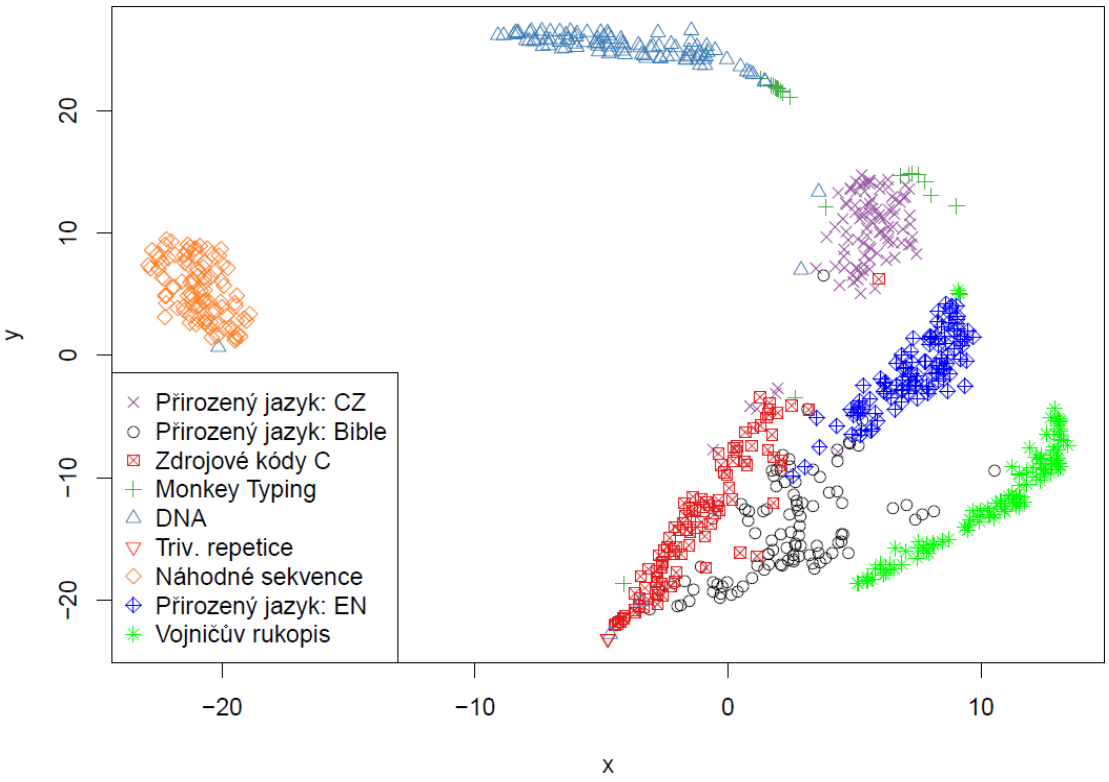
V tomto je právě zásadní problém. Pokud se jedná o uměle vytvořený pseudonáhodný text, který nemá žádný skutečný význam a byl navržen za účelem imitace přirozeného a třeba i šifrovaného textu, pak v této situaci a s dostupným datasetem není možné pomocí metody MKM rozhodnout, co je skutečně Vojničův rukopis zač. Pro další komparaci by tak bylo nutné dataset rozšířit právě o uvažované možné zdroje sekvencí a následně v grafech (či jinak) pozorovat jejich blízkosti. Tedy například do datasetu přidat sekvence, které imitují přirozený jazyk. Dále by se dalo uvážit hledání takových typů sekvencí, které by ohraničily region Vojničova rukopisu a tím diskriminativně určili i jeho typ.

Problémem takového přístupu je samotná diskriminativnost. Výsledky Vojničova rukopisu, které jsou v grafech 16 a 17, porovnáváme a charakterizujeme na základě námi dodaných referenčních sekvencí. V případě, kdy bychom dodali zcela irelevantní referenční data a zdroje, byl by i tak jeden z nich Vojničově rukopisu blíže. Z tohoto je zřejmé, že pro porovnání a charakterizování sekvencí metodou MKM je nutné použít diverzifikovaný a reprezentativní dataset s maximálně relevantními zdroji. I *aposteriorní* přidání zdrojů do datasetu nám může napomoci s charakterizací neznámých sekvencí. Ilustrací může být *aposteriorní* přidání sekvencí stovky vzorků jediné knihy anglické beletrie (Pán prstenů od J. R. R. Tolkiena). Jejich přidáním získáme další představu o tom, jaké rozdíly registruje metoda MKM uvnitř jediné knihy a v jediném jazyce. Poněkud překvapivý výsledek vidíme v grafu 18 rekonstruujícím 99 % původních vzdáleností. Nové vzorky anglické beletrie se umístily přímo do prázdného prostoru mezi stovkou různých českých beletrií, zdrojovými kódy, stovkou Biblí v různých jazycích a Vojničovým rukopisem. Za povšimnutí stojí i ohraničení zleva, které je tvořeno *monkey-typed* texty. Z přidání nového zdroje sekvencí tak získáváme další povědomí o určité stabilitě metody, a především o soběpodobnosti sekvencí jednotlivých zdrojů vedoucí k jejich shlukování a až překvapivé separovatelnosti. O samotném Vojničově rukopisu lze však dále říci jen málo.



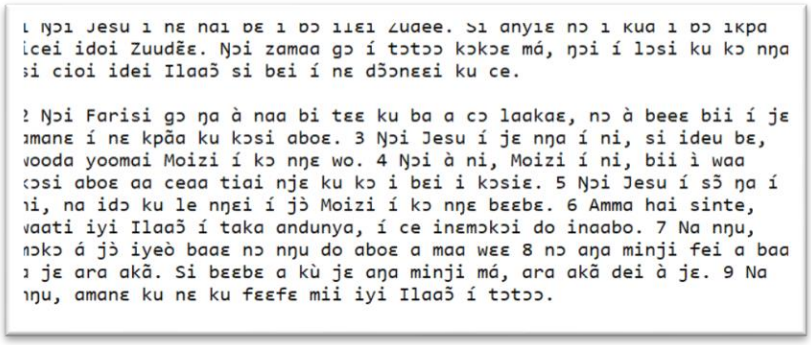
***Graf 18**: Výsledky MDS po a posteriorním přidání sekvencí anglické beletrie.*

Nejistota, která plyne z možnosti, že je Vojničův rukopis jen dostatečně komplexní imitací přirozeného jazyka nebo jde o komplexně šifrovaný a kódovaný přirozený jazyk, nás vede k diskuzi, co je reálným důkazem jednoho nebo druhého tvrzení, tj. k obecné otázce dostatečného důkazu, kterou řešili i D'Imperio, Miller a Currier (1976, 57). Je zřejmé, že nám metoda MKM (a zřejmě ani jí podobné metody) nedokáží pomoci identifikovat s jistotou zdroj sekvencí, ale dokáží nám podat alespoň určité indicie. V tomto ohledu je rovněž otázkou relevance využité metody MDS. Jak už bylo zmíněno výše, 1 % informací v grafu 17 chybí a zároveň v něm pozorujeme náznak problematické redukce rozměrů formou eliptického vzoru podkovy. Metod pro vizualizaci (respektive redukci) vícerozměrových dat je celá řada, nicméně klasické metody jako MDS (za použití euklidovské vzdálenosti), rozklad na hlavní komponenty (PCA) nebo rozklad na singulární hodnoty (SVD) jsou použitými mechanismy blízké a vedou k obdobným výsledkům. Nelineární alternativou k těmto metodám je metoda tSNE (*t-Distributed Stochastic Neighbor Embedding*; Maaten a Hinton 2008), která se od MDS liší zcela odlišným mechanismem, cíli a výsledky. Metoda tSNE nachází cíleně nejvhodnější vizualizaci vícerozměrových dat v zadaném počtu dimenzí tak, aby byly co nejlépe zachovány lokální blízkosti (podobnosti) analyzovaných objektů, zatímco jejich globální vztahy, oproti MDS, nejsou reflektovány a mohou být různě promíchány. Tento způsob vede především k přehlednější vizualizaci podobností sousedů. Umístění výsledků vůči osám grafu, oproti MDS, nelze využít k jejich interpretaci. Metodě tSNE jako vstup předáváme výsledné *embeddingy* sekvencí z metody MKM, výslednou vizualizaci podobností pak vidíme v grafu 19.



***Graf 19**: Výsledky vizualizace vektorů MKM pomocí metody tSNE.*

Výsledky vizualizace tSNE nám poskytují přehlednější pohled na podobnost jednotlivých sekvencí. Zřetelně například vidíme, že jedna ze sekvencí DNA je podobnější náhodným sekvencím než ostatním sekvencím DNA, které vytváří samotný shluk výše. Některé *monkey-typed* sekvence jsou zařazeny do shluku DNA (což je zajímavé vzhledem k fyzikálním či fyziologickým vlivům spojených s procesem tvorby obou sekvencí). Některé *monkey-typed* sekvence jsou podobné sekvencím českých beletrií. Českým beletriím jsou nejblíže vzorky anglické beletrie, kde na předělu mezi oběma jazyky nalezneme dokonce i jeden vzorek Vojničova rukopisu. Anglické beletrie zvolna přecházejí do štěpící se dvojice zdrojových kódů jazyka C a téměř paralelních sekvencí Biblí tvořících tvar pomyslného *šípu*, jehož vrcholem jsou triviální repetice a jedna ze sekvencí DNA. Vojničův rukopis je umístěn paralelně k anglické beletrii a Biblím. Výsledky tSNE jsou tak v tomto případě konzistentní s těmi z aplikace MDS, ovšem vystupují zde některé detaily, které byly u MDS skryty. Prvním z nich, který se týká Vojničova rukopisu, je blízkost k jedné z Biblí, která je zároveň od těch ostatních maximálně odlehlá. Konkrétně se jedná o Bibli v jazyce Mokole (Benin, Guinea, Sierra Leone Afrika), ukázka viz obrázek 19. Podobnost s textem Vojničova rukopisu, respektive jeho sekvencí, tkví pravděpodobně ve strukturaci a užitých krátkých slovech. Dalším detailem, který byl předtím skryt, je viditelné rozdělení Vojničova rukopisu téměř na dva shluky přibližně téměř v pomyslné polovině. Takové rozdělení může být dáno shodou okolností nebo může směřovat ke zmíněnému pozorování Curriera o dvou a více použitých jazycích. Vizualizace pomocí tSNE je přínosná v poskytnutí detailů, nicméně i tak nevede k žádné další odpovědi, která by objasnila Vojničův rukopis.



***Obrázek 19**: Ukázka Bible v jazyce Mokole.*

Ve výsledku získáváme o Vojničově rukopisu několik poznatků. Především lze říci, že jde o velmi specifický typ textu, který je v použitém datasetu unikátní. Tato unikátnost byla zřetelná již v grafu průběhů hodnot TTR, ve kterém jsme si všimli odlišného tvaru křivek a způsobu jejich konvergence. Unikátnost sekvencí byla následně potvrzena i jasnou separovatelností sekvencí rukopisu v grafech metod MDS a tSNE. O rukopisu dále můžeme říci, že je nejblíže strukturovaným přirozeným textům, ale krom jediné odlehlé sekvence jazyka Mokole jsou sekvence rukopisu separovány. Po srovnání si ale v grafu s MDS a tSNE všimněme, že sekvence českých beletrií jsou od sekvencí přirozeného jazyka Biblí rovněž separovány, přitom sekvence Biblí jsou blíže ke zdrojovým kódům programovacího jazyka C. Takový výsledek ovšem vede k otázce, jaké vlastnosti metoda MKM registruje, neboť z aktuálních výsledků a použitého datasetu je zjevné, že její výsledné charakterizace k sobě shlukují spíše obdobně strukturované sekvence než že by registrovala shodu v tom, že se jedná o přirozený jazyk. Tato otázka nás vrací zpět k tomu, co je v kontextu výsledků metody MKM vlastně Vojničův rukopis, neboť jeho popsaná unikátnost může spočívat prakticky jen ve specifické strukturaci a užívání mezer. Je proto zřejmé, že metoda MKM bez dalších rozsáhlých testů a bez rozsáhlejšího datasetu nedokáže o Vojničově rukopisu poskytnout více informací, než kolik je doposud známo. Ovšem v situaci, ve které bychom o Vojničově rukopisu neměli žádnou *apriorní* znalost (tj. byl poskytován anonymně a např. kódovaný), bychom zřejmě dokázali identifikovat, že se jedná o sekvenci blízkou strukturovanému textu přirozeného jazyka. Z výsledků je dále patrné, že výsledné *embeddingy* sekvencí založených na prostém indexu TTR dokáží shlukovat jednotlivé typy zdrojů, což je pro smysl celé metody, kterou je charakterizace neznámých sekvencí, klíčové a můžeme ji tak považovat za úspěšnou, a to nejen z hlediska výsledků, ale i výpočetní náročnosti a takové obecnosti algoritmu, který umožnil definovat i testy náhodnosti sekvencí.

Závěr metody MKM

V úvodu této kapitoly jsme jako hlavní cíl definovali odvození a otestování metody, která by umožnila charakterizovat neznámé, anonymní a zcela obecné sekvence symbolů. V této kapitole jsme proto uvedli, formalizovali a rozvedli novou metodu MKM. Její algoritmické pojetí jsme následně konfrontovali s již existujícími a podobnými metodami analýzy sekvencí, především s metodami z Rao *et al.* 2009, Rao 2010 a Rukhin *et al.* 2001. V testech i pojetí metody jsme reflektovali výtky vůči metodě Rao 2010 ze Sproat 2010 a 2014. MKM je tak novou metodou, která zároveň řeší nevýhody ostatních podobných metod a která zároveň stojí na těch nejjednodušších nástrojích kvantitativní lingvistiky umožňujících vysokou míru transparentnosti a odtud se odvíjející jednodušší interpretaci výsledků.

Pro metodu MKM jsme odvodili řadu vlastností a modelů, které napomáhají při interpretaci jejích výsledků. Rovněž jsme zde pro metodu MKM nalezli několik artefaktů, tj. několik neintuitivních vlastností vyplývajících přímo z principu jejího fungování, které jakýmkoliv způsobem zkreslují výsledná data, která by tak bez znalostí těchto artefaktů mohla být snadno desinterpretována. Stanovili jsme pro MKM několik způsobů vizualizace výsledků, které efektivně umožňují charakterizovat analyzované sekvence a na základě kterých jsme dále odvodili další vlastnosti této metody. Partikulární aplikací metody MKM je i možnost testovat náhodnost sekvence odvozeným statistickým testem. Tento test lze využít jako nutnou, avšak nedostatečnou podmínkou náhodnosti sekvence, což znamená, že metoda MKM není pro tento účel dostatečně striktní. Zajímavé ovšem je, že jím prochází generátor pseudonáhodných sekvencí RANDU, který projde i všemi testy standardu NIST (na což poukazuje Hamano a Yamamoto 2010, kde rovněž nabízí vlastní metodu testování náhodnosti). Odvození statistického testu přitom vedlo k dalším vhledům do samotné metody MKM a poskytlo nám větší porozumění jejím vnitřním principům.

Obecně je výsledkem metody MKM aplikované na konkrétní sekvenci vektor hodnot TTR pro jednotlivé velikostí *n*-gramů. Tyto číselné reprezentace sekvencí neboli *embeddingy*, jsme pro snazší porozumění z počátku interpretovali jako pomyslné body tvořené páry velikosti *n*-gramu a TTR, které jsme za pomocí křivek zobrazovali v grafu. Takové zobrazení nám umožnilo interpretovat výsledné vektory čistě vizuálně a bez nutnosti využívat klasické strojové klasifikační metody. Především jsme z takové vizualizace ale zjistili, že získané hodnoty TTR vytváří specifické průběhy, ze kterých lze odvodit některé konkrétní vlastnosti sekvence, např. její náhodnost, repetitivnost, strukturaci, pseudonáhodnost atd. Dalším důležitým zjištěním byla i podobnost těchto průběhů mezi sekvencemi ze shodného zdroje, které se v grafu shlukovaly do určitých pásem a u kterých jsme dále odvodili jejich obecné logické řazení od repetitivních sekvencí, přes přirozené texty až k těm náhodným.

Zřejmým problémem tohoto zobrazení byla jeho krajní nepřehlednost v případě, kdy bylo zobrazeno více typů sekvencí. Jako alternativu jsme proto k této prvotní a jednoduché vizualizaci nabídli využít vícerozměrové interpretace výsledných *embeddingů*, a tedy především využití metody vícerozměrného škálování. I zde jsme se setkali s určitými nevýhodami pramenícími především z příliš velkých globálních nepodobností mezi sekvencemi analyzovaného datasetu, které mohou být natolik velké, že v jejich porovnání ty menší, avšak stále důležité, zaniknou. Z tohoto důvodu jsme stručně představili a použili metodu tSNE, která rekonstruuje pouze lokální podobnosti a poskytuje náhled na nejbližší okolí za cenu ztráty globálních vztahů.

Prvotním testem a následnou ilustrativní aplikací metod MDS a tSNE na tzv. Vojničův rukopis jsme ověřili, že vícerozměrová interpretace výsledků metody MKM je průchozí a že je až překvapivě úspěšná. Jak jsme mohli vidět v grafu průběhů křivek, výsledné *embeddingy* reprezentující sekvence jsou shlukovány na základě jejich zdroje. Metody MDS a tSNE takové shlukování nejen přehledně ověřily, ale především poukázaly i na vizuální separabilitu těchto shluků vedoucí k náhledu na efektivitu reprezentace sekvencí metodou MKM tak, že i po redukci dimenzionality jsou testované sekvence separovatelné a téměř nepromíchané. Přitom je metoda stále dost obecná na to, aby zaznamenávala rozdíl mezi jednotlivými neekvivalentními sekvencemi a dokázala je s pozorovanou logikou shlukovat k sobě.

Z těchto výsledků lze předpokládat, že užitím vícerozměrových pravděpodobnostních modelů by bylo možné docílit i automatické a pravděpodobně úspěšné metody klasifikace. V tomto ohledu byla uvedená aplikace na Vojničův rukopis velmi přínosná, neboť jsme na jejím základě dokázali identifikovat potřebu využití alternativních metod vizualizace dat, které následně vedly k již popsaným pozitivním pozorováním vlastností metody MKM. Především jsme ale touto ilustrativní aplikací otestovali efektivitu celé metody a navržených postupů, včetně poukázání na problematiku interpretace jejích výsledků. Z grafu průběhů křivek jsme například určili, že sekvence navzorkované z Vojničova rukopisu jsou blízké strukturovanému přirozenému text. Dále jsme z těchto křivek odvodili možnost omezené kombinatoriky symbolů abecedy nebo užití krátkých slov. Z grafu MDS jsme následně pozorovali separovanost Vojničova rukopisu a jeho blízkost k sekvenacím Biblí, anglicky psané beletrii i k *monkey-typed* textům. Aplikace MDS především odhalila separované shluky typů sekvencí, jejich vnitřní podobnost či rozptyl mezi sekvencemi stejného typu, a především vzájemnou podobnost jednotlivých zdrojů.

Užití metody tSNE nám poskytlo detailnější náhled především na Vojničův rukopis, ve kterém byly pozorovány alespoň dva separované shluky hypoteticky odpovídající teorii dvou autorů rukopisu a jejich specifickému jazyku. Důležité je mít také na paměti, že Vojničův rukopis mohl být metodě MKM předložen kódovaný a ano-

nymně, přičemž bychom zřejmě získali podobné nebo ekvivalentní výsledky. Fascinující a důležité také je, že všechny testy, které v této práci na metodě MKM proběhly, byly prováděny na sekvencích dlouhých 6 000 bitů, což je přibližně 750 *písmen*, tj. přibližně 160 slov. Delší či kratší sekvence zde však nebyly testovány a jejich efektivita nebo neefektivita je tak věci dalšího bádání.

Je zřejmé, že metoda MKM má řadu vlastností, které je dále nutné objasnit nebo upravit. Především by bylo výhodné objasnit vztah mezi jednotlivými velikostmi *n*-gramů, neboť je zřejmé, že jejich výsledky na sobě budou závislé, nicméně není zřetelné, jakým způsobem a ani jak tento vztah aktuálně formálně pojmout.

Stejně tak je zřejmá souvislost výsledků velikostí jednotlivých *n*-gramů na způsobu strukturace textu. Například si můžeme představit rozdíl mezi výsledky metody MKM na sekvencích jednolitého českého textu bez mezer, nových řádků a interpunkce, který bude stát proti české výkladové encyklopedii, kde se krátké pasáže textu střídají s dlouhými pasážemi, tabulkami a ilustracemi. Jednotlivé velikosti *n*-gramů v takovém případu budou mít odlišný vztah: shodné velikosti *n*-gramů budou pro obě sekvence potenciálně znamenat registraci odlišných celků, které se budou v případě encyklopedie velmi rychle diverzifikovat. Hodnoty TTR různých délek *n*-gramů budou strukturou velmi ovlivněny a registrací odlišných celků v nich může docházet i k náhlým zvratům.

Ovlivnění výsledků pouhou strukturací textu je poměrně důležitou otázkou a bude nutné ji detailněji prozkoumat, ať už například experimentálně, na přirozených textech s odstraněnými symboly vizuální strukturace, nebo formálně. Další důležitou otázkou je využití indexu TTR. Hlavní výhoda tohoto indexu tkví v jeho jednoduchosti a z ní plynoucí transparentnosti, nicméně poskytuje jen velmi hrubý náhled na opakování slov, který může skrýt důležité detaily.

Příkladem takového opomenutí detailů mohou být dvě sekvence *S a W*, obě čítající 100 slov o délce 2 bity. Potenciální slovník obou sekvencí obsahuje 2^2 slov, přičemž sekvence *S* i *W* tento slovník využijí celý, tzn. obě sekvence realizují typy = { 00, 01, 10, 11 }. Sekvence *S* jednotlivými typy rovnoměrně realizuje všech 100 tokenů, tzn. každý z nich je využit právě 25x. Sekvence *W* naopak realizuje 97 tokenů typem 00 a zbylé tři tokeny realizuje chybějícími typy 01, 10 a 11. Problémem je, že obě sekvence *S* i *W* mají shodné TTR = 4/100.

Taková hrubost v měření je zřejmě dalším z důvodů shody hodnot TTR a grafických průběhů netriviálních sekvencí v intervalu popsaném jako interval vyčerpání. Zároveň jde zřejmě o důvod, proč je test náhodných sekvencí natolik hrubý, že jím prochází prakticky bez povšimnutí popsaný kongruentní generátor RANDU. Samozřejmě lze namítnout, že tento problém řeší a registrují následující *n*-gramy. Nicméně lze uvážit i alternativní způsob kvantifikace opakování slov namísto TTR, například

využitím normalizované entropie nebo Giniho koeficientu, které by reflexí pravděpodobností jednotlivých typů dokázaly odlišnost uvedených dvou sekvencí odhalit. Můžeme proto předpokládat, že užitím entropie či Giniho koeficientu by nabyla metoda MKM větší přesnosti, což je další věcí, kterou je vhodné při dalším bádání otestovat.

Na druhou stranu může být právě určitá hrubost TTR důvodem pozorovaného a téměř ideálního způsobu shlukování sekvencí pocházejícího ze stejného zdroje, a to vzhledem k volnosti, kterou TTR umožňuje, což nás vede k zajímavé reflexi, že tato volnost je dostatečně velká na to, aby pojmula sekvence ze stejného zdroje a zároveň dostatečně malá na to, aby zamezila větší kontaminaci shluku ostatními typy sekvencí. Index TTR se tak může pro obecnost metody MKM prozatím jevit jako velmi výhodná volba.

Využitím entropie namísto TTR se následně dostáváme zpět k metodě z Rao 2010, která by se následně od metody MKM lišila v jediném (avšak jak jsme ukázali zcela kritickém) bodu, a to ve způsobu zohlednění velikosti abecedy. Ilustrovaná chyba metody Rao spočívala v nekorektní práci se sekvencemi obsahujícími různě velké abecedy za pomoci normalizace entropie, která vedla k ilustrované nepodobnosti náhodných sekvencí kódovaných dvěma různými abecedami. Metoda MKM tento problém řeší normalizací sekvence na binární a uvedené náhodné sekvence jsme byli schopni identifikovat jako shodné. Při použití entropie by tak metoda MKM byla stále od metody Rao odlišná. Užití entropie nebo Giniho koeficientu by ovšem znamenalo nutné úpravy a rozšíření jednotlivých modelů, které by nově musely uvažovat pravděpodobnosti každého typu zvlášť, namísto jejich pouhého počtu.

Důležité ovšem je, že metoda MKM i přes zmíněné nedokonalosti dokáže shlukovat sekvence ze stejného zdroje, a to včetně jejich vizuální separace, což bylo jejím hlavním cílem a smyslem. Metodu MKM lze na základě dosavadních zjištění uvažovat k identifikaci nebo charakterizaci různých typů a zdrojů. Při jejím vývoji a testování jsme se dále setkali s řadou zdrojů sekvencí, především se sekvencemi přirozeného jazyka, náhodnými sekvencemi a sekvencemi DNA.

Studium náhodných sekvencí nám poskytlo poměrně zajímavý vhled do jejich důležitosti v mnoha oblastech dotýkajících se každodenního života. Sekvence DNA, se kterými jsme se setkali v referenčním datasetu při testování Vojničova rukopisu, jsou ovšem dalekosáhle důležitější. Tzv. kódující DNA jsme při analýze metodou MKM nacházeli ve shlucích blízkých náhodným sekvencím, *monkey-typed* textům a v některých extrémních případech i zdrojovému kódu nebo triviálním repeticím (viz grafy 18 a 19). Paradoxně tyto sekvence kódují kriticky důležité stavební jednotky organismů a jejich podobnost k náhodným či pseudonáhodným sekvencím značí neregistrování obsažených vzorů metodou MKM (obdobně, jako neregistruje obsažené vzory u sekvencí kongruentního generátoru).

Nad podobností sekvencí DNA s *monkey-typed* texty jsme se dokonce krátce pozastavili, především vzhledem k tomu, že oba typy sekvencí jsou při své tvorbě určitým způsobem omezovány fyzickými faktory. Podobnost DNA se sekvencemi triviálních repetic pak můžeme vysvětlit nejsnáze tak, že některé sekvence mohou obsahovat právě triviální repetice vytvářející specifické fyzické funkce svými chemickými vazbami. Pokud odhlédneme od přirozených jazyků, náhodných a pseudonáhodných sekvencí právě směrem k sekvencím DNA, které se na základě konfrontace s výsledky metody MKM jeví jako určitá směs náhodných, pseudonáhodných a triviálních zdrojů, zjistíme, že se jedná o dalekosáhlou a komplexní problematiku, ve které lze pro jejich detailnější studium využít další poznatky z kvantitativní lingvistiky. V následující kapitole se proto setkáme s úvodem vztahu lingvistiky a oborů studujících genetický kód od molekulární biologie, genetiky a bioinformatiky a nastíníme možnost vnímat tento kód v jiných rovinách než těch běžně uvažovaných.

Analýza sekvencí genetického kódu

V této kapitole se blíže seznámíme se sekvencemi DNA, tedy s lineárními biologickými sekvencemi symbolů A, C, T a G, které uchovávají plány výstavby živých organismů a řídící jejich vnitřní procesy. Jak uvidíme, tyto sekvence nejsou jen určitým druhem biologických *textů*, ale svou komplexitou, využitím vnitřních pravidel a arbitrárních kódů vytvářejí biologickou analogii k přirozeným i formálním jazykům. Principiální blízkost biologických a přirozených textů ilustruje především množství metod, které jazykověda (a aplikovaná disciplína *zpracování přirozeného jazyka* NLP) sdílí společně s obory studujícími DNA. Jak už bylo zmíněno v úvodu této práce, jde například o metody extrakce informací z textů jako je Latentní Dirichletova Alokace (pro NLP nezávisle odvozeno v Blei, Ng a Jordan 2003 a v genetice Pritchard a Donelly 2000), modelování pomocí skrytých markovovských procesů přiřazující slovům slovnědruhové kategorie a v bioinformatice užívaných při predikci genů, využití editačních vzdáleností, jako např. Damerau-Levensthein, pro porovnání blízkosti slov a v biologii pro porovnání sekvencí proteinů. Více sdílených metod obou oborů nalezneme v Yandell a Majoros 2002 nebo Bolshoy *et al*. 2010.

Jak uvidíme podrobněji dále, na genetických textech byla testována řada kvantitativně-lingvistických poznatků. Porozumění textům či sekvencím, ať už na straně jazykovědy nebo na straně biologie, přináší užitek oběma oborům, ať už z hlediska kooperace ve způsobech explanace sdílených jevů nebo i z prosté znalosti jejich přítomnosti na odlišných materiálech. V této kapitole proto nahlédneme na sekvence DNA detailněji skrze kvantitativní lingvistiku, a především skrze jeden z jejích nejznámějších zákonů, tj. tzv. Zipfův zákon, pomocí kterého se zaměříme na teoretickou aplikaci zkoumání kombinatoriky DNA vedoucí k přehodnocení pojetí používaných jednotek a jejich hierarchie. I přes jistou teoretičnost této kapitoly nalezneme hned v následující kapitole aplikaci, která by bez tohoto teoretického posunu zřejmě nenastala. Následující text je převzat z Matlach a Faltýnek 2016.

Genetický kód

Počínaje popsáním struktury DNA mluví biologie, nově vznikající molekulární biologie a obecně věda a společnost o genetickém kódu (viz k tomu Watson a Berry 2003). Dnes je genetický kód všeobecně intuitivně potvrzovaným vědeckým poznatkem. S předpokladem existence genetického kódu dnes bezprostředně vnímáme živé bytosti a biosféru. Na tento kód se pohlíželo s jistými předpoklady: Byly formulovány některé jeho základní vlastnosti a postupem času se pevně ustavilo nahlížení na to, jaký design tento kód má. Výše jsme odkázali k publikaci ozřejmující okolnosti objevu struktury DNA, jejímž autorem je J. Watson. Vlastnosti DNA a genetického kódu

ve svých publikacích a přednáškách představoval, vysvětloval a popularizoval Francis Crick. Využijeme jeho textů jako reprezentativních pro představení dnes běžného pohledu na genetický kód. Vlastnosti genetického kódu Crick formuluje následovně.

Genetický kód k sobě vztahuje aminokyseliny (z nichž se skládají proteiny) a báze, které obsahuje DNA (Crick 1962, 11–12; Crick 1968, 367). V procesu výstavby proteinu se k sobě vztahuje 20 aminokyselin a 64 trojkombinací bází, tzv. tripletů (Crick, 1968, 368). Tento kód je univerzální a až na výjimky, které ale nevybočují z principu vztahu bází a aminokyselin, je společný všem živým organismům (Crick 1962, 8; Crick 1968, 369).

Báze jsou v genetickém kódu spojeny s aminokyselinami arbitrárním vztahem. Ze všech možných aminokyselin je použito jen konkrétních dvacet, které se vztahují k bázím na základě zprostředkování jistými molekulárními prostředky (tzv. adaptorem – tRNA; Crick 1967, 342–343), přičemž mezi aminokyselinami a bázemi není přímá chemická afinita, mluví se zde o zmrzlé náhodě (Crick 1968, 369). Genetický kód je redundantní, a to v tom smyslu, že více tripletů odpovídá jedné aminokyselině. Některé triplety naopak vztah k aminokyselinám nemají a slouží pouze jako signály vymezující hranici začátku a konce využitelné genetické informace. Relevantní roli v tripletech hrají především první dvě báze. Třetí pozice v tripletu často umožňuje variovat báze, aniž by došlo k záměně aminokyseliny. Umístění bází v tripletu tedy není náhodné. Crick (1968, 369) tuto systematičnost detailně popisuje.

Zápis bází je lineární, čte se v jednom směru a bez možnosti přeskakování bází (Crick 1966, 332–333; Crick 1964, 9). Má pevný čtecí rámec, v němž zachovává hranice tripletu. Zároveň se nepřekrývá (z aj. *overlapping*), to znamená, že nenese více informací současně (např. Trifonov ale kód chápe jako překrývající se, důvody popisuje s dalšími spoluautory v práci Popov *et al.* 1996, 66; viz k tomu především Trifonov 1988, 508–510). Proces výstavby proteinu, tzv. proteosyntéza, probíhá (informačně a energeticky) směrem od bází k proteinům. Tento princip je nazván centrální dogma.

Uvedený popis genetického kódu byl dále precizován objevy molekulární biologie. Proces proteosyntézy byl popsán s mnoha dalšími proměnami původního řetězce DNA (do hnRNA, mRNA, v souvislosti s interakcemi s snRNA atd.) a procesy směřujícími k finálním produktům proteosyntézy a jejich funkcím. Byly popsány procesy sestřihu, jejich variace mezi organismy a částmi organismů, konformační procesy proteinů, metylace řetězců DNA, chromatinové interakce atd.

Genetický kód je vnímán jako lineární zápis bází vztahující se k tvaru a funkci proteinu. Báze jsou v praxi charakterizovány jako písmena (Crick 1967, 331; Crick 1962, 16). Tato písmena tvoří triplety (trojice bází), které jsou pojímány jako slova, tzv. kodony – kódová slova. Tato slova mají kódem zprostředkovaný vztah ke konkrétní aminokyselině. Soubor tripletů (kódových slov) tvoří gen, jednotku,

která má vztah k celému proteinu, jeho tvaru a z něj plynoucí funkci v organismu (Crick 1962, 8; Crick 1964, 2). O bázích jakožto písmenech a tripletech jako slovech se hovoří např. v následujících publikacích (vybíráme reprezentativní doklady pouze pro ilustraci): Stanford (1975, 74) ve svých Základech biofyziky hovoří o tripletech jakožto třípísmenných slovech, Weaver (2002, 569) ve své Molekulární biologii říká, že kodony jsou kódová slova a skládají se ze tří písmen, Twyman (1998, 205) ve své syntéze molekulární biologie označuje báze jako písmena, triplety jako slova a geny jako věty a stejně tak se vyjadřují i Hartl a Ruvolová (2013, 10).

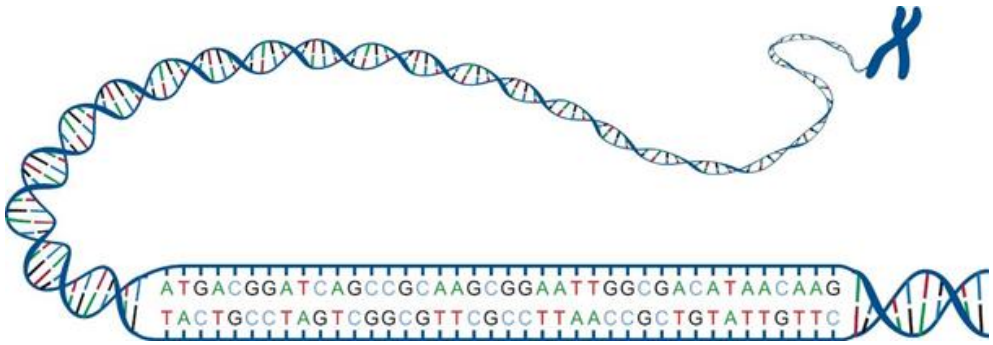
Genetický kód a přirozený jazyk

O genetickém kódu se vždy uvažovalo ve vztahu k přirozenému jazyku. Vědní obor molekulární biologie je pevně svázán s pojetím DNA a proteosyntézy jakožto analogie přirozeného jazyka. Analogie molekulárně genetických procesů s jazykem jsou v molekulární biologii do jisté míry instrumentální, představují tradiční přístup k terminologii. Částečně ale analogie s jazykem molekulární biologii zprostředkovává přístup ke genetickému kódu, jeho struktuře a funkci. Užívání jazykové metafory v molekulární biologii názorně ukázal Raible (2001). Ten provedl korpusové šetření na desítkách tisíc molekulárně genetických textů, z něhož jasně vyplývá, že termíny analogické k popisu přirozeného jazyka (písmeno, slovo, čtení, zápis, překlad atd.) jsou široce používány v běžné praxi této vědy. Searls (2002) ukázal, že molekulárně genetický výzkum nesdílí s lingvistikou jen terminologii, ale i výzkumné metody. Jakobson (1971, 655–696) dokonce potvrdil molekulární biologii korektnost využívání analogie genetického kódu a jazyka a hovořil přímo o struktuře genetického kódu, který se dle něj skládá z písmen (bází), slov (tripletů) a vět (genů). Dále poukazuje na to, že v genetickém kódu nacházíme vlastnosti jako je synonymie, suprasegmentální nebo syntaktická delimitace, systém distinktivních rysů či pružná stabilita. Jakobsonův jazykový výklad genetického kódu je pak přejímán dál (např. Katz 2008). O jazykové metafoře v biologii referuje Markoš a Faltýnek (2011).

V tomto textu chceme ukázat, že běžně přijímaný design genetického kódu, jak jsme jej představili výše, je možné zpochybnit. Domníváme se, že pojetí struktury genetického kódu, a to především v jazykové analogii, je od prvopočátku chybné. Chceme jej odmítnout, a to na základě popření analogie bází a písmen. Ze sémiotického hlediska jsme to již udělali (viz Faltýnek, 2012). K tomuto účelu využijeme metodu kvantitativní analýzy textu, kterou představíme níže. Nejdříve ale čtenáře seznámíme se základním instrumentáriem molekulární biologie, které je nutné k vyložení našich závěrů.

Instrumentárium molekulární biologie

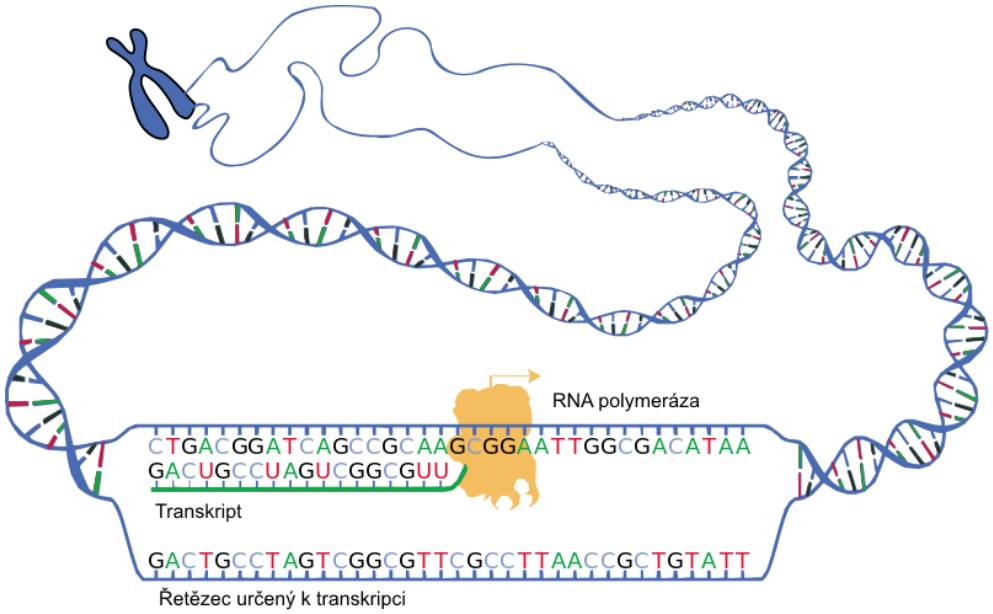
Dědičná informace, obsahující instrukce k výstavbě organismu a řízení jeho biologických pochodů, je fyzicky zapsána v každé buňce ve formě deoxyribonukleové kyseliny mající podobu dvoušroubovice a známé pod zkratkou DNA. Způsob, jakým DNA uchovává informace, je založen na principu variací čtyř specifických makromolekul nukleových kyselin, konkrétně thyminu (T), guaninu (G), adeninu (A) a cytosinu (C). Střídáním těchto tzv. bází dochází k záznamu informace obdobně, jako když Morseova abeceda zaznamenává informace střídáním teček a čárek. Každá z bází má svůj chemicky afinitní (vzájemná vazba vodíkovými vazbami) protějšek, thymin stojí v DNA vždy proti adeninu a guanin proti cytosinu.



Obrázek 1: Dvoušroubovice DNA s jejími dvěma rameny nesoucí jednotlivé báze. A značí adenin, T thymin, G guanin, C cytosin. Dvoušroubovice DNA zaujímá strukturu v tzv. chromozomu (viz: <http://www.genome.gov/dmd/img.cfm?node=Photos/Graphics&id=85259>).

Lineární zápis bází DNA nám dovoluje celou DNA přečíst a přepsat formou textu, tj. zapsat zleva doprava, písmeno za písmenem (viz např. Cvrčková 2006, 17). Takový přepis je kopií řetězce bází zkoumané DNA. Tento postup je v praxi nazýván jako sekvenování (detaily viz in Berg *et al*. 2012, 140–148). Sekvenováním je řetězec DNA zprostředkován ke zkoumání nejrůznějšími nástroji.

Proces, kterým se z lineární sekvence DNA (tedy z určitého textu) stane protein – reálný fyzický nástroj využitelný v organismu – označujeme jako proteosyntézu. Celý tento proces můžeme popsat v několika krocích (viz např. Alberts *et al*. 2008, 329; Weaver 2002, 39):

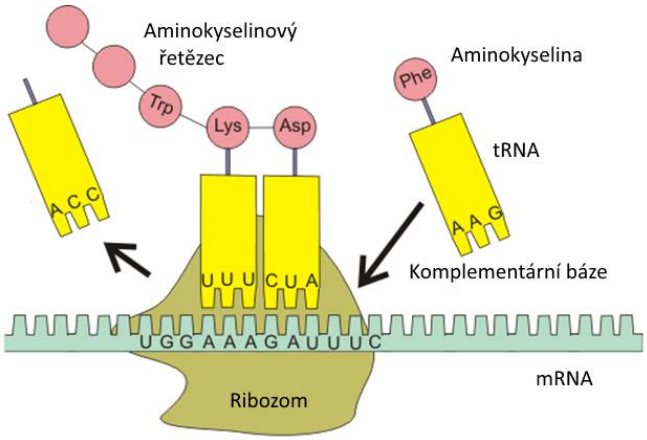


Obrázek 2: Transkripce. Přepis sekvence DNA na komplementární protějšky jejích bází RNA polymerázou, thymin je přepisován na uracil, zbylé báze na své komplementární protějšky (viz: <http://commons.wikimedia.org/wiki/File:DNA_transcription.svg>).

1. Transkripce. Po naplnění specifických podmínek uvnitř buňky se na začátek sekvence DNA (tzv. genu) připne protein RNA-polymeráza, který se po této sekvenci pohybuje v zadaném směru a nukleotid po nukleotidu tuto sekvenci přepisuje. Výsledkem je tzv. transkript – samostatná „pracovní kopie" DNA ve formě ribonukleové kyseliny RNA, která je určena k zamýšlenému použití v proteosyntéze.

2. Úprava transkriptu. Transkript může být dále upraven (například vystřihnutím částí, které jsou v sekvencích vloženy a slouží jiným účelům). Finální verze transkriptu, tzv. mRNA, je pak přemístěna k ribozomu, kde je konstruován protein.

3. Translace. Ribozom čte mRNA lineárně po trojicích nukleotidů (tripletech). Každému tripletu mRNA je donesena pro něj specifická aminokyselina, která je připojena k předchozí. Takto vytvořený aminokyselinový řetězec se při výstupu z ribozomu začne na základě fyzikálních vlastností jednotlivých konstituentů a fyzikálních vlastností molekul v prostředí (tedy na základě určitého kontextu) formovat, dokud nevytvoří stabilní konformaci (tvar) proteinu. Funkce a vlastnosti proteinu jsou determinovány fyzikálními vlastnosti jeho makromolekulární konformace (Twyman 2004, 103).



Obrázek 3: Translace. Proces vzniku aminokyselinového řetězce proteinu. Tripletům mRNA je na ribozomu přiřazována tRNA s komplementárním tripletem, která nese aminokyselinu. Takto přiřazené aminokyseliny tvoří řetězec budoucího proteinu (viz: <Boumphreyfr/Wikipedia>).

Tripletům je aminokyselina přiřazena pomocí zprostředkujícího elementu – tzv. adaptorové molekuly tRNA. Adaptor tRNA získává svůj tvar při transkripci z DNA (Weaver 2002, 51). Funkcí této adaptorové molekuly je vázat na jedno ze svých vazebných míst specifický triplet a na své druhé vazebné místo konkrétní aminokyselinu. Vztah aminokyselin a nukleotidů je tak zapsán přímo v DNA. Popsaný vztah byl nazván jako genetický kód (Weaver 2002, 12).

Genetický kód je tvořen variacemi čtyř různých nukleotidů v každé pozici tripletu. Triplet může nabývat 43 (64) možných unikátních kombinací, které tak mohou kódovat 64 různých aminokyselin. Všemi šedesáti čtyřmi realizovanými triplety je kódováno dvacet různých aminokyselin (Alberts et al. 2008, 367), mnoho tripletů kóduje stejnou aminokyselinu (kód je tzv. degenerovaný, viz obrázek 4). Některé triplety mají využití jako označení počátku a konce kódující sekvence.

1. pozice	2. pozice U	2. pozice C	2. pozice A	2. pozice G	3. pozice
U	Phe	Ser	Tyr	Cys	U
	Phe	Ser	Tyr	Cys	C
	Leu	Ser	stop	stop	A
	Leu	Ser	stop	Trp	G
C	Leu	Pro	His	Arg	U
	Leu	Pro	His	Arg	C
	Leu	Pro	Gln	Arg	A
	Leu	Pro	Gln	Arg	G
A	Ile	Thr	Asn	Ser	U
	Ile	Thr	Asn	Ser	C
	Ile	Thr	Lys	Arg	A
	Met	Thr	Lys	Arg	G
G	Val	Ala	Asp	Gly	U
	Val	Ala	Asp	Gly	C
	Val	Ala	Glu	Gly	A
	Val	Ala	Glu	Gly	G
			Aminokyseliny		

Obrázek 4: Genetický kód. Umístění bází v první, druhé a třetí pozici tripletu. Zkratky (Phe, Leu atd.) označují aminokyselinu kódovanou daným tripletem (viz: <www.genome.gov>).

Pojmenování vztahu bází řazených v mRNA a aminokyselin jakožto genetického kódu se vztahuje k jednomu z klasických pojmů lingvistiky. Pojem kód vyjadřuje vztah dvou veličin daný určitým územ (Monod 1970, 159–160). Od pojmenování genetický kód se konzistentně odvíjí i další názvosloví – kromě samotného kódování aminokyseliny tripletem (kodonem) je celý proces tvorby aminokyselinového řetězce podle nukleotidového vzoru při proteosyntéze pojmenován jako translace. Ta se staví do opozice vůči jednoduchému přepisu vzájemně chemicky afinitních molekul při transkripci.

Lingvistická paralela

V instrumentáriu jsme popsali strukturu DNA a proces proteosyntézy. V souvislosti s tím jsme představili také tradiční molekulárně-biologickou terminologii. Tato terminologie často využívá lingvistických termínů (transkripce, translace, kód, text) nebo termínů založených na jazykové metafoře (zápis, čtení, zpráva, informace (neterminologicky)). Jakobson (1971) potvrzuje, že využití jazykové metafory v molekulární biologii je korektní a že genetický kód má vlastnosti přirozeného jazyka. Analogizuje báze s písmeny (respektive fonémy), triplety se slovy a geny s větami. Nachází mnohé další společné vlastnosti genetického kódu a přirozeného jazyka. Ji (1999, 412) postupuje dále a analogizuje širokou škálu vlastností genetického kódu a přirozeného jazyka: písmena s nukleotidy a aminokyselinami, slova s geny, řetězce slov se souborem společně exprimovaných genů. Dále k sobě vztahuje gramatiku a fyzikální a chemické zákony, fonetiku a řízení energetického toku, sémantiku a genově řízené procesy v buňce. V případě obou kódů explicitně hovoří o dvojí artikulaci. Ji v nalézání protějšků procesů v buňce a konceptů popisujících přirozený jazyk představuje extrémní případ. Jeho přístup sugeruje, že libovolnému lingvistickému konceptu lze nalézt odpovídající proces či strukturu v buňce.

Trifonov (1988) popisuje soustavy kódů zajišťujících interakci DNA, RNA a proteinů, u jiných autorů můžeme nalézt další obdobné metafory (viz např. Barbieri, 2002; Collado-Vides, 1992; 1993; Markoš, 1997; Markoš, 2002).

Diskurz těchto znakových popisů procesů v buňce je rozvíjen biosémiotikou, mladou vědní disciplínou. Problém současné biosémiotiky spočívá v tom, že výše zmíněné a mnohé další znakové přístupy k buňce jsou vzájemně nekonzistentní. Každý autor rozvíjí specifický přístup a neexistuje jednotná metoda, která by platnost těchto přístupů ověřovala. Na příkladu Jiho lze vidět, že analogizovat přirozený jazyk a genetický kód lze libovolně, přičemž posouzení korektnosti takových analogií není snadné.

Motivace

Metafory a analogie nám mohou poskytnout nadhled nad určitou problematikou. To je ale v kontextu jazykových metafor a analogií DNA problematické. Nevíme, které z těchto metafor jsou relevantní a užitečné a které nikoliv. Popis procesů v buňce využívá jazykovou metaforu a analogii. Uvažování genetiků, bioinformatiků, makromolekulárních biologů, biochemiků a dalších tak může být ovlivněno zavádějící metaforou. Z epistemologického hlediska by korekce těchto metafor měla pro jejich uživatele velký význam. Cílem tohoto článku je představit experimentální metodu, která by mohla ověření některých ze zmíněných metafor umožnit a zároveň poskytnout vhled do struktury genetického kódu.

Metodika kvantitativní analýzy DNA

Pro analýzu nukleotidových sekvencí reprezentovaných zápisem v textu jsou užívány různé lingvistické metody a kvantitativní analytické přístupy. Představíme některé z nich a ukážeme, jak se vztahují k naší metodě analýzy struktury genetického kódu. Pokusíme se o využití těchto metod pro podložení nového designu genetického kódu.

Mantegna et al. (1995, viz též Havlin et al. 1995) analyzovali projevy Zipfova zákona na kódující a nekódující DNA. Kódující DNA dle Mantegni *et al*. Zipfův zákon vykazuje. Nekódující DNA projevy Zipfova zákona vykazuje také, ale pouze do určité míry. Mantegnova analýza byla motivována poznatkem, že pouze malá část (pro homo sapiens uvažováno 5,33 %; Mantegna *et al*. 1995, 2940) genomu je kódující, a tedy nese informaci k výstavbě proteinu (viz naše instrumentárium výše). Zbylá část genomu nemá takovou jasně zadanou funkci a od šedesátých let se pro ni zažil termín junk DNA. Nekódující DNA měla být v genomu historicky neseným reliktem bez využití v organismu (viz např. Watson – Berry 2003, 253; Palazzo 2014; Mantegna *et al*. 1995 o junk DNA píše jako o silent DNA).

Mantegna *et al*. (1995, 2949) dále tvrdí, že se nekódující DNA podobá v některých vlastnostech přirozenému jazyku; viz též Niyogi – Berwick, 1995). Mantegna *et al*. (1995) mluví o tom, že nekódující DNA nese určitý jazyk, z hlediska jeho redundance oproti kódující DNA dokonce bližší přirozeným jazykům (tím Mantegna *et al*. rozšířili analogie DNA a přirozeného jazyka, o nichž hovoříme níže, a opět uplatnil jazykovou metaforu DNA). Tato zjištění Mantegnu et al. vedou k hypotéze, že nekódující DNA má také funkci, kterou prozatím neregistrujeme a nepopisujeme, a tedy že nekódující DNA je nějakým způsobem použita pro uchování informací „biologických struktur" (Mantegna *et al*., 1995, 2949). Pozdější rozvoj molekulární biologie dal Mantegnově domněnce za pravdu (viz Alberts *et al*. 2008, s. 31–42; The ENCODE Project Consortium 2012, 57).

Potvrzení výskytu Zipfova zákona u kódující DNA odpovídalo tomu, že kódující řetězec nese informaci k výstavbě funkčního tvaru proteinu, tj. určité struktury s určitou funkcí v organismu. Analogicky k tomu se v textech přirozeném jazyce projevuje Zipfův zákon z důvodů naplňování určité funkce textu (to se můžeme pokusit vysvětlit např. v souvislosti s informační strukturou textu zajišťující přenos signálu prostředím a výrazovou a obsahovou strukturou a soudržností textu; viz Zipf 1949, 19–47).

K Mantegnově et al. analýze je ale nutné poznamenat následující: V analýze Mantegna et al. využívají dlouhé řetězce nekódující DNA (delší než 50 tisíc bází). Nekódující DNA sice informaci nese, ale nese také množství reliktních řetězců bez využití (nekódují protein ani se nepodílejí na regulaci proteosyntézy). Projekt ENCODE (2012) odhaduje, že až 80 % nekódující DNA má funkční využití. Zbylých 20 % muselo Mantegnovu et al. analýzu ovlivnit, a to proto, že v jeho analyzované nekódující DNA musely být obsaženy složky regulace proteosyntézy a také reliktní DNA (pro niž můžeme stále používat termín junk DNA a která obsahuje mnoho repetitivních a z informačního hlediska redundantních sekvencí). Tato kontaminace by pak posilovala hodnocení nekódující DNA jako podobné přirozenému jazyku z hlediska redundance.

Mantegna et al. byli při zacházení s nekódující DNA postaveni před následující problém: koncept genetického kódu přisuzuje kódující DNA strukturní roli tripletů (viz vztah tripletů a aminokyselin popsaný výše). Nekódující DNA však takové striktní ohraničení a priori přisuzovat nelze, funkční roli zde mohou zastávat sekvence o různé délce. Proto se pro kvantitativní analýzu kódující i nekódující DNA rozhodl využít techniku tzv. n-gramů (tuto techniku využívají např. také Bolshoy *et al*. 2010, 26). Představíme ji na ilustračním příkladu.

Mějme následující řetězec ABCDEFGH. Tento řetězec segmentujeme na 3-gramy, jimiž jsou: ABC, BCD, CDE, DEF, EFG, FGH; 4-gramy mají podobu: ABCD, BCDE, CDEF, DEFG, EFGH. N-gramová analýza tedy segmentuje řetězec tak, že postupuje lineárně jednotku po jednotce a delimituje vždy v řetězci následující n-tici (n-gram). V analýze přirozených textů n-gramová segmentace postupuje bez registrace hranic slov či vět. Sekvence „Zipfův zákon“ je rozdělena na tyto 5-gramy hlásek: zipfů, ipfův, pfůvz, fůvzá, ůvzák atd. Při analýze kódující i nekódující DNA jsou podobně zanedbány jakékoliv dříve stanovené hranice. N-gramový přístup tak dovoluje analyzovat řetězec nezávisle na jeho vnitřní strukturaci tím, že registruje jednotlivé sousedící prvky. Jednotlivé n-gramy představují v analýze analogie slov, která nejsou vydělena mezerou, ale hranicí délky n-gramu (zipfů, ipfův, pfůvz atd. představují slova vstupující do analýzy).

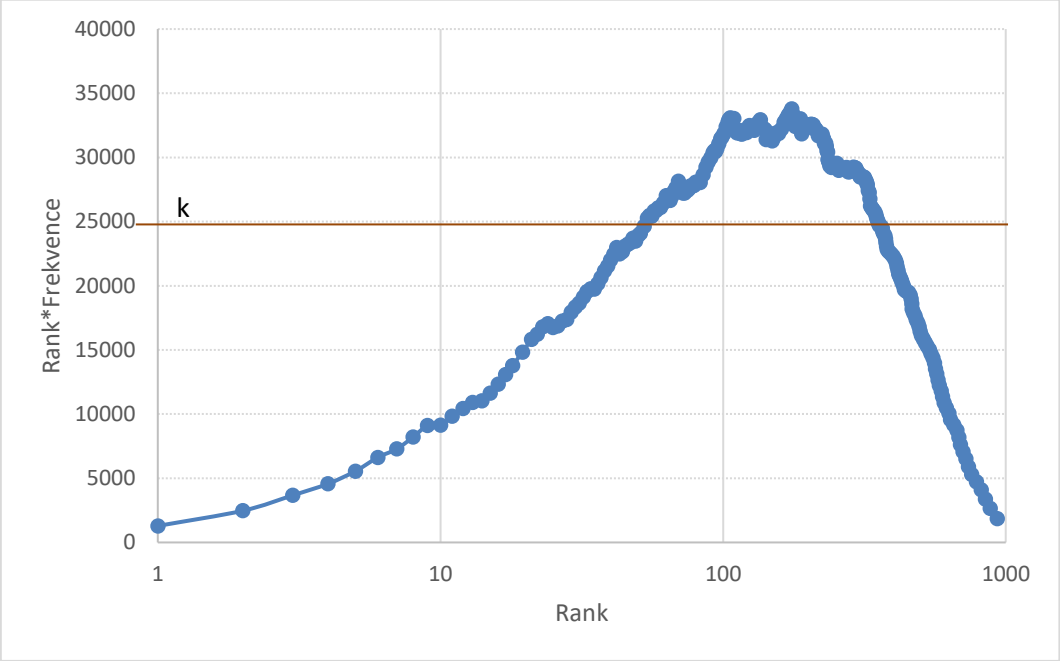
Tento přístup zvolili k oběma typům DNA Mantegna et al.. My tímto způsobem postupujeme také, a to z toho důvodu, abychom se vyhnuli apriornímu určení hranic kódovaných složek řetězce (viz dále). Jednotky delimitované n-gramovou technikou budeme stejně jako Mantegna et al. analyzovat z hlediska projevů Zipfova zákona. Výsledky analýzy nás mají vést k potvrzení aktuálního designu genetického kódu, nebo případně k jeho odmítnutí a následné reformulaci.

Je však nutné poznamenat, že využití Zipfova zákona v naší analýze může být sporné. Kvantitativní charakteristiky textu, jako jsou projevy Zipfova zákona nebo n-gramové analýzy, mohou být jako důkaz o určité vlastnosti či povaze tohoto textu (je kódující / nese informaci / nese instrukci; znaková funkce v genetickém kódu) chápány jako nepřímé. Např. Konopka (1995) referuje o závěrech Mantegnova týmu a doplňuje poznámky, které mohou být analogicky vztaženy i k našim závěrům. V prvním případě vyvstává problém s tím, že projevy Zipfova zákona jsou identifikovány na mnoha jevech, jako je trh, velikost měst nebo biologických populací atd. Zipfův zákon se pak jeví jako epifenomén jakéhokoliv systémového chování fyzikálních, sociálních, biologických apod. soustav. Zipfův zákon se tedy nemusí vztahovat ke kódujícím funkcím řetězce, ale naopak k jiným jevům jeho konstrukce. V případě řetězce bází by to mohla být jejich kombinatorika daná termostabilitou jednotlivých bází. V případě přirozených textů by se mohlo jednat např. o fonetické důvody kombinovatelnosti hlásek, které představují typ systematizace.

Zipfův zákon byl dokonce napadán v obecném rozměru, a to na základě jeho projevů na náhodných a nenáhodných textech (Li, 1992); diskuse v této oblasti zahrnuje problém generování náhodného vzorku a to, že mechanismus tvorby náhodného vzorku může produkovat projevy, které se z důvodů ne zcela náhodného generování textu přiblíží Zipfovu zákonu – to by opět potvrzovalo domněnku, že Zipfův zákon je projeven libovolného systémového, auto/regulovaného chování). Stále je ale vnímán jako projev znakovosti, kódovosti či jazykovosti. Na základě něj se hodnotí např. dorozumívání zvířat nebo struktura textu pacientů s postižením způsobujícím jazykový deficit (Ferrer-i-Cancho, 2006; Ferrer-i-Cancho – McCowan, 2009; Ferrer-i-Cancho – Elvevåg, 2010). I přes veškeré zmíněné výhrady použijeme v naší analýze Zipfův zákon, navážeme tak na lingvisticko-kvantitativní diskurz ověřování kódové povahy DNA, jako je tomu u Mantegni et al. a dalších. K tématu Zipfova zákona a DNA viz Tsonis et al. 1997.

Zipfův zákon na přirozených textech

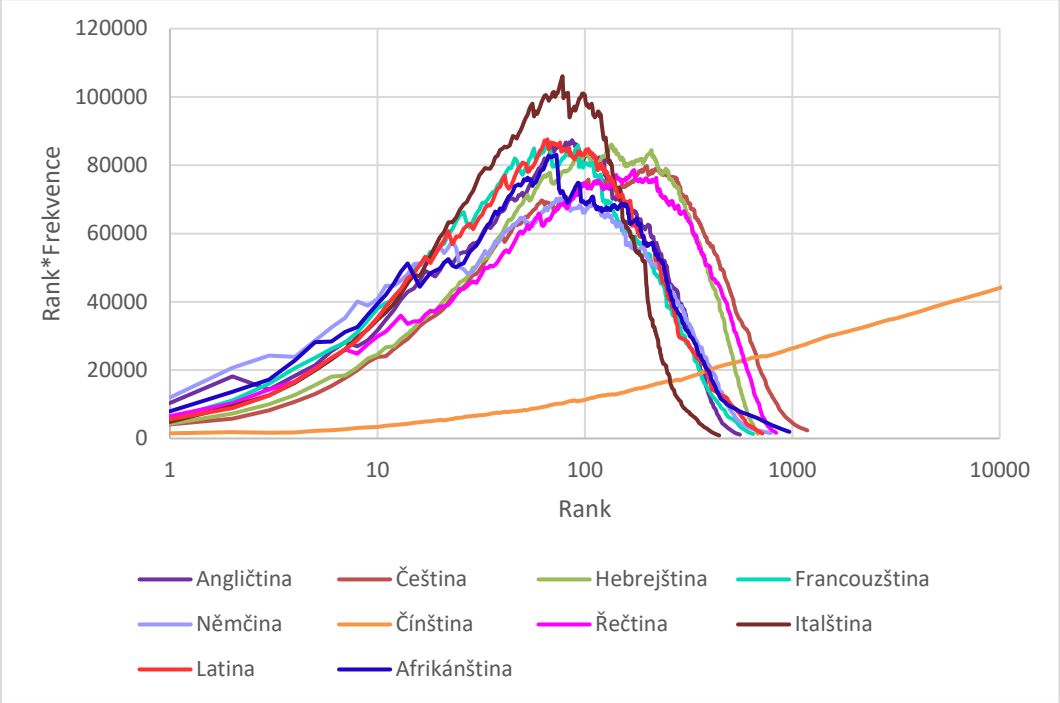
Zipfův zákon formuluje následující vztah: vezmeme-li určitý text a seřadíme-li počty výskytů (neboli frekvence) jeho entit (např. slov) od nejvyšší po nejnižší, pak pokud vynásobíme frekvenci každé této entity jejím pořadím (tzv. rankem), bude se výsledek *p* vždy blížit určité hodnotě reprezentované tzv. konstantou *k* (Zipf 1949, 22–25). Jak si ale můžeme všimnout na Obrázek 5, výsledky násobků ranků a frekvencí jsou u přirozených jazyků značně proměnlivé. Pomocí vodorovné úsečky proto v obrázku ilustrativně zobrazujeme konstantu *k*, jak ji vyjadřuje Zipfův zákon – reálné hodnoty násobků ranků a jejich frekvencí jsou od ní různě vzdáleny.



Obrázek 5: Vztah ranku a násobku ranku a frekvence písmen českého textu a ilustrace konstanty *k* Zipfova zákona. Text: M. Viewegh – Účastníci zájezdu.

Experimentální metoda založená na Zipfově zákonu spočívá ve sledování průběhů grafů hodnot násobků ranků a frekvencí entit textu (tj. spočívá ve vizuální komparaci průběhů grafů a registrování jeho vlastností, např. konkávnost, konvexnost, strmost, charakter maxim, definiční obory, monotónnost atd.; prozatím jsme neaplikovali žádnou formální metodu vyjádření podobnosti grafů, použitá kritéria jsou však pro naše účely dostačující). Tyto entity vybíráme z jedné konkrétní jazykové roviny textu – písmena, slova, věty apod. Klíčovým aspektem této metody a naší experimentálně ověřenou tezí je, že se u různých přirozených jazyků jednotky konkrétních jazykových rovin (písmena, slova, věty apod.) projevují podobně. Máme-li pak text v neznámém jazyce či neznámém zápisu, můžeme díky této metodě identifikovat jazykovou povahu jeho jednotek. Připomínáme jen, že je při této analýze využita n-gramová technika. Příklad

uvádíme na obrázku 6. Na něm můžeme sledovat projevy Zipfova zákona u deseti různých jazyků. Studovanými jednotkami jsou zde dvojice písmen textu (2-gramy; nejsou registrovány žádné spřežky, pracuje se s nimi jako s kombinací jednotlivých písmen, např. spřežka *ch*. Tečky, mezery, pomlčky atd. nejsou registrovány, registrována jsou pouze písmena). Všechny texty byly před analýzou redukovány na stejný počet znaků (270 000). Čeština, angličtina, němčina, afrikánština, latina, italština, francouzština, řečtina a hebrejština mají obdobný průběh grafu, čínština se od ostatních průběhů výrazně liší. Z tohoto pozorování můžeme usuzovat, že znaky čínštiny mají oproti ostatním zkoumaným jazykům zcela jinou jazykovou roli.



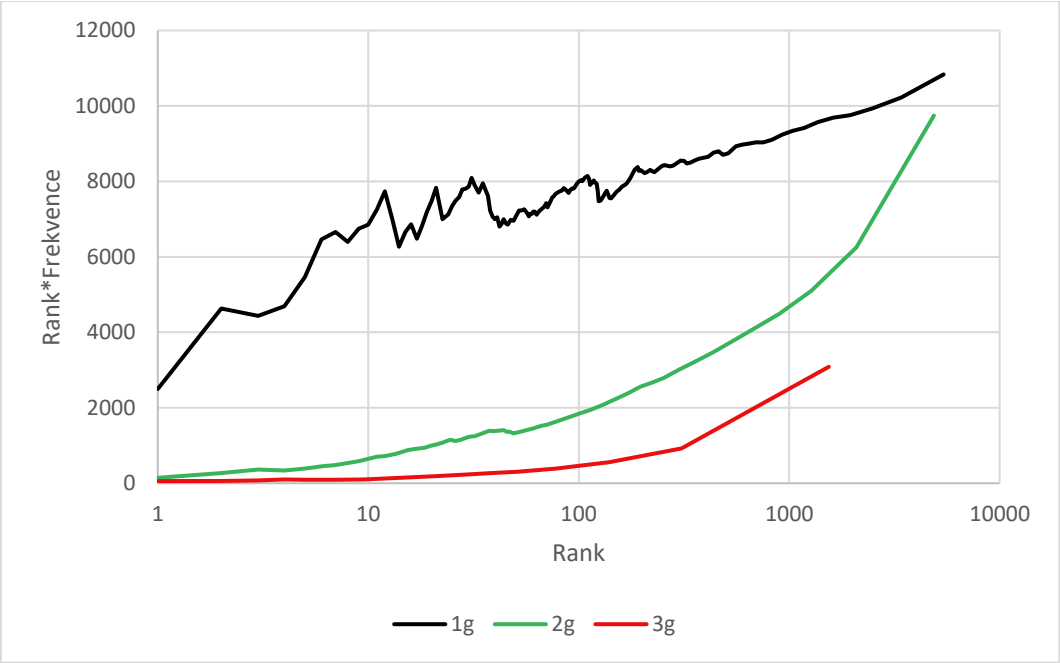
Obrázek 6: Projev Zipfova zákona na 2-gramech písmen textů vybraných jazyků. Texty: afikánština <18203-8>; řečtina <28658-0>; hebrejština <8cewa10>; latina <27219-0>; němčina <30695-0>, italština <28910-8>; francouzština <15371-8>; čínština <25350-0>; angličtina. J. R. R Tolkien – The Lord of the Rings; čeština M. Viewegh – Účastníci zájezdu. Kód ve špičatých závorkách je identifikátor textu volně stažitelného v projektu Guttenberg (<www.gutenberg.org>).

V předchozím odstavci jsme užili naši experimentální metodu na vzorku textů různých jazyků segmentovaných v tomto případě na 2-gramy písmen. Náš experimentální postup však provádí stejnou analýzu za užití 1-gramů, 2-gramů, 3-gramů atd. (experimentálně jsme ověřili signifikantnost nejvýše 20-gramů písmen). Tento způsob analýzy nám umožňuje sledovat, jak postupně se zvětšující n-tice odrážejí strukturu textu, a to v určité kontinuitě proměn podoby grafu. Postupné zvětšování n-tic nám dává možnost sledovat strukturu textu na stále vyšších jazykových rovinách.

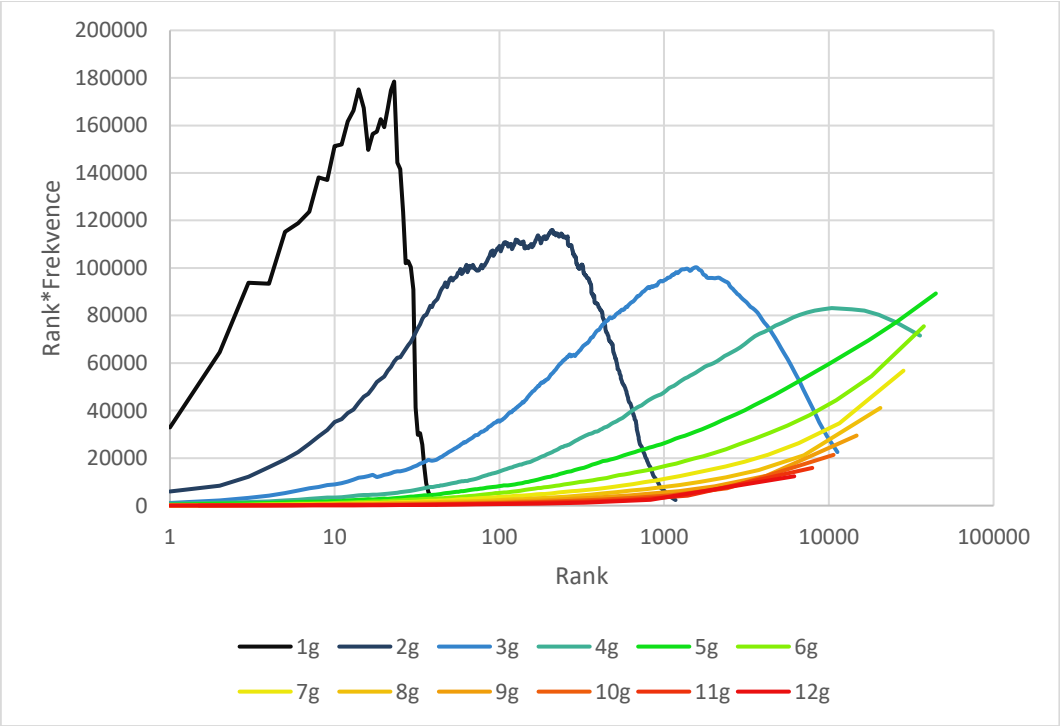
Pozorujeme tak chování písmen (jakožto stanovené základní úrovně popisu), kombinací písmen, slov (daných jejich průměrnou délkou v určitém jazyce, pro češtinu je to 5-gram písmen; viz dále), dále vět atd. Všechny tyto jednotky jsou reprezentovány jako n-gramy písmen. Tento experimentální postup je plausibilní, v mnoha analýzách jsme ověřili, že průběhy grafů n-gramů odpovídajících průměrné délce dané jednotky v určitém jazyce (např. slov) jsou signifikantně podobné průběhům grafů těchto jazykových jednotek textu. Tuto proceduru nazýváme mapování.

Výsledkem procedury mapování určitého textu je x průběhů (zobrazujeme je do jednoho grafu), které můžeme použít pro porovnání s výsledkem procedury mapování jiného textu. Porovnáním se míní zjištění podobnosti jednotlivých navazujících n-gramových průběhů u obou textů. Například pokud známe jazykové jednotky určitého textu, který dále zmapujeme, můžeme použít výsledný graf jako referenci k porovnání s výsledným grafem jiného textu, u kterého neznáme povahu jeho jazykových jednotek. Tímto způsobem se můžeme pokusit o jejich určení. Konkrétní příklady srovnání českého a čínského textu uvádíme níže.

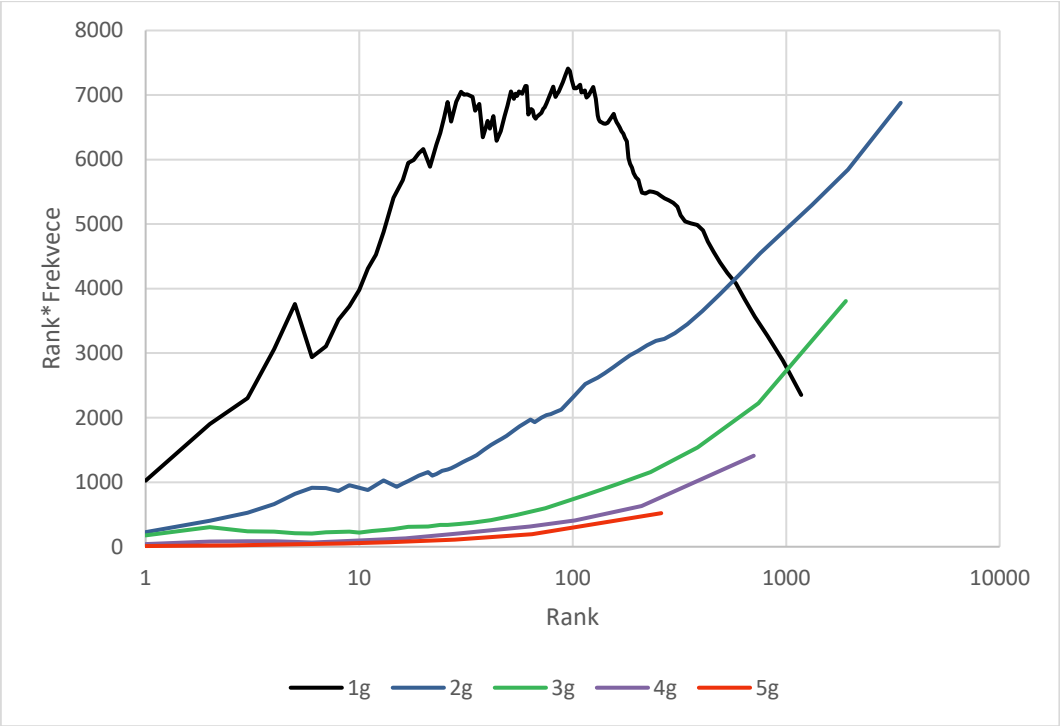
První graf (obrázek 7) ukazuje mapování slovních forem českého textu. Druhý graf (obrázek 8) ukazuje mapování písmen rovněž českého textu. Na základě korelace jejich průběhů zjišťujeme, že průběh 1-gramů slov se podobá průběhu 4-gramů až 5-gramů písmen. Průběhy následujících n-gramů (u slov 2-gramy a vyšší, u písmen 5-gramy a vyšší) si svými průběhy také odpovídají. Korektnost korelace 1-gramů slov s 4-gramy a 5-gramy písmen je zajištěna také podobností vývoje grafů v obou mapováních, nikoliv pouze podobností dvou konkrétních průběhů grafů. K nalezené hranici 4-gramů až 5-gramů písmen můžeme poznamenat, že průměrná délka českého slova, jak jsme experimentálně zjistili (na vzorku 243 českých literárních textů velikosti od 140 slovních forem do 149 070 slovních forem), má hodnotu 4,768 písmen. Srovnání obou uvedených mapování této hodnotě odpovídá, n-gramová analýza identifikuje hranici počtu písmen odpovídající slovům.



Obrázek 7: Mapování slovních forem českého textu. Text: M. Viewegh – Účastníci zájezdu.



Obrázek 8: Mapování písmen českého textu. Text: M. Viewegh – Účastníci zájezdu.



Obrázek 9: Mapování znaků čínského text. Text: čínština <25350-0>. Kód v ostrých závorkách je identifikátor textu volně stažitelného v projektu Guttenberg (<www.gutenberg.org>).

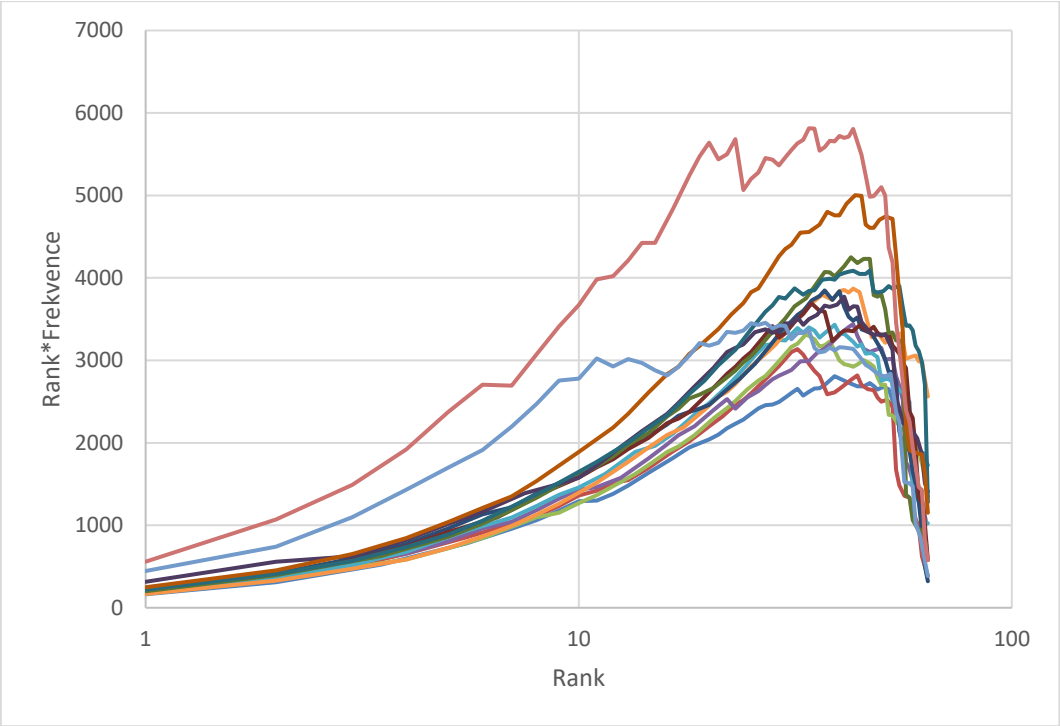
V obrázku 9 je zobrazeno mapování znaků čínského textu. To srovnáme s předchozím mapováním písmen českého textu (obrázek 8). Sledujeme-li průběhy grafů mapování čínského textu, identifikujeme podobnost průběhu 1-gramů znaků čínského textu s 2-gramy až 3-gramy písmen českého textu. Podobně 2-gramy (a vyšší) čínského textu pak odpovídají průběhu grafů 5-gramů (a vyšších) písmen, respektive 2-gramů slov českého textu. Zjišťujeme tak, že znaky čínského textu nemají povahu písmen, ale spíš jejich dvoj až trojkombinací, což odpovídá charakteru čínského znakového písma. Na základě výše stanovených kritérií porovnání průběhů grafů je korektnost tohoto závěru opět potvrzena.

Ověření metafory DNA

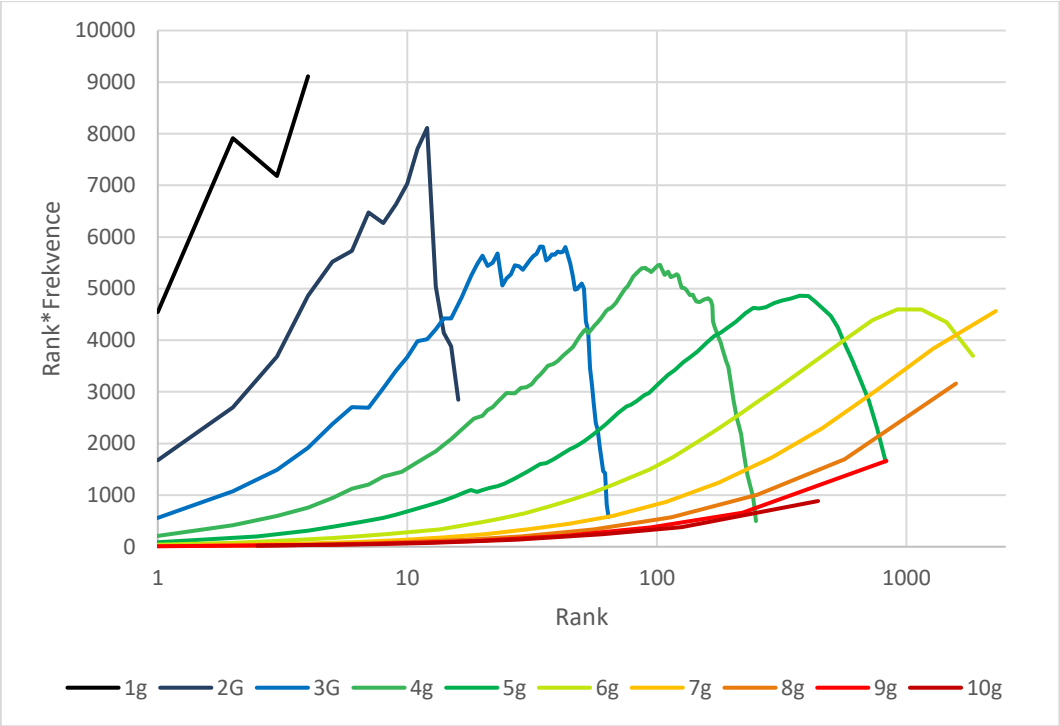
Výše uvedenou metodou mapování budeme analyzovat sekvence mRNA. Představili jsme standardní pojetí genetického kódu a různé analogie DNA a jazyka, včetně společných metod jejich zkoumání. Nejužívanější z analogií DNA a jazyka je pojetí bází DNA jakožto písmen – následně tripletů jakožto slov a genů jakožto vět (Jakobsonova metafora). Aplikací naší metody mapování vzorků sekvencí mRNA chceme ve srovnání s mapováním textů přirozeného jazyka prověřit, zda báze DNA a konsekventně mRNA hrají ve struktuře genetického kódu obdobnou roli jako písmena ve struktuře přirozených jazyků.

Aplikace metody

Výše představenou metodou bylo zmapováno přibližně 1000 náhodně vybraných mRNA sekvencí homo sapiens (EMBL-EBI, 2014; použité mRNA sekvence mají různou délku, experimentálně jsme ověřili, že délka textu nemá vliv na mapování, pouze na obor hodnot, který není kritériem porovnání grafů). Zvolenou jednotkou je v této analýze báze. Aplikací naší metody (testováno na 1-gramech až 20-gramech) jsme zjistili signifikantní podobnost grafů různých sekvencí (viz např. obrázek 10 s grafy 3-gramů). Pro názornost grafu mapování však uvádíme analýzu pouze jedné náhodně vybrané sekvence (obrázek 11). Pro přehlednost ještě zopakujme, že obrázek 10 ukazuje 3-gramovou analýzu Zipfova zákona více vzorků mRNA a obrázek 11 1-gramovou až 10-gramovou analýzu Zipfova zákona jedné konkrétní sekvence.



Obrázek 10: Projev Zipfova zákona 3-gramů bází 20 náhodně vybraných sekvencích mRNA z analyzovaného vzorku. Text: ENA <AAA59187> 1 Homo sapiens (human) ras GTPase-activating-like protein, ENA <AAA59483> 1 Homo sapiens (human) epiligrin alpha 3 subunit, ENA <AAA59486> 1 Homo sapiens (human) laminin B1, ENA <AAA60554> 1 Homo sapiens (human) sodium channel alpha subunit, ENA <AAA59504> 1 Homo sapiens (human) lactase phlorizinhydrolase, ENA <AAA18895> 1 Homo sapiens (human) voltage-gated sodium channel, ENA <AAA51901> 1 Homo sapiens (human) calcium channel L-type alpha 1 subunit, ENA <AAA35629> 1 Homo sapiens (human) calcium channel alpha-1 subunit, ENA <AAA51898> 1 Homo sapiens (human) N-type calcium channel alpha-1 subunit, ENA <AAA15448> 1 Homo sapiens (human) DNA polymerase epsilon catalytic subunit, ENA <AAA60225> 1 Homo sapiens (human) protein tyrosine phosphatase zeta-polypeptide, ENA <AAA18639> 1 Homo sapiens (human) p300 protein, ENA <AAA59866> 1 Homo sapiens (human) mannose 6-phosphate receptor, ENA <AAA59924> 1 Homo sapiens (human) GAP-related protein, ENA <AAA58965> 1 Homo sapiens (human) collagen type VII, ENA <AAA52700> 1 Homo sapiens (human) heparan sulfate proteoglykan. Kód v ostrých závorkách je identifikátor textu volně stažitelného v genové bance EMBL-EBI (<www.ebi.ac.uk>).



Obrázek 11: mRNA. Text: ENA <AAA52700> AAA52700 1 Homo sapiens (human) heparan sulfate proteoglykan. Kód v ostrých závorkách je identifikátor textu volně stažitelného v genové bance EMBL-EBI (<www.ebi.ac.uk>).

Věnujme se nyní obrázku 11, který porovnáme s obrázkem 8 zobrazujícím mapování písmen českého textu. Můžeme si všimnout, že průběhy grafů mapování jsou si podobné od 3-gramů bází u DNA a 1-gramů písmen českého textu. Průběh 2-gramů bází je vůči průběhu 1-gramů písmen neúplný a až 3-gram bází realizuje křivku, kterou nacházíme u 1-gramů písmen (viz obrázek 12 zobrazující mapování distinktivních rysů hlásek, o kterých hovoříme níže – i zde je rozhodujícím kritériem úplnost průběhu grafu a podobnost průběhu grafu jako taková). Z tohoto důvodu klademe hranici písmene k 3-gramům bází. Následující průběhy obou mapování mají podobný vývoj. Průběhy 1-gramů a 2-gramů bází DNA nejsou u písmen realizovány, 3-gramy bází odpovídají 1-gramům písmen, 6-gramy až 7-gramy bází odpovídají 4-gramům až 5-gramům písmen. Další průběhy mají totožnou povahu.

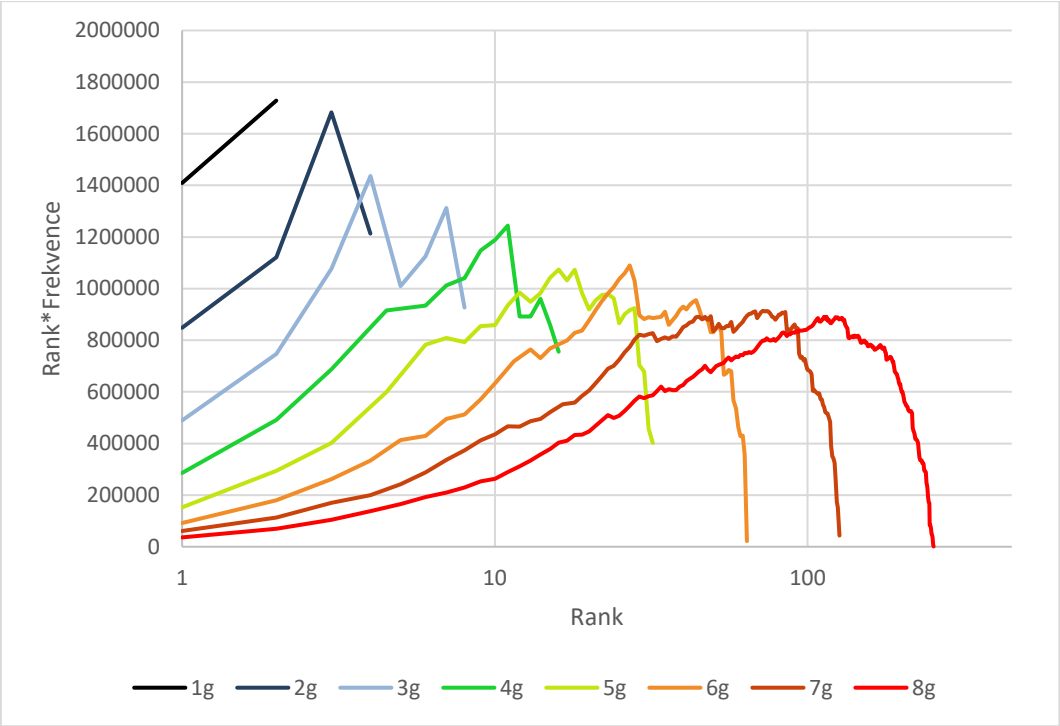
Z průběhu obou mapování můžeme implikovat následující. Báze jsou jednotkou konstitučně nižší než písmena. U 2-gramů bází můžeme sledovat podobný, avšak neúplný průběh, jako mají 1-gramy hlásek (viz obrázek 12 zobrazující mapování distinktivních rysů hlásek, kde totéž platí pro 1-gramy až 5-gramy distinktivních rysů). To je vysvětlitelné degenerovaností genetického kódu, kde je pro kódování aminokyseliny často třetí báze redundantní (obdobně mnoho hlásek odlišují pouze jeden či dva distinktivní rysy). Trojici bází (tripletům) přisuzujeme na základě podobnosti průběhů grafů roli písmen, sedmice bází (tedy více než 2 triplety) již tvoří obdobný typ konstituentu jako čtyř až pětice písmen – tj. tvoří obdobu slov.

Vezměme první implikaci, která říká, že báze jsou jednotkou konstitučně nižší než písmena. Z této implikace vyplývá otázka, jakou roli hrají báze v genetickém kódu, analogizujeme-li jej s přirozeným jazykem a předpokládáme-li obdobný design obou kódů.

Písmena zcela intuitivně vnímáme jako nedělitelná. Tvořena jsou ovšem na základě vzájemných vztahů, které diferencují jedno písmeno od druhého. Každé písmeno je pak tvořeno souborem vlastností, který jej charakterizuje a zároveň odlišuje od ostatních. Lingvistika tyto vlastnosti nazývá distinktivními rysy. Písmena textu tak můžeme chápat jako soubory distinktivních rysů. S touto rovinou jsme při analýze přirozeného jazyka prozatím nepracovali.

Pro zavedení roviny distinktivních rysů do analýzy postupujeme následujícím způsobem: každé z písmen charakterizujeme jeho vlastnostmi (distinktivními rysy), které jej odlišují od ostatních. Neužili jsme tradičně lingvistikou popisované distinktivní rysy ve smyslu akustických vlastností hlásek, kterých je ve fonologických popisech více než deset. Použili jsme nejmenší možný počet k písmenům arbitrárně přiřazených distinktivních rysů schopných odlišovat písmena češtiny. Pro českou abecedu je takových opozic nutných pouze šest. V textu je každé písmeno reprezentováno unikátním řetězcem šesti pozic obsazených jedničkou nebo nulou (tedy přítomných nebo nepřítomných vlastností arbitrárních distinktivních rysů). Takto nově reprezentovaný text zmapujeme a porovnáme s výsledky mapování mRNA.

Výsledek mapování distinktivních rysů českého textu (viz obrázek 12) nám odhaluje typy průběhů, které se projevily u 1-gramů a 2-gramů bází DNA (obrázek 11). U mapování písmen podobné průběhy nenacházíme. Průběhům 1-gramů až 5-gramů distinktivních rysů písmen českého textu ovšem odpovídají průběhy grafů 1-gramů a 2-gramů bází. U distinktivních rysů je z důvodu jejich počtu mezi průběhy pozvolnější přechod. Průběhu grafu 6-gramů distinktivních rysů odpovídá průběh 3-gramů bází. Z toho můžeme usuzovat, že báze mají v designu genetického kódu obdobnou roli jako distinktivní rysy písmen v designu přirozeného jazyka.



Obrázek 12: Mapování distinktivních rysů českého textu.

Závěr

Na základě aplikace Zipfova zákona na n-gramy bází DNA, písmena textu přirozeného jazyka a distinktivních rysů písmen textu přirozeného jazyka můžeme usuzovat, že nukleotidové báze DNA plní v designu genetického kódu roli analogickou distinktivním rysům písmen textu přirozeného jazyka. Triplety bází jsou dále analogické písmenům. Kombinace tripletů jsou následně analogií slov.

Naším cílem bylo ověření analogie DNA a přirozeného jazyka. Jakobson formuloval analogii bází DNA a písmen, tripletů a slov, genů a vět. Naše analýzy tento design genetického kódu zpochybňují a ukazují analogii bází DNA s distinktivními rysy, tripletů s písmeny a kombinací tripletů se slovy. Všeobecně užívaná analogie bází jakožto písmen genetického textu se tak jeví jako chybná. Reprezentace genetického zápisu sledem písmen odpovídajících bázím (A, C, G, T) zřejmě zapříčinila pevné ukotvení této analogie ve vědecké praxi a v laickém pojímání genetického kódu. Naše výsledky však ukazují, že tato analogie je vzhledem k designu přirozeného jazyka nesprávná, a báze tak neodpovídají písmenům. Takový závěr, ačkoliv se může jevit jako banální, nás zavede k myšlence aplikace jiného kvantitativně-lingvistického zákona s potenciálem jeho aplikace na jeden z největších problémů týkajících se genetického kódu a proteomiky.

Menzerath-Altmannův zákon a sekvence proteinů

V předchozí kapitole jsme představili teoretickou možnost reinterpretace paralely DNA a přirozených jazyků týkající se především role uvažovaných jednotek genetického kódu. Nově navržená reinterpretace spočívá především v posunu jednotek tzv. bází, původně a s jistou intuicí chápaných jako analogie písmen, k vyšším jednotkám, tj. ke kodonům či jinak řečeno tripletům kódujících aminokyseliny. Upravená analogie nukleotidových bází DNA, které se svou povahou blíží spíše distinktivním rysům a uvažovaný posun k jejich tripletům analogizujícím písmena jistě zapříčiňuje i posun v chápání vyšších jednotek, především jejich konstruktorů, tj. slov, vět a textů.

V teoretické rovině tak lze považovat aminokyseliny za analogii písmen a do možné role slov emergentně vstupují z nich tvořené sekundární struktury. Taková analogie je založena na fyzické realizaci distinktivních rysů, jejich arbitrárnímu propojení s aminokyselinami tvořícími analogii písmen, vytvářející různé sekundární struktury plnící různé funkce, tedy určitou analogii slov, které jsou lineárně řazeny uvnitř sekvence proteinu, tedy možné analogie věty či textu.

Tento posun a reinterpretace přináší především změnu způsobu náhledu na genetické texty a umožňuje je vnímat novým a jazykovědně přijatelnějším způsobem, což nám umožňuje smysluplnější aplikace kvantitativně lingvistických metod vyžadujících znalosti jednotek textu. Kromě heuristického testu popsaného v předchozí kapitole je myšlenka o posunu celé analogie shodně diskutována i čistě teoreticky v Lacková *et al*. 2017.

Jak uvidíme dále, uvedená reinterpretace nás vede k unikátnímu pohledu na sekvence kódujících DNA a umožňuje nám položit novou otázku, zda uvnitř genetických textů nalezneme specifický vztah empiricky přítomný na přirozených jazycích mezi velikostmi textů (proteinů, tzv. konstruktory) a velikostmi jejich slov (sekundárními strukturami, tzv. konstituenty) popsaný tzv. Menzerath-Altmannovým zákonem. Důležité rovněž je, že případná přítomnost Menzerath-Altmannova zákona může být nejen velmi důležitým teoretickým poznatkem vzhledem k možnému upevnění lingvistické analogie, ale především může mít zcela praktické důsledky v analýze proteinů od faktorů ovlivňujících jejich vznik z evolučního hlediska, způsobů a zdůvodnění designu sekundárních struktur proteinů, což je poznatek, který potenciálně může vést i k problematice designu *de novo* proteinů. Předtím než se na tyto aplikace a jejich význam zaměříme, se podívejme na definici Menzerath-Altmannova zákona, ze které následně odvodíme myšlenku, proč je tento zákon zajímavé hledat právě na genetických textech a především proteinech.

Menzerath-Altmannův zákon

Menzerath-Altmannův zákon je, obecně řečeno, specifický vztah velikosti celků (konstruktů) a jejich konstituentů (či komponent, tj. prvků, ze kterých jsou konstrukty tvořeny). Tento vztahů definuje Altmann (1980, 1) slovní (či heuristickou) definicí:

Čím větší je jazykový konstrukt, tím menší má komponenty.

Tento vztah můžeme ilustrovat na jednoduchém lingvistickém příkladu vět, slov a slabik. Věty se skládají ze slov, slova ze slabik a slabiky z jednotlivých hlásek. Každou tuto jednotku můžeme považovat za konstrukt obsahující vlastní komponenty. Pokud budeme například zkoumat věty, pak nám slovní definice Menzerath-Altmannova říká, že čím delší tyto věty (konstrukty) budou, tím kratší slova v nich budou obsaženy. Analýza textů z hlediska Menzerath-Altmannova zákona nám tedy umožňuje nahlédnout na vztahy uvnitř textu a způsob užívání konstituentů vzhledem k jejich tvořenému celku.

Z hlediska sekvencí tato analýza ovšem vyžaduje *apriorní* znalost charakteru studovaného textu a schopnost rozlišit nejen přítomné jednotky v jejich několika rovinách, ale i identifikovat jejich hierarchii, ze které následně vyvstávají definice konstruktů, konstituentů, a i jejich konstituentů atd. Z předchozí teoretické kapitoly a z předchozích odstavců je však zřejmé, že takové znalosti u sekvencí kódujících proteiny v důsledku máme a můžeme je využít k testování přítomnosti Menzerath-Altmannova zákona.

Menzerath-Altmannův zákon je v lingvistice pozorován na řadě jazyků a na různých rovinách, historicky především na fonémech (Menzerath 1928) a slovech (Altmann 1980) a dále například na zápisu japonských a čínských znaků (Benešová *et al*. 2016). Obdobně je popsaný vztah pozorován například i v hudbě (Boroda a Altmann 1991), architektuře (Lorenz *et al*. 2017) nebo, jak uvidíme dále, právě i v genetice. Přehled toho, kde můžeme Menzerath-Altmannův zákon pozorovat nalezneme v Altmann 2014. Menzerath-Altmannův zákon proto do jisté míry můžeme považovat za jistým způsobem univerzální a jak z této kapitoly také vyplyne, jeho přítomnost není triviální (viz např. Ferrer-i-Cancho *et al*. 2012).

Samotné porozumění Menzerath-Altmannovu zákonu z hlediska jeho slovní definice, tedy jako určitého vztahu *nepřímé úměry*, není zcela bezproblémové. Altmann (1980, 2-3) sám argumentuje i pro možnost, že za určitých okolností a výběru kombinace konstruktů a konstituentů trend *většího konstruktu* a *menších komponent* může přestat platit a může dojít i k jeho otočení. Z původní uvedené heuristické definice

Altmann (1980, 2) formalizuje několik modelů vyplývajících z řešení diferenciální rovnice reflektující intuitivní myšlenku: se změnou velikosti konstruktu x se průměrná délka konstituentů y (formálně zapsáno jako $\frac{dy}{dx}$) sníží o násobek konstantou c, tj. $-cy$:

$$\frac{dy}{dx} = -cy \quad .$$

Z této formalizace víme, že čím větší je velikost konstruktu x, tím je průměrná velikost konstituentů nižší. Poměr c, se kterým se průměrná velikost konstituentů snižuje, Altmann (1980, 3) doplňuje o parametr b, který vztahem k velikosti konstruktu x dokáže korigovat *rychlost*, se kterou průměrná délka konstituentů klesá. Tímto způsobem doplněná diferenciální rovnice (1) je finální formalizací změn velikosti konstituentů vůči konstruktu:

$$\frac{dy}{dx} = \left(-c + \frac{b}{x}\right) y \quad . \qquad (1)$$

Řešením (1) jako obyčejné diferenciální rovnice prvního řádu získáváme obecný model vztahu Menzerath-Altmannova zákona (2):

$$y = A * x^b e^{-cx} \quad , \qquad (2)$$

kde x je délka konstruktu (původně uvažovaná pro $\mathbb{N}$, nicméně platí i pro $\mathbb{R}$), y je průměrná délka konstituentů, $A \in \mathbb{R}$ je konstanta vzniklá při integraci, $b, c \in \mathbb{R}$ jsou konstanty upravující průběh funkce. Za určitých okolností lze uvedenou obecnou formuli (2) zjednodušit. Altmann (1980, 3) tyto okolnosti shrnuje do tří formulí (I. – III.):

$b = 0$		$y = A * e^{-cx}$	,	(I.)
$b \neq 0$	$c = 0$	$y = A * x^b$	,	(II.)
	$c \neq 0$	$y = A * x^b e^{-cx}$	.	(III.)

Původní formalizace Menzerath-Altmannova zákona z Altmann (1980) byly rozšířeny i o další modely se stále otevřenou otázkou, který z nich a pro jaké parametry nejlépe odpovídá přirozenému jazyku (např. Andres *et al*. 2012). Jedním z možných rozšíření pojetí Menzerath-Altmannova zákona je formalizace z Milička (2014, 87), která oproti modelům (I.-III.) nepracuje s exponentem a pro parametry A a $b \in \mathbb{R}$ a vyjadřuje téměř jen nepřímou úměru (IV.):

$$y = A + \frac{b}{x} \quad . \qquad (IV.)$$

Empirické testování jednotlivých formalizací má však svá úskalí. Prvním z nich, zřetelným především u přirozených jazyků, je nejistota v určení a definici jednotek rovin, tj. nejistota, jak text segmentovat a jak následně měřit velikosti konstituentů. Ta vyplývá z neostrosti a arbitrárnosti určení jazykových jednotek, přitom jejich volba dokáže kriticky ovlivnit výsledky a věrohodnost testů (viz např. Andres a Benešová 2012). Způsob segmentace textu tak svým důsledkem do testování Menzerath-Altmannova zákona vstupuje jako další parametr, se kterým je nutné operovat.

Oproti přirozeným textům má studium genetických textů popsaných výše, tj. kódujících sekvencí DNA, v rovině aminokyselin, sekundárních struktur a proteinů, praktickou výhodu. V analýze, kterou vzápětí představíme, existuje již předem daná strukturace jednotek i jejich segmentace, včetně předem známých nejistot v jejich hranicích. Druhým úskalím je problematika samotné kvantifikace přítomnosti a úspěšnosti modelů Menzerath-Altmannova zákona na konkrétních datech. Test přítomnosti Menzerath-Altmannova zákona na genetických textech často paradoxně spočívá pouze v otestování přítomnosti vztahu vyplývajícího ze slovní heuristické definice pomocí negativní Pearsonovy a případně Spearmanovy korelace logaritmizovaných velikostí konstruktů a konstituentů (např. Li 2012, 1; Baixeries *et al*. 2013, 95; Shahzad *et al*. 2015, 2 ad.). Takové testy nicméně prakticky obchází model (III.), který umožňuje obrácení celého trendu a testují tak pouze přítomnost modelů (I., II. a IV.).

Je zřejmé, že už pouhý test přítomnosti Menzerath-Altmannova zákona není bezproblémový a že dochází ke konfliktu mezi slovní heuristickou definicí, formálními modely a způsobem testování jejich přítomnosti. Abychom se vůči tomuto problému vymezili, budeme v následujícím textu označovat původní slovní definici snižování délky konstituentů s narůstající délkou konstruktů jako *trend* či *manifestaci Menzerathova zákona* (oproti Menzerath-Altmannově zákonu). Problematické je následně i vyhodnocení, který z modelů či formulí data vystihuje nejlépe, což je obecně heuristický problém vyhodnocování modelů, kdy lze využít i řadu kvalitativních metrik zároveň (např. Andres *et al*. 2014). Jak dále uvidíme, plánovaná aplikace analýzy Menzerath-Altmannova zákona na sekvencích DNA bude vyžadovat nejen rozhodnutí o přítomnosti tohoto zákona v datech, ale i vyhodnocení nejlepšího modelu. Formálně metodiku a podmínky testování přítomnosti tohoto vztahu uvedeme hned po představení cíle a smyslu celé analýzy a dat.

Důvod, proč hledat Menzerath-Altmannův zákon na sekvencích kódující DNA, respektive na sekvencích proteinů tkví především v jejich důležitosti. Proč jsou proteiny důležité a jak jsou v DNA uloženy a kódovány si stručně připomeňme. Proteiny jsou fyzické, trojrozměrné makromolekuly plnící v organizmu zásadní funkce od navazování a přenášení látek, roli nástrojů při zpracování DNA, její replikaci při dělení buněk, vytvářející různé druhy pojiv např. ve svalech, plnící funkce hormonů, stavebních prvků od svalů až po kosti a mnohé další.

I přesto, že jde o trojrozměrné fyzické *věci*, jsou zapsány lineárně v tzv. kódujících regionech DNA pomocí nukleotidových bází A, C, T a G původně a intuitivně vnímaných jako *písmena*. Pro vytvoření nové instance proteinu je nejprve nutné tuto část kódující DNA z dvoušroubovice zkopírovat do nové, fyzicky nezávislé sekvence v procesu transkripce. Replikovaný řetězec bází následně při procesu translace slouží jako vzor vytvoření druhé lineární sekvence tentokrát tvořené složitějšími a arbitrárně zprostředkovanými molekulami aminokyselin. Sekvence či nově vzniklý řetězec aminokyselin tvoří tzv. primární strukturu, která se začne následně vlivem vazeb vlastních prvků, prostředí nebo pomocných proteinů lokálně formovat do typických tvarů označovaných jako sekundární struktury (Dill a McCallum 2012, 1043). Tvary či konformace sekundárních struktur tvoří především tzv. alfa šroubovice (*alpha-helix*), překrývající se skládané listy (*beta-sheet, beta-strand*) a otočky (*turn*; otáčející směr polypeptidu). Původně lineární řetězec, tentokrát už obsahující první trojrozměrné sekundární struktury, se dále formuje do výsledného tvaru, kterým je tzv. terciální struktura odpovídající konečnému proteinu.

Tvar proteinu, jak už bylo řečeno, určuje jeho chemické vazby, a tedy i jeho funkci v organismu. Odlišný tvar proteinu, než je ten původně plánovaný, proto může vést k odlišné, a potenciálně i destruktivní funkci. Chybné konformace proteinů jsou následně příčinou řady onemocnění jako Creutzfeldt-Jakobova nemoc, Parkinsonova nemoc, Alzheimerova nemoc, diabetes druhého typu, cystická fibróza a mnohé další, shrnuté pod zastřešující název proteopatie (blíže např. Walker a LeVine 2000; Valastyan a Lindquist 2014; DeToma 2012). Porozumění, jak se z jednorozměrné lineární sekvence 20 symbolů, tj. v jistém smyslu pouhého textu, stávají sekundární struktury a z nich ve finále trojrozměrný fyzický nástroj, je problémem zkoumaným od 70. let 20. století (viz Anfinsen 1973), přičemž řešení této otázky vede k potenciálu léčby uvedených onemocnění, vývoji a designu nových léků a zvýšení množství anotovaných proteinů v genových bankách. Právě způsob, kterým se z lineárního řetězce DNA stává konkrétní trojrozměrný nástroj se specifickou funkcí v organizmu je kritickou otázkou – přitom už predikce *pouhých* sekundárních struktur není spolehlivě vyřešenou úlohou (viz Yang *et al*. 2018).

Způsoby predikce sekundárních struktur ze sekvencí DNA zahrnují řadu metod od těch čistě statistických, metod využívaných ve zpracování přirozeného jazyka (NLP) jako jsou markovovské procesy, generativní gramatiky a dále obecných metod strojového učení, nově především hluboké neuronové sítě (Yang *et al*. 2018). Predikce těchto struktur je problematická, přitom jde o důležitý mezikrok při určování kriticky důležité finální terciální struktury (Heffernan *et al*. 2015 a chyby v predikci sekundárních struktur mohou zapříčinit i chyby v predikci terciální struktury.

Predikce terciální struktury je navíc algoritmicky těžkou úlohou vzhledem k enormnímu množství potenciálních konformací daných kombinatorikou chemických vazeb, které může postupně se balící protein zaujímat. Tato složitost vedla k vytvoření projektů distribuovaných výpočtů jako rosetta@home zaměřených na využití co největšího výpočetního výkonu jejich dobrovolných uživatelů pro simulace a hledání terciálních struktur proteinů (Baker 2006).[18] Nicméně i distribuování výpočtů není vzhledem k enormní složitosti řešením. Z tohoto i dalších důvodů došlo k paradoxní situaci, kdy je výpočetně těžký problém přenesen z počítačů zpět na lidi pomocí *hry* Foldit. V této *hře* hráči nachází stabilní konformace proteinů vlastním úsudkem. Odměňováni jsou body vypočítanými na základě energetické stability nalezené konformace (Cooper *et al.* 2010).[19] Pomocí projektu Foldit byl například vyřešen způsob konformace krystalické struktury štěpícího proteinu Mason-Pfizerova viru způsobujícího onemocnění AIDS u opic (Khatib 2011).

Úspěšnost takových metod ovšem závisí na kvalitně segmentace (anotaci) sekundárních struktur a jejich predikci (Heffernan 2015, 1; Yang *et al*. 2018, 483). Každá chybná predikce sekundárních struktur tak může být příčinou ztráty výpočetního času i času vědecké síly. Yang *et al*. 2018 uvádí přehled současného stavu predikce sekundárních struktur i s dalšími účely, kterým validní predikce sekundárních struktur může dále sloužit. Především jde o nalézání proteinů podobných nové či neznámé sekvenci, ze kterých je možné inferovat její funkci nebo dále umožnit rozlišovat mezi neutrálními sekvencemi a sekvencemi způsobujícími onemocnění. Pro srovnání uvádí počty evidovaných sekvencí, kterých je přibližně 200 000 000 a počet sekvencí opatřených anotací sekundárních struktur, kterých je přibližně 100 000. Predikce prováděná počítači je dle Yang *et al*. „jediným praktickým řešením", a to především vzhledem k ceně fyzické alternativy (uvádí cenu přibližně 100 000 amerických dolarů za kus). Programové či výpočetní způsoby dnes dokáží predikovat sekundární struktury s úspěšností 82 % (Heffernar 2015) a 84 % (Wang 2016), stále tedy existuje nejistota, zda je sekundární struktura proteinu anotovaná správně.

Smyslem aplikace Menzerath-Altmannova zákonu na proteiny v této práci, jak již byla tato myšlenka a metoda prezentována v Matlach *et al*. 2016b a 2017, je zjistit, zda existuje vztah mezi proteinem a jeho sekundárními strukturami, tj. mezi konstruktem a jeho konstituenty obdobně, jako je tomu u přirozených jazyků. V případě, že by byl takový vztah nalezen, může jeho přítomnost sloužit cílům 1-3.

[18] Dostupné online https://boinc.bakerlab.org cit 11. 8. 2018
V době psaní tohoto textu je v projektu aktivních více než 50 000 počítačů s výkonem 194 000 TeraFLOP. Aktuální statistiky poskytovaného výkonu uživateli viz https://boinc.bakerlab.org/ cit 11. 8. 2018
[19] Dostupné ke stažení online: https://fold.it/portal cit.: 11. 8. 2018

(1) Znalost, že sekundární struktury mají vůči svému celku konkrétní a formálně vyjádřitelný vztah, by nám umožnil vnímat výsledky predikcí sekundárních struktur s novou možností evaluace její teoretické *kvality*. Kvantifikace „jak moc" a s jakou pravděpodobností daná predikce odpovídá vztahu Menzerath-Altmannova zákona by mohla sloužit jako kritérium přednostního výběru konkrétních anotací sekundárních struktur v případě, kdy je pro určitý protein dostupných více různých anotací a zároveň je nutné některou z nich vybrat jako kandidáta či kandidáty k další analýze. Kvantifikace vzdálenosti proteinu od modelu Menzerath-Altmannova zákona by zde mohla sloužit právě jako heuristika přednostního výběru a model sám by sloužil jako určitý etalon *přirozenějšího chování* vyvstávající z empirických dat, interpolující i chybějící pozorování.

(2) V případě, že by Menzerath-Altmannův zákon pro proteiny platil, je důležitou otázkou, zda a jak by tento vztah platil i pro proteiny pocházející z poškozených sekvencí DNA nebo obecně proteiny s poškozenou sekundární strukturou. V případě, že by pro tyto partikulární případy vztah neplatil, bylo by opět možné vytvořit model kvantifikující pravděpodobnost chybné strukturace proteinu a tím napomoci například *a priori* identifikovat kódující sekvence DNA s potenciálem tvorby chybných proteinů, a tedy potenciálně zdroj možného proteopatického onemocnění.

(3) Případné pozorování vztahu Menzerath-Altmannova zákona na proteinech a jejich sekundárních strukturách nám přináší i řadu teoretických znalostí. Především by se jednalo o poznatky týkající se způsobů ukládání informací v kontextu živých organismů na molekulární úrovni, dále užitých zohlednění ekonomizačních a konzervačních faktorů daných fyzikálním prostředím buněk nebo týkající se evolučních tlaků a faktorů na tvorbu specifické segmentace. Nalezení takového vztahu by dále umožnilo rozvést tyto teoretické faktory v paralele s přirozenými jazyky, u kterých je Menzerath-Altmannův zákon pozorován, a odvodit tak nové interpretační rámce pro oba obory.

Myšlenka, že by na úrovni proteinů a sekundárních struktur mohl existovat vztah pozorovaný v přirozených jazycích není nijak nová a například testování Zipfova zákona na DNA jsme představili již v předchozí kapitole. Přitažlivost testování lingvistických metod na DNA je zřejmě dána popsanou podobností obou typů dat a především tím, že se jedná o lineární zápis komplexních informací užívající arbitrární kód k realizaci jednotek s variabilní délkou směřujících ke konkrétním efektům. Oba systémy, tj. jak přirozený jazyk, tak i DNA, se navíc mohou potýkat s obdobným komunikačním schématem, ve kterém je nutné řešit vhodné navržení kódu vzhledem k dostupným zdrojům, množství a typu šumu, a řešit tak partikulární problémy předestřené ve formální disciplíně teorie komunikace. Jak poukazují Mian a Rose (2011), nalezené poznatky z biologie mohou navíc napomoci i v technickém návrhu síťové komunikace, vzhledem k evolučně ověřené biologické spolehlivosti těchto technik.

Obdobné úvahy o podobnosti DNA a přirozeného jazyka samozřejmě vedly i k testování projevů Menzerath-Altmannova zákona na různých typech konstruktů a konstituentů:

- Li (2012) testuje a potvrzuje přítomnost Menzerath-Altmannova zákona (dále jako MAL) na exonech, tj. částech genu, které po sestřihu původní transkripce vytváří finální sekvenci mRNA. Konstruktem je zde gen a konstituenty jsou zde exony měřené v počtu bází.

- Hernández-Fernández *et al*. (2011) testují MAL na velikostech genomu (konstrukt) a počtu chromozomů (konstituenty) na různých živočišných říších, opět s pozitivními výsledky.

- Ferrer-i-Cancho *et al*. (2012) testují vztah počtu chromozomů a jejich délky v bázích, opět potvrzující Menzerath-Altmannův zákon u různých živočišných druhů.

- Baixeries *et al*. (2013) testují vztah velikosti genomu (konstrukt) a velikosti chromozomů (konstituent) měřené v počtu bází a neutrálně uvádějí MAL jako možné vysvětlení nalezeného vztahu.

- Eroglu (2015) analyzuje délky proteinů (konstituenty) měřené v počtu aminokyselin vůči proteomům (konstrukty). Výsledkem analýzy je přijetí MAL jako modelu vysvětlujícího pozorované chování.

- Shahzad *et al*. (2015) studují vztah proteinů (konstrukty) a velikosti domén (konstituenty) počítaných v aminokyselinách s potvrzením trendu delšího proteinu a kratších domén.

Pozoruhodné je, že žádný ze zmíněných článků neuvažuje sekundární struktury jako rovinu či cíl analýzy Menzerath-Altmannova zákona, přičemž, jak jsme již uvedli, je rovina proteinů a sekundárních struktur kombinatoricky i teoreticky nejblíže. Žádný z uvedených článků také neuvažuje aplikaci těchto poznatků směrem k možnému využití při anotaci proteinů nebo jejich hodnocení. Myšlenka aplikovat nalezený vztah sekundárních struktur a proteinů byla prezentována spolu s prvotními výsledky v Matlach *et al*. 2016b a Matlach *et al*. 2017. Zde uvedené postupy však trpěly řadou metodologických nedostatků a nebyl zde ani podán konkrétní návrh či formalizace uvažované aplikace. Kompletní analýzu Menzerath-Altmannova zákona na proteinech a jeho sekundárních strukturách zde proto zavedeme přímo směrem k načrtnutým aplikacím v bodech 1-3 uvedených výše.

Prvním krokem, který při analýze a testování Menzerath-Altmannova zákona uděláme, je identifikace rovin a jednotek. Intuitivně bychom mohli roviny jednotlivé roviny a jejich hierarchii určit jako:

proteiny → sekundární struktury → aminokyseliny,

ale strukturace proteinů je v tomto ohledu poněkud složitější. Samotné proteiny mohou, kromě sekundárních struktur, obsahovat jejich více nezávislých částí plnících vlastní funkci, tj. obsahovat tzv. domény. Domény sice jsou součástí jediného řetězce tvořící protein, ale evolučně šlo zřejmě o vlastní nezávislé proteiny, které byly evolucí propojeny do větších celků (Kyte 2006, 345-346). Pokud od proteinu například pomocí proteolýzy doménu odstraníme, může stále plnit její funkce (Buxbaum 2015, 34).

Pojetí proteinů jako možných původně samostatných konglomerátů nás proto vede k možnému přehodnocení rovin a výběru konstruktu. Přidáním domén se metaforicky a s nadsázkou situace ohledně analýzy proteinů posunula od analýzy knih s jednolitým textem k analýze sborníků obsahujících několik izolovaných částí tvořících tematické celky. Pokud nás naivní strukturace proteinů vedla k analýze roviny *protein → sekundární struktura → aminokyseliny*, pak zanesením konceptu domén vzniká další možná rovina:

proteiny → domény → sekundární struktury → aminokyseliny.

V případě analýzy sekundárních struktur jsou v tomto případě jejich bezprostředním konstruktem domény. Z evolučního hlediska proto dává smysl za konstrukty považovat pouze domény, a nikoliv jejich *aglomerativní celky*. Paradoxně ale můžeme evolucí argumentovat i pro zanedbání roviny domén a analyzovat proteiny tak, jak jsou, neboť jejich spojením vzniká jediný funkční celek, jehož vznik navíc mohl být také řízen nebo ovlivněn ekonomizačními principy projevujícími se jako Menzerath-Altmannův zákon. Na základě této nejistoty je proto nejvýhodnější volbou provést testování MAL na obou typech konstruktů a zkoumat roviny:

domény → sekundární struktury → aminokyseliny

a

proteiny → sekundární struktury → aminokyseliny.

Z hlediska zamýšlených aplikací je však nutné tento vztah nalézt právě na rovině proteinů, neboť vztah platící jen na jejich částech neumožňuje analyzovat obecné a *a priori* neznámé sekvence. Zanesení roviny domén by znamenalo zanesení další nejistoty vyplývající z jejich heuristické predikce. Z tohoto důvodu je možné analýzu MAL na doménách považovat za přínosnou jen vzhledem k možnému zisku teoretických poznatků. Překvapivé je, že Shahzad *et al*. (2015) v definici jednotlivých rovin analýzy MAL sekundární struktury zcela vynechávají a analyzují roviny:

proteiny → domény → aminokyseliny.

Velikost domén tak měří přímo v aminokyselinách. Z jeho výsledků však víme, že na této rovině je MAL přítomen a platí, že čím více má protein domén, tím jsou tyto domény v počtu aminokyselin kratší. Tento jev je podle něj způsoben ekonomizačními důvody a principy, pomocí kterých je možné nahlédnout na způsoby a omezení, se kterými proteiny vznikly (Shahzad *et al*. 2015, 2). Altmann (1980, 5) vztahu MAL rovněž předvídal ekonomizační a balanční principy, které Shahzad *et al*. (2015, 7) reformulují do kontextu genetiky a Caetano-Anollés *et al.* (2017, 163-165) je dále rozvádí.

Druhou otázkou, kterou musíme v kontextu segmentace a identifikace rovin vyřešit, je interpretace sekundárních struktur. Jak bylo popsáno výše, typicky jsou prvky sekundární struktury popsány jejich třemi základními typy, kterými jsou alfa šroubovice, skládané beta listy a otočky. Tyto třídy pochází z 50. let 20. století a s příchodem nových predikčních metod a způsobů, jak sekundární struktury chápat, vznikly další a rozšířené notace. Zejména se jedná o notaci zavedenou predikčním softwarem DSSP obsahující 8 detailnějších typů struktur rozšiřujících ty původní na základě analýzy vodíkových a elektrostatických vazeb na atomové úrovni (Bienkowska *et al*. 2002). Problematické ovšem je, že různé predikční metody a definice sekundárních struktur vedou i k odlišným anotacím (či segmentacím), které se ve výsledku liší na jejich hranicích, a tedy délkách v počtu aminokyselin (srov. Martin *et al*. 2005). Problémem je také to, že už pouhá detailnější anotace vede (nebo může vést) ke změně, a především navýšení počtu konstituentů. Od původní intuice ostrosti segmentace proteinů se tak dostáváme velmi blízko problémům, které řeší lingvistika u přirozených textů. Množství typů segmentací a pojetí sekundárních struktur nás vede k nejistotě, kterou z nich pro analýzu MAL vybrat, zda tu základní nebo rozšířenou, tj. DSSP. Jistým řešením je proto analyzovat oba typy anotací a porovnat je.

Výběr vzorku dat proteinů

Výše jsme odvodili, že vzhledem k určitým nejistotám v anotaci sekvencí a nejistotě v určení rovin proteinů, i přes pouhou teoretickou využitelnost výsledků z roviny domén, budeme přítomnost Menzerath-Altmannova zákona testovat na celkem čtyřech typech sekvencí. První dva typy odpovídají sekvencím proteinů anotovaných dvěma způsoby, a to základními třemi třídami sekundárních struktur a osmi třídami DSSP. Třetí a čtvrtý typ dat odpovídají sekvencím domén anotovaných základními třemi třídami a osmi třídami DSSP.

Abychom se vyhnuli možnému negativnímu efektu převážení či zkreslení výsledků skupinou sobě podobných domén/proteinů, využijeme pro každý zmíněný typ dat pouze takové sekvence, které jsou si nepodobné, tj. neredundantní a na základě kterých bychom měli získat výsledky platné obecně, nikoliv jen pro nejpočetnější třídu proteinů/domén. Pro všechny čtyři typy dat proto získáme předzpracované a pro tyto účely používané seznamy neredundantních sekvencí domén a proteinů. Identifikátory neredundantních domén získáme z projektu CATH (Dawson *et al*. 2017), který zaznamenává domény podobné maximálně ze 40 % na základě metody BLAST.[20] Seznam identifikátorů neredundantních proteinů získáme pomocí nástroje VAST (Madej *et al*. 2014), ve kterém jsou evidovány proteiny s BLAST podobností na p-hodnotě $10e^{-7}$.[21] Seznamy získané z obou nástrojů obsahují pouze identifikátory domén a proteinů, jejichž sekvence včetně obou typů anotací dále musíme získat separátně. Sekvence a anotace sekundárních struktur pomocí DSSP získáme z banky RSCB PDB (Berman *et al*. 2002)[22] a sekvence anotované třemi třídami z databáze Uniprot (UniProt Consortium 2018) [23] . Z obou zdrojů využijeme jen ty sekvence odpovídající filtru CATH/VAST.[24]

Zběžnou analýzou výsledných datasetů z obou zdrojů sekvencí po odfiltrování redundantních záznamů zjistíme několik rozdílů – oba datasety jsou jinak velké (viz dále) a dle identifikátorů obsahují sekvence odlišných proteinů/domén – shodná je pouze podmnožina. Již dopředu nelze očekávat ideální srovnání mezi oběma typy

[20] Dostupné online: http://www.cathdb.info/download cit. 12. 8. 2018, konkrétně soubor: ftp://orengoftp.biochem.ucl.ac.uk/cath/releases/latest-release/non-redundant-data-sets/cath-dataset-nonredundant-S40.list cit. 12. 8. 2018.

[21] Dostupné online: https://structure.ncbi.nlm.nih.gov/Structure/VAST/nrpdb.html, cit 12. 8. 2018.

[22] Dostupné online: https://www.rcsb.org/pdb/static.do?p=download/http/index.html cit. 12. 8. 2018, konkrétně soubor: https://cdn.rcsb.org/etl/kabschSander/ss.txt.gz cit. 12. 8. 2018.

[23] Dostupné online: https://www.uniprot.org/ cit 12. 8. 2018, staženy byly veškeré dostupné proteiny ve formátu XML. Anotace tří struktur databáze Uniprot vychází z anotace DSSP, kterou Uniprot zobecňuje na základě určitých pravidel, která jsou k nahédnutí online: https://www.uniprot.org/help/structure_section cit. 12. 8. 2018.

[24] Pro tyto účely bylo nutné vytvořit specializované programy zpracovávající surová data z obou bank do formátu využitelném při analýze MAL, oba programy UniprotToMA a DomainsToStructs jsou dostupné v datové příloze: Programy.

anotací, neboť jimi nejsou anotovány shodné sekvence. Kromě této neshody nalezneme i druhý neideální jev, kterým jsou neshody přímo v samotných sekvencích. Obě databáze, tj. RSCB PDB a Uniprot uchovávají pod stejnými identifikátory sekvence, které nemusí být nutně shodné a mohou obsahovat např. bodové mutace, a to kvůli vzorkování těchto sekvencí z odlišných organismů (tento problém dále ilustrujeme). V tomto ohledu je proto zřejmé, že výsledky pro oba typy anotací (či segmentací) nemohou jednoznačně vést k rozhodnutí, který způsob anotace je nejvýhodnější. Pokud bude učiněno, stane se tak pouze na základě aktuální empirie. Dále se blíže seznámíme se sekvencemi, které budeme zkoumat.

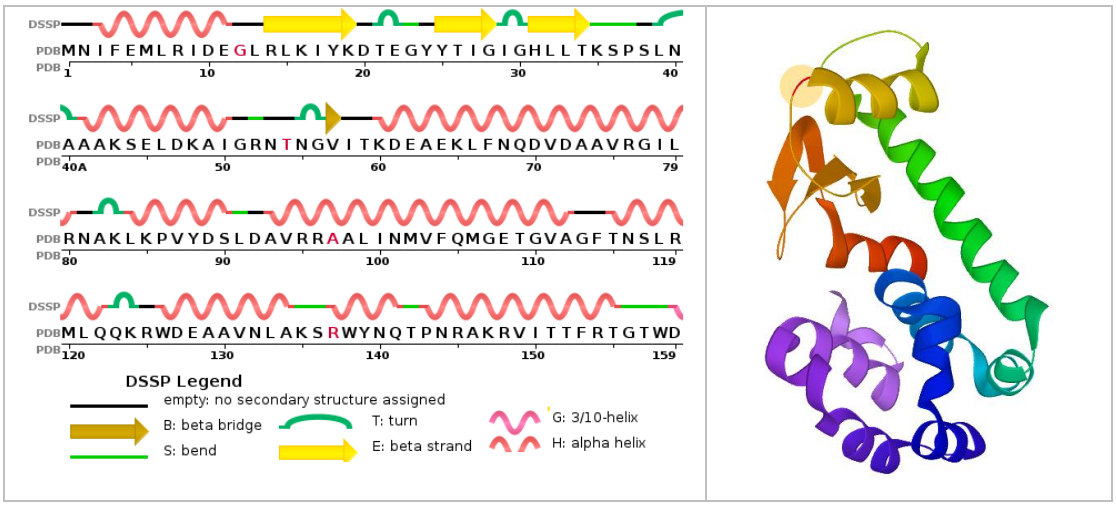
Pro bližší náhled na sekvence proteinů a jejich anotace se podíváme na protein identifikovaný jako 102L, tj. enzym *lysozym* obsažený např. v krevní plazmě. Sekvenci tohoto proteinu, společně s anotací DSSP sekundárních struktur získané z banky RSCB PDB v textovém formátu, vidíme v tabulce 1. První a třetí řádek této tabulky obsahuje identifikátor sekvence a informaci, jaký typ dat je zapsán na následujícím řádku, tj. aminokyselinová sekvence v případě *sequence* nebo sekvence anotující sekundární strukturu v případě *secstr*. Jednotlivá písmena v první ze sekvencí odpovídají jednotlivým aminokyselinám. V případě druhé sekvence zastupují jednotlivá písmena konkrétní typy sekundárních struktur. Graficky můžeme sekvenci i její sekundární struktury zobrazit diagramem (A) nebo pomocí trojrozměrného modelu (B) na obrázku 1.

Samotná sekvence proteinu 102L má 165 aminokyselin a 29 anotovaných sekundárních struktur dle DSSP s jejich šesti realizovanými typy. U textové anotace sekundárních struktur si můžeme všimnout značně vyčnívajících mezer. Ty označují části sekvence bez přiřazených sekundárních struktur, tj. tzv. smyčky (klubka / náhodná klubka; anglicky *loops*, *coils* a *random coils*) sloužící k propojení sousedních sekundárních struktur (např. Xiong 2006, 179). Pro porovnání se na tentýž protein podíváme do databáze Uniprot, ve které jsou proteiny anotovány třemi základními typy struktur, tj. alfa-šroubovicemi, skládanými beta listy a otočkami. Tuto jednodušší anotaci vidíme na obrázku 2.

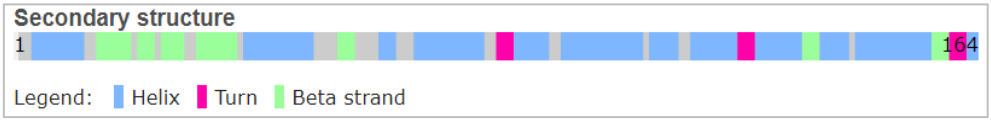
Zběžným porovnáním obou anotací odhalíme několik rozdílů. Sekvence z Uniprot je oproti sekvenci z RSCB o jednu aminokyselinu kratší, kdy RSCB registruje mutaci inzercí alaninu na pozici 41 před alfa-šroubovici (pozice mutace je v trojrozměrném schématu proteinu na obrázku 1 zvýrazněna oranžovým kroužkem v horní části schématu; blíže k této inzerci viz Heinz *et al*. 1993). Porovnáním anotací sekundárních struktur zjistíme, že obě sekvence začínají shodně šroubovicí (*helix*) následovanou smyčkou a beta listy (*beta strand*). V anotaci RSCB pak následuje otočka, namísto které je v anotaci Uniprot pouze smyčka. Takové rozdíly budou hrát v analýze Menzerath-Altmannova zákona určitou roli, neboť otočky (obsažené v anotaci RSCB) jsou součástí sekundárních struktur, zatímco náhodná klubka (registrované v anotaci Uniprot) ne.

```
1: >102L:A:sequence
2: MNIFEMLRIDEGLRLKIYKDTEGYYTIGIGHLLTKSPSLNAAAKSELDKAIGRNTNGV…
3: >102L:A:secstr
4:  HHHHHHHHH  EEEEEE TTS EEEETTEEEESSS TTTHHHHHHHHHHTS  TTB…
```

***Tabulka 1**: Ilustrace záznamu proteinu 102L včetně aminokyselinové sekvence a anotace sekundárních struktur.*



***Obrázek 1**: Vizualizace proteinu 102L pomocí wiring diagramu (A) a trojrozměrného schematického modelu (B).[25,26]*



***Obrázek 2**: Vizualizace anotace sekundárních struktur proteinu 102L z Uniprot.[27]*

Pokud se dále zaměříme jen na ta místa, ve kterých má sekvence z Uniprot anotované beta skládané listy, všimneme si, že anotace z RSCB přibližně od první třetiny sekvence už žádné beta skládané listy neanotuje. Anotace RSCB obsahuje celkem 29 anotovaných sekundárních struktur, anotace z Uniprot jich obsahuje pouze 22. Z tohoto je zřejmé, že rozdíly v anotacích skutečně nastávají (blíže o porovnání různých způsobů anotací viz Martin *et al*. 2005). Předem nicméně nelze odhadnout, jaké důsledky budou mít tyto rozdíly na případné projevy Menzerath-Altmannova zákona, což je právě důvod, proč budeme testovat obě varianty anotace zvlášť. Jednotlivá data a úrovně, na kterých budeme projevy Menzerath-Altmannova zákona testovat, nyní shrneme a popíšeme.

[25] Dostupné online: http://www.ebi.ac.uk/thornton-srv/databases/cgi-bin/pdbsum/GetPage.pl?pdbcode=102l cit. 15. 8. 2018.
[26] Dostupné online: https://www.ebi.ac.uk/pdbe/entry/pdb/102l/protein/1 cit 15.8.2018.
[27] Dostupné online: https://www.uniprot.org/uniprot/P00720 cit 15.8.2018.

První dvojice dat k testování se týká domén anotovaných oběma diskutovanými způsoby a druhá dvojice se následně týká proteinů opět anotovaných oběma způsoby. Připomeňme, že všechny sekvence jsou filtrovány tak, aby v datech byly obsaženy pouze neredundantní sekvence. Důležité také je, že pro plánovanou aplikaci jsou rozhodující výsledky z proteinů, zatímco výsledky z domén nám poslouží jako teoretická opora poskytující informaci o možné přítomnost Menzerath-Altmannova zákona na jednodušších a evolučně starších konstruktech.

1. Domény: Data z RSCB filtrovaná CATH

Rovina: domény → sekundární struktury (anotace DSSP) → aminokyseliny.

Prvními daty k testování přítomnosti MAL jsou neredundantní domény anotované pomocí DSSP získané z databáze RSCB. Celkem se jedná o 30 693 domén, jejichž základní přehled si můžeme prohlédnout v tabulce 2. U těchto dat je důležité, že obsahují 8 tříd sekundárních struktur, přitom minimální délkou, kterou může nabývat nejkratší z nich (izolované beta listy), je pouhá jedna aminokyselina. Průměrná doména zde má přibližně 316 aminokyselin a obsahuje v průměru 61 sekundárních struktur. V datasetu jsou obsaženy domény obsahující i jedinou sekundární strukturu.

	Průměr	Medián	Min	Max
Délka sekvence [amin.]	316,5	273	40	3367
Počet sekundárních struktur	60,55	51	1	551
Délka s. struktur [amin.]	3,9	2	1	154
Počet pozorování:	30 693 domén.			

***Tabulka 2**: Popis neredundantních sekvencí domén anotovaných DSSP pocházející z RSCB.*

2. Domény: Data z Uniprot filtrovaná CATH

Rovina: domény → sekundární struktury (tři základní typy) → aminokyseliny.

Data neredundantních domén z databáze Uniprot jsou anotována pouze třemi základními typy sekundárních struktur. Oproti RSCB je zde obsaženo pouze 13 204 domén (tj. méně než polovina; tento rozdíl je zapříčiněn získáním pouze tzv. *reviewed* sekvencí, které prošly manuální kontrolou), nicméně dramatické rozdíly dále nalezneme u počtu sekundárních struktur na doménu. Průměrným počtem struktur je zde 26,2 a maximálním 288 oproti průměrným 60,55 a maximálním 551 strukturám RSCB. Důvodem nižších kvantit sekundárních struktur je zřejmě splynutí hruběji anotovaných struktur do jediného tokenu, který má za následek i zvýšení průměrné délky sekundárních struktur z RSCB průměru 3,9 na 7,067 v Uniprot. Projev odlišné definice sekundárních struktur pak nalezneme i na jejich minimální délce, kterou jsou 3 aminokyseliny.

	Průměr	Medián	Min	Max
Délka sekvence [amin.]	378,5	345	7	3332
Počet sekundárních struktur	26,2	22	1	288
Délka s. struktur [amin.]	7,067	5	3	148
Počet pozorování:	13 204 domén.			

***Tabulka 3**: Popis neredundantních sekvencí domén anotovaných třemi základními typy sekundárních struktur pocházejících z Uniprot.*

3. Proteiny: Data z RSCB filtrovaná VAST

Rovina: proteiny → sekundární struktury (DSSP) → aminokyseliny.

Třetím typem dat, na kterých budeme testovat Menzerath-Altmannův zákon a která jsou pro nás z aplikačního hlediska spolu s následujícími daty kritická, jsou sekvence proteinů. Po odfiltrování redundantních proteinů pomocí filtru VAST z databáze RSCB získáváme 14 567 anotovaných proteinů popsaných v tabulce 4. Oproti datům domén (tabulka 2) si můžeme všimnout především poklesu průměrné délky sekvencí i počtu sekundárních struktur (přibližně 1,6násobně).

	Průměr	Medián	Min	Max
Délka sekvence [amin.]	199	153	20	3245
Počet sekundárních struktur	37,49	27	1	498
Délka s. struktur [amin.]	4,033	2	1	152
Počet pozorování:	14 567 proteinů.			

***Tabulka 4**: Popis neredundantních sekvencí proteinů anotovaných DSSP a pocházející z RSCB.*

4. Proteiny: Data z Uniprot filtrovaná VAST

Rovina: proteiny → sekundární struktury (tři základní typy) → aminokyseliny.

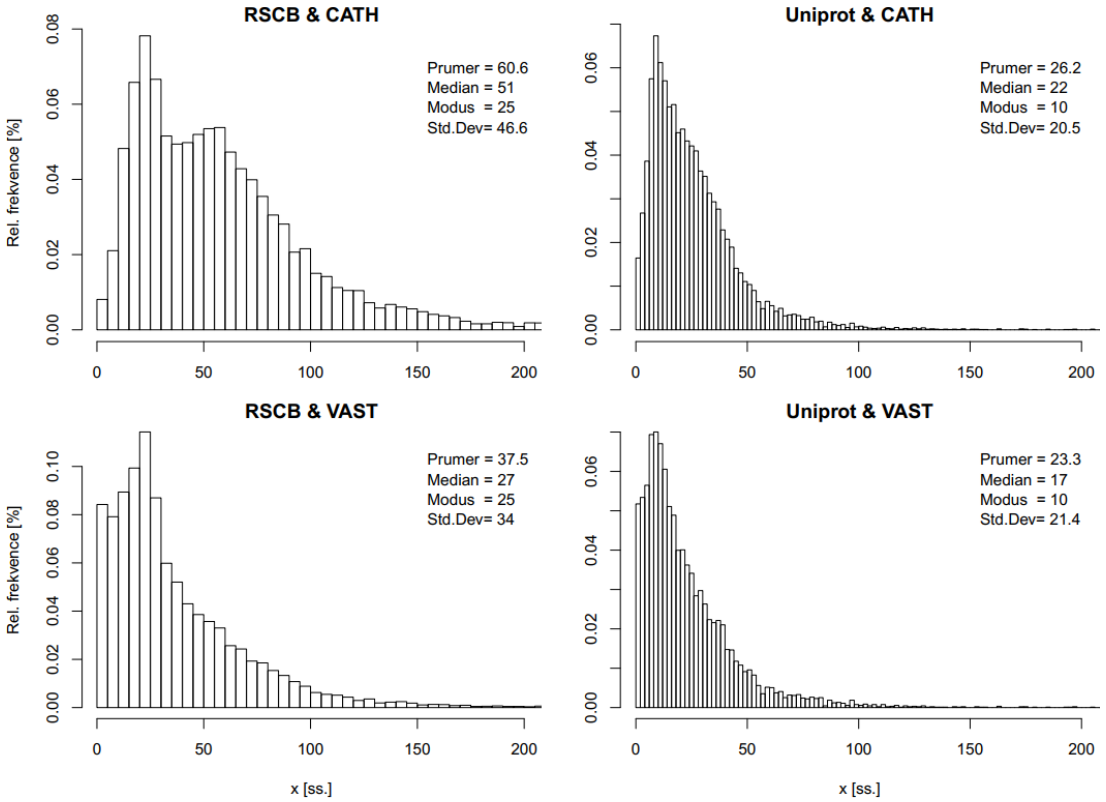
Posledním typem dat k testování přítomnosti MAL jsou neredundantní sekvence proteinů anotované třemi základními typy sekundárních struktur z databáze Uniprot. Shrnutí 9 082 sekvencí nalezneme v tabulce 5. Oproti výše diskutovaným doménám už při porovnání počtů sekvencí s konkurenční databází RSCB nenalézáme tak dramatický rozdíl. Rozdíl však můžeme nově pozorovat především na průměrných délkách sekvencí. Zde obsažený protein *Titin* (obsažený ve svalových vláknech) zvyšuje maximální délku proteinů až na 34 tisíc aminokyselin. Nejmenší protein je zde i u sekvencí z RSCB shodný na 20 aminokyselinách. Ostatní rozdíly, například v minimální velikosti sekundárních struktur a jejich průměrné délce odpovídají poznatkům z porovnání domén výše.

	Průměr	Medián	Min	Max
Délka sekvence [amin.]	492,4	340	20	34350
Počet sekundárních struktur	23,33	17	1	288
Délka s. struktur [amin.]	7,154	5	3	148
Počet pozorování:	9 082 proteinů.			

***Tabulka 5**: Popis neredundantních sekvencí proteinů anotovaných třemi základními typy sekundárních struktur z Uniprot.*

Na výše uvedené čtveřici souborů dat (či *datasetech*) budeme testovat přítomnost Menzerath-Altmannova zákona. U obou typů konstruktů, tj. u domén a proteinů, budeme měřit jejich velikost počtem obsažených sekundárních struktur. Tuto velikost (počet sekundárních struktur) budeme pro jednodušší referenci značit písmenem x. Každému konstruktu následně odpovídá hodnota y vyjadřující průměrnou velikost jeho konstituentů, tj. průměrnou velikost všech obsažených sekundárních struktur měřených v počtu aminokyselin. V případě pozorování Menzerathova zákona (tj. původního trendu) by tak se s narůstajícími hodnotami x (velikostmi konstruktů) měly snižovat hodnoty y (průměrné velikosti konstituentů). Před samotným testováním Menzerath-Altmannova zákona se napřed seznámíme s hodnotami x a y jednotlivých datasetů a jejich distribucemi, které nám poskytnou důležité poznatky nejen pro konstrukci testů přítomnosti MAL, ale i pro pozdější stochastické simulace uvažovaných aplikací.

Hodnoty x zastupují počty sekundárních struktur jednotlivých proteinů a domén. Histogramy těchto hodnot pro jednotlivé datasety vidíme v grafu 1. Zde zjišťujeme, že rozdělení u domén (horní řada grafů) i u proteinů (dolní řada) je blízké negativně-binomiální distribuci. Takové rozdělení není příliš překvapivé, ovšem jeho znalost je pro nás důležitá vzhledem k možnému zkreslení proložení modelů Menzerath-Altmannova zákona.



***Graf 1**: Histogramy velikostí konstruktů, tj. hodnot x (proteinů/domén) v počtu sekundárních struktur.*

Prokládání nerovnoměrně rozdělenými daty pak vede k parametrům reflektujícím právě ty nejfrekventovanější hodnoty *x*, kterým se model snaží svým průběhem *vyhovět* minimalizací některé z kvantifikací predikčních chyb reflektujících právě ty nejfrekventovanější intervaly. Výsledek proložení takových dat pak nutně nemusí odpovídat přítomnému obecnému či průměrnému trendu v pozorovaných datech. V metodice testování přítomnosti Menzerath-Altmannova zákona proto budeme muset tuto problematiku zohlednit. Kromě náhledu na konkrétní tvary distribuce jsme se s popisem jejich základních statistických vlastností již setkali při představení jednotlivých dat výše.

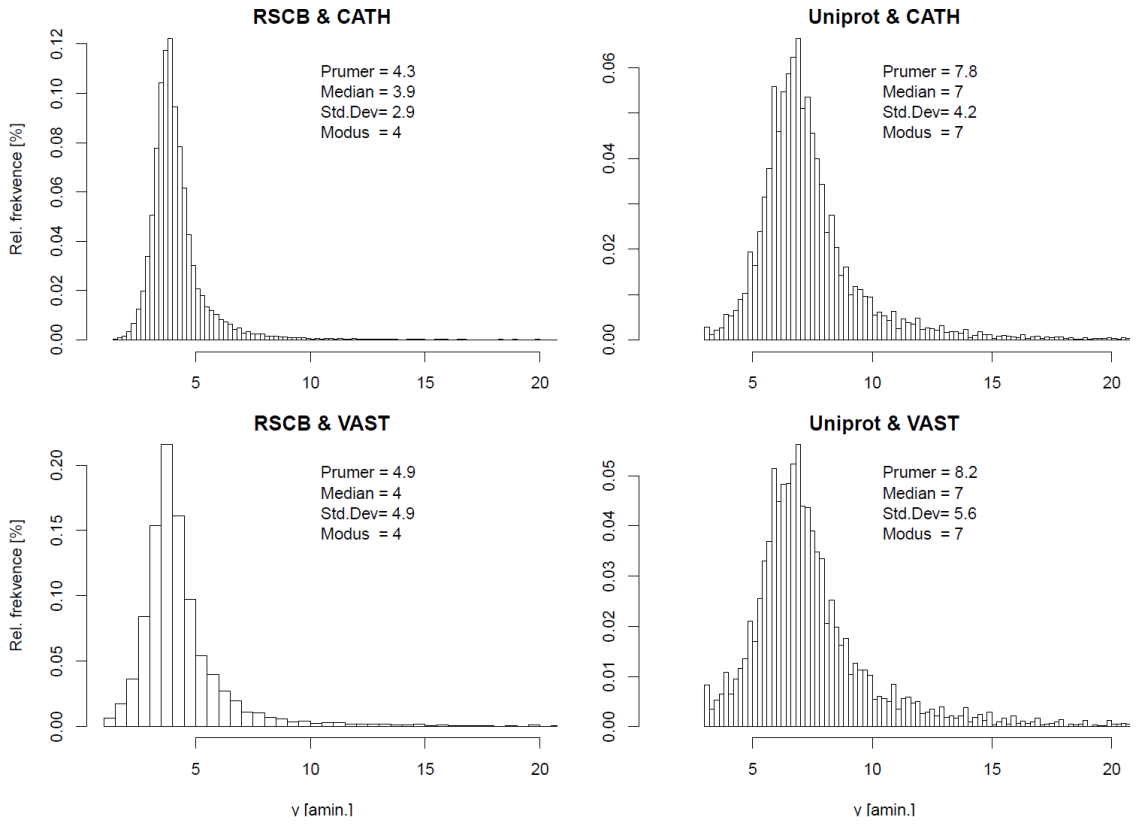
Pozorované rozdělení je pro nás ovšem zajímavé i z čistě teoretických důvodů vedoucích k otázce designu proteinů a jejich sekundárních struktur. Obdobné dělení je pozorováno u poločasu rozpadu proteinů (Cambridge *et al*. 2011, 5278) a nelze také opomenout, že tento typ průběhu je specifický i pro Birnbaum-Saundersovu distribuci, typicky využívanou k modelování pravděpodobnosti rozpadu materiálu a součástek při opakovaném namáhání (Leiva 2016, 3–6). Z pohledu namáhání, rozpadu a kolizních interakcí jsou proteiny více než zajímavé. V prostorách buněk běžně dochází k jevu přeplnění (*crowding*), při kterém je cytosol natolik přeplněn, že dochází k fyzickým kolizím proteinů s ostatními makromolekulami, a to v takovém měřítku, že jsou jím ovlivněny procesy od transkripce DNA, translace, balení proteinů, rozkladu proteinů a je ovlivněna i jejich funkční interakce (přehled viz Zhou 2013). Tento

jev v buňkách nastává běžně a z hlediska evoluce musel být v organismech řešen. Překvapivé je, že s tímto jevem nebylo při modelování proteinů většinou pracováno (Ellis 2001, 597). Podstata interakce proteinů je stejně tak zajímavá. Pro konání specifické funkce, například přenášení molekul nebo při interakci s jinou makromolekulou, musí dojít k fyzické kolizi obou uvažovaných prvků, a to zároveň, ve správném natočení a pořadí (Nussinov a Schreiber 2009, 87-90; Kleanthous 2000, 20-23). Navíc i samotné balení proteinů je v pojetí *diffusion-collision* modelu (Karplus a Weaver 1994, 651) chápáno jako proces kolizí.

Funkce proteinů, jak už bylo řečeno, je dána jejich tvarem, který je dle výše uvedeného, neustále vystavován rozmanitým fyzickým kolizím. Z hlediska evoluce proto design proteinů musí následovat určitá pravidla napomáhající jejich stabilitě v prostředí, jejich správnému balení, interakcím i dekompozici. Pozorované rozdělení nás tak přivádí k myšlence specifického designu proteinů, ve kterém je místo pro řadu principů, mezi kterými by mohl být i Menzerath-Altmannův zákon.

Počty sekundárních struktur, tedy hodnoty x, mají svůj protějšek v hodnotách y odpovídajících jejich průměrným velikostem počítaným v aminokyselinách. Proteiny i domény mají v použitém datasetu, jak jsme zjistili, nejpravděpodobněji 10 nebo 25 sekundárních struktur dle typu anotace. Rozdělení jejich průměrných délek vidíme v histogramech grafu 2 a výpočty vzdáleností pozorování od průměru a modu v tabulce 6. I zde můžeme vidět rozdělení blízká negativně-binomiální nebo Poissonově, u kterých si můžeme všimnout zejména jejich špičatosti. V případě domén (horní řada histogramů) je 95 % hodnot y od nejpravděpodobnější velikosti (značené $Mod(Y)$) vzdáleno maximálně 2,5 aminokyselin v případě anotace DSSP a 5,5 aminokyselin v případě anotace třemi třídami. V případě proteinů je 95 % hodnot y od nejpravděpodobnější velikosti vzdáleno maximálně 5,2 aminokyselin v případě anotace DSSP a 7,9 aminokyselin v případě anotace třemi třídami. Vzdálenosti hodnot y od průměru jsou ještě nižší: 95 % z nich leží pro domény a anotaci DSSP v maximální vzdálenosti 2,1 aminokyselin a pro anotaci třemi třídami maximálně 4,7 aminokyselin. V případě proteinů pak jde maximálně 4,3 a 6,7 aminokyselin pro DSSP a tři třídy sekundárních struktur. Takové pozorování má velmi důležité důsledky pro plánované modelování predikcí. V případě, kdy bychom měli určit průměrnou velikost sekundárních struktur např. neznámého proteinu anotovaného DSSP, můžeme tipovat hodnotu průměru, tj. 4,3 aminokyselin a v 95 % případů se nespleteme o více než o 2,1 aminokyselin, což je, vzhledem k jejich maximální velikosti kolem 150 aminokyselin (viz výše) velmi přesný odhad.

Z těchto údajů vyplývá, že jakýkoliv model nelze automaticky považovat za kvalitní jen tím, že bude produkovat nízkou chybu v počtu aminokyselin, neboť tato úspěšnost je dána už samotnou distribucí dat, kterou je nutné překonat. Z důvodů dalšího porovnání úspěšnosti modelů jsou proto v tabulce 6 uvedeny i kvantifikace chyb predikcí průměru a modu, včetně hodnot RMSE a MAE (viz dále).



***Graf 2**: Histogramy průměrných velikosti sekundárních struktur proteinů a domén v počtu aminokyselin.*

RSCB & CATH		95% CI
y-E(Y)	-	-1,5—2,1
y-Mod(Y)	0,33	-1,2—2,5
MAE(y-E(Y))	1,013	0,988—1,039
MAE(y-Mod(Y))	0,930	0,905—0,957
RMSE(y-E(Y))	2,93	2,7—3,2
RMSE(y-Mod(Y))	2,94	2,7—3,2

Uniprot & CATH		95% CI
y-E(Y)	-	-2,9—4,7
y-Mod(Y)	0,75	-2,1—5,5
MAE(y-E(Y))	1,977	1,923—2,031
MAE(y-Mod(Y))	1,809	1,754—1,865
RMSE(y-E(Y))	4,21	3,9—4,5
RMSE(y-Mod(Y))	4,28	4—4,6

RSCB & VAST		95% CI
y-E(Y)	-	-2,5—4,3
y-Mod(Y)	0,92	-1,6—5,2
MAE(y-E(Y))	1,961	1,900—2,025
MAE(y-Mod(Y))	1,679	1,615—1,742
RMSE(y-E(Y))	4,95	4,5—5,4
RMSE(y-Mod(Y))	5,03	4,6—5,5

Uniprot & VAST		95% CI
y-E(Y)	-	-3,8—6,7
y-Mod(Y)	1,16	-2,6—7,9
MAE(y-E(Y))	2,673	2,591—2,760
MAE(y-Mod(Y))	2,381	2,294—2,469
RMSE(y-E(Y))	5,55	5,2—5,9
RMSE(y-Mod(Y))	5,67	5,3—6,1

***Tabulka 6**: Hodnoty vztahující se k predikci velikostí sekundárních struktur dle průměru a modu.*

Nyní již známe základní charakteristiku dat určených k testování přítomnosti Menzerath-Altmannova zákona včetně jejich možných dopadů na modelování a vyhodnocování výsledků. Obecně můžeme říci, že hlavním problémem je zde nevyvážení datasetu, kdy pro všechny velikosti konstruktů x nemáme shodný počet pozorování, což vede ke zkreslení proložení, tj. problému, se kterým se budeme muset vypořádat. Dalším získaným a velmi důležitým poznatkem z náhledu na data je *apriorní* blízkost průměrných velikostí sekundárních struktur y, která bude v případě nalezení Menzerath-Altmannova zákona hrát roli určitého *benchmarku* kvality predikce jednotlivých modelů. Pohled na hodnoty x a y nám proto přinesl velmi důležité poznatky, které zohledníme v metodice testování uvedené dále.

Nyní již můžeme definovat způsob testování Menzerath-Altmannova zákona a způsoby vyhodnocení jeho přítomnosti. V případě přítomnosti tohoto zákona dále definujeme metodiku výběru nejlepšího modelu. Po dokončení analýzy přítomnosti Menzerath-Altmannova zákona a v případě jeho nalezení následně definujeme metodiku jeho využití načrtnutého výše, tj. pro heuristický skóring počtu a délek sekundárních struktur anotovaného proteinu a vyhodnotíme jeho úspěšnost a aplikovatelnost.

Metoda analýzy Menzerath-Altmannova zákona

Metoda analýzy přítomnosti a následně i využitelnosti Menzerath-Altmannova zákona vyžaduje několik separátních kroků. Prvním z nich je zjištění, zda je v datech přítomný obecný trend Menzerathova zákona, tj. *čím větší jsou sekundární struktury, tím menší bude jejich průměrná velikost*. K otestování tohoto vztahu dle Li 2012 (1), Baieries *et al*. 2013 (95) a Shahzad *et al*. 2015 (2) využijeme statisticky významné negativní korelace logaritmizovaných hodnot x a y. Vzhledem k nerovnoměrným distribucím studovaných hodnot test uskutečníme i na zprůměrovaných hodnotách, ve kterých pro každý počet sekundárních struktur $x \in X$ bude jediná, průměrně pozorování hodnota $\bar{y}$. V případě, že nalezneme obecný trend Menzerathova zákona, přistoupíme k dalšímu kroku, kterým je nalezení nejlepšího modelu (I.-IV.), od kterého následně odvodíme a teoreticky ověříme aplikace popsané na začátku této kapitoly. Postup testování shrneme do tří kroků.

1) Surová data proteinů/domén připravíme tak, abychom měli k dispozici:

a) identifikátor proteinu/domény,
b) celkový počet obsažených sekundárních struktur, tj. velikost konstruktu x,
c) průměrnou velikost sekundárních struktur v aminokyselinách, tj. průměrnou velikost konstituentů y.

2) Vypočítáme Pearsonův korelační koeficient r, Spearmanův koeficient pořadové korelace ρ a koeficient determinace R^2 lineární regrese pro zlogaritmizovaná data x a y. Pro tyto tři ukazatele pomocí *bootstrapu*[28] získáme 95% konfidenční intervaly. Tímto krokem zjistíme, zda jsou data v souladu s obecným trendem Menzerath-Altmannova zákona manifestovaného zápornými hodnotami koeficientů r a ρ. Koeficient determinace R^2 bude sloužit pouze pro ilustrativní kvantifikaci vysvětlitelnosti dat log-lineárním modelem. Výpočty hodnot r, ρ a R^2 aplikujeme na původní a zprůměrovaná data. Zprůměrovaná data vytvoříme tak, že pro každou velikost konstruktů $x \in X$ vypočítáme průměr velikostí jejich pozorovaných konstituentů $y_x \in Y$. Zprůměrování vícenásobných hodnot y nám umožní nahlédnout na *průměrný* a potenciálně tak i skrytý trend, který by nemusel být v množství dat jednoduše pozorovatelný a důležitěji, pomůže nám alespoň částečně pracovat s pozorovaným nerovnoměrným rozdělením hodnot x a y. V případě zprůměrovaných dat tedy bude každé partikulární velikosti konstruktu x odpovídat pouze jediná a průměrná hodnota $\bar{y}$. Užitím bootstrapu budou převzorkována původní *podkladová* data, ze kterých jsou průměry vypočítány tak, aby došlo k variaci především u málo zastoupených velikostí x. Pro odlišení budeme výpočty a výsledky prováděné na zprůměrovaných datech značit přidáním apostrofu, tj. r', $R^{2\prime}$ a ρ'.

   Hodnoty r, ρ a R^2 a jejich zprůměrované verze tedy budou opatřeny konfidenčním intervalem a pro následně splnění obecného trendu Menzerathova zákona a uznání jeho přítomnosti, je nutné splnit podmínku:

$$r, \rho, r', \rho' < 0 \quad \text{s 95\% spolehlivostí.}$$

   Výstupy tohoto kroku opatříme bodovým grafem originálních a logaritmizovaných dat pro vizuální kontrolu trendu a detailnější náhled na samotná data.

3) V případě, že na testovaném datasetu bude pozorován obecný trend Menzerathova zákona, provedeme proložení modelů (I.-IV.) na původních datech, a to pomocí optimalizačního algoritmu Levenberg-Marquardt s cílem minimalizovat součet kvadrátů chyb. V tomto případě, v protikladu se zjištěním obecného trendu v kroku (2), aplikujeme celou analýzu na původní nezprůměrovaná data. Důvodem tohoto rozhodnutí je vhodnost zatížení modelu vůči nejfrekventovanějším dosud pozorovaným počtům sekundárních struktur x tak, aby nejčastější hodnoty měly co nejkvalitnější predikci. Jedná se tedy čistě o pragmatický krok s cílem zvýšit kvalitu predikce u těch nejpravděpodobnějších velikostí konstruktů x. Pro kvantifikaci kvality proložení jednotlivých modelů vypočítáme metriky MAE (*mean-absolute-error*), RMSE (*root-mean-square-error*) a MAPE (*mean-absolute-percentage-error*):

[28] Metoda tzv. převzorkování, kdy jsou původní data náhodně permutována (tj. některá pozorování nemusí být vybrána, některá mohou být vybrána vícekrát). Tato metoda umožňuje simulovat odlišné situace z dostupných dat a je univerzální metodou právě při tvorbě konfidenčních intervlaů. Blíže o bootstrapu viz např. Efron a Tibshirani (1994). Pomocí bootstrapu budeme permutovat původní vzorek 6000x.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \quad ,$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \quad ,$$

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \quad ,$$

kde n je počet pozorování, y pozorovaná hodnota a $\hat{y}$ predikovaná hodnota. Každý nalezený parametr a metriku opatříme 95% konfidenčním intervalem pomocí *bootstrapu* permutujícího podkladová data. Abychom získali vizuální představu o tom, jak je model schopen reflektovat samotný průměrný trend v datech, vykreslíme pro každý model jeho proložení zprůměrovanými daty, která budou bootstrapována tak, abychom získali představu o alternativních proloženích, které do grafu zaneseme formou pásma extrémů. Zprůměrované hodnoty původních dat opatříme chybovými úsečkami znázorňující směrodatnou odchylku jejich vzorkování. Takový graf nám umožní ohodnotit způsob proložení průměrného trendu v datech a poskytne nám i informace o tom, jaký by měl model průběh s permutovaným datasetem.

4) Po získání parametrů a jednotlivých kvantifikací kvality proložení z předchozího kroku 3 vybereme model s nejnižší hodnotou RMSE, a to vzhledem k její penalizaci velkých chyb. Takto vybraný nejúspěšnější model dále otestujeme, zda je vůbec přínosný pro predikci průměrných délek sekundárních struktur (tj. hodnot y) a to tím, že musí být statisticky významně lepší než *tip* na průměr hodnot y nebo jejich modus, tj. musí být úspěšnější než *validační modely*. Pozorovanou chybu RMSE a MAE testovaného modelu tedy porovnáme s chybou RMSE a MAE validačních modelů predikujících vždy hodnoty *E(Y)* a *Mod(Y)* a jejich chybami uvedenými v tabulce 6. Pro kvalifikaci modelu jako přínosného (nebo také úspěšného) je nutné splnit podmínku, že RMSE testovaného modelu je statisticky významně menší než RMSE nejlepšího validačního modelu a totéž musí platit i pro chybu MAE, tj. pro úspěšný model musí platit:

$$RMSE(y - \hat{y}) < \min\left(RMSE\big(y - Mod(Y)\big), RMSE\big(y - E(Y)\big)\right)$$

$$\wedge$$

$$MAE(y - \hat{y}) < \min\left(MAE\big(y - Mod(Y)\big), MAE\big(y - E(Y)\big)\right) \quad ,$$

kde y jsou pozorované průměrné velikosti sekundárních struktur, $\hat{y}$ predikované průměrné velikosti sekundárních struktur testovaným modelem a *E(y)* je průměr hodnot y.

5) V případě, že některý z testovaných modelů Menzerath-Altmannova zákona bude statisticky úspěšný, budou dále zkoumány jeho vlastnosti a možné aplikace načrtnuté v úvodu této kapitoly. Zde je však nutné připomenout, že pro zmíněné aplikace dávají smysl pouze modely úspěšné na datasetech proteinů a nikoliv domén, jelikož slouží pouze jako doprovodná analýza evolučně starších celků.

Definovaná analýza vede v první řadě k otestování, zda se na úrovní proteinů či domén a sekundárních struktur projevuje Menzerathův zákon a zda je některý z dostupných modelů statisticky výhodnější než model *kvalifikovaného odhadu* tipem na průměr. V případě, že Menzerathův zákon v datech nalezneme a některý z modelů bude skutečně vystihovat data lépe než jejich pouhý průměr nebo modus, využijeme nejlepší nalezený model k otestování myšlenky vytvoření hypotetické skórovací funkce a jejímu testování. Tento krok je ovšem závislý na tom, zda je na úrovni proteinů a sekundárních struktur tento trend obsažen a v jaké kvalitě jej ty nejlepší modely dokáží predikovat. Nyní se proto podívejme na výsledky analýzy jednotlivých rovin.

Analýza domén

První, avšak z hlediska aplikací čistě doprovodnou analýzou, je analýza domén, anotovaných oběma definovanými typy.

Domény RSCB PDB filtrované CATH

Prvními analyzovanými daty jsou domény anotované osmi třídami sekundárních struktur DSSP pocházející z databáze RSCB PDB. V terminologii Menzerath-Altmannova zákona jsou zde konstruktem domény měřené v počtu sekundárních struktur, tj. hodnoty x. Konstituenty jsou zde sekundární struktury s délkou měřenou v počtu aminokyselin, tj. hodnoty y.

Prvním krokem definované metodiky testování je zjištění, zda je v logaritmizovaných datech a jejich zprůměrované verzi manifestován Menzerathův zákon. Test jeho přítomnosti je založen na zjištění Pearsonova korelačního koeficientu r a r' a Spearmanova koeficientu ρ *a* ρ', které pro potvrzení trendu musí vyjít se svými 95% konfidenčními intervaly záporné. Ilustrativně uvedeme i hodnotu koeficientu determinace R^2 a $R^{2'}$, který by měl poukázat na vysvětlitelnost dat logaritmicko-lineárním modelem. Výsledky výpočtů si prohlédněme v tabulce 7.

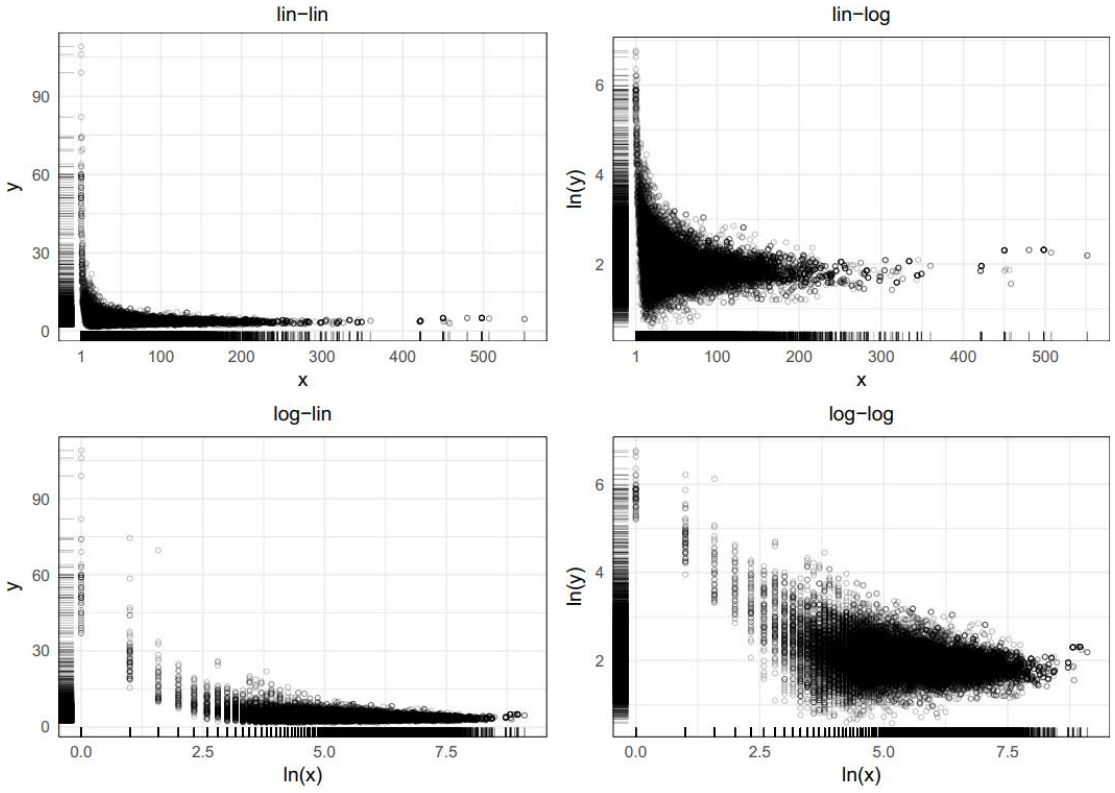
	Průměr	Dolní 95% CI	Horní 95% CI
r	**-0,4327642**	-0,4342517	-0,4312766
r'	**-0,7111040**	-0,7112536	-0,7109545
ρ	**-0,3113187**	-0,3114649	-0,3111724
ρ'	**-0,6362652**	-0,63668200	-0,6358485
R^2	0,2015123	0,2002665	0,202758
$R^{2'}$	0,5057039	0,50549074	0,505917

***Tabulka 7**: Výsledky korelací a koeficientu determinace datasetu RSCB & CATH.*

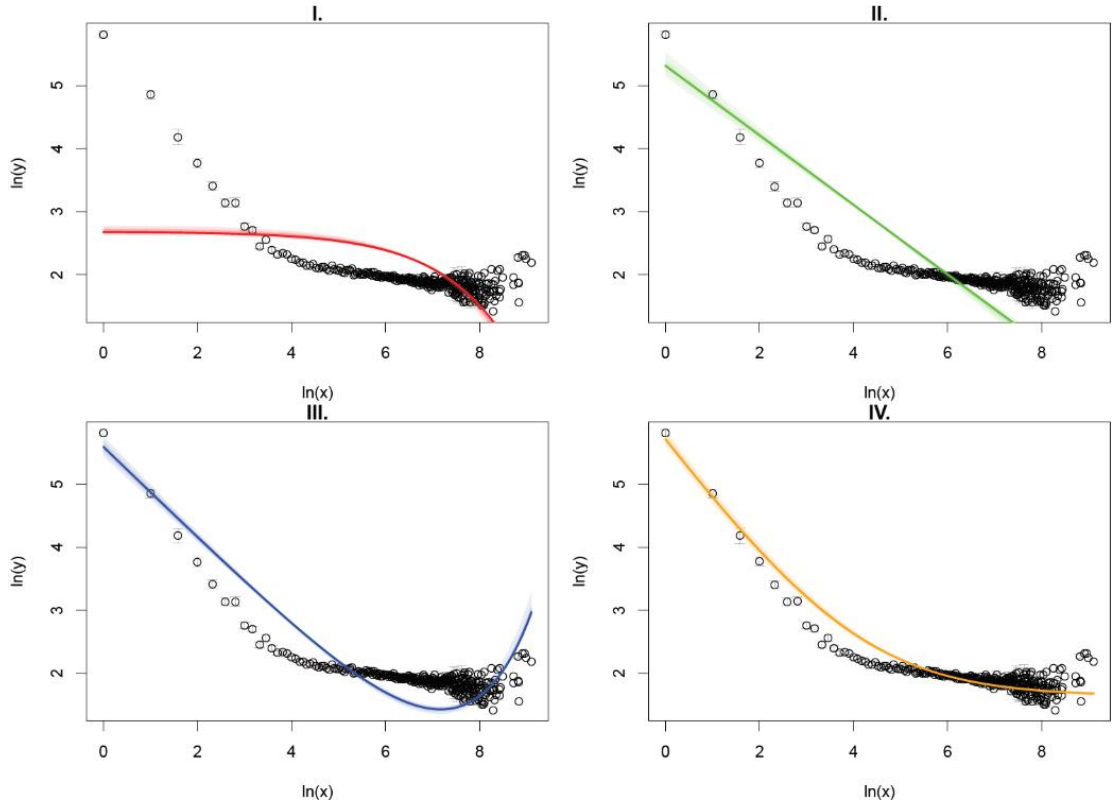
V této tabulce si můžeme všimnout, že v obou případech – na původních i na zprůměrovaných datech – jsou korelace r a ρ záporné včetně 95% spolehlivostního intervalu. V datech tedy identifikujeme obecný trend Menzerathova snižování průměrné velikosti sekundárních struktur s jejich narůstajícím počtem. V této tabulce si také můžeme všimnout, že zprůměrovaná data vykazují hodnoty blíže ideální hodnotě -1 než data tvořená všemi pozorováními. Takový výsledek můžeme interpretovat tak, že v doménách existuje *průměrný* trend lépe vyhovující Menzerathovu zákonu, než by mohlo být patrné z jednotlivých pozorování. Obdobně vychází i koeficient determinace, který v důsledku říká, že 50,6 % pozorované variance zprůměrovaných dat je vysvětlitelných log-lineárním modelem. Platí tedy nutné předpoklady pro uznání přítomnosti Menzerathova zákona a pokračování analýzy, tj. platí $r, \rho, r', \rho' < 0$ na 95% spolehlivostních intervalech.

Abychom měli nad výsledky korelací určitou kontrolu, zobrazíme testovaná data pomocí grafu. Velikosti konstruktů (počty sekundárních struktur) zobrazíme na ose *x* a průměrné délky jejich konstituentů (počty aminokyselin) na ose *y*. Výsledek zobrazení vidíme v grafu 3. V něm, a především v lineárním (lin-lin) a logaritmicko-lineárním zobrazení (lin-log), je zřetelně vidět trend snižování průměrné velikosti sekundárních struktur s jejich narůstajícím počtem. Na velikostech domén odpovídajících 400 sekundárním strukturám a dále na ose *x* si však můžeme všimnout, že některé domény tomuto trendu částečně uhýbají a začínají délku svých sekundárních struktur naopak zpět navyšovat. Takové pozorování můžeme vysvětlit minimálně dvěma způsoby. Prvním z nich je nedostatečný počet vzorků s danou velikostí konstruktů, které by po zprůměrování v čistě Menzerathově trendu pokračovaly. Další možností je, že se jedná o jev popsaný Altmannem (1980, 2), kdy za určitých okolností trend Menzerathova zákona přestane platit a průměrná velikost konstituentů začne s délkou konstruktu opět nabývat, tj. jev explicitně začleněný do modelu III. parametrem *c* (graficky viz v grafech dále).

Z grafů je tedy zřejmé, že v datech je přítomen trend ovlivňující velikost sekundárních struktur na základě jejich počtu, ovšem zda je tento trend možné modelovat některou z uvedených formulí Menzerath-Altmannova zákona a s jakou úspěšností vyzkoušíme dále. Nejprve se však podíváme, jak jsou jednotlivé modely schopné reflektovat průměrný trend uvnitř dat.



***Graf 3**: Zobrazení vztahu počtu sekundárních struktur domén (anotovaných pomocí DSSP) a jejich průměrných velikostí.*



***Graf 4**: Proložení zprůměrovaných a bootstrapovaných dat RSCB & CATH.*

Jednotlivé modely (I.-IV.) proto nyní proložíme zprůměrovanými daty. Každá hodnota x odpovídající počtu sekundárních struktur bude nyní zobrazena s jedinou průměrnou hodnotou y_x, což nám umožňuje nahlédnout na potenciální skrytý trend uvnitř dat a zároveň čistě kvalitativně posoudit, zda a jak dokáží jednotlivé modely přítomný trend reflektovat. Podkladová data hodnot x a y *bootstrapujeme*, takže získáme přehled o možné variabilitě proložení.

Kromě samotných proložení modelů v grafech nalezneme i další tři praktické informace. První a druhou informací budou zprůměrovaná data zobrazená pomocí kroužků a opatřená chybovou úsečkou o velikosti směrodatné odchylky, ve kterých se průměry při *bootstrapu* vyskytovaly – tímto získáme přehled o variabilitě průměrů. Třetí informací je doplnění proloženého modelu o pásmo, ve kterém se na základě *bootstrapu* nachází všechna alternativní proložení. Pro zvýšení přehlednosti tyto grafy zobrazíme v logaritmickém měřítku.

Pohledem na výsledek v grafu 4 zjišťujeme, že je zde skutečně přítomen určitý průměrný trend, který je lépe čitelnější než původní surová data zobrazená v grafu 3. U těchto dat si také můžeme všimnout, že prakticky nejsou zobrazeny žádné chybové úsečky a průměry hodnot y se tedy *bootstrapem* prakticky nemění.

U prvního modelu (I.) vidíme, že i přes nejlepší možné proložení nevystihuje viditelný průměrný trend, a to už přímo svým odlišným tvarem. Druhý a čistě mocninný model (II.) odpovídá v logaritmickém zobrazení přímce. Tento model vystihuje

pouze obecný trend snižování průměrné délky sekundárních struktur, avšak nevysvětluje „prohnutí“ dat pod proloženou přímkou. Model (III.) je zajímavější, neboť data vystihuje prozatím nejlépe i s registrací diskutované možnosti obrácení trendu pomocí parametru *c*. Můžeme si však všimnout dvou intervalů na ose *x*, které modelem zůstávají nevysvětleny, a to intervalu $x \approx \langle 1{,}8; 4{,}5 \rangle$, na kterém model nevysvětluje pozorování umístěné pod křivkou a ve druhém intervalu $x \approx \langle 6; 8 \rangle$ pak nevysvětluje pozorování umístěná nad křivkou. Důležité je, že tento model zmíněná pozorování nedokáže vysvětlit ani v případě permutace podkladových dat, protože v takovém případě by pozorování nebo chybové úsečky ležely ve vykresleném pásmu. Poslední model (IV.) vystihuje data subjektivně nejlépe, včetně určité plynulosti. Při bližším pohledu ale také zjistíme, že i pro tento model tu jsou pozorování, která zůstávají nevysvětlena, a to ani v případě odlišné permutace podkladových dat. Kvantifikace kvality proložení by tato subjektivní pozorování měla jen vyčíslit na původních surových datech.

Jednotlivé modely Menzerath-Altmannova zákona (I.-IV.) proložíme, tentokrát původními, nezprůměrovanými, tak, abychom nalezli nejvýhodnější konfiguraci, kterou porovnáme s validačním modelem. Data před prokládáním a samotnými výpočty *bootstrapujeme*, abychom získali přehled o možné variabilitě parametrů a metrik určujících velikosti chyb. Výsledkem jsou parametry modelu a metriky v tabulkách 8 a 9.

Model	*A*	95% CI	*b*	95% CI	*c*	95% CI
I.	5,6217	5,6192—5,6242			0,0047	0,0047—0,0047
II.	38,6904	38,6225—38,7584	-0,6270	-0,6276—-0,6265		
III.	45,7996	45,7486—45,8506	-0,7590	-0,7594—-0,7587	-0,0070	-0,00707—-0,00706
IV.	2.8739	2.8729—2.8749	46.8432	46.8069—46.8795		

***Tabulka 8**: Výsledné parametry proložení dat RSCB & CATH.*

Model	RMSE	95% CI	MAE	b 95% CI	MAPE	c 95% CI
I.	2,8481	2,8444—2,8519	1,03287	1,03210—1,03364	0,2270	0,2269—0,2271
II.	2,1237	2,1219—2,1255	1,47542	1,47422—1,47663	0,3615	0,3612—0,3617
III.	1,8015	1,7998—1,8031	1,14675	1,14610—1,14739	0,2725	0,2723—0,2726
IV.	1,5502	1,5484—1,5519	0,87933	0,87890—0,87976	0,2091	0,2090—0,2092

***Tabulka 9**: Výsledné metriky proložení dat RSCB & CATH.*

V tabulce 9 pomocí chyby RMSE rozhodneme o nejlepším modelu. Model s nejnižší chybou RMSE je model (IV.), a to nejen na základě RMSE, ale i MAE a MAPE, které jsou na 95% konfidenčních intervalech nižší než chyby ostatních modelů. Model (IV.) tak kvantitativně nejlépe odpovídá datům – průměrnou velikost sekundárních struktur dokáže na 95 % určit s chybou pouze 0,879 aminokyselin (chyba MAE). Takový výsledek můžeme interpretovat z aplikačního hlediska následovně: pokud modelu předáme počet sekundárních struktur neznámé domény, model se při predikci jejich průměrné délky v 95 % případů mine maximálně o 0,879 aminokyselin. Taková přesnost se zdá být až neuvěřitelně úspěšná, nicméně je nutné ji relativizovat vůči rozložení samotných dat (viz graf 2 a tabulka 6) a validačním modelům. Pouhý tip na průměrnou velikost sekundárních struktur $\bar{y}$ nám v tomto případě přináší v 95 % případů minimální chybu MAE 0,988 aminokyselin, což znamená, že model (IV.) je lepší přibližně jen o 11 %. Pro přijetí modelu (IV.) jako úspěšného metodika ukládá nutnost splnit statistickou významnost rozdílu RMSE a MAE od validačního modelu, kdy musí dle metodiky a tabulek 6 a 9 pro data RSCB & CATH platit:

$$1{,}5519 < \min(2{,}7; 2{,}7) \wedge 0{,}87976 < \min(0{,}988; 0{,}905)$$

Tato podmínka pro model (IV.) platí jako pro jediný z testovaných. Ostatní modely (I.-III.) lze nahradit validačním modelem tipujícím průměr nebo modus pozorovaných dat a nejsou tedy pro teoretickou aplikaci modelování a predikci přínosné. Důležité také je, že kvantitativně vybraný model je i ve shodě s kvalitativní interpretací vycházející z grafu proložení průměrného trendu dat výše. Dále se podíváme opět na domény, avšak anotované pouze třemi obecnými třídami.

Domény Uniprot filtrované CATH

Oproti předchozím výsledkům sekvencí domén získaných z databáze RSCB anotovaných DSSP jsou následující sekvence pocházející z databáze Uniprot anotované pouze třemi obecnými typy sekundárních struktur. Tato anotace sice vychází z anotace DSSP, nicméně redukovaným počtem typů zasahuje do množství a průměrných délek sekundárních struktur. Touto analýzou proto získáme určitý, avšak omezený náhled na to, která z anotací je výhodnější pro případné modelování dat pomocí Menzerath-Altmannová zákona. Musíme však dodat, že oba datasety obsahují odlišné vzorky a jejich odlišné počty. Srovnání je tak nutno brát s rezervou.

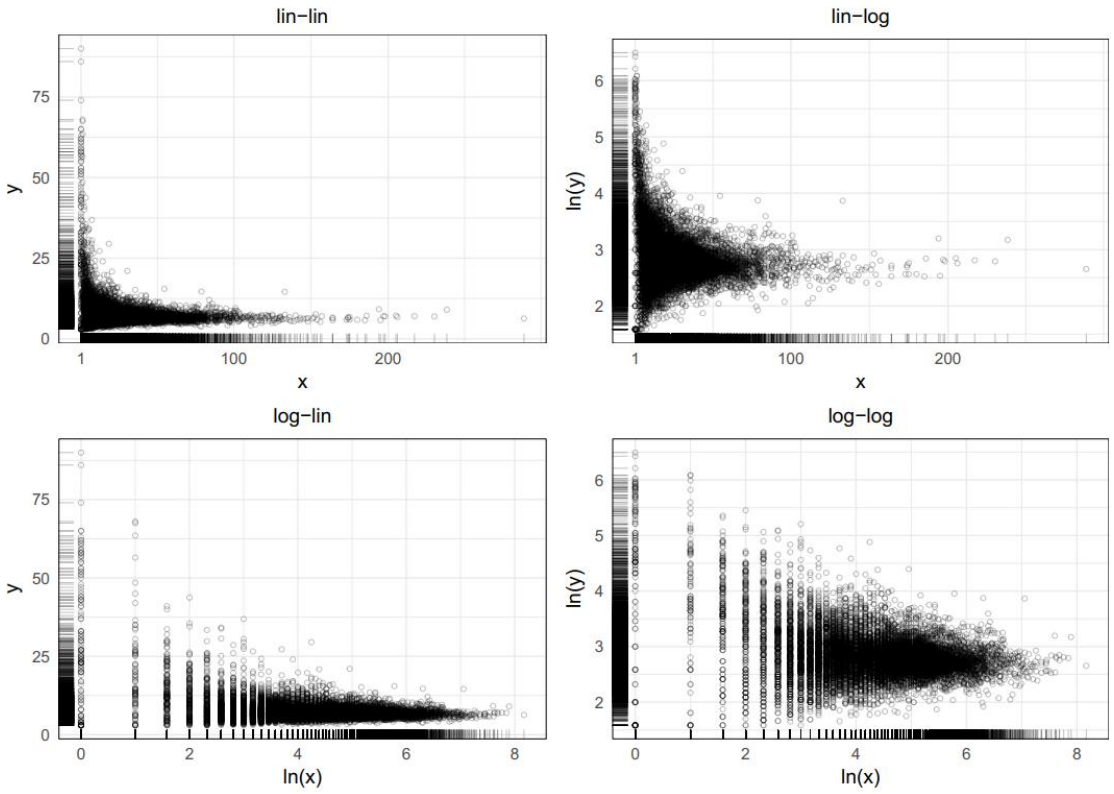
Prvním krokem analýzy je ověření trendu Menzerathova zákona, kdy na 95% konfidenčních intervalech musí platit negativní korelace Pearsonova korelačního koeficientu r a Spearmanova koeficientu ρ, včetně jejich variant pocházejících ze zprůměrovaných dat, tj. hodnoty r' a ρ'. Výsledky těchto výpočtů nalezneme v tabulce 10, ve které vidíme, že hodnoty jednotlivých koeficientů jsou v požadovaných intervalech záporné a splňují tím požadavek pro přijetí pozorování Menzerathova zákona.

Opět si zde můžeme všimnout, že se jednotlivé koeficienty blíží ideální hodnotě -1 v případě zprůměrovaných dat a můžeme tak očekávat, že průměrný trend bude opět lépe čitelný a konzistentní. Zajímavý je však propad u všech hodnot k horšímu ve srovnání s anotací DSSP z tabulky 7, kdy Pearsonův korelační koeficient na zprůměrovaných datech vycházel $r' = -0{,}71$, zatímco zde vychází $r' = -0{,}62$. Obdobný propad je čitelný u každé z vypočítaných hodnot.

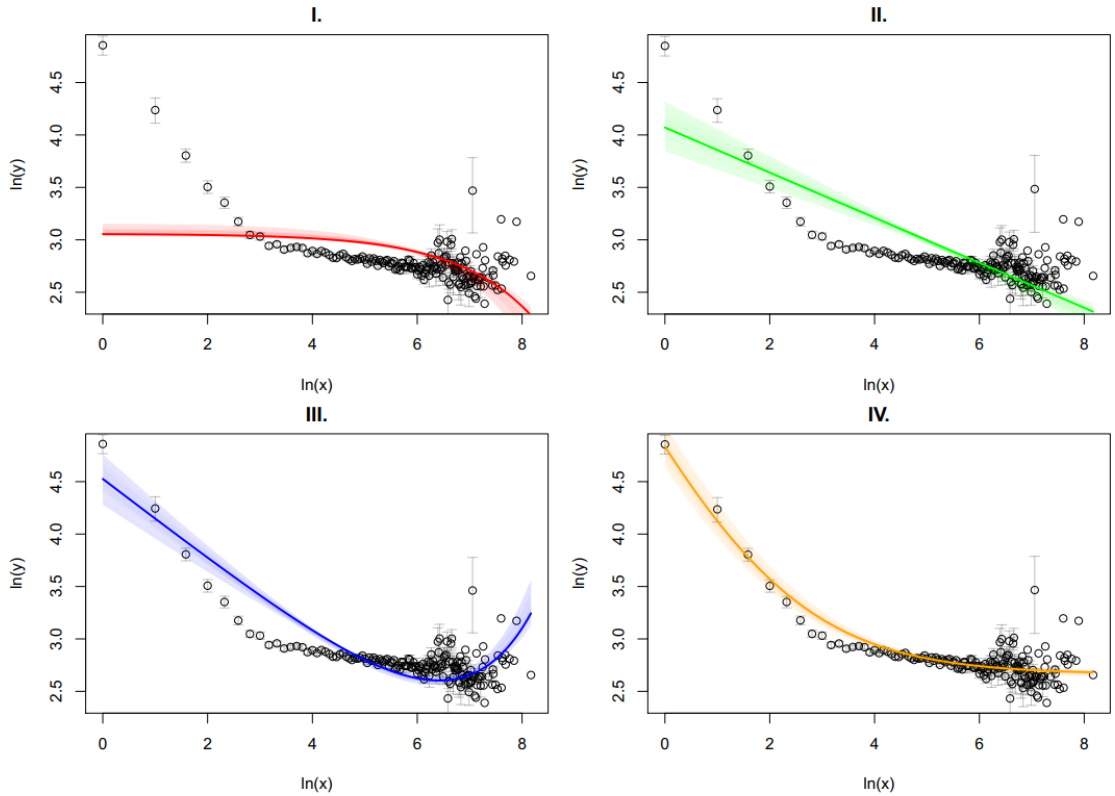
	Průměr	Dolní 95% CI	Horní 95% CI
r	**-0,35139265**	-0,3517233	-0,3510619
r'	**-0,62588877**	-0,6268513	-0,6249263
ρ	**-0,206306960**	-0,2065444	-0,2060695
ρ'	**-0,483959416**	-0,4849877	-0,4829311
R^2	0,12364753	0,1234153	0,1238797
$R^{2'}$	0,39318285	0,3919797	0,3943860

***Tabulka 10**: Výsledky korelací pro domény anotované třemi typy sekundárních struktur.*

Pohledem do grafu 5 zobrazujícího surová data zjišťujeme důvod zhoršení kvality pozorovaného trendu. V čistě lineárním zobrazení je dobře patrný trend Menzerath-Altmannova zákona. Tím potvrzujeme výsledky korelací, ale v logaritmicko-lineárním zobrazení (*log-lin*), oproti anotaci DSSP (graf 3) přibyly pozorování s nízkým počtem sekundárních struktur a současnou nízkou průměrnou velikostí. Kvalita pozorování trendu je však dostačující na to, abychom mohli provést test jednotlivých modelů. Nejprve se opět podíváme, zda a jak dokáží jednotlivé modely reflektovat průměrný trend v datech.



***Graf 5**: Zobrazení vztahu počtu sekundárních struktur domén (anotovaných pomocí tří typů s. struktur) a jejich průměrných velikostí.*



***Graf 6**: Proložení zprůměrovaných a bootstrapovaných dat Uniprot & CATH.*

Zprůměrovaná data permutovaná *bootstrapem* proložíme modely (I.-IV.) a výsledky zobrazíme včetně chybových úseček průměrů jednotlivých pozorování y a pásem proložení do grafu 6. Zde můžeme vyčíst několik důležitých poznatků o datech a kvalitě proložení. Nejprve si můžeme všimnout rozdílů v datech, kdy oproti *bootstrapu* domén anotovaných DSSP výše vytváří *bootstrap* anotace třemi třídami znatelně větší rozdíly v možných průměrech indikovaných chybovými úsečkami. Hlavní pozorovatelný rozdíl je u velkých domén pozorovatelných na ose x v intervalu $\langle 6; 8 \rangle$, které se oproti anotaci DSSP chovají chaotičtěji a s vyšší variabilitou.

Zhodnotíme-li však jednotlivé modely, nalezneme řadu podobností s předchozími výsledky. Nejvýhodnější proložení modelu (I.) se opět míjí s celkovým trendem pozorovaných dat, u kterých tento model neregistruje strmost poklesu průměrných délek sekundárních struktur. Model (II.) vystihuje pouze obecný trend, kdy si můžeme všimnout větší variability v možných proloženích. Model (III.) reflektuje data lépe, avšak stále nedokáže vysvětlit pozorovanou strmost poklesu. Model (IV.) následně můžeme, kromě relativně malého intervalu rovněž neregistrujícího strmost poklesu, opět subjektivně hodnotit jako model s nejlepším proložením. Tento model podtrhuje poněkud chaotický konec průměrného trendu, který vede k otázce, zda namísto kritické hranice otočení trendu Menzerath-Altmannova zákona (tak, jak je to řešeno v modelu III. pomocí parametru c) nepůjde spíše o určitou kritickou hranici, která nabízí možnost výběru dále setrvat v klesajícím trendu, nebo tento trend zcela opustit a zvyšovat průměrnou velikost konstituentů.

Lze tak spekulovat, že po překročení určité kritické meze přestane pro některé případy dávat využití Menzerathova zákona (respektive systému, který se jím manifestuje) význam a je vhodné jej opustit nebo se jej držet jen v určité míře. Dále jednotlivé modely porovnáme kvantitativně, abychom identifikovali ten nejlepší z nich.

Jednotlivé modely (I.-IV.) proložíme na původních bootstrapovaných datech, čímž získáme parametry a výsledky měření chyb v tabulkách 10 a 11. Z tabulky 11 s výsledky měření chyb modelů dle RMSE zjišťujeme, že nejlepšího proložení dosahuje model (IV.), a to opět dle RMSE, MAE i MAPE zároveň. Tento model dokáže určit průměrnou délku sekundárních struktur na 95 % s chybou maximálně 1,73359. Pouhý tip na nejfrekventovanější průměrnou délku sekundárních struktur validačním modelem však přináší nejlepší 95% přesnost 1,754 aminokyselin, což znamená, že model (IV.) je výhodnější jen o 1,16 %.

Model	*A*	95% CI	*b*	95% CI	*c*	95% CI
I.	9,5963	9,5921—9,6005			0,0088	0,00877—0,00879
II.	20,1633	20,1347—20,1920	-0,3376	-0,3381—-0,3371		
III.	24,0690	24,0359—24,1020	-0,4833	-0,4839—-0,4827	-0,0101	-0,01021—-0,01018
IV.	6,0584	6,0562—6,0605	22,1970	22,1626—22,2314		

***Tabulka 10**: Výsledné parametry proložení dat Uniprot & CATH.*

Model	RMSE	95% CI	MAE	95% CI	MAPE	95% CI
I.	4,0826	4,0785—4,0868	2,00119	1,99979—2,00259	0,2541	0,2539—0,2542
II.	3,6639	3,6608—3,6670	2,05912	2,05747—2,06077	0,2791	0,2788—0,2793
III.	3,5012	3,4984—3,5041	1,87642	1,87534—1,8775	0,2524	0,2522—0,2525
IV.	3,3793	3,3766—3,3820	1,73281	1,73203—1,73359	0,2320	0,2319—0,2321

***Tabulka 11**: Výsledné metriky proložení dat Uniprot & CATH.*

Abychom mohli model (IV). označit jako úspěšný, musí dosahovat nižší chyby RMSE a MAE než validační modely tipující průměr a modus. Dle metody a tabulek 6 a 11 proto musí platit:

$$3{,}3820 < \min(3{,}9; 4) \wedge 1{,}73359 < \min(1{,}923; 1{,}754)\ .$$

Tato podmínka platí a model (IV.) proto můžeme označit jako úspěšný v porovnání s validačními modely. Modely (I.-III.) naopak lze nahradit validačním modelem tipujícím průměr nebo modus dat a nejsou proto pro uvažovanou aplikaci nijak přínosné.

Shrnutí výsledků domén

Nyní lze shrnout, že na úrovní domén nacházíme vztah popsatelný jako Menzerath-Altmannův zákon. V případě obou typů anotací jsme dále zjistili, že modelem nejlépe reflektujícím pozorovaná data je model (IV.), což je zajímavé, neboť se nejedná o původní model, ale o model navržený v Milička 2014. Pozorování Menzerath-Altmannova zákona na rovině domén a sekundárních struktur sice není pro vytyčený cíl zcela klíčové, nicméně je přínosné z několika důvodů.

Prvním přínosem je doplnění existujících výzkumů uvedených výše – především pak doplnění Shahzad *et al*. 2015, kdy je potvrzeno pozorování Menzerath-Altmannova zákona na doménách i se započítáním sekundárních struktur. Druhým přínosem je nalezení modelu, který prozatím nejlépe odpovídá pozorovaným datům a může tak sloužit k přesnější interpretaci souvislosti velikosti domén a jejich sekundárních struktur. Třetím přínosem je možnost porovnání jednotek časově předcházejících proteiny. Čtvrtý přínos je podmíněn nalezením Menzerath-Altmannova zákona i na rovině proteinů. Pokud některé proteiny kombinují více domén, pak by pozorování shody i na těchto kombinacích něco vypovídalo o systému či způsobu jejich propojování. Dále ověříme, zda je Menzerath-Altmannův zákon přítomný i na úrovni proteinů.

Analýza proteinů

V předešlé části jsme analyzovali domény, u kterých jsme identifikovali obecnou tendenci snižovat průměrnou délku sekundárních struktur s jejich narůstajícím počtem, tj. tendenci, kterou můžeme považovat za shodnou s Menzerath-Altmannovým zákonem. Tento krok nám poskytl informace o evolučně starších konstruktech, než kterými jsou proteiny. Těm budeme nyní věnovat další pozornost.

Sekvence proteinů jsou obecně a pro uvažovanou aplikaci zcela kritické, neboť hypotetická aplikace spočívá ve schopnosti pro libovolný protein, který může nebo také nemusí být složen z více domén, dokázat určit míru *přirozenosti* designu jeho sekundárních struktur. Domény jsou v tomto ohledu pouze podmnožinou proteinů a pozitivní výsledky u nich nemusí nutně znamenat i pozitivní výsledky na úrovni proteinů.

Nalezení Menzerath-Altmannova zákona na doménách můžeme proto sice chápat jako úspěch, ten je však z uvažovaného aplikačního hlediska nevyužitelný. Prvními sekvencemi, na které se zaměříme, jsou proteiny s anotací DSSP pocházející z banky RSCB PDB a následně se podíváme na zobecněnou tří třídovou variantu pocházející z databáze Uniprot. Vzhledem ke komplexnosti proteinů lze jen nesnadno předvídat výsledky.

Proteiny RSCB PDB filtrované VAST

První sekvence proteinů určených k analýze pochází z databáze RSCB PDB s anotací sekundárních struktur osmi třídami DSSP. Prvním krokem je test přítomnost Menzerathova zákona na základě negativních hodnot Pearsonovy a Spearmanovy korelace na 95% konfidenčních intervalech, a to na původních i zprůměrovaných datech. Výsledky těchto korelací vidíme v tabulce 12. Zde můžeme pozorovat, že výsledky korelací potvrzují přítomnost Menzerathova zákona na rovině proteinů a sekundárních struktur svými negativními hodnotami.

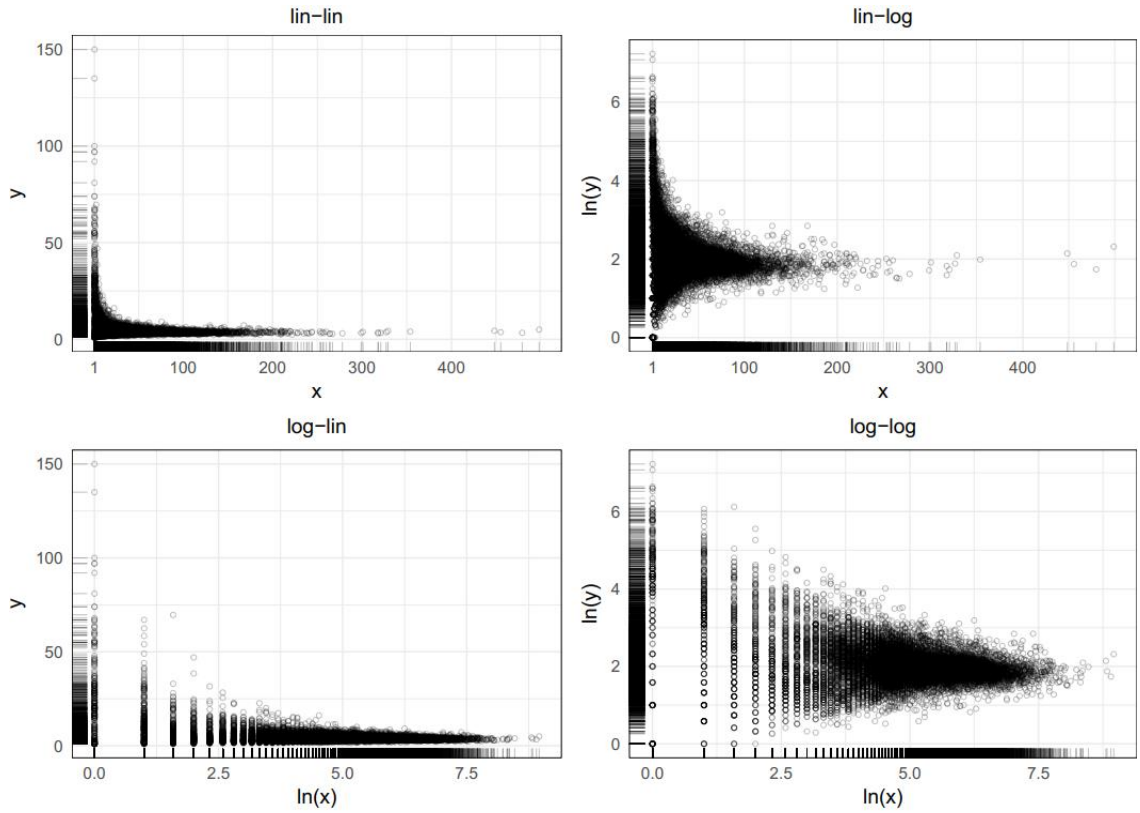
Hodnoty korelací jsou zde nižší než v případě domén užívající stejný typ anotace (viz tabulka 7), a to až o 12,8 % v případě hodnoty r' a o 18 % v případě ρ'. Zda je důsledkem pouze fakt, že jsou zahrnuty i proteiny evolučně kombinující více domén nebo je příčinou jiný faktor, např. způsob filtrace redundantních proteinů, prozatím nemůžeme vysvětlit. Rozdíly v datech však můžeme porovnat pomocí grafu 7. Zde zjišťujeme, že je skutečně viditelný obecný trend snižování průměrné velikosti sekundárních struktur na základě jejich počtu. Popsaný pokles kvality korelací, v po-

rovnání s doménami v grafu 3, má zřejmě příčinu v nové přítomnosti proteinů s nízkým počtem sekundárních struktur a jejich nízkou průměrnou velikostí. Ty jsou dobře viditelné v logaritmicko-lineárním zobrazení (*log-lin*) jako vyplnění prostoru mezi osou x a pomyslným hlavním trendem. I přes narušení kvality korelací je však tento trend signifikantně přítomen a můžeme přistoupit k identifikaci nejlepšího modelu, předtím se však podívejme, jak tyto modely vůbec dokáží reflektovat průměrný trend obsažený v datech.

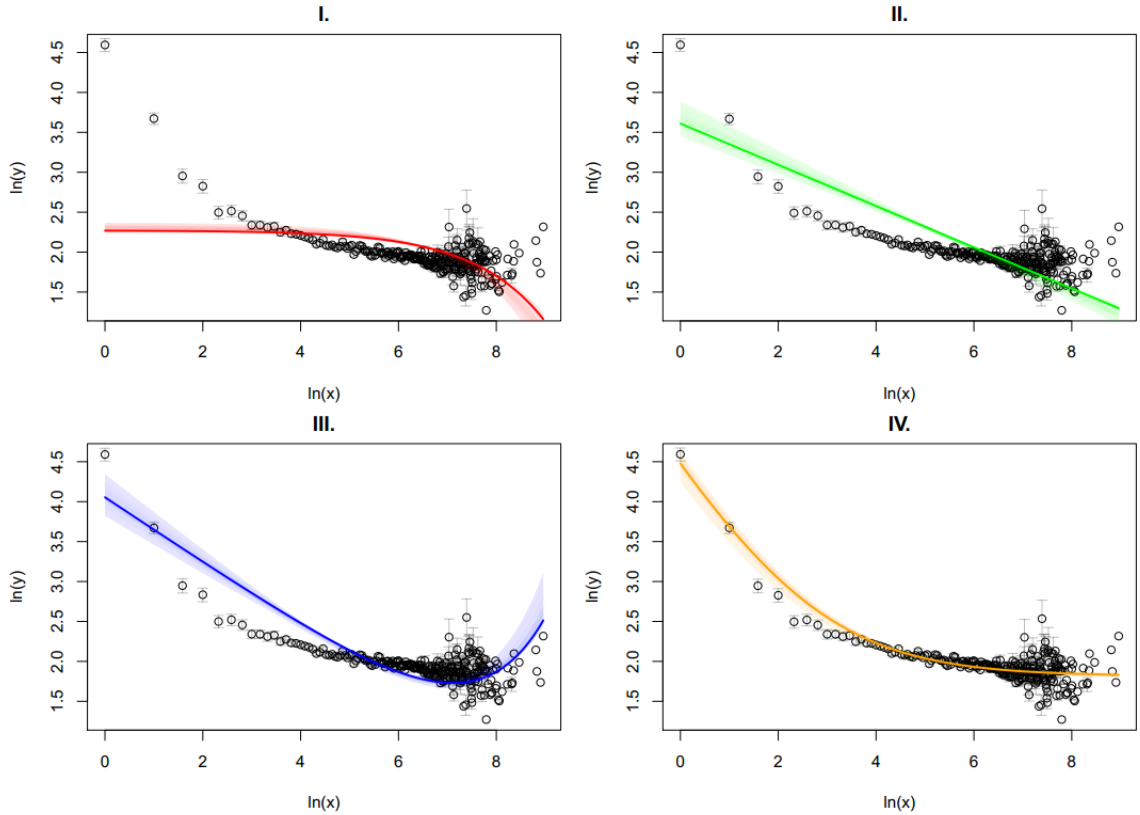
	Průměr	Dolní 95% CI	Horní 95% CI
r	**-0,37705056**	-0,37736922	-0,37673190
r'	**-0,62086406**	-0,62157204	-0,62157205
ρ	**-0,22409579**	-0,22432392	-0,22386766
ρ'	**-0,52124501**	-0,52215827	-0,52033174
R^2	0,14232563	0,14208557	0,142565692
$R^{2'}$	0,38625462	0,38537431	0,387134935

***Tabulka 12**: Výsledky korelací pro proteiny anotované DSSP.*

Jednotlivé modely (I.-IV.) proložíme zprůměrovanými a bootstrapovanými daty tak, abychom získali přehled o alternativních průbězích proložení a variabilitě průměrů jednotlivých počtů sekundárních struktur x. Výsledky vidíme v grafu 8. Pozorovaná data i průběhy proložených modelů se nápadně podobají výsledkům získaným z prokládání domén anotovaných třemi obecnými typy struktur, viz graf 6 a popis výše. Samotná zprůměrovaná data mají velmi podobný průběh, včetně chaotického zakončení, které od pohledu postrádá jednoznačný trend. Všimněme si však, že délky proteinů a domén jsou u této chaotické části odlišné, u proteinů jde o délky 6,5 – 8,5 v přirozeném logaritmu (tj. 90 až 362 sekundárních struktur) a u domén o délky 6 – 8 v přirozeném logaritmu (tj. 64 až 256 sekundárních struktur). Obdobná je i variabilita dat, která je shodně nejvyšší v chaotické části zmíněných intervalů. Jednotlivé modely rovněž vykazují prakticky stejné průběhy se stejnými nedostatky, jako je např. neregistrace strmosti poklesu u modelů (I.-III.). Nejlépe odpovídajícím modelem je zde opět model (IV.) a lze tak předpokládat, že bude nejúspěšnějším i na základě kvantitativních měřítek.



***Graf 7**: Zobrazení vztahu počtu sekundárních struktur proteinů (anotovaných pomocí DSSP; x) a jejich průměrných velikost (y).*



***Graf 8**: Proložení zprůměrovaných a bootstrapovaných dat RSCB & VAST.*

Původní data proložíme jednotlivými modely a pomocí *bootstrapu* získáme parametry a metriky kvality proložení. Nalezené parametry jsou uvedeny v tabulce 13 a metriky kvality v tabulce 14. V tabulce 14 opět identifikujeme model (IV.) jako model s nejnižší chybou RMSE, MAE i MAPE zároveň. Model (IV.) dokáže určit průměrnou velikost sekundárních struktur v 95 % s chybou maximálně 1,6 aminokyselin. Oproti tipu na nejčastější velikost validačním modelem je však chyba modelu nižší (v nejhorším případě) jen o 0,654 %.

Model	*A*	95% CI	*b*	95% CI	*c*	95% CI
I.	7,80949	7,80085—7,81813			0,01588	0,01584—0,01592
II.	18,72937	18,70364—18,75510	-0,46971	-0,47023—-0,46918		
III.	20,10708	20,08108—20,13308	-0,56366	-0,56422—-0,56310	-0,00735	-0,00736—-0,00733
IV.	3,38537	3,38392—3,38681	19,25031	19,22401—19,27662		

***Tabulka 13**: Výsledné parametry proložení dat RSCB & VAST.*

Model	RMSE	95% CI	MAE	95% CI	MAPE	95% CI
I.	4,75949	4,75306—4,76592	2,16032	2,15792—2,16272	0,49378	0,49324—0,49432
II.	4,18041	4,17460—4,18622	1,983	1,98138—1,98461	0,49247	0,49210—0,49285
III.	4,08925	4,08350—4,09499	1,78418	1,78281—1,78556	0,43643	0,43612—0,43674
IV.	3,96698	3,96132—3,97265	1,60342	1,6024—1,60443	0,39131	0,39106—0,39156

***Tabulka 14**: Výsledné metriky proložení dat RSCB & VAST.*

Abychom model (IV.) mohli označit za úspěšný a přínosný pro predikci průměrné délky sekundárních struktur pro neznámý protein, musí tato být výhodnější než validační modely tipující průměr a modus dat na základě chyb RMSE a MAE. Pro model (IV.) tak musí dle tabulek 6 a 13 platit podmínka:

$$3{,}97265 < \min(4{,}5; 4{,}6) \wedge 1{,}60443 < \min(1{,}9; 1{,}615)\ ,$$

což platí a model (IV.) můžeme považovat za úspěšný. Modely (I.-III.) lze pro predikci průměrného počtu sekundárních struktur dle jejich počtu nahradit tipem na průměr či modus empirických dat.

Prvotní výsledky na rovině proteinů jsou tedy alespoň částečně slibné v tom, že jsou konzistentní s poznatky získaných z domén a nalézáme na nich průměrný i obecný trend interpretovatelný jako Menzerath-Altmannův zákon a také, že tato data nejlépe modeluje právě model (IV.). Dále se podíváme na výsledky pro anotaci třemi třídami struktur.

Proteiny Uniprot filtrované VAST

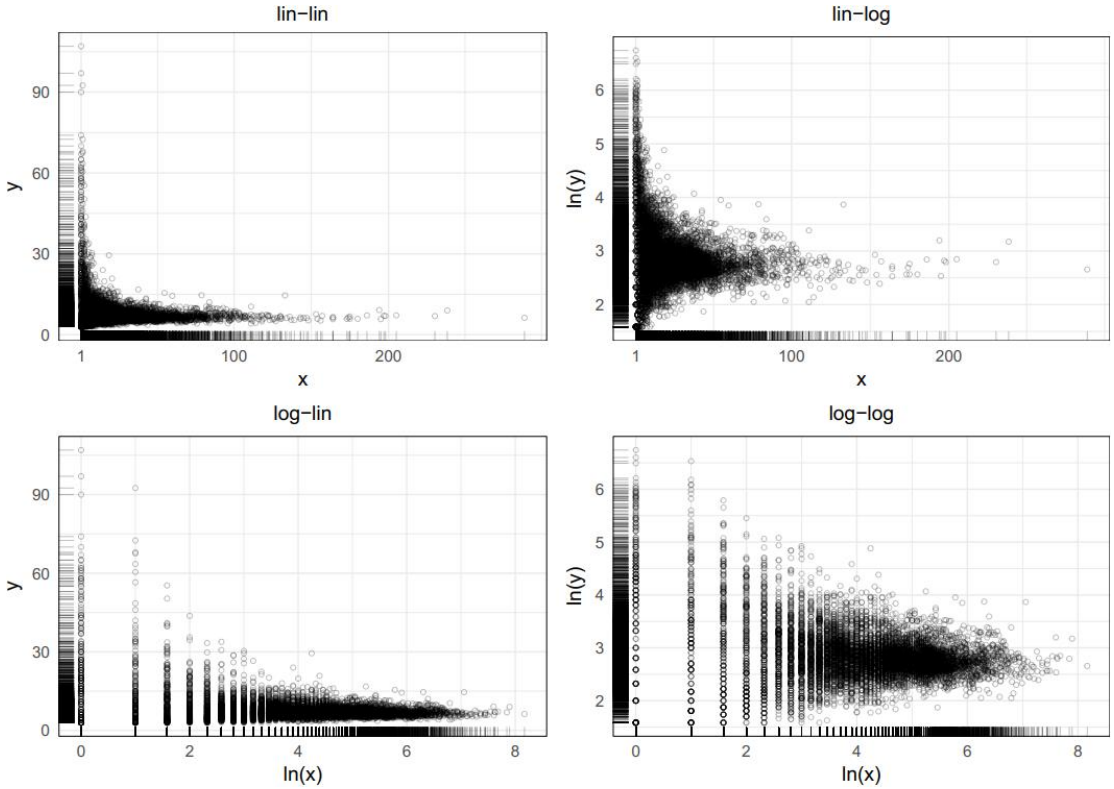
Posledními daty, na kterých otestujeme přítomnost Menzerath-Altmannova zákona, jsou sekvence získané z databáze Uniprot. Tyto sekvence, jak už bylo uvedeno dříve, jsou anotovány pouze třemi typy sekundárních struktur, které Uniprot zobecňuje z anotace DSSP. Na úrovni domén jsme u tohoto typu anotace zaznamenali pokles kvality trendu interpretovatelného jako Menzerath-Altmannův zákon a pokles jsme registrovali i přechodem na úroveň proteinů. Intuitivně tak můžeme předvídat, že u výsledků dojde k dalšímu zhoršení.

Prvním krokem analýzy je výpočet a ověření korelačních koeficientů r, r', ρ a ρ', které musí být na svých 95% konfidenčních intervalech záporné, abychom přijali možnost pozorování Menzerathova zákona. Pohledem na výsledky korelací v tabulce 15 zjišťujeme, že je trend Menzerathova zákona v datech přítomen. Oproti doménám se shodnou anotací jsou naměřené hodnoty dle očekávání horší, v případě r' o 18,88 % a v případě ρ' o 9 %. Pohledem na graf 9 se můžeme přesvědčit o kvalitě a obecnosti trendu Menzerath-Altmannova zákona. Data se v porovnání s detailněji anotovaným datasetem zobrazeným v grafu 7 příliš neliší, což nás vede k relativní předvídatelnosti výsledků.

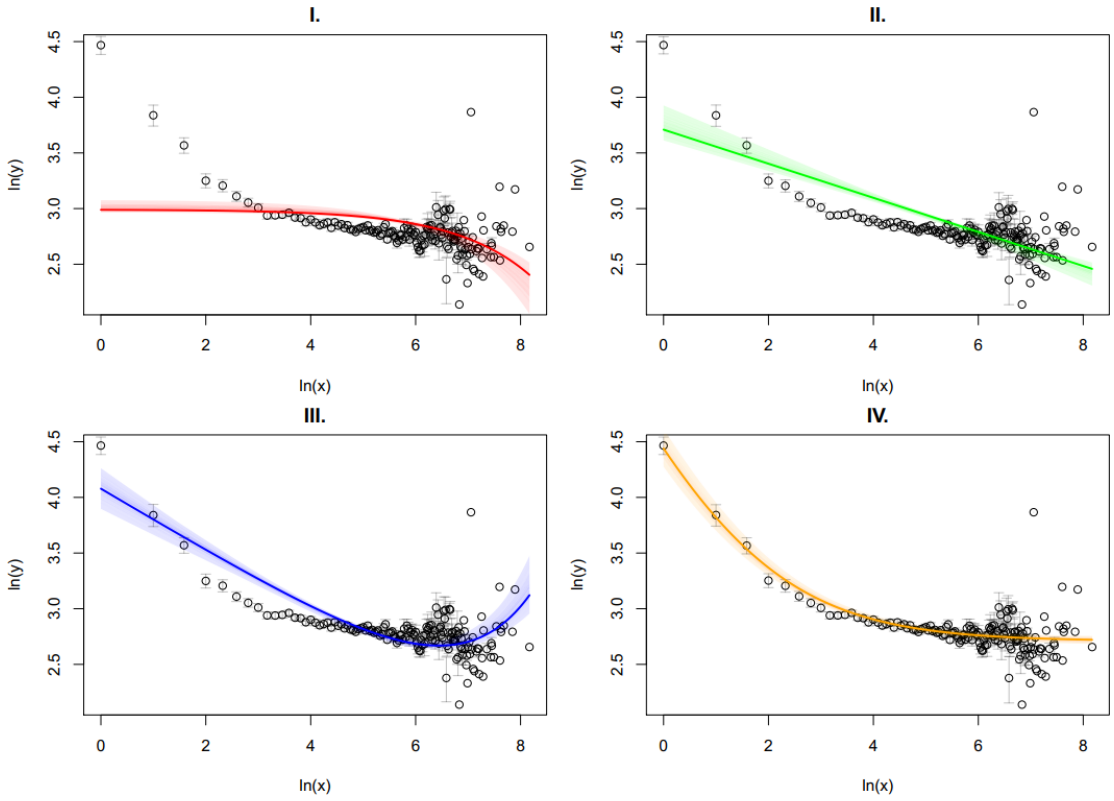
Dále se podíváme, jak dokáží jednotlivé modely reflektovat průměrný trend v datech. Výsledky proložení modelů na zprůměrovaných datech vidíme v grafu 9. Zde můžeme pozorovat vizualizace *bootstrapovaných* zprůměrovaných dat a jejich proložení. Grafy se opět jeví velmi podobně s těmi předcházejícími, v číslech 6 a 8, tj. nacházíme podobnosti jak na úrovni proteinů s anotací DSSP, tak i na úrovni domén a anotace třemi třídami sekundárních struktur. Jedinými snadno pozorovatelnými změnami je vyšší chaotičnost pozorování v rozmezí $x \approx \langle 6; 8 \rangle$ a přítomnost odlehlé hodnoty na $x \approx 7$. Způsob proložení jednotlivých modelů zůstává prakticky beze změny, kromě subjektivně lepší polohy modelu (IV.), který lépe odpovídá způsobu klesání.

	Průměr	Dolní 95% CI	Horní 95% CI
r	**-0,30336240**	-0,30373830	-0,30298650
r'	**-0,50775337**	-0,50922385	-0,50628296
ρ	**-0,17265327**	-0,17295054	-0,17235599
ρ'	**-0,43987670**	-0,44102430	-0,43872909
R^2	0,09224932	0,09202155	0,09247708
$R^{2'}$	0,26118856	0,25964930	0,26272782

***Tabulka 15**: Výsledky korelací pro proteiny anotované třemi typy sekundárních struktur.*



***Graf 9**: Zobrazení vztahu počtu sekundárních struktur proteinů (anotovaných pomocí tří typů sekundárních struktur; x) a jejich průměrných velikost (y).*



***Graf 10**: Proložení zprůměrovaných a bootstrapovaných dat Uniprot & VAST.*

Původní data dále proložíme všemi čtyřmi modely. Nalezené parametry jsou uvedeny v tabulce 16 a výsledky jednotlivých metrik v tabulce 17. Pohledem do tabulky 17 zjišťujeme, že dle chyby RMSE, ale i MAE a MAPE je opět nejlepší model (IV.). Ovšem v tabulce 6 a 17 také zjišťujeme, že validační model tipující modus dosahuje průměrné absolutní chyby MAE v nejlepším případě pouhých 2,294 aminokyselin, zatímco model (IV.) v tom nejlepším případě 2,35869 aminokyselin. Podmínka pro označení modelu jako úspěšného, tj.:

$$4{,}82391 < \min(5{,}3; 6{,}1) \wedge 2{,}36137 < \min(2{,}591; 2{,}294)$$

tak neplatí – model nemůžeme označit jako úspěšný, neboť jej lze nahradit tipem na modus dat, tj. tip na vlastnost vyplývající přímo z dat. Vzhledem k obsažení celého konfidenčního intervalu chyby MAE modelu (IV.) uvnitř konfidenčního intervalu validačního modelu tipujícího modus pak lze rovněž tvrdit, že rozdíl obou metod není z tohoto pohledu signifikantní. Nelze však říci, že by tento model selhal úplně, a to vzhledem k významně nižší chybě RMSE zohledňující velké chyby predikce. Model (IV.) tak lze, alespoň částečně, stále vnímat jako úspěšný, avšak ne aplikovatelný.

Model	*A*	95% CI	*b*	95% CI	*c*	95% CI
I.	10,29177	10,28592—10,29762			0,01123	0,01120—0,01125
II.	17,65472	17,63452—17,67492	-0,30038	-0,30081—-0,29994		
III.	19,36402	19,34255—19,38549	-0,40439	-0,40492—-0,40387	-0,00872	-0,00874—-0,00870
IV.	6,39829	6,39629—6,40029	15,49290	15,46765—15,51816		

***Tabulka 16**: Výsledné parametry proložení dat Uniprot & VAST*

.

Model	RMSE	95% CI	MAE	95% CI	MAPE	95% CI
I.	5,39094	5,38521—5,39668	2,72035	2,7181—2,7226	0,34549	0,34525—0,34573
II.	4,99438	4,98932—4,99943	2,62267	2,62067—2,62467	0,35505	0,35480—0,35530
III.	4,89733	4,89248—4,90218	2,46576	2,46418—2,46734	0,33161	0,33143—0,33180
IV.	4,81913	4,81436—4,82391	2,36003	2,35869—2,36137	0,31441	0,31425—0,31457

***Tabulka 17**: Výsledné metriky proložení dat Uniprot & VAST.*

Shrnutí výsledků proteinů

V tuto chvíli již víme, že nejen na doménách, ale i na celcích, které tyto domény mohou kombinovat, tj. na proteinech, pozorujeme vztah popsatelný jako Menzerath-Altmannovův zákon. Takové zjištění je pro cíl vytyčený na začátku této kapitoly kritické, neboť MAL by měl v této práci sloužit jako rukověť v heuristickému skóringu anotací sekundárních struktur. Výběr vhodného kandidáta na takový model je zde zjednodušen prakticky bezvýhradným úspěchem modelu (IV.), který, i přes některé pozorované nedostatky, můžeme prohlásit za možného kandidáta pro uvažovanou aplikaci.

Otázku výběru anotace pak můžeme opustit skrze nedostatečnou prediktivní kvalitu modelů na anotaci proteinů třemi třídami sekundárních struktur. Kromě identifikace vhodného modelu a anotace směřujících k plánované aplikaci však již máme v rukou poznatky opět doplňující řadu výzkumů potvrzením přítomnosti Menzerath-Altmannova zákona na úrovni proteinů, sekundárních struktur a aminokyselin. Je však otevřenou otázkou, jaký vliv na získané výsledky mají proteiny s více doménami.

Skóring anotace sekundárních struktur

Analýzou domén a proteinů jsme identifikovali model (IV.) jako prozatím nejvýhodnější způsob reflexe vztahu počtu sekundárních struktur $x \in \mathbb{N}$ a jejich průměrné délky $y \in \mathbb{R}$, parametrizovaný dvěma empiricky stanovenými parametry $A, b \in \mathbb{R}$ ve vzorci (Milička 2014):

$$y = A + \frac{b}{x} \quad .$$

Cílem, kterého bychom chtěli pomocí této formule modelující Menzerath-Altmannův zákon dosáhnout, je vytvoření určitého etalonu – modelového způsobu vztahu velikosti proteinu v počtu sekundárních struktur a jejich průměrné velikosti. Takový vztah by mohl být přínosný pro explanaci jevů v genetice a dále by mohl být aplikovatelný v případě predikce, kdy známe počet sekundárních struktur a chceme ověřit, zda se jejich průměrná délka chová v souladu s modelem. Představit si tak můžeme situaci, kdy řada predikčních softwarů sekundárních struktur vytvoří různé anotace vytvářející různé velikosti sekundárních struktur, na základě kterých bude vytvářen odhad terciální struktury sloužící pro odhad funkce proteinu v organismu.

Nyní už víme, že model Menzerath-Altmannova zákona dokáže poměrně dobře vystihnout průměrný trend vztahu počtu sekundárních struktur a jejich průměrných délek. Na základě tohoto modelu by bylo možné různým alternativám anotací přiřadit skóre určující blízkost k modelu. Tento prostý skóring tak může u časově náročných analýz pomoci vybrat a seřadit jednotlivé anotace podle toho, jak se jeví *přirozeně* a napomoci tak lépe zaměřit časově omezené prostředky. Takový cíl aplikace neříká, která anotace je špatně a která správně, ale napovídá, které z nich jako první věnovat pozornost.

Abychom uvažovanou aplikaci Menzerath-Altmannova zákona mohli přijmout z teoretického hlediska za možnou, musíme nejprve provést několik testů. První z nich musí odpovědět na to, zda není jednodušší a efektivnější ke stejnému cíli namísto modelu využít přímo vypočítané průměry či mody jednotlivých hodnot y_x konkrétních proteinů. Taková možnost tu skutečně je, nicméně je při jejím užití nutné zvážit několik pragmatických faktorů.

Prvním z nich a velmi důležitým je absence některých délek proteinů v dostupných datasetech. Například u datasetu RSCB chybí počty sekundárních struktur x = {186, 189, 196, 203, 206} a další. Absence takových pozorování vede k jejich případné interpolaci, a to opět pomocí určitého modelu.

Druhým pragmatickým faktorem je problém určení skutečného průměru v případě nízkého počtu pozorování. Průměry sice lze opatřit konfidenčními intervaly, ale v případě jednoho či dvou pozorování i tato metoda vede k nejistotě (viz např. proložení v grafu 10 zobrazující zprůměrovaná data).

Z těchto důvodů je vhodné použít model, který co nejlépe vystihuje pozorovaný trend dat. Absentující pozorování počtů sekundárních struktur x jsou následně interpolována na základě globálního, a nikoliv lokálního trendu. Pragmatickou námitkou proti užití modelu je tedy rozhodně jeho kvalita.[29]

Uvažované skóre anotace sekundárních struktur λ můžeme, zcela experimentálně, definovat jako kvadrát vzdálenosti mezi predikcí modelu $\hat{y}$ a pozorovanou hodnotou y:

$$\lambda = (y - \hat{y})^2 \quad ,$$

tj. v případě užití modelu (IV.) s definovanými parametry a a b:

$$\lambda = \left(y - \left(a - \frac{b}{x}\right)\right)^2 \qquad (1)$$

Interpretace výsledných hodnot λ je pak následující. Nulová hodnota λ znamená dokonalou shodu s modelem, tj. anotace sekundárních struktur proteinu odpovídá modelu. Růst hodnoty λ pak odpovídá postupnému vzdalování anotace od modelu, tj. porušování nalezeného vztahu. Vyšší hodnoty skóre λ tak odpovídají přisuzované horší kvalitě anotace/proteinu.

Z hlediska zápisu budeme používat zkrácenou verzi zápisu pomocí funkce $\mathcal{A}(m)$ určující skóre λ pro anotaci proteinu m. Uvedenou metodu zde můžeme pracovně nazvat jako *λ skóring*. Při výběru anotací jsou upřednostňovány ty, které minimalizují hodnotu skóre λ. Dále provedeme několik testů, které nám ukáží, zda je popsaná metoda a aplikace za naivních předpokladů schopna plnit vytyčený cíl. Kritéria nastavíme tak, abychom získali, v rámci možností, co nejvěrnější obraz schopností této metody.

[29] Pomocí genetických algoritmů (Schmidt a Lipson 2009) zde byly vyhledávány i výhodnější modely pro původní data proteinů RSCB. Dvěma nejúspěšnějšími nalezenými modely byly rozšíření modelu (IV.) o parametr c, a to $y_{ModelAlt1} = a + c * x + \frac{b}{x}$ a $y_{ModelAlt2} = a - c * L(x) + \frac{b}{x}$, kde L(x) je logistická regrese $L(x) = \frac{1}{1+e^{-x}}$ využívaná pro modelování exponenciálního růstu v prostředí s omezenými zdroji. Statisticky významné zlepšení vůči modelu (IV.) bylo pozorováno u zprůměrovaných dat z RSCB i Uniprot pro model ModelAlt2, na originálních datech je statistické zlepšení nevýznamné.

Testování metody náhodnými anotacemi

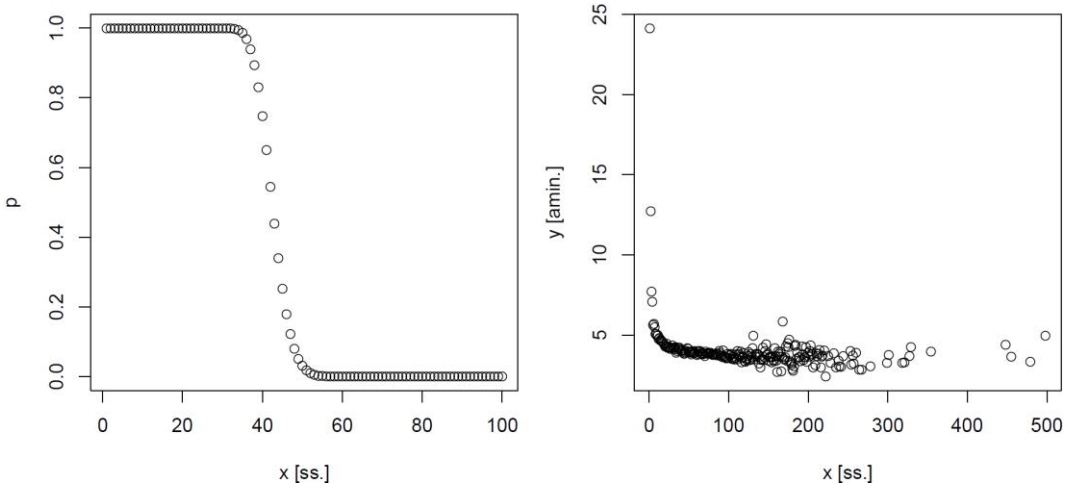
První test, který má odpovědět na to, zda je aplikace modelu Menzerath-Altmannova zákona na určování *přirozenosti* anotace sekundárních struktur proteinů pomocí λ skóre vůbec reálná, je zaměřen na relativně hrubé a zároveň však striktní předpoklady. Test spočívá ve vytvoření vzorku náhodných či také *falešných* anotací sekundárních struktur „proteinů". U těchto náhodných anotací následně zjistíme, s jakou pravděpodobností by došlo k jejich chybné záměně se skutečnou anotací reálného proteinu na základě nižšího λ skóringu. Tuto pravděpodobnost získáme pro každý pozorovaný počet sekundárních struktur x a tím nahlédneme na míru nejistoty, se kterou metoda dokáže pro tyto počty rozeznat náhodnou anotaci od té reálné.

Algoritmus testu je následující. Pro zadaný počet sekundárních struktur x náhodně z uniformní distribuce vybereme jejich délky. Možné délky sekundárních struktur (v aminokyselinách) jsou omezeny pozorovaným minimem L a maximem P. V našem případě se jedná o hodnoty z tabulky 4 a 5. Dále náhodně vybrané délky sekundárních struktur musí být v součtu menší, než empiricky známá největší délka proteinu (v aminokyselinách). Tímto alespoň částečně zaručíme, že vytvořené anotace budou svými vnějšími vlastnostmi odpovídat empirii. Je však zajímavé, že vytvoření náhodné anotace tímto způsobem není zcela jednoduché. Pravděpodobnost, že pro x sekundárních struktur o jejich maximální velikosti s (v aminokyselinách) náhodně z uniformní distribuce vybereme takovou kombinaci, která bude mít součet menší nebo roven maximální délce Z (v aminokyselinách), vypočítáme pomocí (2; odvozeno a upraveno z de Moivrovy pravděpodobnostní funkce, dle DasGupta 2010, 84):

$$p(x,s,Z) = \sum_{p=1}^{Z} \left(\frac{1}{s^x} \sum_{k=0}^{\left\lfloor \frac{p-x}{s} \right\rfloor} (-1)^k \binom{x}{k} \binom{p-ks-1}{x-1} \right) \quad . \qquad (2)$$

Enumerací (2) pro jednotlivé počty sekundárních struktur x získáme pravděpodobnosti, se kterými mohou náhodně a z uniformní distribuce vzniknout anotace sekundárních struktur proteinů odpovídajících dosavadní empirii. Na výsledek enumerace (2) pro specifikace z RSCB (viz tabulka 4), tj. $x = 1, \ldots, 100$; $s = 152$; $Z = 3245$ se podívejme do grafu 11. Zde vidíme, že do počtu 34 sekundárních struktur může s touto konfigurací vzniknout s naprostou jistotou anotace, která neporušuje empiricky stanovené parametry. Například pro $x = 20$ je prakticky jisté, že lze náhodně vybrat právě 20 délek sekundárních struktur z uniformní distribuce tak, aby jejich součet nepřesáhl empiricky známou maximální délku proteinů. Již při 55 sekundárních strukturách je pravděpodobnost takového náhodného výběru téměř nulová a u 60 sekundárních struktur nulová. To znamená, že při 60 sekundárních strukturách prakticky nelze náhodně a z uniformní distribuce vytvořit anotaci, která by neporušovala empiricky dané maximum délky proteinu. Přitom se můžeme z grafu 12 s daty z RSCB přesvědčit, že jsou zde obsaženy proteiny i o 500 sekundárních strukturách.

Problém výše navrženého testu tedy je, že pro počty sekundárních struktur ~60 a více nejsme schopni náhodně vytvořit anotaci, která by neporušovala empirické údaje a zůstala tak z hlediska základních kvantitativních vlastností shodná s těmi reálnými. Technicky proto generování délek sekundárních struktur $\delta_1, \delta_2, \ldots, \delta_x$ upravíme tak, aby v případě, kdy na první pokus nebude platit $\sum\delta < Z$, budeme pokračovat v náhodném generování těchto délek tak dlouho, dokud nebude tato podmínka splněna, nebo bude překročen počet pokusů F. V případě překročení počtu pokusů F, bude vybráno takové nalezené nastavení, které minimalizuje $\sum\delta$.



***Graf 11 (vlevo) a 12 (vpravo)**: Pravděpodobnost vytvoření náhodné anotace o x sekundárních strukturách limitovaných maximální délkou výsledného proteinu empiricky známým maximem. (vlevo) Počty sekundárních struktur proteinů z RSCB a jejich průměrná velikost. (vpravo)*

Tímto způsobem vytvoříme náhodné anotace $\mathcal{m}$, které by svými vnějšími parametry (počtem aminokyselin a minimální a maximální délkou sekundárních struktur) měly přibližně odpovídat či se blížit empirii. Tímto způsobem vytvoříme vzorek n náhodných anotací $\mathcal{m}_{1,\ldots,n}$ a vzorek jejich skóringů $\lambda_{1,\ldots,n}$, který označíme $\mathbb{L}$. Za ideálních okolností pro aplikaci by skóringy náhodných anotací $\mathbb{L}$ měly být oproti λ skóringu skutečných proteinů $\mathbb{T}$ vždy vyšší. To znamená, že i skutečný protein, který je od modelu nejdále a má tedy největší skóre λ, bude mít toto skóre stále nižší než ten nejbližší náhodný protein s nejnižším skóre λ.

Formálně tedy očekáváme, že pro každý počet sekundárních sekvencí x bude platit $\min(\mathbb{L}_x) > \max(\mathbb{T}_x)$. Vzhledem k nahodilosti však existuje vždy nenulová pravděpodobnost, že některý z náhodných proteinů bude přeci jen „lepší“ než ten „nejhorší“ reálný protein a metodou λ skóringu bychom tak chybně upřednostnili špatný protein (či anotaci). Pravděpodobnost, se kterou taková chybná záměna může nastat, označíme jako p_{Accept}. Pro množinu skóringů náhodně vygenerovaných proteinů $\mathbb{L}$, množinu skóringů skutečných proteinů $\mathbb{T}$ a zadaný počet sekundárních struktur x ji vypočítáme jako:

$$p_{Accept} = \frac{|W_x|}{|\mathbb{L}_x|}$$

$$W_x = \{\lambda \in \mathbb{L}_x \mid \lambda < \max(\mathbb{T}_x)\}$$

Pravděpodobnost $p_{Aaccept}$ tedy jednoduše říká, s jakou pravděpodobností bude jakýkoliv náhodně vygenerovaný protein „lepší“ v λ skóringu než-li ten nejhorší z reálných proteinů pro daný počet sekundárních struktur x.

Aplikací tohoto testu na anotace z RSCB i z Uniprot (pro srovnání, vzhledem k odlišným vlastnostem anotací) získáme první náhled na to, zda a s jakou pravděpodobností dokáže λ skórování rozeznat anotaci reálného proteinu od náhodně vytvořené anotace. Jak bylo popsáno, anotace jsou pro tento účel vytvářeny se specifiky blížícími reálným proteinům tak, aby bylo jejich rozeznání ztíženo. V testu pro každou délku sekundárních struktur x vygenerujeme 1 000 náhodných anotací proteinů a na jejich základě zjistíme pravděpodobnost chybné záměny p_{Accept}. Model, na základě kterého budou počítány hodnoty λ pro oba zdroje (RSCB a Uniprot), proložíme zvlášť na původních datech tak, aby odrážel jejich specifika. Výsledky hodnot p_{Accept} pro jednotlivé hodnoty x vidíme v tabulce 20, odkud můžeme vyčíst zásadní informace.

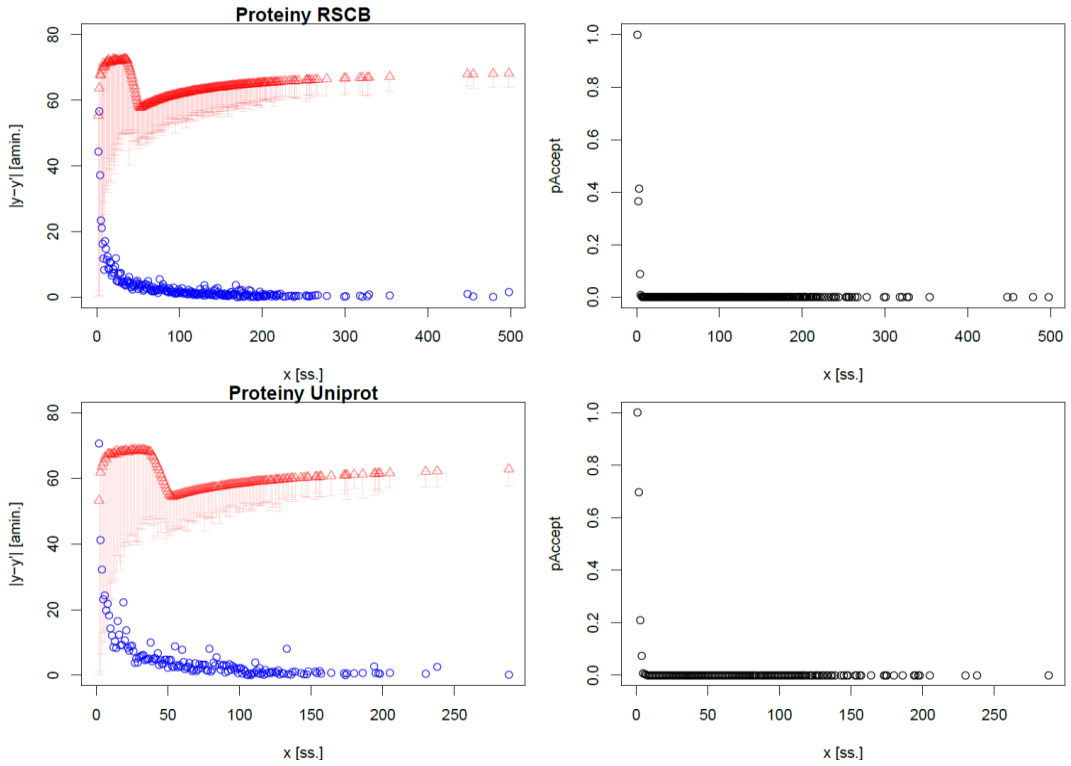
Zdroj	x [ss.]	1	2	3	4	5	6	7	8	9	...
RSCB	p_{Accept}	0,978	0,392	0,411	0,099	0,010	0,004	0	0	0	0
Uniprot	p_{Accept}	0,730	0,704	0,196	0,060	0,011	0,003	0,001	0,001	0	0

***Tabulka 20**: Pravděpodobnost chybného přijetí p_{Accept} náhodné anotace namísto anotace skutečného proteinu pro datasety RSCB a Uniprot.*

V případě proteinu obsahujícího pouze jedinou sekundární strukturu je velmi pravděpodobné, že metoda λ skórování nedokáže rozeznat náhodné anotace od těch skutečných, a to vzhledem k téměř 100% pravděpodonosti jejich záměny. S větším počtem sekundárních struktur tato pravděpodobnost následně klesá, až přibližně u počtu 9 zkonverguje k hodnotě 0. Z tohoto vyplývá, že od devíti a více sekundárních struktur ani jedna z 1 000 náhodných či falešných anotací nebyla proloženému modelu Menzerath-Altmannova zákona blíže než anotace skutečného proteinu. Od tohoto počtu tedy platí ideální předpoklad pro užitečnost této metody dané nulovou klasifikační chybou. U menších počtů sekundárních struktur, u kterých docházelo k chybě s pravděpodobností $p_{Accept} < 0{,}5$, by bylo nutné zvážit přínos užití této metody, pro menší počty s pravděpodobností chyby> 0,5 pak tato metoda nemá aplikační význam.

Zjištěná data a situaci lze pro větší vhled ilustrovat pomocí grafu 13, ve kterém vidíme dvě dvojice grafů s řádky odpovídajícími zdrojům RSCB a Uniprot. Levá dvojice grafů pro oba zdroje zobrazuje počty sekundárních struktur a λ skóre pro reálné anotace (modré kroužky) a průměrné λ skóre náhodných anotací (červené body) s vyznačením jejich minimálních hodnot (červené úsečky). Ose x zde tedy odpovídá model

Menzerath-Altmannova zákona. Pravý graf z dvojice následně zobrazuje pravděpodobnost p_{Accept} (tj. data z tabulky 20). U obou zdrojů můžeme v grafech zobrazujících λ skóre (vlevo) sledovat odchylku reálných anotací od modelu Menzerath-Altmannova zákona (odpovídající ose x), která se s rostoucím počtem sekundárních struktur postupně vytrácí. Pro náhodně vytvořené anotace je tento trend téměř opačný. Z počátku je z empiricky nastavené uniformní distribuce snadné vytvořit anotace, které jsou modelu blíže než ty skutečné, a to díky jejímu vysokému průměru ($\mu = 76$ aminokyselin pro data z RSCB a $\mu = 75{,}5$ aminokyselin pro Uniprot). Následně toto nastavení distribuce interferuje s rostoucím počtem sekundárních struktur a heuristickým omezením součtu jejich délek na empirické maximum. Pravděpodobnost záměny náhodné anotace se skutečnou tak rychle klesá (viz popsaná tabulka 20 a pravé grafy v grafu 13) spolu s tím, jak u náhodných anotací opět začne docházet vlivem počtu sekundárních struktur ke konvergenci jejich průměrné délky k průměru distribuce.



***Graf 13**: Vizualizace výsledků testování metody λ skórování na falešných proteinech vytvářených z uniformní distribuce. Vlevo vidíme vzdálenosti skutečných a zároveň nejvzdálenějších proteinů (modře) a průměrných falešných proteinů (červeně) od modelu (osa x). Červené chybové úsečky sahají k hodnotě umístění nejlepších falešných proteinů.*

Na základě těchto výsledků lze říci, že metoda λ skórování je v úloze odhalení náhodně vytvořených anotací z uniformní distribuce spolehlivá, a to přibližně od devíti a více sekundárních struktur, v závislosti na typu anotace.

Dále jsme odvodili, že náhodné vytváření anotací proteinů z této distribuce odpovídající vnějším vlastnostem skutečných anotací je od jistého počtu sekundárních struktur těžké a následně i nepravděpodobné. Důvodem je především takový výběr délek sekundárních struktur, které v součtu nesmí překročit empiricky pozorovanou maximální délku sekvencí reálných proteinů, což je komplikováno zmíněným vysokým průměrem distribuce.

Úloha vytvořit náhodnou anotaci splňující všechny vnější vlastnosti skutečných anotací (minimální a maximální délky sekundárních struktur a jejich maximální délka v součtu) bude snazší v případě, kdy jsou tyto délky získávány z normálního či jiného unimodálního rozdělení, především pak takového, které přebírá parametry přímo ve shodně s empirií. V takovém případě by nemělo být těžké získat anotace, které nepřesáhnou empiricky stanovené maximum délky proteinů, a to vzhledem k nízké empiricky známé průměrné velikosti sekundárních struktur, což jsou přibližně 4 aminokyseliny pro RSCB a 7 aminokyselin pro Uniprot (viz tabulky 4 a 7).

Dopředu tak můžeme spekulovat, že užití unimodální distribuce bude při tvorbě falešných anotací úspěšnější v jejich záměnitelnosti s těmi skutečnými. Ovšem již můžeme také předvídat, že s unimodalitou distribuce dojde k následujícímu jevu: průměr hodnot y_x bude s narůstajícím počtem sekundárních struktur x nebo s narůstajícím počtem vzorků anotací konvergovat ke konstantě a nedá tak možnost vzniknout trendu Menzerath-Altmannova zákona, který tak zůstane tímto procesem dále nevysvětlen.

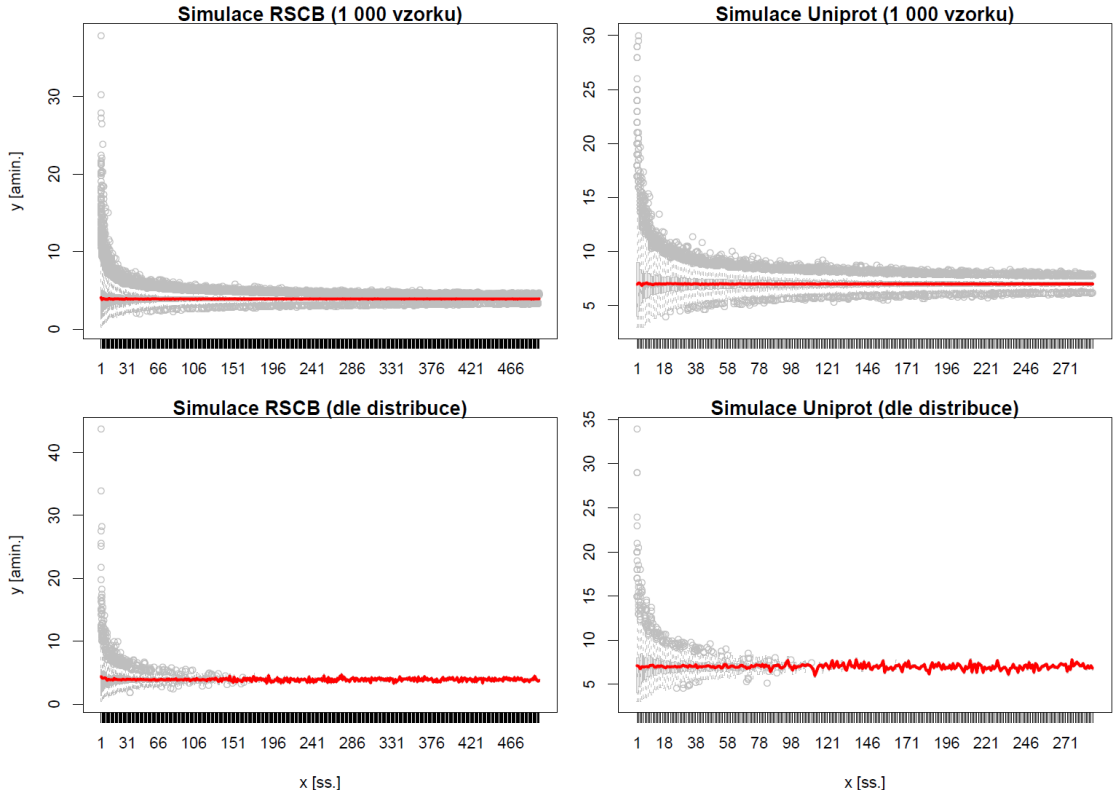
Vytvoření náhodných anotací budeme pro následující testování provádět náhodným výběrem z modelových distribucí sekundárních struktur reflektujících empirická data a identifikovaných metodou maximální věrohodnosti (*likelihood maximalization*). Pro data z RSCB bylo jako nejvěrohodnější nalezeno inverzní Gaussovo rozložení (3) s parametry $\mu = 3{,}8998$ a $\lambda = 3{,}14$ a pro data z Uniprot byla jako nejvěrohodnější nalezena negativní binominální distribuce (4) s parametry $\mu = 4$ a $p = 0{,}9$. Vytvářené anotace opět musí dodržet pravidlo maximální délky součtu obsažených sekundárních struktur pomocí výše uvedený heuristiky.

$$IG(x, \mu, \lambda) = \left[\frac{\lambda}{2\pi x^3}\right]^{1/2} \exp\left\{\frac{-\lambda(x-\mu)^3}{2\mu^2 x}\right\} \tag{3}$$

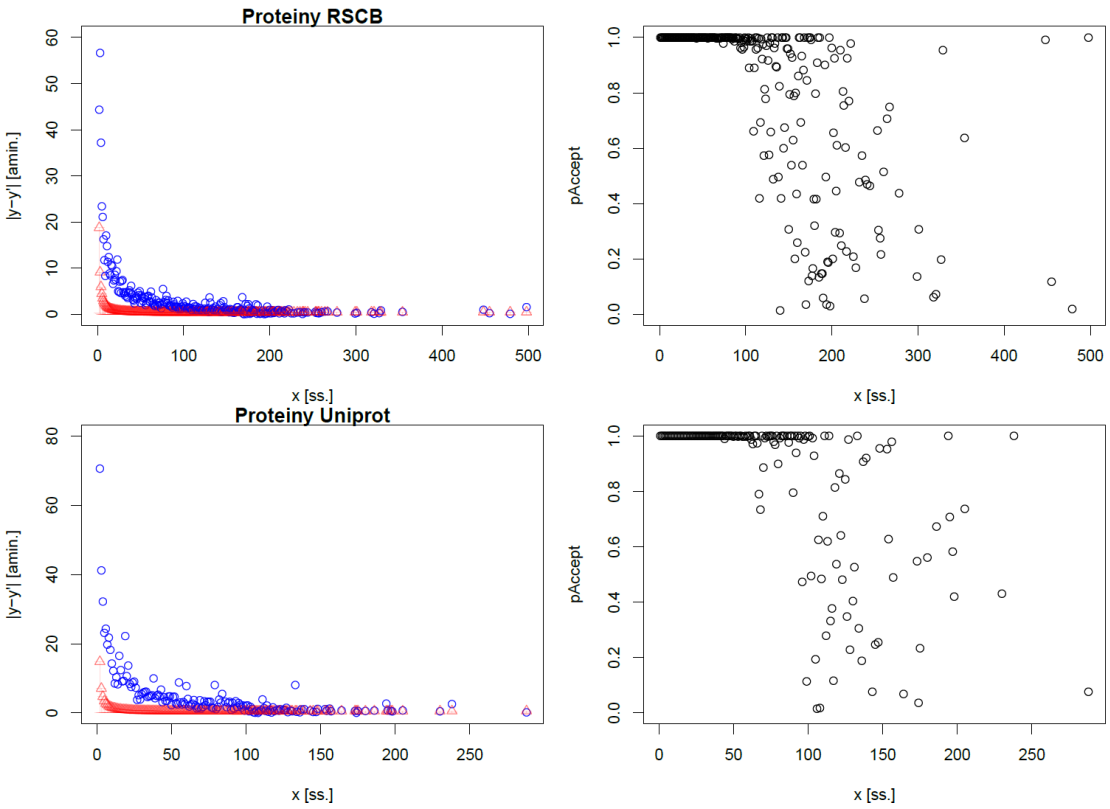
$$NB(k, \mu, p) = \binom{\mu + k - 1}{k} p^k (1-p)^{\mu} \tag{4}$$

Důležité ovšem také je, že samotná distribuce vzorků pro každý počet sekundárních struktur x není uniformní (viz histogramy v grafu 1), což by mohlo vést k úpravě hodnot výsledných průměrů. Kromě shodného počtu vzorků proto testování provedeme i na základě jejich empirické distribuce dle dat z grafu 1 a tabulek 4 a 5.

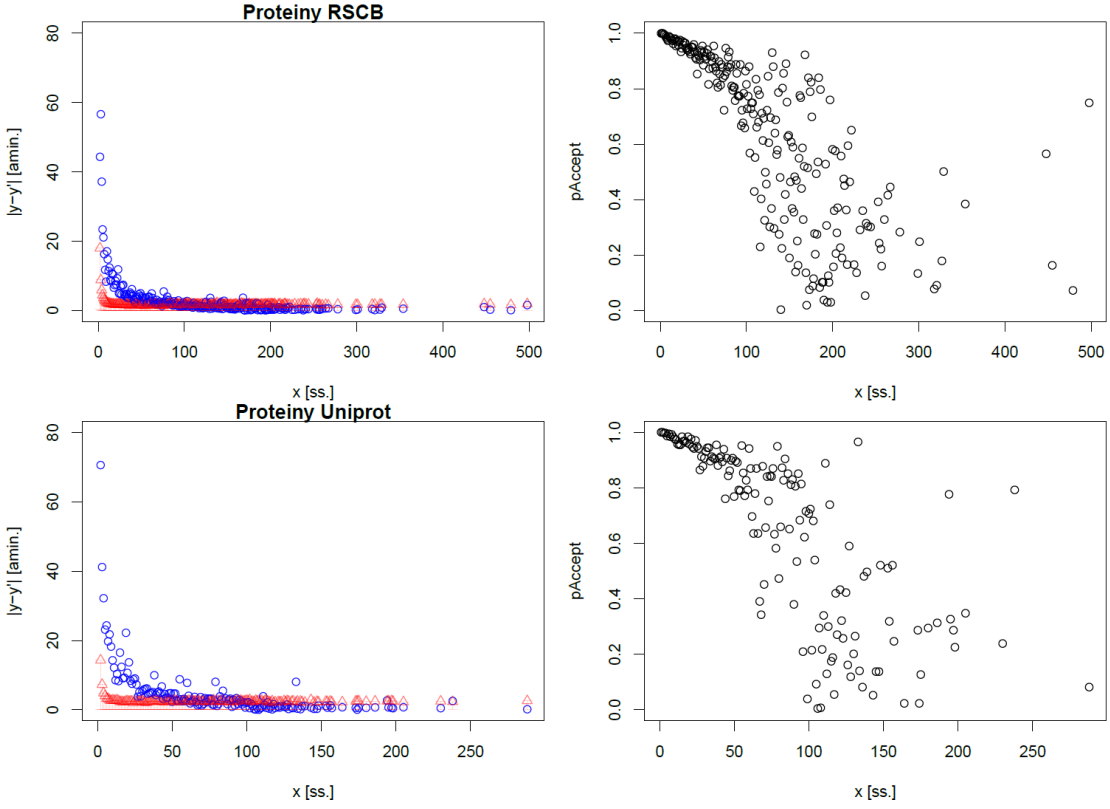
Výsledek generování náhodných sekvencí unimodálními distribucemi pro oba zdroje si následně můžeme prohlédnout v krabicovém grafu 14, ve kterém horní řada obsahuje pro každé x vždy 1 000 vzorků náhodných anotací a spodní řada v těchto počtech reflektuje empirickou distribuci. Červenou linkou jsou zde zvýrazněny průměrné hodnoty y. Ze všech grafů je právě na základě této linky zřejmé, že i přes užití věrohodných distribucí nedošlo k vytvoření průměrného projevu Menzerath-Altmannova zákona, neboť linka zůstává téměř konstantní. Vliv empirické distribuce počtu vzorků pro x je viditelný v množství odlehlých hodnot znázorněných kroužky vytvářejícími zdání Menzerath-Altmannova zákona. Z tohoto pozorování lze říci, že MAL není důsledkem náhodného výběru z jedné nebo ze dvou unimodálních distribucí. Vznik projevů MAL je nyní vedlejší otázkou. Hlavní otázkou, ke které se opět vrátíme, je, zda metoda λ skórování dokáže rozlišit mezi takto vytvořenými nejlepšími *falešnými* anotacemi a těmi *nejhoršími* skutečnými. Vzhledem k popsanému množství odlehlých hodnot kopírujících Menzerath-Altmannův trend však můžeme i nadále očekávat, že dojde k selhání celé metody. Na výsledek testu záměny *nejhorších* skutečných anotací proteinů od těch *nejlepších* falešných se můžeme podívat v grafech 15 a 16.



***Graf 14**: Náhodné anotace proteinů vytvořené z modelových distribucí délek sekundárních struktur s fixním počtem vzorků pro každý počet sekundárních struktur x (horní řada) a s počtem odpovídajících empirii (dolní řada). Červená linka odpovídá průměru hodnot y.*



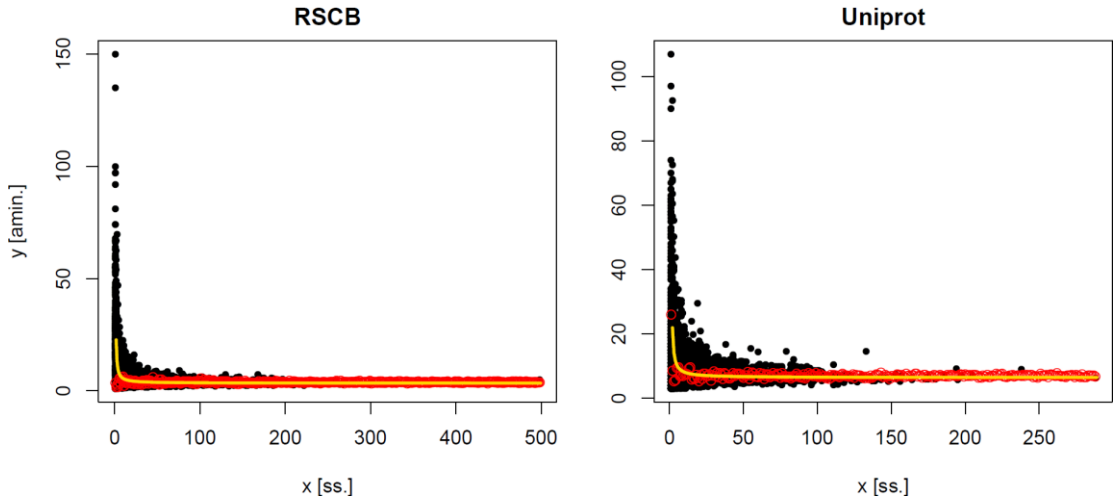
***Graf 15**: Vizualizace výsledků testování metody λ skórování na falešných proteinech vytvářených z uniformní distribuce. Vlevo vidíme vzdálenosti skutečných nejvzdálenějších proteinů (modře) a průměrných falešných proteinů (červeně) od modelu (osa x). Červené chybové úsečky sahají k hodnotě umístění nejlepších falešných proteinů.*



***Graf 16**: Vizualizace výsledků testování metody λ skórování na falešných proteinech vytvářených z empirické distribuce. Vlevo vidíme vzdálenosti skutečných nejvzdálenějších proteinů (modře) a průměrných falešných proteinů (červeně) od modelu (osa x). Červené chybové úsečky sahají k hodnotě umístění nejlepších falešných proteinů.*

Z výsledků v grafu 15 i 16 zjišťujeme, že se falešné anotace ze začátku svým průměrem umísťují modelu Menzerath-Altmannova zákona mnohem blíže než ty skutečné (grafy vlevo), a to v případě dat z RSCB i z Uniprot. Grafy pravděpodobnosti záměny (vpravo) tuto blízkost jednoduše vystihují, kdy si můžeme všimnout, že pro shodné počty vzorků x (graf 15) je přibližně do první stovky sekundárních struktur prakticky jisté, že v průměru bude náhodně vytvořená anotace modelu Menzerath-Altmannovu zákonu blíže než skutečná anotace. Pro počty vzorků x odpovídajících empirii (graf 16) je tato jistá chybovost omezena pouze pro několik prvních délek sekundárních struktur. Chybovost klasifikace metody se však s narůstajícím počtem sekundárních struktur nezlepšuje, neboť pravděpodobnost chyby stále fluktuuje.

Z tohoto výsledku lze odvodit, že vytváření náhodných anotací pomocí empiricky zjištěné i uniformní distribuce počtu vzorků nevede k Menzerath-Altmannově trendu a zároveň zjišťujeme, že užití modelu (IV.) nevystihuje ty nejodlehlejší (*nejhorší*) skutečné anotace natolik dobře, aby je dokázal od těch falešných odlišit. Na takové pozorování můžeme přímo nahlédnout v grafu 17 zobrazujícím hodnoty x a y skutečných (černě) a falešných (červeně) anotací, včetně proloženého modelu (oranžová křivka). Zde si můžeme všimnout, že modelu na začátku unikají extrémně vysoké hodnoty y skutečných anotací, a proto považuje falešné proteiny držící se u nízkého průměru za bližší. Následně je zřetelná konvergence obou dat k průměru a zaměnitelnosti.



***Graf 17**: Skutečné proteiny (černé) a vygenerované falešné proteiny z distribucí (červené) včetně proloženého modelu (IV.) (žlutě).*

Z dosavadních výsledků je tak zřejmých několik důležitých poznatků. Již je jasné, že původně zamýšlená aplikace projevů případně se vyskytujícího Menzerath-Altmannova zákona pro identifikaci skutečné či *přirozené* anotace není z hlediska těch nejstriktnějších testovaných vlastností bez problémů. Jakmile bude anotátor sekvencí využívat empiricky známé distribuce délek a počtů sekundárních struktur proteinů, dosáhne do jejich určitého počtu lepšího λ skóre než skutečné proteiny. Užití λ skóre tak sice může vést k nutné, ale nedostačující, a především neostré podmínce identifikace nestandardního proteinu.

Dalším nalezeným a důležitým poznatkem je, že trend, který můžeme sledovat v průměru hodnot y anotací skutečných proteinů nevzniká triviálně náhodným výběrem z jedné či dvou distribucí, ale je důsledkem systému upravujícího jejich využití.

Posledním důležitým poznatkem je možná irelevance modelu (IV.), který nedokáže svým průběhem vystihnout extrémně vysoké hodnoty y pro nízké počty sekundárních struktur x a tím zapříčiňuje téměř jistou záměnu skutečných a falešných anotací. Takové prosté selhání modelu (IV.) nám bylo prozatím skryto, neboť dosud uvedené grafy 8 a 10 zobrazovaly jen průměry těchto dat. Nicméně z tohoto vyplývá, že λ skóre využívající modelu (IV.) je v ohledu klasifikace i skóringu anotací netriviálně využitelné, neboť dokážeme generovat náhodné anotace tak, aby byly blíže modelu než ti *nejhorší* zástupci skutečných.

Ačkoliv je tento test velmi striktní a automaticky původní myšlenku aplikace projevu Menzerath-Altmannova zákona zcela nevylučuje, je minimálně nutné nalézt výhodnější model, který by pozorovaná data reflektoval lépe a redukoval počáteční ignoraci pozorování. Takové hledání pak obnáší provést všechny výše uvedené testy znovu a identifikovat nový nejlepší model.

V této práci proto ponecháme původní zamýšlenou aplikaci Menzerath-Altmannova zákona na skórování *přirozenosti* proteinů jako statisticky neprůkaznou, nicméně zcela nezavrhujeme její potenciál s odlišným modelem.[30] Tyto prvotní negativní výsledky nás však vedou k možnosti na celý problém proteinů a jejich anotací nahlédnou odlišným způsobem a s myšlenkou zcela nové aplikace.

[30] Lze však předeslat, že byl v této fázi pro porovnání testován i v předchozí poznámce pod čarou představený alternativní model využívající logistickou regresi a v testu využívajícím generování náhodných anotací z empirických distribucí selhal obdobným způsobem jako model (IV.). Zároveň byl test upraven tak, aby byly využity průměry skutečných proteinů namísto těch *nejhorších*. Výsledky a především fluktuaci úspěšnosti klasifikací to však příliš neovlivnilo.

Testování metody náhodnými sekvencemi s reálnou anotací

Předchozí test měl za úkol odpovědět, zda metoda λ skóringu dokáže registrovat rozdíly mezi anotacemi skutečných proteinů a anotacemi vzniklými náhodně s reflexí vnějších empirických specifik. Prvotní pozitivní výsledky naivně tvořené z rovnoměrného rozložení nám poskytly iluzi, že metodou λ skórování dokážeme s vysokou jistotou a prakticky ideálně rozlišit náhodné anotace od extrémů těch skutečných. Taková myšlenka a prvotní možná aplikace byly posléze vyvráceny důvtipnějším způsobem vytváření náhodných sekvencí vycházejících ze zjištěných distribucí.

Cílem následujícího testu je ovšem zjistit, zda metoda λ skórování dokáže registrovat rozdíly mezi náhodnými sekvencemi aminokyselin (*falešnými proteiny*) a sekvencemi skutečně kódujícími proteiny, a to na základě anotace vytvořené prediktorem sekundárních struktur. Smyslem je zjistit, zda by dokázala rozlišit mezi sekvencemi kódující proteiny a sekvencemi, které funkční jednotku mohou vytvářet jen prostou shodou okolností. V následujícím testu proto vyzkoušíme, jaké zastoupení *falešných* proteinů bude pro jednotlivá x svým λ skóre spadat do 95% kvantilu skutečných proteinů, což můžeme interpretovat jako určení množství falešných proteinů zaměnitelných s těmi skutečnými, neboť spadají do blízkosti modelu stanoveného empirií. Abychom odhalili případnou tendenci k falešne-negativním výsledkům neboli tendenci ve většině případů odpovídat v otázce kódování proteinu sekvencí negativně, aplikujeme stejný test i na sekvence skutečně kódující proteiny, které necháme znovu a anonymně anotovat shodným prediktorem. Tento test nám tak pro dva datasety – dataset falešných sekvencí a dataset skutečných sekvencí proteinů – zjistí, zda je možné Menzerath-Altmannův zákon aplikovat při identifikaci kódujících a nekódujících sekvencí a s jakou nejistotou. Výsledkem testu budou pro jednotlivé počty sekundárních struktur pravděpodobnosti přijetí testovaného proteinu za reálný na základě překonání prahové hodnoty dané 95% kvantilem skutečných proteinů.

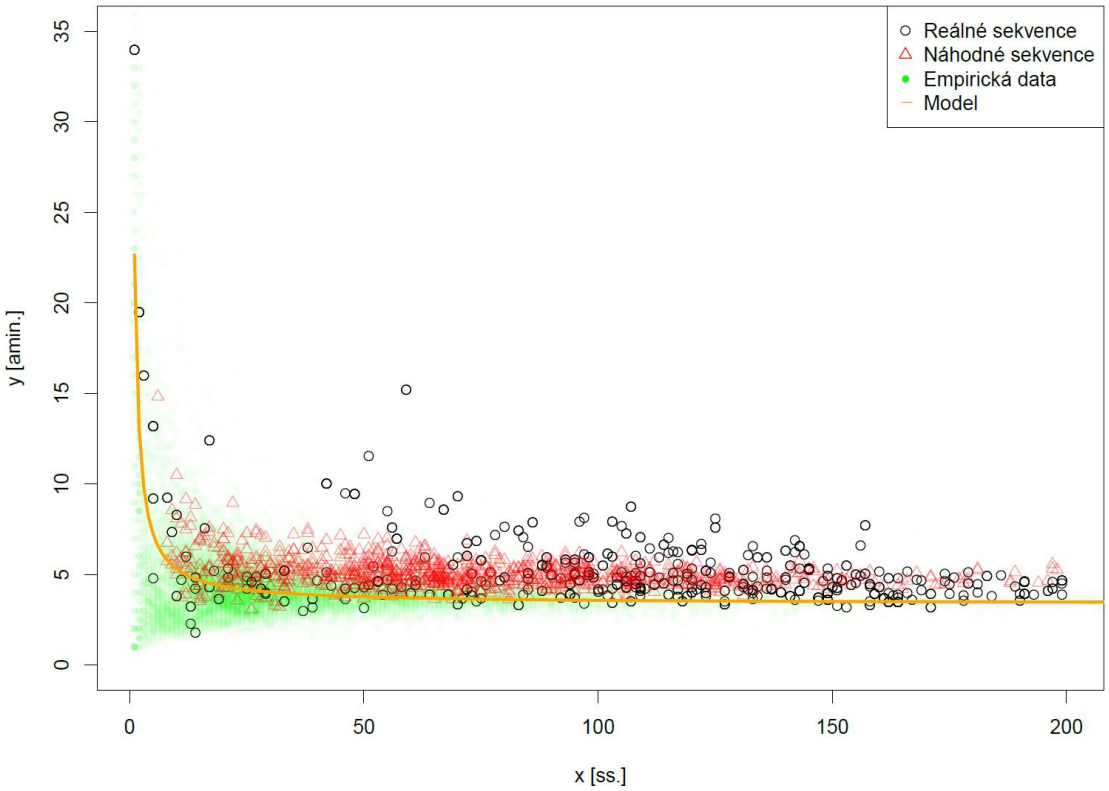
Algoritmus testu sekvencí je následující: Je vygenerován vzorek Z náhodných sekvencí aminokyselin S o minimální délce L a maximální délce K. Výběr aminokyselin probíhá náhodně z empiricky známé distribuce. Vytvořené sekvence by tak měly opět odrážet vnější specifika skutečných sekvencí. Každé sekvenci je pomocí predikčního softwaru přiřazena anotace domnělých sekundárních struktur. Dále, pro dataset skutečných proteinů R bude nalezeno nejlepší proložení MAL modelem (IV.). Pro každý počet sekundárních struktur x vypočítáme λ skóre skutečných proteinů a určíme 95% kvantil vytvářející prahovou hodnotu T_x^λ. Pro jednotlivé počty sekudnárních struktur x získáme seznam testovaných sekvencí z S se shodným počtem sekundárních struktur, u kterých následně vypočítáme λ skóre neboli S_x^λ. Pro tyto následně určíme procentuální zastoupení sekvencí, jejichž λ skóre je nižší než prahová hodnota T_x^λ, neboli hodnota $p_{Accept\ x} = \left|S_x^\lambda < T_x^\lambda\right| / \left|S_x^\lambda\right|$, kterou lze interpretovat jako pravděpodobnost označení testované sekvence P za skutečný protein.

Veškerá data, tj. dataset skutečných proteinů na základě kterého je prokládán model a je stanovován práh S_x^{λ} a dále dataset testovaných sekvencí, jsou bootstrapována. Výsledkem tohoto algoritmu jsou hodnoty $p_{Accept\ x}$ pro jednotlivé počty sekundárních struktur x, u kterých zároveň existují vzorky skutečných (podkladových) proteinů a testovaných proteinů či sekvencí. Pro označení sekvence jako kódující je nutné splnit výše uvedenou podmínku nižšího λ skóre, než je pro dané x určená prahová hodnota. Pro testování skutečných (znovu predikovaných) sekvencí je tento algoritmus dále shodný, ovšem s výjimkou jejich vytváření.

Pro test náhodných sekvencí vygenerujeme vzorek celkem 1 300 sekvencí o minimální délce 100 a maximální délce 3 400 aminokyselin s empiricky daným zastoupením jednotlivých aminokyselin[31]. Získaný vzorek sekvencí anotujeme pomocí prediktoru DeepCNF[32], který je založen na kombinaci hlubokých konvolučních neuronových sítí a podmíněných neuronových polí (Wang *et al*. 2016a a 2016b), tj. nástrojem aktuálně dosahujícím nejlepších výsledků při predikci sekundárních struktur (Yang *et al*. 2018, 486–488). Výsledné anotace jsou uvedeny dle DSSP. Jako dataset skutečných proteinů definujících proložení modelu a prahové hodnoty proto využijeme dataset RSCB. Vzhledem k nedostatečnému zastoupení získaných anotací převyšujících 200 sekundárních struktur je test omezen pouze do této hodnoty.

Před provedením testů nahlédneme na samotná data falešných i nově anotovaných skutečných proteinů, včetně proložení podkladových dat z RSCB modelem (IV.). Všechny tři datasety a jejich proložení modelem vidíme v grafu 18. Z tohoto grafu získáváme rychlý náhled na to, že všechny tři skupiny proteinů, ať už jsou falešné (červeně), skutečné s původní (zeleně) nebo novou anotací (černě), jsou modelu Menzerath-Altmannova zákona velmi blízko. Paradoxně můžeme sledovat, že vzdálenost od modelu je vyšší pro skutečné, znovu anotované sekvence, a to v případě několika prvních desítek sekundárních struktur, což platí i pro původní proteiny z RSCB, které nalezneme pod modelem. Čistě vizuálně se také může jevit, že model určitým způsobem vytváří podporu pro náhodné sekvence, které se objevují téměř jen nad ním. Již z těchto výsledků můžeme soudit, že separabilita obou skupin sekvencí, tj. falešných a skutečných proteinů bude těžká, ne-li nemožná.

[31] Distribuce aminokyselin je převzata z vydání UniProtKB/Swiss-Prot 2013_04, duben 2013; dostupné online https://web.expasy.org/protscale/pscale/A.A.Swiss-Prot.html, cit. 20. 8. 2018.
[32] Metoda je dostupná online v aplikaci http://raptorx.uchicago.edu, cit 20.8. 2018.



***Graf 18**: Vizualizace vztahu počtu sekundárních struktur a jejich průměrné délky pro znovu anotované reálné sekvence kódující proteiny, anotované náhodné sekvence a data z RSCB včetně proložení modelem Menzerath-Altmannova zákona.*

Aplikací výše definovaného testu na 1 300 náhodných sekvencí získáváme pro každé dostupné x pravděpodobnost přijetí náhodné sekvence jako kódující, tj. získáváme v podstatě pravděpodobnosti chybné klasifikace. Výsledky testu nalezneme v grafu 19 a konkrétní hodnoty v tabulce 21.

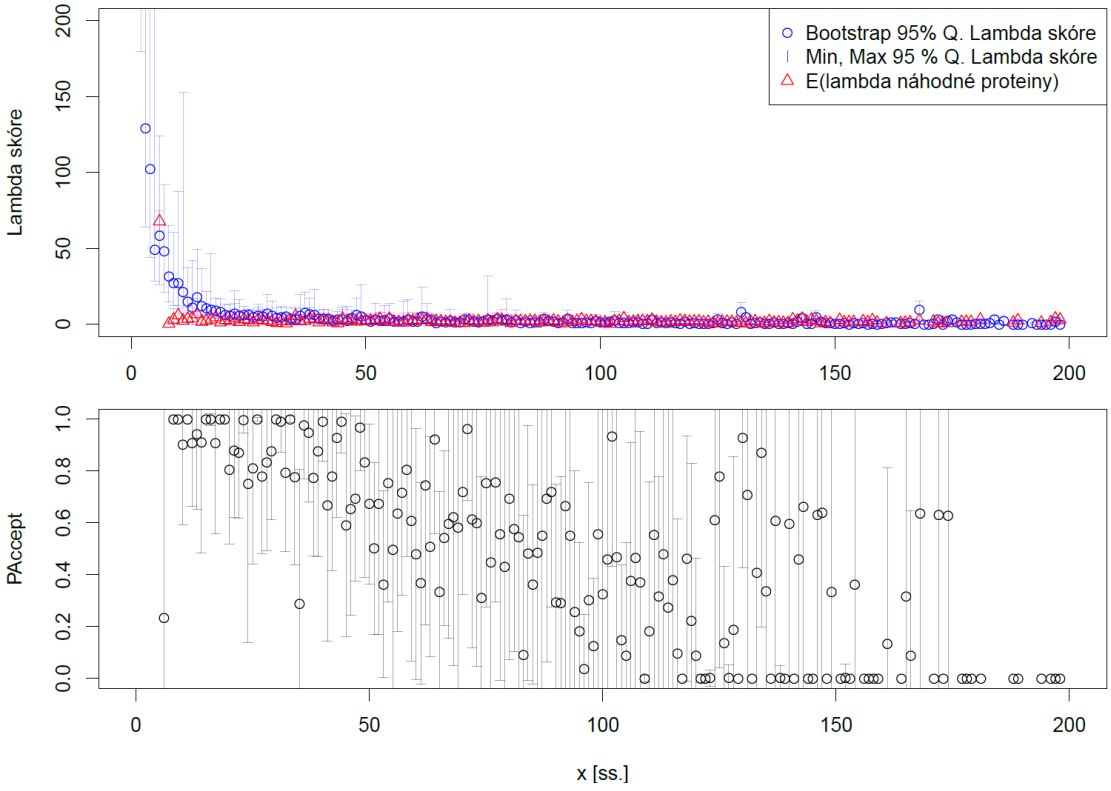
V horní polovině grafu 19 nalezneme znázorněné prahové hodnoty λ skóre pro jednotlivá x (modře) a průměrná λ skóre falešných proteinů (červeně). Zaměříme-li se na umístění průměrů náhodných proteinů, zjistíme, že znovu nacházíme pozorovaný trend, ve kterém jsou sekvence falešných proteinů modelu paradoxně blíže, než jsou mu sekvence podkladové, tj. ty, na jejichž základě je model proložen (srov. grafy 15 a 16 s testy náhodných anotací). Pohled na spodní polovinu grafu zobrazující jednotlivé pravděpodobnosti přijetí p_{Accept} nás o této podobnosti dále přesvědčuje, nicméně v užitém datasetu není dostatek krátkých falešných proteinů (především x = 1 až 10) na to, abychom dokázali říci, zda je jejich vzdálenost od modelu přibližně konstantní, nebo je z počátku odchýlená stejně, jako jsme viděli v předchozích grafech 15 a 16.

Takové pozorování nás proto znovu vede k otázce relevance užitého modelu a nově i k otázce fungování prediktoru DeepCNF. V případě datasetu většího a doplněného o tyto nízké počty x bychom mohli zjistit, zda ve vytvořených anotacích uvažuje Menzerath-Altmannův zákon (ať už z jakéhokoliv důvodu) anebo zda délky sekundárních struktur vybírá pouze z empirické distribuce. Obě protichůdná zjištění

by pak byla mimořádně zajímavá – v případě reflexe Menzerath-Altmannova zákona prediktorem by to znamenalo jeho známost, ať už implicitně či explicitně, a jeho důležitost při anotaci proteinů a v případě jeho nereflektování možnost tento jednoduchý princip do prediktorů doimplementovat.

Pokud se však vrátíme zpět k nově uvažované aplikaci, tj. odhadu, zda neznámá sekvence kóduje protein, pohled na blízkost falešných proteinů a pravděpodobnosti p_{Accept} v grafu 19 nám prozrazuje, že i tato aplikaci nebude statisticky schůdná. Pro prvních 50 sekundárních struktur k chybné klasifikaci náhodné sekvence jako kódující dochází v průměru s pravděpodobností 85,5 % (výpočet z tabulky 21). Dále se tato pravděpodobnost sice snižuje a od 120 sekundárních struktur je blízká nule, ovšem vzhledem k pozorované fluktuaci je tato pravděpodobnost 25%. Ovšem celkem je průměr pravděpodobnosti pro všechny počty *x* přibližně 50 %, což znamená, že metoda není v důsledku pro klasifikaci kódující sekvence lepší než hod mincí.

Na základě sekundárních struktur a užitého modelu Menzerath-Altmannova zákona tak nedokážeme rozeznat falešné proteiny od těch skutečných tím, zda spadají či nespadají do intervalu vzdáleností od modelu stanovených 95% kvantilem skutečných proteinů.



***Graf 19**: Vizualizace lambda skóre reálných proteinů (modře) a proteinů pocházejících z náhodných sekvencí (červeně). Modré chybové úsečky znázorňují.*

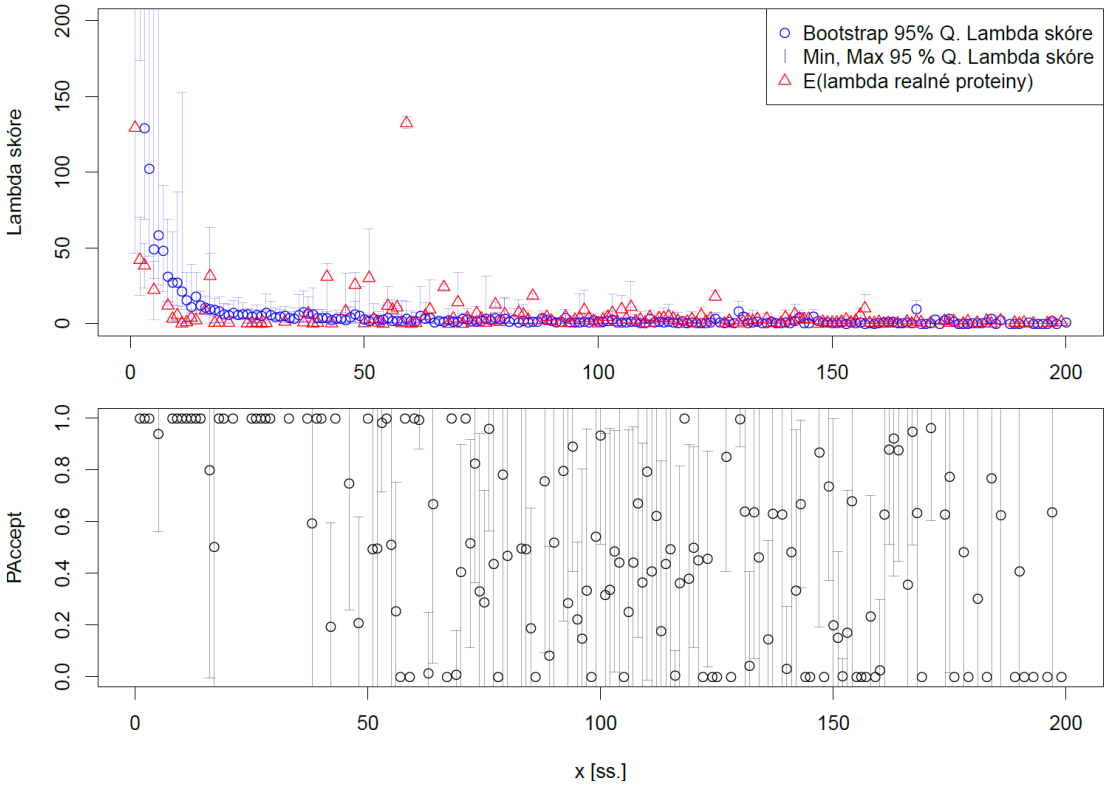
x	*pAcc.*	*x*	*pAcc.*	*x*	*pAcc.*	*x*	*pAcc.*	*x*	*pAcc.*	*x*	*pAcc.*	*x*	*pAcc.*	*x*	*pAcc.*
1	-	**26**	1,00	**51**	0,50	**76**	0,44	**101**	0,46	**126**	0,14	**151**	0,00	**176**	-
2	-	**27**	0,78	**52**	0,68	**77**	0,77	**102**	0,93	**127**	0,00	**152**	0,00	**177**	0,00
3	-	**28**	0,83	**53**	0,37	**78**	0,56	**103**	0,47	**128**	0,18	**153**	0,00	**178**	0,00
4	-	**29**	0,88	**54**	0,76	**79**	0,42	**104**	0,15	**129**	0,00	**154**	0,36	**179**	0,00
5	-	**30**	1,00	**55**	0,50	**80**	0,70	**105**	0,09	**130**	0,93	**155**	-	**180**	-
6	0,23	**31**	0,99	**56**	0,64	**81**	0,58	**106**	0,37	**131**	0,71	**156**	0,00	**181**	0,00
7	-	**32**	0,79	**57**	0,71	**82**	0,56	**107**	0,46	**132**	0,00	**157**	0,00	**182**	-
8	1,00	**33**	1,00	**58**	0,81	**83**	0,09	**108**	0,37	**133**	0,40	**158**	0,00	**183**	-
9	1,00	**34**	0,78	**59**	0,61	**84**	0,49	**109**	0,00	**134**	0,85	**159**	0,00	**184**	-
10	0,90	**35**	0,29	**60**	0,49	**85**	0,36	**110**	0,18	**135**	0,34	**160**	-	**185**	-
11	1,00	**36**	0,97	**61**	0,37	**86**	0,49	**111**	0,55	**136**	0,00	**161**	0,13	**186**	-
12	0,91	**37**	0,94	**62**	0,74	**87**	0,55	**112**	0,31	**137**	0,61	**162**	-	**187**	-
13	0,94	**38**	0,78	**63**	0,51	**88**	0,68	**113**	0,45	**138**	0,00	**163**	-	**188**	0,00
14	0,91	**39**	0,88	**64**	0,93	**89**	0,72	**114**	0,26	**139**	0,00	**164**	0,00	**189**	0,00
15	1,00	**40**	0,99	**65**	0,33	**90**	0,30	**115**	0,38	**140**	0,59	**165**	0,31	**190**	-
16	1,00	**41**	0,67	**66**	0,54	**91**	0,29	**116**	0,10	**141**	0,00	**166**	0,09	**191**	-
17	0,91	**42**	0,78	**67**	0,59	**92**	0,66	**117**	0,00	**142**	0,45	**167**	-	**192**	-
18	1,00	**43**	0,93	**68**	0,62	**93**	0,54	**118**	0,46	**143**	0,67	**168**	0,63	**193**	-
19	1,00	**44**	0,99	**69**	0,58	**94**	0,26	**119**	0,22	**144**	0,00	**169**	-	**194**	0,00
20	0,80	**45**	0,59	**70**	0,72	**95**	0,18	**120**	0,09	**145**	0,00	**170**	-	**195**	-
21	0,88	**46**	0,66	**71**	0,96	**96**	0,03	**121**	0,00	**146**	0,64	**171**	0,00	**196**	0,00
22	0,87	**47**	0,70	**72**	0,60	**97**	0,31	**122**	0,00	**147**	0,61	**172**	0,63	**197**	0,00
23	1,00	**48**	0,97	**73**	0,60	**98**	0,12	**123**	0,00	**148**	0,00	**173**	0,00	**198**	0,00
24	0,75	**49**	0,83	**74**	0,30	**99**	0,56	**124**	0,62	**149**	0,33	**174**	0,63	**199**	-
25	0,81	**50**	0,68	**75**	0,75	**100**	0,32	**125**	0,78	**150**	-	**175**	-	**200**	-
Průměr $p_{Accept} = 0{,}5041790$, 95% CI=$\langle 0{,}5035648\,;\,0{,}5047933\rangle$															

***Tabulka 21**: Výsledky hodnot pAccept pro jednotlivé počty sekundárních struktur x.*

Pro úplnost, i přesto, že je již zřejmé, že uvedená aplikace nemůže na tomto principu fungovat, uveďme i výsledky pro znovu anotované sekvence skutečně kódující proteiny. Jejich výsledky si můžeme prohlédnout v grafu 20 a konkrétní hodnoty v tabulce 22.

Opět se jako první zaměříme na graf 20 a jeho horní polovinu znázorňující prahové hodnoty λ skóre pro přijetí skutečné kódující sekvence jako kódující (modře) a pozorovaná λ skóre testovaných, tentokrát skutečných proteinů (červeně). Zde si můžeme všimnout, že predikce kódujících sekvencí spadají v pořádku pod vytyčené prahové hodnoty a splňují tak podmínku pro označení kódujících sekvencí náležitostí do 95% kvantilu. S narůstajícím počtem sekundárních struktur ovšem tento správný trend klesá, ovšem se značnými fluktuacemi, až k nulové pravděpodobnosti, tj. situaci, kdy prediktor nedokáže s jistotou produkovat anotace, které by do užitého 95% kvantilu empirických dat spadaly. Pohledem do tabulky 22 se můžeme přesvědčit, že pravděpodobnost označení kódující sekvence touto metodou jako kódující je v průměru

pouhých 50 %, což je shodná hodnota jako u určení, že náhodná sekvence není kódující. Tímto je přesnost metody potvrzena i z druhé strany. Posledním negativním výsledkem je i to, že nedokážeme označit jako kódující i sekvence s vysokým počtem sekundárních struktur, jejichž falešné protějšky byly v testu výše správně odmítnuty. Uvažovaná metoda tak z tohoto pohledu nedokáže o kódování či nekódování proteinu danou sekvencí podat žádnou informaci.



***Graf 20**: Vizualizace lambda skóre reálných proteinů (modře) a proteinů pocházejících ze skutečných kódujících sekvencí (červeně). Modré chybové úsečky znázorňují.*

Z výsledků obou testů můžeme konstatovat, že Menzerath-Altmannův zákon modelovaný pomocí (IV.) a za použití výše uvedeného testu není schopen z anotací sekundárních struktur určit, zda sekvence kóduje či nekóduje protein, neboť výsledné predikované anotace jsou z hlediska užitého datasetu vždy natolik blízké skutečným proteinům, že je není možné v 50 % rozlišit. Takový výsledek uvažovanou aplikaci, která měla na základě anotace sekundárních struktur rozhodnout o kódujícím či nekódujícím charakteru sekvence, zcela znevalidňuje. Lze tak říci, že kvantifikace této jediné vlastnosti nevede k požadovanému rozlišení a aplikaci.

Vzhledem k poznatkům, které jsme získali z testů náhodných sekvencí, takový výsledek nemusí být zcela překvapivý, neboť užitý prediktor mohl skutečně vytvářet anotace dle empirických distribucí, které implikovaly blízkost modelu. Tímto se dostáváme do paradoxní situace, ve které by mohl pomoci jednodušší prediktor sekundárních struktur, který nevytváří *a priori* takové anotace, které se musí blížit empirii. Problémem však je, že takový prediktor by byl zřejmě v praxi těžko použitelný. Nicméně je nutné dodat, že predikce kódujících a nekódujících sekvencí na základě kvantifikace vlastností sekundárních struktur je možná, viz např. Washietl *et al*. 2005.

x	*pAcc.*	***x***	*pAcc.*	***x***	*pAcc.*	***x***	*pAcc.*	***x***	*pAcc.*	***x***	*pAcc.*	***x***	*pAcc.*	***x***	*pAcc.*
1	1,00	**26**	1,00	**51**	0,50	**76**	0,96	**101**	0,32	**126**	-	**151**	0,15	**176**	0,00
2	1,00	**27**	1,00	**52**	0,48	**77**	0,44	**102**	0,33	**127**	0,85	**152**	0,00	**177**	-
3	1,00	**28**	1,00	**53**	0,98	**78**	0,00	**103**	0,49	**128**	0,00	**153**	0,17	**178**	0,50
4	-	**29**	1,00	**54**	1,00	**79**	0,79	**104**	0,44	**129**	-	**154**	0,71	**179**	0,00
5	0,94	**30**	-	**55**	0,50	**80**	0,47	**105**	0,00	**130**	1,00	**155**	0,00	**180**	-
6	-	**31**	-	**56**	0,25	**81**	-	**106**	0,25	**131**	0,63	**156**	0,00	**181**	0,30
7	-	**32**	-	**57**	0,00	**82**	-	**107**	0,45	**132**	0,05	**157**	0,00	**182**	-
8	1,00	**33**	1,00	**58**	1,00	**83**	0,50	**108**	0,68	**133**	0,64	**158**	0,24	**183**	0,00
9	1,00	**34**	-	**59**	0,00	**84**	0,50	**109**	0,37	**134**	0,47	**159**	0,00	**184**	0,78
10	1,00	**35**	-	**60**	1,00	**85**	0,19	**110**	0,80	**135**	-	**160**	0,03	**185**	-
11	1,00	**36**	-	**61**	1,00	**86**	0,00	**111**	0,41	**136**	0,14	**161**	0,63	**186**	0,63
12	1,00	**37**	1,00	**62**	-	**87**	-	**112**	0,61	**137**	0,64	**162**	0,88	**187**	-
13	1,00	**38**	0,59	**63**	0,01	**88**	0,75	**113**	0,18	**138**	-	**163**	0,92	**188**	-
14	1,00	**39**	1,00	**64**	0,67	**89**	0,08	**114**	0,43	**139**	0,63	**164**	0,88	**189**	0,00
15	-	**40**	1,00	**65**	-	**90**	0,53	**115**	0,49	**140**	0,04	**165**	-	**190**	0,41
16	0,80	**41**	-	**66**	-	**91**	-	**116**	0,01	**141**	0,48	**166**	0,35	**191**	0,00
17	0,50	**42**	0,20	**67**	0,00	**92**	0,79	**117**	0,36	**142**	0,34	**167**	0,95	**192**	-
18	1,00	**43**	1,00	**68**	1,00	**93**	0,29	**118**	1,00	**143**	0,66	**168**	0,64	**193**	0,00
19	1,00	**44**	-	**69**	0,01	**94**	0,89	**119**	0,38	**144**	0,00	**169**	0,00	**194**	-
20	-	**45**	-	**70**	0,40	**95**	0,22	**120**	0,50	**145**	0,00	**170**	-	**195**	-
21	1,00	**46**	0,75	**71**	1,00	**96**	0,15	**121**	0,45	**146**	-	**171**	0,96	**196**	0,00
22	-	**47**	-	**72**	0,51	**97**	0,34	**122**	0,00	**147**	0,87	**172**	-	**197**	0,63
23	-	**48**	0,20	**73**	0,83	**98**	0,00	**123**	0,45	**148**	0,00	**173**	-	**198**	-
24	-	**49**	-	**74**	0,33	**99**	0,54	**124**	0,00	**149**	0,73	**174**	0,64	**199**	0,00
25	1,00	**50**	1,00	**75**	0,29	**100**	0,93	**125**	0,00	**150**	0,20	**175**	0,77	**200**	-
Průměr $p_{Accept} = 0{,}5023772$, 95% CI=$\langle 0{,}5016832 ; 0{,}5030713 \rangle$															

***Tabulka 22**: Výsledky hodnot pAccept pro jednotlivé počty sekundárních struktur x.*

Shrnutí výsledků testů aplikovatelnosti

Výše jsme definovali a aplikovali dva různé testy využitelnosti zjištěné přítomnosti manifestace Menzerath-Altmannova zákona na proteinech a jejich sekundárních strukturách. Úkolem prvního testu bylo ověřit, zda je na základě dodržení tohoto *zákona* možné odlišit skutečné (správné) anotace od těch chybných či náhodných. Druhý test měl dále zjistit, zda je možné na základě jeho dodržení rozlišit přímo sekvence, jejichž biologickou interpretací vznikne dle empirie funkční protein. Obě tyto aplikace by byly v případě jejich potvrzení přínosné pro praxi molekulární genetiky či bioinformatiky. Jak testy nicméně prokázaly, je využití přítomného Menzerath-Altmannova trendu pro dané účely statisticky nevýznamné. Tyto testy, včetně jejich variant testujících partikulární konfigurace celého problému, však alespoň poukázaly na dvě kritické vlastnosti genetického kódu, kterými jsou komplexita a přítomnost systému ovlivňujícího návrh proteinů.

Z prvního uvedeného testu možnosti rozeznání náhodně vytvořených anotací od těch skutečných jsme zjistili, že délky sekundárních struktur proteinů nemohou být vybírány náhodně z uniformní a ani unimodální distribuce v kombinaci s jinou distribucí. Využitím uniformní distribuce je vyloučena minimální pravděpodobnost vzniku anotací neporušujících empiricky dané limity a náhodný výběr z unimodálních distribucí je vyloučen tím, že s narůstajícím počtem vzorků průměr délek sekundárních struktur konverguje k průměru distribuce, což odporuje pozorovanému průměrnému trendu. Tato zjištění nejsou překvapivá, ovšem nejsou ani triviální, neboť dle zjištěných dat tvrdí, že za výběrem délek sekundárních struktur proteinů není jen náhodný výběr z konkrétní distribuce, ale stojí za ním pravidlo ovlivňující výběr distribuce, její parametry nebo způsob jejich kombinace. Délky sekundárních struktur tak nejsou na základě testů jen pouhým důsledkem náhody, ale důkazem o přítomnosti evolučních filtrů, které design proteinů překonává. Původní myšlenka, že tuto systematičnost bude možné jednoduše modelovat a využít ji tak ke kvantifikaci *přirozenosti* anotace, byla vyvrácena tím, že jen náhodný výběr z vhodné distribuce umožní vytvořit anotace modelu paradoxně bližší než anotace skutečných proteinů. Takové zjištění vede k nedůvěryhodnosti celé metody a uvažovaného způsobu modelování.

Posléze navržené alternativní využití manifestace Menzerath-Altmannova zákona diskutovalo možnost, že sekvence nekódující proteiny by mohly vytvářet při jejich „proteinové" interpretaci natolik empiricky se vymykající sekundární struktury, které by mohly být z hlediska tohoto zákona zachytitelné. Výsledky testů náhodných a skutečných sekvencí však ukázaly, že prediktor pro libovolnou sekvenci v polovině případů vytvoří takovou anotaci sekundárních struktur, která bude spadat přímo mezi 95% skutečných proteinů. Takový výsledek není příliš překvapivý, ale opět není triviální, neboť znamená, že už samotné predikce sekundárních struktur vychází buď

z empirické distribuce nebo prediktor implementuje některé z diskutovaných pravidel hypoteticky vedoucích k projevům Menzerath-Altmannova zákona. Která z těchto zajímavých alternativ nastává, však není vzhledem k užitému datasetu zřejmé. V důsledku ovšem tento výsledek znamená, že i navržená alternativní aplikace nelze s jakýmkoliv ziskem využít.

Prozatím tedy lze tvrdit, že explikované aplikace v oblasti testování sekundárních struktur proteinů s cílem jejich řazení dle *přirozenosti* nebo identifikace kódující či nekódující sekvence za pomoci nejvýhodnějšího nalezeného modelu Menzerath-Altmannova zákona nejsou s tímto modelem a metodikou možné.

Shrnutí MAL na proteinech

V této části práce jsme plynule navázali na předchozí teoretickou kapitolu, která nám poskytla poněkud odlišný náhled na genetický kód a jeho typické vnímání jednotek. Získaná změna perspektivy nám přinesla otázku, zda je na nově vnímaných jednotkách genů, které lze určitým způsobem nahlížet jako možnou analogii k jednotkám užívaným v přirozeném jazyce, přítomen jeden z klasických empirických zákonů kvantitativní lingvistiky upravujících vztah velikosti konstruktů a jejich konstituentů, a to Menzerath-Altmannův zákon. Důvodů, proč by zjištění přítomnosti tohoto vztahu na proteinech bylo zajímavé, bylo v této kapitole představeno několik a týkaly se především praktických i teoretických důsledků pro molekulární biologii a posléze i lingvistiku.

Připomeňme, že podstata genů tkví v lineárním zápisu textu uchovávajícího informace nejen o tom, jak vytvořit funkční trojrozměrné nástroje, ale i o způsobu, jak tyto informace dekódovat a jak s nimi za různých okolností pracovat. Užitý lineární zápis je možno navíc uchovávat ve formě textu s konečnou abecedou a stejně tak i tímto způsobem lze, až anekdoticky řečeno, i v textovém editoru vytvářet nové a funkční biologické nástroje. V tomto ohledu, ať už je či není jazyková metafora DNA platná, přináší i tak potenciál pro vytváření heuristických metod schopných analyzovat tyto sekvence a poskytnout teoretická vodítka vyvstávající z porovnání ekonomizačních principů a dalších faktorů ovlivňujících design obou lineárních systémů pro ukládání informací. Výsledky získané v této kapitole lze však nahlížet z obou disciplín i jejich kombinace.

Manifestace Menzerath-Altmannova zákona je jedním z netriviálních poznatků o přirozeném jazyce a nyní je i takovým poznatkem o designu proteinů, který může nabídnout oběma disciplínám náhled na faktory, které tento projev způsobují. Je zřejmé, že tyto důvody mohou být u obou typů dat odlišné, ovšem i tato odlišnost by měla zajímavou dohru ve vnímání Menzerath-Altmannova zákona jako dalšího obecného principu pozorovatelného napříč různými typy dat, disciplín atd., jak se tomu již jeví např. ve zmíněné hudbě nebo architektuře.

Ověření přítomnosti projevu tohoto zákona na úrovni proteinů a jejich sekundárních struktur nám dále posunuly možnosti porovnání DNA s přirozeným jazykem. Důvod přítomnosti či manifestace tohoto zákona na proteinech je ovšem věcí interpretace molekulárních biologů a až posléze či paralelně věcí jazykovědců, kteří mohou přispět vysvětlením vycházejícím například ze způsobu kódování struktur (Köhler 2005, 84-87). Z biologického hlediska však můžeme v této práci alespoň naivně odhadnout několik faktorů, které by mohly mít na evoluční zisk manifestace Menzerath-Altmannova zákona nějaký vliv.

Prvním z takových faktorů mohou být neustálé kolize proteinů uvnitř buňky, kdy lze zcela hypoteticky předvídat, že by snižování průměrné velikosti konstituentů s jejich rostoucím počtem mohlo vést k jejich vyšší stabilitě. Dalším možným faktorem může být korektní a cílený způsob kolizí (neboli interakcí), které by v případě absence Menzerath-Altmannova zákona mohly být složitější tím, že s rostoucí velikostí proteinu by bylo jednak těžké vytvořit *detailní* konformace umožňující správné interakce menších makromolekul a dále by bylo náročné skrývat vazby, které by neměly být exponovány a které by mohly negativně ovlivňovat ostatní proteiny. Třetím faktorem může být případné usnadnění balení proteinů v případě dodržení Menzerath-Altmannova zákona z opět hypotetického důvodu, kdy jsou velké počty sekundárních struktur dostatečně malé na to, aby se s nimi v kontextu buňky dalo snadno pracovat. Čtvrtým hypotetickým důvodem může být opak balení, tj. rozklad proteinů, který probíhá například v případě jejich poničení. S rostoucí velikostí proteinu, tj. s rostoucím počtem sekundárních struktur se zkracuje, na základě Menzerath-Altmannova zákona, jejich průměrná délka a lze tedy předvídat, že síla vazeb mezi strukturami zůstává tímto vztahem konstantní s libovolnou velikostí proteinů, což by mohlo umožnit jejich snazší rozklad.[33]

K předestřeným biologickým důvodům lze nalézt i paralely v přirozeném jazyce, především v přítomnosti šumu (kolizí), fyzické, fyziologické i kognitivní limitaci mluvčích a recipientů (množství materiálu, energie, správné způsoby interakce atd.), tj. faktorech s potenciálem ovlivňovat strukturu a délky jednotlivých konstituentů textů přirozeného jazyka. V tomto ohledu je například přínosem testování uvažovaných aplikací poznatek, že k tvorbě proteinů a designu jejich sekundárních struktur nemůže docházet náhodným výběrem z uniformních či unimodálních distribucí, které nejsou parametrizovány počtem sekundárních struktur. Z evolučního hlediska proto můžeme říci, že v průměru zbyly právě takové proteiny, které Menzerath-Altmannův zákon dodržely.

Nalezení projevů Menzerath-Altmannova zákona a jeho vysvětlení tak může mít vliv nejen na způsob vnímání specifických procesů uvnitř buňky a jejich ekonomizaci, ale může ovlivňovat i aktuální praktické úlohy, například na metodiku designu nových proteinů (což je kritický bod návrhu nových léků) nebo způsob predikce sekundárních struktur ze sekvencí. Menzerath-Altmannův zákon tak může být přidán jako formální složka do různých metod molekulární biologie a bioinformatiky.

Směřováním k ověření zamýšlených aplikací jsme ovšem dále z empirických dat identifikovali i nejvýhodnější model Menzerath-Altmannova zákona. Pozoruhodným výsledkem je, že tento model pochází z Milička 2014 a nahrazuje tak předchozí tradiční modely uváděné v Altmann 1980. I přes využití modelu výhodnějšího nežli

[33] Množství z uvedených hypotetických důvodů byly stanoveny při diskuzích s Danem Faltýnkem, Jiřím Miličkou, Daliborem Pavlasem a Lukášem Zámečníkem.

těch klasických, následující testy prokázaly, že obě uvažované aplikace nejsou, navzdory specifičnosti nalezeného vztahu a pravidel, využitelné. Lze tedy shrnout, že pouhá kvantifikace průměrné délky sekundárních struktur vůči jejich počtu ve vztahu k empirickému modelu není dostatečnou vlastností rozhodující o kódování či kvalitě anotace proteinů. Důležité ovšem je, že přítomnost Menzerath-Altmannova zákona a blízkost k modelu může být pro tyto účely stále nutnou, avšak nedostačující podmínkou (v heuristickém významu), tj. podmínkou, která vynucuje s určitou pravděpodobností splnění blízkosti k empirickému modelu tak, aby bylo možné anotaci uvážit za reálnou, avšak není dostačující k tomu, aby její povahu potvrdila.

Posledním poznatkem, který nelze opomenout, je nalezený alternativní model (viz poznámka pod čarou 29), který ve výpočtu využívá logistickou funkci aplikovanou při modelování exponenciálního růstu v prostředí s omezenými zdroji a který by tak bylo vhodné dále v kontextu proteinů prozkoumat. V tomto ohledu je rovněž nutné uvážit, zda a jaký vliv měl využitý model na výsledky testů aplikovatelnosti, protože lze spekulovat, že s výhodnějším modelem by bylo možné dosáhnout lepších výsledků, především u proteinů s nízkými počty sekundárních struktur. Ovšem jak uvádí poznámka pod čarou 29, předběžné testy tohoto modelu na zvýšenou úspěšnost neukazují.

Další hypotetický poznatek využitelný v teorii evoluce proteinů vyplývá z paralelních testů přítomnosti Menzerath-Altmannova zákona na evolučně starších konstruktech, než jsou proteiny, tj. na doménách, na kterých byl tento zákon také potvrzen. Takový poznatek je zajímavý, jelikož může poukazovat na stáří evolučního vynucování konformity designu sekundárních struktur tímto zákonem nebo na jeho důležitost. Zároveň jeho přítomnost může poukazovat i na další vlastnost evolučního procesu: Jak již víme, proteiny i domény vykazují Menzerath-Altmannův zákon, přitom proteiny jsou vytvářeny právě i kombinací domén. Dle Milička 2016 ovšem kombinace dvou konstruktů dodržujících Menzerath-Altmannův nemusí nutně znamenat, že tato nově vzniklá kombinace bude tento zákon dále dodržovat. Vzhledem k tomu, že na obou z nich Menzerath-Altmannův zákon v průměru platí, můžeme uvážit možnost, že evolučně byly v průměru vybrány do kombinace proteinů takové domény, které v součtu tento zákon opět dodržovaly. Takovou možnost je nicméně možné dále ověřit a testovat, především ve vztahu proteinů kombinujících více domén a jejich konformity s Menzerath-Altmannovým zákonem. Takový výzkum však ponecháváme na další práci.

Poznatky získané z testování Menzerath-Altmannova zákona na proteinech i doménách tak doplnily nejen existující studie testující jeho přítomnost na různých konstruktech a komponentách v kontextu genů s potvrzením přítomnosti na úrovni domén/proteinů, sekundárních struktur a aminokyselin, ale poukázaly i na další teoretické a praktické implikace, které lze z tohoto pozorování učinit.

Závěr

Analýza sekvencí je místem, ve kterém se střetává zájem mnoha vědních disciplín a zároveň je i místem, které je blízké jazykovědě a její kvantitativní části. V této problematice se tak setkává lingvistika, bioinformatika, sociologie i kryptoanalýza. Na mnoho jevů, ať už se týkají mezilidské komunikace, chůze po sekcích obchodu analyzované pro potřeby marketingu, záblesků kvasarů, zápisů biologických dat nebo řady symbolů na svitku papíru, lze nahlížet jako na sekvence, u kterých hledáme odpovědi na přítomnost vzorů, nahodilosti nebo komplexity. Šíře množství možných aplikací nám tak poskytla motivaci k nalezení a popisu dalších metod analýzy sekvencí, a to skrze optiku nástrojů kvantitativní lingvistiky, od studia opakování *n*-gramů až po testování přítomnosti lingvistických zákonů.

V první kapitole této práce jsme se setkali s metodou, kterou jsme pracovně pojmenovali jako metodu MKM, jejímž cílem je umožnit analyzovat sekvence v co nejobecnějším pojetí, s libovolnou abecedou, délkou, bez jakýchkoliv *apriorních* znalostí o jednotkách nebo jejich jiném členění. Metoda MKM odvozená za tímto účelem využila ty nejjednodušší nástroje kvantitativní lingvistiky, kterými jsou *n*-gramy a poměr mezi velikostí slovníku a počtem realizací tokenů neboli TTR. Způsob užití a kombinace těchto nástrojů zde byl představen, formálně ukotven a ilustrován na řadě partikulárních problémů, ke kterým se dále vrátíme. Pro metodu MKM byly odvozeny její formální vlastnosti vysvětlující její výsledky a mapující její artefakty, které jsou při analýze sekvencí vytvářeny a mohly by bez jejich znalosti zkreslovat úsudek.

Jedním z důležitých bodů odvození této metody bylo i její porovnání s již existujícími a algoritmicky podobnými metodami. Zejména pak u té algoritmicky nejpodobnější (Rao *et al.* 2009, Rao 2010) byly demonstrovány její hlavní nevýhody v souladu s její existující kritikou a byla porovnána s řešeními nabídnutými metodou MKM. Námitky, které vůči porovnávané metodě vznikly lze označit u metody MKM za vyřešené, pokud nastaly, nebo za neexistující. Lze tak říci, že navržená metoda je výhodnější, stabilnější a dosahuje vyšší míry transparentnosti než nejbližší konkurenční metoda.

Aplikovatelnost odvozené metody MKM jsme demonstrovali na řadě příkladů, od přirozených textů, přes genetické sekvence, strukturované texty, triviální repetice a další, včetně různě kvalitních náhodných sekvencí pocházejících z řady zdrojů entropie určených pouze ke generování té nejkvalitnější náhody.

Výsledkem této kapitoly je metoda, která pro danou sekvenci vytvoří číselnou vektorovou reprezentaci o fixní délce (tzv. *embedding*), který dále umožní porovnání této sekvence s ostatními čistě kvantitativní cestou. Tyto reprezentace jsou vytvářeny pouze na základě vyčíslení kombinatorických vlastností symbolů a jejich kombinací

uvnitř sekvence. Výsledky jsou i přes jednoduchost této metody a užitých nástrojů skutečně zajímavé. Zjistili jsme, že i jednoduchým vykreslením *embeddingů* do grafu pomocí křivek definovaných indexem složky a její hodnotou je možné vizuálně shlukovat řadu typů textů, od těch přirozených, přes silně strukturované či triviální. Dále k sobě dokáže tato metoda shlukovat jazyky, sekvence DNA i náhodná či pseudonáhodná data. Pro výsledné *embeddingy* a jejich analýzu jsme experimentálně ověřili aplikovatelnost vícerozměrných analýz a od nich odvíjejících se vizualizací, které nám poskytly nové poznatky o celé metodě, a především nám umožnily lépe interpretovat výsledky analýzy sekvencí.

Využité vizualizace vícerozměrných dat (především metodou vícerozměrného škálování a metodou tSNE), nám poskytly náhled na preciznost, se kterou kvantifikovaná kombinatorika sekvencí dokáže rozlišovat mezi jednotlivými typy textů, a to na základě vizuální separovatelnosti jejich shluků. Zcela překvapivý je tento výsledek i proto, že užité vzorky sekvencí, ze kterých byly jejich *embeddingy* vytvářeny, měly pouze 6 000 bitů (přibližně 750 grafémů). Aplikace a výsledky metody MKM v tomto ohledu předčily očekávání a ukázaly na efektivitu klasických nástrojů kvantitativní lingvistiky.

K aplikaci metody jsme připojili i experimentální analýzu tzv. Vojničova rukopisu, u kterého jsme pozorovali blízkost k přirozeným jazykům a demonstrovali jím především problematiku interpretability těchto metod plynoucích z diskriminační povahy užitých vizualizačních nástrojů.

Jednou z ověřovaných a uvažovaných aplikací metody MKM je i možnost testovat náhodnost sekvencí. V tomto ohledu jsme pro metodu MKM odvodili popisné modely a z nich následně statistický test, který dokáže rozeznat náhodné či pseudonáhodné sekvence od těch zcela nenáhodných. Zjistili jsme však, že pro detekci a odlišení pseudonáhodných sekvencí od těch skutečně náhodných je metodika MKM příliš benevolentní, paradoxně shodně s oficiálními metodami detekce náhodných sekvencí NIST, na jejichž slabost bylo již dříve poukázáno.

Lze však předvídat, že metoda MKM může být upravena, především nahrazením výpočtu TTR pomocí silnějšího nástroje, například pomocí normalizované entropie, která by potenciálně dokázala zvýšit přesnost, se kterou jsou *embeddingy* vytvářeny, především vzhledem k započítání pravděpodobnosti užití jednotlivých slov ze slovníku. Takové změny by však vynutily opětovné vyhodnocení tvorby artefaktů inherentních nové metodě a paradoxně by mohly znamenat i ztrátu popsaných schopností shlukovat typy sekvencí vzhledem k vyšší striktnosti či zaměření na detail. Takové úpravy a výzkum ponecháváme na další práci.

Pozoruhodnou možností, jak metodu dále rozvést a rozšířit její aplikovatelnost, je využití heuristického vyhodnocování shody mezi pozorovanými tokeny, což by mohlo vést především k identifikaci sekvencí se vzory tvořenými specifickými kontexty (tj. využití tzv. *fuzzy n-grams*).[34] Dále je potřeba pro celou metodu prozkoumat vliv délky textu, a především na větším vzorku ověřit vliv velikosti abecedy a její skutečnou invarianci vůči výsledkům. Stejně tak je vhodné analýzy zopakovat se zastoupením většího množství jazyků. Celkově však lze metodu MKM pro analýzu zcela obecných, neznámých sekvencí bez libovolných *apriorních* znalostí hodnotit jako přínosnou.

V následující kapitole jsme se od sekvencí s *a priori* neznámou přítomností jednotek vyšších než samotné symboly abecedy, dostali k otázce, zda je možné u sekvencí, u kterých máme určitou informaci o obsažení vyšších jednotek, určit a nalézt jejich délky. Pro tento účel jsme navrhli specifickou metodu využívající projevy Zipfova zákona na bitové kombinatorice sekvencí, která na základě experimentálních testů odhaluje především předěl mezi kombinacemi symbolů abecedy a kombinacemi tvořící morfémy či slova v případě přirozeného jazyka. Metodu jsme aplikovali na sekvence, u kterých diskuze o umístění jednotek analogických ke slovům přirozeného jazyka existuje a je řešena, a to na kódující sekvence DNA.

V této kapitole jsme z důvodu experimentů s genetickým kódem explikovali základní postupy jeho chápání a jeho analogie k přirozenému jazyku, včetně popisu existující spolupráce jazykovědy s obory studujícími genetické texty jako je molekulární biologie a poukázali na jejich historickou provázanost. Pomocí metody aplikující Zipfův zákon, kterou jsme pracovně označili jako metodu mapování, jsme hypoteticky identifikovali možný předěl v tom, co lze chápat v kontextu genetických textů jako analogii slov, písmen a distinktivních rysů přirozeného jazyka.

Původní chápání této analogie, která přisuzovalo roli písmen nukleotidovým bázím známých pod písmeny ACTG, bylo ve výsledku posunuto na jejich trojice, které kódují komplexnější molekuly aminokyselin. Z nukleotidových bází se tak přirozeně stala analogie distinktivních rysů a z jejich trojic, respektive aminokyselin, analogie písmen. Tato změna v chápání jednotek nás následně posunula k identifikaci slov, tedy takové kombinace písmen, která plní konkrétní funkci a pro kterou kombinací aminokyselin nalezneme analogii v tzv. sekundárních strukturách.

Posun v této jazykové metafoře, jak je označována, ačkoliv je čistě teoretická, vede k myšlenkovému přehodnocení rolí jednotek ve struktuře a hierarchii DNA. Myšlenka přehodnocení jednotek odpovídajících *písmenům, slovům* a *větám* uvnitř sekvencí DNA nás zavedla přímo k následující kapitole a dalšímu lingvistickému zákonu s pozoruhodným potenciálem aplikací.

[34] Využití tzv. *fuzzy n-gramů* bylo doporučeno při diskuzích s Alexandrem Bolshoyem.

Ve třetí kapitole nás nově nalezená analogická strukturace a hierarchizace jednotek uvnitř genetických textů zavedla k otázce, zda i na těchto nových jednotkách – tj. na rovině aminokyselin, sekundárních struktur a proteinů, metaforicky blízkých písmenům, slovům a větám – nalezneme projevy Menzerath-Altmannova zákona. Motivace, které nás k této otázce i testování jeho přítomnosti vedly, byly především aplikační, neboť Menzerath-Altmannův zákon je specifický formální vztah ovlivňující kombinatoriku užitých délek jednotek (tzv. komponent, zde sekundárních struktur) tvořících celek (tzv. konstrukty, zde proteiny). Odhalení přítomnosti tohoto zákona by znamenalo nejen důležitý zisk teoretických poznatků o možných faktorech způsobujících shodné kvantitativní projevy nacházené na přirozeném jazyce a které by tak dokázaly poskytnout oběma disciplínám (tj. jazykovědě i molekulární biologii či bioinformatice) nové náhledy a vysvětlení jeho přítomnosti, ale hlavně by nám tento zákon umožnil testovat dvě kritické otázky týkající se praktických molekulárně biologických aplikací.

První taková aplikace uvažovala využít vztahu Menzerath-Altmannova zákona k hodnocení *přirozenosti* designu, nebo jinak řečeno anotace, sekundárních struktur v případě výběru z mnoha jejich nabízených variant, kdy je vybrána varianta nejlépe odpovídající popsanému vztahu. Druhá aplikace pak na základě obdobného principu uvažovala rozeznání sekvencí DNA kódujících proteiny, a to na základě *přirozeně* se chovající predikované anotace sekundárních struktur profesionálním predikčním nástrojem.

Cílem této kapitoly tedy bylo primárně otestovat přítomnost Menzerath-Altmannova zákona na kódujících sekvencích DNA a v případě jeho nalezení následně ověřit jeho využitelnost pro uvedené aplikace. Sekundární motivací byl zmíněný teoretický přínos, který by nalezení shodné manifestace v datech odlišné disciplíny poskytl oběma oborům. Nalezení takové shody mezi obory by mohlo vést především k diskuzi nad motivací vzniku těchto projevů a jejich alternativních vysvětlení. Motivace k testování Menzerath-Altmannova zákona pak byla zdůrazněna ověřením jeho netriviálnosti a poznatky z jeho možné simulace nám poskytly cenné zázemí při testování obou uvažovaných aplikací.

Projevy Menzerath-Altmannova zákona jsme na studované rovině proteinů i jejich evolučních předchůdců nalezli, včetně jeho zřetelnější manifestace na jejich průměrech. Takové zjištění doplnilo řadu existujících publikací o Menzerath-Altmannově zákonu v kontextu DNA zcela novými poznatky. Tyto pozitivní výsledky nám následně umožnily testovat obě zamýšlené aplikace s potenciálem poskytovat důležité informace o genetických textech. Jedním z největších pozitiv těchto uvažovaných aplikací byla navíc jejich triviálnost a transparentnost, která by usnadňovala intepretaci výsledků a zvážení jejich rizik.

Jak jsme ovšem ukázali, obě aplikace tak, jak zde byly se vší striktností navrženy, nejsou statisticky průkazné a nepřináší dostatečně jisté informace, které by byly pro účinnou aplikaci třeba. Při dosahování bodu, ve kterém jsme mohli obě aplikace zavrhnout, jsme však postupně nalezli a odvodili řadu důležitých teoretických informací, které mohou sloužit oběma oborům tak, jak jsme předvídali.

Jedním z těchto poznatků je především netriviálnost, se kterou se v textech může Menzerath-Altmannův zákon projevit. Na základě simulací i na základě formálních důkazů jsme odvodili, že přítomnost tohoto vztahu na proteinech či jiných textech nemůže být dána jen náhodným výběrem z distribuce, která není přímo parametrizovaná počtem plánovaných výběrů. Takové zjištění vede k tomu, že je design sekundárních struktur proteinů řízen konkrétním systémem nebo k existenci evolučních filtrů, kterými proteiny v průměru nedodržující Menzerath-Altmannův zákon neprojdou a následně zanikají. Překvapivým zjištěním, které jsme dále učinili, byla výhodnost užití nově představeného modelu Menzerath-Altmannova zákona než těch klasických.

Závěrem této kapitoly je, že i na genetických textech a jejich nově analogizovaných jednotkách nacházíme vztah, jehož přítomnost je empiricky ověřována na textech přirozeného jazyka. Zároveň však tento vztah není natolik silný, aby měl přímočaré predikční schopnosti v kontextu analýzy proteinů a uvažovaných aplikací. Přesto, jak jsme již diskutovali výše, je zde stále možnost, že přítomnost tohoto pravidla je nedostatečnou, ale nutnou podmínkou, kterou proteiny svou strukturací musí splňovat. Taková myšlenka, která si nutně žádá další zkoumání, může sloužit jako formální doplnění existujících metod pro predikci sekundárních struktur nebo design proteinů *de novo*.

Cílem této práce bylo nalézt nové aplikace poznatků kvantitativní lingvistiky využitelné v lingvistických, ale i nelingvistických oborech, nebo případně vylepšit stávající. Překvapivým průsečíkem zájmu mnoha oborů, ať už těch čistě přírodovědných nebo těch humanitních, je analýza sekvencí, která jim umožňuje studovat řadu rozmanitých jevů transformovatelných do lineární, textové podoby.

Ve třech kapitolách této práce jsme ukázali, že poznatky z kvantitativní lingvistiky i její klasické nástroje dokáží nabídnout univerzální a zajímavé analytické postupy sekvencí a vzhledem k jejich lingvistickému původu i jisté interpretační zázemí.

Nově formalizovanou a zavedenou metodou MKM jsme dokázali vyhodnotit podobnost sekvencí různých typů, o různých délkách, vytvořených různou abecedou, bez nutných znalostí segmentace obsažených jednotek, a to se zcela nepředpokládanou efektivitou, pouze na základě vnitřní kombinatoriky obsažených symbolů.

Heuristickou aplikací Zipfova zákona na sekvence jsme byli schopni navrhnout metodu s potenciálem identifikovat rovinu odpovídající rovině slov a jejich kombinací. Výstupem aplikace této metody je přehodnocení způsobu vnímání jednotek v genetickém kódu, které nás v tomto kontextu posunulo k testování zcela nových hypotéz.

Na základě nově získaného náhledu na jednotky sekvencí genetického kódu a jejich hierarchie jsme na úrovni proteinů i domén identifikovali projevy Menzerath-Altmannova zákona, a to s potenciálem poskytnout nové teoretické poznatky jak jazykovědě, tak i molekulární biologii. Jeho přítomnost na obou systémech lineárně ukládajících komplexní informace může vést k diskuzi nad evolučními či jinými faktory vedoucími k jeho manifestaci.

Lze proto tvrdit, že cíle práce byly splněny a v některých případech vedly k zisku výsledků překonávajících původní očekávání. Je zřejmé, že řada z uvedených metod či výstupů musí být doplněna o další testy na rozsáhlejších datech tak, aby byla ověřena jejich skutečná univerzálnost a stabilita. Především je v tomto ohledu třeba doplnit znalosti o představené metodě MKM, u které je nutné ověřit skutečnou invarianci vlivu velikosti abecedy sekvencí na výsledky. Dále je pak nutné doplnit statistické testy, včetně rozsáhlejšího množství dat, ověřující průkaznost heuristické metody aplikace Zipfova zákona. Stejně tak je vhodné k metodě aplikace Menzerath-Altmannova zákona ověřit vhodnost užitých modelů, nalézt jejich případné alternativy, přeformulovat a znovu ověřit testy uvažované aplikovatelnosti.

Takové úlohy však ponecháme na další práci, především pak takové, která kooperativně zahrne kapacity v partikulárních oborech plánovaných aplikací.

Summary

The purpose of this work is to introduce the possibilities made available by the use of quantitative linguistics methods in the context of interdisciplinary applications focusing on the analysis of texts and sequences. Sequence analysis is a discipline with a diverse array of applications for a variety of disciplines. 'Sequences' here refers to linear texts using a finite alphabet, without necessary prior knowledge of the presence of any units (except the alphabet itself) and their hierarchy. Linguistics itself is referred to in terms of methods of sequence analysis in cases where there is no prior segmentation method for texts, such as texts of unknown or putative languages. Methods of sequence analysis, often corresponding to linguistic methods, are extensively used in bioinformatics and molecular biology in DNA sequence analysis, to give another example. Biology is not the only discipline that draws on linguistic methods in sequence analysis. Sequence analysis also has its position in sociology: this includes analyzing interpersonal interactions by Markov processes, *n*-grams, editing distances, and other methods (analogous to gene analysis in biology). Such interactions are transformed into a linear symbolic text, allowing for quantitative search for patterns and similarities, and consequently deriving clusters that allow for complex interpretation and inference. For such tasks, not only methods from quantitative linguistics and natural language processing are called upon, but also from communication theory. Surprisingly, it may be important to analyze sequences in the field of computer security, which is nowadays dependent on a particular type of sequences, namely random sequences. Perfectly random sequences are used to generate cryptographic keys and session keys, which are also used to identify users logged in online services ranging from e-mail to online banking. The randomness of these sequences is intended to prevent hackers from guessing this key, and hence against asserting its identity by an attacker. In this context, sequence analysis is directed precisely at detecting the predictability of these sequences and verifying their source. Sequence analysis can be also found, for example, in astronomy, in which signals from celestial bodies are analyzed. This thesis deals with the analysis of sequences from the perspective of quantitative linguistics, and investigates the possibilities of presenting novel information about sequences. In three of its main parts, we will present an analysis of general and *anonymous* sequences, about which we do not know anything in advance except for alphabet, through a chapter examining the possibility of revealing in the sequences of the unit similar to the natural language, to testing the presence and usability of one of the classic laws quantitative linguistics on one of the most important sequences, the DNA sequences. We will describe the individual parts in detail.

The first part of the thesis introduces and characterizes a method called 'the MKM method', which creates a standardized vector representation for any given sequence in order to represent the sequence characteristic for the classification of its source. Creation of such embedding is based only on the simplest tools of quantitative linguistics, namely n-gram for n from 1 to empirically selected constant Z, where TTR values (type to token ratios) are calculated for each n. These resultants form a Z-dimensional vector that can be used to visualize or aggregate sequences. Visualization of such embeddings can be done in a number of ways, from the simplest rendering of n-gram n dependence with their TTR values to multidimensional methods such as MDS (classical multidimensional scaling), SVD (singular value decomposition) or tSNE (t-Distributed Stochastic Neighbor Embedding). The MKM method is illustrated and tested on several sequence types in which it demonstrated the surprising ability to group or cluster the types clearly together. We deduce some formal properties of this method, and derive and verify the possibility to apply it in the determination of a sequence's randomness. The MKM method is also compared with the selected and closest competitive methods, and its development reflects the existing discussion of their main disadvantages and shortcomings. By using the simplest tools, the MKM method has achieved high transparency in the principles of producing results, which, despite the simple principle described, has shown surprisingly successful clustering results for many types of sequences in tests and illustrative applications. As a result, this part of the work is a method capable of providing information about sequences in which we do not know any information other than the alphabet used to help identify the most useful empirically known sequence sources by this method.

The second part of the thesis deals with a specific type of sequences, namely sequences of genetic code storing information about the construction of biological instruments – the proteins – which are of key importance for all living organisms. This part also describes historical links between linguistics and molecular biology, both from the point of view of the methods employed and shared theoretical background. However, the specific application of linguistic methods within coding DNA sequences is complicated by the absence of units with a clear analogy to those of natural language. In this work, we have presented an experimental method with which units analogous to those of the natural language can be found by heuristic application of Zipf's law based on the study of sequences and genetic code. The heuristic application of this law resides, as with the method MKM, in the application of n-grams for n from 1 to a specified constant X, where for each size of n-grams the results of approximation to the hypothetical constant are determined by the multiples of the rank and the frequency of the individual contained types. By drawing these values into the graph using curves, we get the waveforms that tell us, based on empirical comparison, what size of n-grams approximates hypothetical units of words. Empirical knowledge of such waveforms was based on several languages, and on that basis, we have identified a specific number of units that make up the analogy of words in coding sequences

in order to fulfill the natural language analogy of genetic texts. Although this result is metaphorical or very theoretical, it also allows for the at least somewhat justifiable application of linguistic methods a priori requiring knowledge of units close to those of natural language.

The third and final part of this work deals with whether it is possible to observe Menzerath-Altman's Law (MAL) on gene sequences coding proteins with newly designated units of approximately equivalent analogues of those in natural language. This law, present in natural languages, simply says that the larger the construct (a whole), the smaller its elements (so-called 'components'). Example of this relationship may be sentences and words: With the increasing number of words in the sentence, the length of the words should be, according to the MAL, reduced on average. The presence of MAL associated with conservation and economy principles has been found at the level of coding sequences. The search and testing of this relationship was, however, motivated by the purely practical potential of implied applications resulting from the knowledge of this relationship and by the possible gain of theoretical knowledge that would allow for a further and purely quantitative comparison of the behavior of both systems that store complex information in a linear manner. The two applications considered were based on the idea that when the MAL is present at the level of proteins and their secondary structures and amino acids (i.e., at the level of segmentation of the coding sequences) it might mean that: (1) proteins with a damaged secondary structure (e.g., which cause proteopathic disease) may be so varied in the law that they could easily be identified as outlier values or anomalies; (2) that the non-coding sequence of the proteins would, by predicting their hypothetical secondary structures, be so stray to the law that they would be readily recognizable from real proteins. The first application would lead to the possibility of identifying damaged proteins or more natural annotations of secondary protein structures in the case of several available versions from different prediction software. The second of the intended applications would then be able to detect sequences coding the proteins within the DNA. Verifying that this law is present at the level of proteins and domains has led to great theoretical advancement and, above all, to the testing of both applications. However, the following tests have shown that both ideas for the practical applications of the MAL statistically fail and have no proven efficacy. The conclusion nevertheless is that fulfilling this law is necessary, however, insufficient condition for classifying the sequence as coding an actual protein. For the purpose of testing and maximizing the effectiveness of individual applications, however, the new formal model of the MAL, which surpassed statistically its predecessors, was identified as more advantageous. This work has supplemented existing knowledge about coding sequences, proteins, and the existence of MAL in genetics – these existing knowledge have so far avoided the level of protein and secondary structures, and the findings, including hypothetical applications tested, can be perceived as beneficial.

This thesis aimed to apply the tools of quantitative linguistics to analysis of intersectional intersection segments. From the method of analyzing any sequences with the ability to aggregate these sequences based on similarity to the analysis of genetic sequences with the potential to contribute to the knowledge of molecular biology, only the classical tools of quantitative linguistics and its empirical laws were used. The objectives of this work have been fulfilled and we have verified that quantitative linguistics tools can contribute to the study of different types of sequences and that several notable findings can be obtained by analyzing the presence of empirical linguistic laws.

Anotace

Cílem této práce je nalézt aplikace metod kvantitativní lingvistiky s mezioborovým přesahem. Nosným tématem je zvolena analýza sekvencí, která je využívána v řadě oborů od samotné lingvistiky analyzující například neznámé či domnělé jazyky, dále genetiky analyzující sekvence DNA, až po počítačovou bezpečnost, ve které je například třeba analyzovat kvalitu generátorů náhodných sekvencí. Tato práce obsahuje tři kapitoly, ve kterých se postupně věnuje obecné a univerzální analýze sekvencí a následně pokračuje k jejich konkrétním typům i aplikacím. První kapitola této práce představuje metodu odvozenou za účelem kvantitativní charakterizace obecných sekvencí a umožnění jejich shlukování či další analýzy, a to na základě běžných nástrojů kvantitativní lingvistiky. Druhá kapitola se zabývá teoretickou možností odhadu jednotek sekvencí tvořících analogii ke slovům přirozeného jazyka na základě testování projevů Zipfova zákona. Třetí a finální kapitola se zabývá možností využití Menzerath-Altmannova zákona, jakožto specifického pravidla upravujícího vztah konstruktů a konstituentů na kódujících sekvencích DNA s cílem vytvoření heuristické metody testující věrohodnost segmentace sekundárních struktur. Výsledky této práce poukazují na přínos metod a empirických zákonů kvantitativní lingvistiky s potenciálem získávat praktické i teoretické poznatky v kontextu mezioborových aplikací.

Bibliografie

Abramowitz, Milton a Irene A. Stegun. *Handbook Of Mathematical Functions: With Formulas, Graphs, And Mathematical Tables*. Vol. 55. Courier Corporation, 1965.

Alberts, Bruce, Johnson, Alexander, Lewis, Julian, Raff, Martin, Keith, Roberts a Peter Walter: Molecular Biology Of The Cell [5. Vydání]. New York, NY: Garland Science, 2008.

Alkassar, A., Nicolay, T. a M. Rohe. Obtaining True-Random Binary Numbers From A Weak Radioactive Source. *International Conference On Computational Science And Its Applications* (Pp. 634-646). Springer, Berlin, Heidelberg, 2005.

Altmann, G., Wimmer, G. Review Article: On Vocabulary Richness. *Journal Of Quantitative Linguistics*, 6 (2), str. 1-9, 1999.

Altmann, Gabriel. Bibliography: Menzerath's Law. *Glottotheory*, 5 (1), str. 121-123, 2014.

Altmann, Gabriel. Prolegomena To Menzerath's Law. *Glottometrika* 2, str. 1–10, 1980.

Andres, Jan a Martina Benešová. Fractal Analysis Of Poe's Raven, II. *Journal Of Quantitative Linguistics*, 19 (4), str. 301-324. 2012.

Andres, Jan, Benešová, Martina, Chvosteková, Martina a Eva Fišerová. Optimization Of Parameters In The Menzerath–Altmann Law, II. *Acta Universitatis Palackianae Olomucensis*, 53 (2): str. 5-28, 2014.

Andres, Jan, Kubáček, L., Machalová, J. a M. Tučková. Optimization Of Parameters In The Menzerath–Altmann Law. *Acta Universitatis Palackianae Olomucensis*, 51 (1), str. 5-27, 2012.

Anfinsen, Christian B. Principles That Govern The Folding Of Protein Chains. *Science* 181 (4096): str. 223-230, 1973.

Baixeries, J., Hernandez-Fernandez, A., Forns, N. a Ramon Ferrer-I-Cancho. The Parameters Of The Menzerath-Altmann Law In Genomes. *Journal Of Quantitative Linguistics*, 20 (2), str. 94-104, 2013.

Baker, David. Proteins By Design. *The Scientist*, 20 (7), 2006.

Barbieri, Marcello. *The Organic Codes: An Introduction To Semantic Biology*. Cambridge: Cambridge University Press, 2002.

Bellhouse, David R. *Abraham De Moivre: Setting The Stage For Classical Probability And Its Applications*. AK Peters/CRC Press, 2011.

Benešová, Martina, Birjukov, Denis, Kovalová, Jana, Matoušková, Lenka, Motalová, Tereza, Schusterová, Denisa a Petra Vaculíková. *Text Segmentation For Menzerath-Altmann Law Testing*. Olomouc: Palacký University, Faculty Of Arts, 2016.

Berg, Jeremy M., Tymoczko, John L. a Lubert Stryer. *Biochemistry*. New York, NY: W. H. Freeman, 2012.

Berman, Helen M., J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov a P.E. Bourne. The Protein Data Bank. *Nucleic Acids Research*, 28: str. 235-242, 2002.

Bienkowska, Jadwiga, He, Hongxian, Rogers, Robert G. Jr. A Lihua Yu. Bayesian Approach To Protein Fold Recognition: Building Protein Structural Models From Bits And Pieces. *Protein Structure Prediction: Bioinformatic Approach*, 3, str. 54-60, 2002.

Blei, David M., Ng, Andrew Y. a Michael I. Jordan. Latent Dirichlet Allocation. *Journal Of Machine Learning Research*, 3 (4–5), str. 993–1022, 2003.

Bolshoy, Alexander, Volkovich, Zeev (Vladimir), Kirzhner, Valery a Zeev Barzily. *Genome Clustering From Linguistic Models To Classification Of Genetic Texts*. Berlin – Heidelberg: Springer, 2010.

Borel, Émile. La Mécanique Statique Et L'irréversibilité. *Journal De Physique Théorique Et Appliqué*, 3 (1): str. 189-196, 1913.

Boroda, Moisei G. a Gabriel Altmann. Menzerath's Law In Musical Texts. Musikometrica 3, str. 1–13, 1991.

Caetano-Anollés, Gustavo Et Al. The Compressed Vocabulary Of The Proteins Of Archaea. In: Witzany G. (Eds*) Biocommunication Of Archaea*. Springer, Cham, 2017.

Callas, J., Donnerhacke, L., Finney, H., Shaw, D. A Thayer, R. *Openpgp Message Format (No. RFC 4880)*, 2007.

Cambridge, Sidney B., Gnad, F., Nguyen, C., Bermejo, J. L., Krüger, M. a Matthias Mann. Systems-Wide Proteomic Analysis In Mammalian Cells Reveals Conserved, Functional Protein Turnover. *Journal Of Proteome Research*, 10 (12), str. 5275-5284, 2011.

Collado-Vides, Julio. Grammatical Model Of The Regulation Of Gene Expression. *Proceedings Of The National Academy Of Sciences Of The United States Of America*, 89 (20), str. 9405–9409, 1992.

Collado-Vides, Julio. A Linguistic Representation Of The Regulation Of Transcription Initiation. I. An Ordered Array Of Complex Symbols With Distinctive Features. *Biosystems*, 29 (2–3), str. 87–104, 1993.

Conroy, Matthew M. *A Collection Of Dice Problems*. 2018. Dostupné Online: https://www.madandmoonly.com/doctormatt/mathematics/dice1.pdf, cit. 16. 8. 2018.

Cooper, Seth, Khatib, Firas, Treuille, Adrien, Barbero, Janos, Lee, Jeehyung, Beenen, Michael, Leaver-Fay, Andrew, Baker, David, Popović, Zoran a Foldit Players. Predicting Protein Structures With A Multiplayer Online Game. *Nature*, 466, str. 756-760, 2010.

Cornwell, Benjamin. *Social Sequence Analysis: Methods And Applications*, 37. Cambridge University Press, 2015.

Crick, Francis H. C. Towards The Genetic Code. *Discovery*, 22, str. 8–16, 1962.

Crick, Francis H. C. On The Genetic Code: Nobel Lecture, December 11, 1962. In: *Nobel Lectures: Physiology Or Medicine: 1942–1962*. Singapore – New Jersey, NJ – London – Hongkong: World Scientific, str. 811–821, 1964.

Crick, Francis H. C. The Croonian Lecture, 1966: The Genetic Code. *Proceedings Of The Royal Society Of London B: Biological Sciences*, 167 (1009), str. 331–347, 1967.

Crick, Francis H. C. The Origin Of The Genetic Code. *Journal Of Molecular Biology*, 38, str. 367–379, 1968.

Currier, Prescott. New Research On The Voynich, Proceedings Of A Seminar, 1976. Dostupné online: https://www.nsa.gov/news-features/declassified-documents/voynich/assets/files/proceedings-of-a-seminar-30-november-1976.pdf, cit 13. 8. 2018.

Cvrčková, Fatima: *Úvod Do Praktické Bioinformatiky*. Praha: Academia, 2006.

Čech, R., Popescu, I. I., Altmann, G. *Metody Kvantitativní Analýzy (Nejen) Básnických Textů.* Olomouc: Univerzita Palackého V Olomouci, 2014.

Dasgupta, Anirban. *Fundamentals Of Probability: A First Course*. Springer Science & Business Media, 2010.

Dawson, Natalie L. *et al*. CATH: An Expanded Resource To Predict Protein Function Through Structure And Sequence. *Nucleic Acids Research*, 45 (D1), 2017.

De Moivre, Abraham. *The Doctrine Of Chances: A Method Of Calculating The Probabilities Of Events In Play*. Chelsea Publishing Company, 1967.

Detoma, Alaina S., Salamekh, S., Ramamoorthy, A. a M. H. Lim. Misfolded Proteins In Alzheimer's Disease And Type II Diabetes. *Chemical Society Reviews*, 41 (2), str. 608-621, 2012.

Dill, Ken A. a Justin L. Maccallum. The Protein-Folding Problem, 50 Years On. *Science*, 338 (6110), str. 1042-1046, 2012.

d'Imperio, Mary E. *The Voynich Manuscript: An Elegant Enigma*. National Security Agency/Central Security Service Fort George G Meade Md, 1978.

Dürmuth, Markus, Fabian Angelstorf, Claude Castelluccia, Daniele Perito a Abdelberi Chaabane. OMEN: Faster Password Guessing Using An Ordered Markov Enumerator. In *Engineering Secure Software And Systems*, str. 119-132, 2015.

Diaconis, Persi, Sharad, Goel a Susan Holmes. Horseshoes in multidimensional scaling and local kernel methods. *The Annals of Applied Statistics* 2 (3), str. 777-807, 2008.

Efron, Bradley a Robert Tibshirani. *An Introduction To The Bootstrap*. New York: Chapman & Hall, 1994.

Ellis, R. John a Allen P. Minton. Cell Biology: Join The Crowd. *Nature* 425 (6953), str. 27-28, 2003.

Ellis, R. John. Macromolecular Crowding: Obvious But Underappreciated. *Trends In Biochemical Sciences* 26 (10), str. 597-604, 2001.

Eroglu, Sertac. Language-Like Behavior Of Protein Length Distribution In Proteomes. *Complexity* 20 (2), str. 12-21, 2014.

Faltýnek, Dan. *Sémiotické Primitivy V Konstrukci Gramatik: Testování Gramatik Jazyka A DNA*. Olomouc: Univerzita Palackého V Olomouci, 2012.

Ferrer-I-Cancho, Ramon – Elvevåg, Brita: Random Texts Do Not Exhibit The Real Zipf's Law-Like Rank Distribution. *Plos ONE*, 5 (3), E9411, 2010. Cit. 8. 1. 2016.

Ferrer-I-Cancho, Ramon – Mccowan, Brenda. A Law Of Word Meaning In Dolphin Whistle Types. *Entropy*, 11 (4), str. 688–701, 2009.

Ferrer-I-Cancho, Ramon. When Language Breaks Into Pieces: A Conflict Between Communication Through Isolated Signals And Language. *Biosystems*, 84(3), str. 242–253, 2006.

Ferrer-I-Cancho, Ramon a Fermín Moscoso Del Prado Martín. Information Content Versus Word Length In Random Typing. *Journal Of Statistical Mechanics: Theory And Experiment*, 12, L12002, 2011.

Ferrer-I-Cancho, Ramon, Forns, N., Hernández-Fernández, A., Bel-Enguix, G. a Jaume Baixeries. The Challenges Of Statistical Patterns Of Language: The Case Of Menzerath's Law In Genomes." *Complexity*, 18 (3), str. 11-17, 2013.

Fleiss, Joseph L. a Jacob Cohen. The Equivalence Of Weighted Kappa And The Intraclass Correlation Coefficient As Measures Of Reliability. *Educational And Psychological Measurement*, 33 (3), str. 613-619, 1973.

Gastwirth, Joseph L. The Estimation Of The Lorenz Curve And Gini Index. *The Review Of Economics And Statistics*, str. 306-316, 1972.

Govindan, Vidya, Chakraborty, R.S., Santikellur, P. a Aditya Kumar Chaudhary. A Hardware Trojan Attack On FPGA-Based Cryptographic Key Generation: Impact And Detection. In *Journal Of Hardware And Systems Security*, str. 1-15. 2018.

Ha, Le Quan, Elvira Sicilia-Garcia, Ji Ming a Jack Smith. *Extension Of Zipf's Law To Words And Phrases*, 2002.

Haahr, Mads. *True Random Integer Generator, RANDOM.ORG: True Random Number Service. Randomness And Integrity Services Ltd.*, 2018.

Hamano, K. a Yamamoto, H. A Randomness Test Based On T-Complexity. IEICE Transactions On Fundamentals Of Electronics. *Communications And Computer Sciences*, E93-A (7), str. 1346-1354, 2010.

Hamid, R., A. Johnson, S. Batta, A. Bobick, C. Isbell A G. Coleman. Detection And Explanation Of Anomalous Activities: Representing Activities As Bags Of Event N-Grams. In *IEEE Computer Society Conference On Computer Vision And Pattern Recognition* (1), str. 1031-1038, 2005.

Hartl, Daniel L. – Ruvolo, Maryellen. Genetics: *Analysis Of Genes And Genomes* [8. Vydání]. Burlington, MA: Jones And Bartlett Learning, 2013.

Havlin, Shlomo, Buldyrev, Sergey V., Goldberger, Ary L., Mantegna, Rosario N., Peng, Chung-Kang, Simons, Michael a Stanley H. Eugene. Statistical And Linguistic Features Of DNA Sequences. *Fractals*, 3 (2), str. 269–284, 1995.

Haw, J. Y., Assad, S. M., Lance, A. M., Ng, N. H. Y., Sharma V., Lam, P. K. a T. Symul. Maximization Of Extractable Randomness In A Quantum Random-Number Generator. *Physical Review Applied*, 3, 054004, 2015.

Heffernan, Rhys *et al*. Improving Prediction Of Secondary Structure, Local Backbone Angles, And Solvent Accessible Surface Area Of Proteins By Iterative Deep Learning. *Scientific Reports*, 5, 11476, 2015.

Hernández-Fernández, A., Baixeries, J., Forns, N. a Ramon Ferrer-I-Cancho. Size Of The Whole Versus Number Of Parts In Genomes. *Entropy*, 13 (8), str. 1465-1480, 2011.

Jain A. a N. S. Chaudhari. A New Heuristic Based On The Cuckoo Search For Cryptanalysis Of Substitution Ciphers. In *Neural Information Processing*. ICONIP 2015. Lecture Notes In Computer Science, (9490). Springer, Cham, 2015.

Jakobson, Roman. Linguistics In Relation To Other Sciences. In: Roman Jakobson, *Selected Writings: Vol. 2: Word And Language*. The Hague – Paris: Mouton, str. 655–696, 1971.

Ji, Sungchul. Isomorphism Between Cell And Human Languages: Molecular Biological, Bioinformatic And Linguistic Implications. *Biosystems*, 44 (1), str. 17–39, 1997.

Ji, Sungchul. The Linguistics Of DNA: Words, Sentences, Grammar, Phonetics, And Semantics. *Annals Of The New York Academy Of Science*, 870, str. 411–417, 1999.

Ji, Sungchul. The Proteome As An Autonomous Molecular Language: "Proteinese"; A Poster Accepted For Presentation At The DIMACS Workshop On Sequences, Structure And Systems Approaches To Predict Protein Function, Rutgers University, Piscataway, NJ, 2006.

Juola, Patrick, George K. Mikros a Sean Vinsick. Correlations And Potential Cross-Linguistic Indicators Of Writing Style. *Journal Of Quantitative Linguistics*, str. 1-26, 2018.

Karplus, Martin a David L. Weaver. Protein Folding Dynamics: The Diffusion-Collision Model And Experimental Data. *Protein Science* 3 (4), str. 650-668, 1994.

Katz, Gregory. The Hypothesis Of A Genetic Protolanguage: An Epistemological Investigation. *Biosemiotics*, 1 (1), str. 57–73, 2008.

Katz, Jonathan a Yehuda Lindell. *Introduction To Modern Cryptography*. Boca Raton: CRC Press/Taylor & Francis, 2015.

Khatib, Firas, Dimaio, Frank, Foldit Contenders Group, Foldit Void Crushers Group, Cooper, Seth, Kazmierczyk, Maciej, Gilski, Miroslaw, Krzywda, Szymon, Zabranska, Helena, Pichova, Iva, Thompson, James, Popović, Zoran, Jaskolski, Mariusz a Davif Baker. Crystal Structure Of A Monomeric Retroviral Protease Solved By Protein Folding Game Players. *Nature Structural & Molecular Biology* Vol. 18, str. 1175–1177, 2011.

Kleanthous, Colin, Eds. Protein-Protein Recognition, 31. *Frontiers In Molecular Biology*, 2000.

Kneusel, Ronald T. *Random Numbers And Computers*. Cham: Springer International Publishing Imprint Springer, 2018.

Köhler, Reinhard. *Quantitative Syntax Analysis*, 65. Walter De Gruyter, 2012.

Konopka, Andrzej K. Noncoding DNA, Zipf's Law, And Language. *Science*, 268 (5212), str. 789, 1995.

Koopman, Philip J. Jr. *Random Number Generating System And Process Based On Chaos, U.S. Patent No. US5696828A. (N.D.).* Washington, DC: U.S. Patent And Trademark Office, 1995.

Kumar, Uma, Vinod Kumar a J. N. Kapur. Normalized Measures Of Entropy. *International Journal Of General System* 12 (1), str. 55-69, 1986.

Kyte, Jack. *Structure In Protein Chemistry*. Boca Raton, FL: CRC Press, 2006.

Landini, Gabriel. Evidence Of Linguistic Structure In The Voynich Manuscript Using Spectral Analysis. *Cryptologia*, 25 (4), str. 275-295, 2001.

Lasry, George. *A Methodology For The Cryptanalysis Of Classical Ciphers With Search Metaheuristics*. Kassel, Hess: Kassel University Press, 2018.

Law, Averill M. *Simulation Modeling And Analysis*. Dubuque: Mcgraw-Hill Education, 2013.

Leiva, Víctor. *The Birnbaum-Saunders Distribution*. Amsterdam: Elsevier, 2016.

Li, Wentian. Random Texts Exhibit Zipf's-Law-Like Word Frequency Distribution. *IEEE Transactions On Information Theory*, 38 (6), str. 1842–1845, 1992.

Li, Wentian. Menzerath's Law At The Gene-Exon Level In The Human Genome. *Complexity* 17 (4), str. 49-53, 2012.

Lorenz, Wolfgang E., Jan Andres a Georg Franck. *Fractal Aesthetics In Architecture*. Applied Mathematics & Information Sciences 11 (4), str. 971-981, 2017.

Ma, W. J. Campbell, D. Tran a D. Kleeman. Password Entropy And Password Quality. In *Fourth International Conference On Network And System Security*, Melbourne, VIC, str. 583-587, 2010.

Maaten, Laurens Van Der a Geoffrey Hinton. Visualizing Data Using t-SNE. *Journal Of Machine Learning Research* 9 (9), str. 2579-2605, 2008.

Madej T, Lanczycki CJ, Zhang D, Thiessen PA, Geer RC, Marchler-Bauer A, Bryant SH. MMDB and VAST+: Tracking Structural Similarities Between Macromolecular Complexes. *Nucleic Acids Res*, Jan 1. 42 (1), Str. 297-303, 2014.

Mahadevan, Iravatham. *The Indus Script*. New Delhi: Archaelogical Survey Of India.Manning, Ch, 1977.

Manly, John Matthews. Roger Bacon And The Voynich MS. In *Speculum* 6 (3), str. 345-391, 1931.

Mantegna, Rosario N., Buldyrev, Sergey V., Goldberger, Ary L., Havlin, Shlomo, Peng, Chung-Kang, Simons, Michael a Stanley, H. Eugene. Linguistic Features Of Noncoding Sequences. *Physical Review Letters*, 73 (23), str. 3169–3172, 1994.

Mantegna, Rosario N., Buldyrev, Sergey V., Goldberger, Ary L., Havlin, Shlomo, Peng, Chung-Kang, Simons, Michael a Stanley, H. Eugene. Systematic Analysis Of Coding And Noncoding DNA Sequences Using Methods Of Statistical Linguistics. *Physical Review: E, Statistical Physics, Plasmas, Fluids, And Related Interdisciplinary Topics*, 52(3), str. 2939–2950, 1995.

Markoš, Anton a Dan Faltýnek. Language Metaphors Of Life. *Biosemiotics*, 4 (2), str. 171–200, 2011.

Markoš, Anton. *Povstání Živého Tvaru*. Praha: Vesmír, 1997.

Markoš, Anton. *Readers Of The Book Of Life: Contextualizing Developmental Evolutionary Biology*. New York, NY: Oxford University Press, 2002.

Marsaglia, George. *The Marsaglia Random Number CDROM Including The Diehard Battery Of Tests Of Randomness*. Http://Www. Stat. Fsu. Edu/Pub/Diehard/ (2008).

Marsland, Stephen. *Machine Learning: An Algorithmic Perspective*. CRC Press, 2015.

Martin, Juliette, Letellier, G., Marin, A., Taly, J. F., De Brevern, A. G. a J. F. Gibrat. Protein Secondary Structure Assignment Revisited: A Detailed Analysis Of Different Assignment Methods. *BMC Structural Biology*, 5 (1), 17, 2005.

Matlach, Vladimír a Dan Faltýnek. Báze Nejsou Písmena. *Studie z Aplikované Lingvistiky* 1 (7), 2016

Matlach, Vladimír a Diego G. Krivochen. Measuring String „Randomness“: Applications Of Quantitative Methods To Natural And Formal Languages. In *Olomouc Linguistics Colloqium Book Of Abstracts*, 2016.

Matlach, Vladimír, Faltýnek, Dan a L'udmila Lacková. Text Dependency Between Length Of Protein Secondary Structure And The Protein Size. In *Gatherings In Biosemiotics*, str. 62-63, 2017.

Matlach, Vladimír, Faltýnek, Dan a Zámečník Lukáš. Statistical Trends Manifested In Protein Analysis. *QUALICO Information And Language: Coding, Extraction And Applications Book Of Abstracts*, 2016b.

Matlach, Vladimír, Krivochen, Diego G., Milička, Jiří a Lukáš H. Zámečník. Randomness Classification. *QUALICO Information And Language: Coding, Extraction And Applications Book Of Abstracts*, 2018.

Menzerath, Paul. Über Einige Phonetische Probleme. *Actes Du Premier Congres International De Linguistes*. Leiden: Sijthoff, 1928.

Mian, I. S. a C. Rose. Communication Theory And Multicellular Biology. *Integrative Biology*, 3 (4), str. 350-367, 2011.

Milička, Jiří. Is Menzerath's Law A Consequence Of Segment Inventory Inhomogeneity? *Czech And Slovak Linguistic Review*. 2/2015. 2016, str. 62–71, 2016.

Milička, Jiří. Menzerath's Law: The Whole Is Greater Than The Sum Of Its Parts. *Journal Of Quantitative Linguistics* 21 (2), str. 85-99, 2014.

Miller, Doris. *Cryptolog* 2 (8-9), str. 10-11, National Security Agency, Fort George G. Meade, Maryland, 1975.

Mitzenmacher, Michael a Eli Upfal. *Probability And Computing: Randomized Algorithms And Probabilistic Analysis*. Cambridge University Press, 2005.

Monod, Jacques. *Le Hasard Et La Nécessité: Essai Sur La Philosophie Naturelle De La Biologie Moderne*. Paris: Éditions Du Seuil, 1970.

Narayanan, A. a V. Shmatikov. Fast Dictionary Attacks On Passwords Using Time-Space Tradeoff. In *CCS '05: Proceedings Of The 12 ACM Conference On Computer And Communications Security.* ACM, 2005, str. 364-372, 2005.

Niyogi, Partha, Berwick, Robert C. A Note On Zipf's Law, Natural Languages, And Noncoding DNA Regions. *A. I. Memo*, (1530) / *C.B.C.L. Paper*, (118), 1995.

Noll, Landon Curt, Mende, Robert G. a Sanjeev Sisodiya. *Method For Seeding A Pseudo-Random Number Generator With A Cryptographic Hash Of A Digitization Of A Chaotic System*, U.S. Patent No. US5732138A. (N.D.). Washington, DC: U.S. Patent And Trademark Office, 1996.

NSA. *Cryptolog* 2 (8-9), str. 12, National Security Agency, Fort George G. Meade, Maryland, 1975.

Nussinov, Ruth a Gideon Schreiber. *Computational Protein-Protein Interactions*. CRC Press, 2009.

Palazzo, Alexander F. a Ryan Gregory. The Case For Junk DNA. *Plos Genetics*, 8, 10 (5), E1004351, 2014.

Popescu, I. I., Altmann, G., Grzybek, P., Jayaram, B. D., Köhler, R., Krupa, V., Mačutek, J., Pustet, R., Uhlířová, L., Vidya, M. N. *Word Frequency Studies.* Berlin-New York: Mouton De Gruyter, 2009.

Popov, O. S., Segal, Daniel M. a Edward N. Trifonov. Linguistic Complexity Of Protein Sequences As Compared To Texts Of Human Languages. *Biosystems*, 38 (1), str. 65–74, 1996.

Pritchard, J. K., Stephens, M. a P. Donnelly. Inference Of Population Structure Using Multilocus Genotype Data. *Genetics*. 155 (2), str. 945–959, 2000.

Raible, Wolfgang. Linguistics And Genetics: Systematic Parallels. In: Martin Haspelmath – Ekkehard König – Wulf Oesterreicher – Wolfgang Raible (Eds.), *Language Typology And Language Universals: An International Handbook / Sprachtypologie Und Sprachliche Universalien: Ein Internationales Handbuch / La Typologie Des Langues Et Les Universaux Linguistiques: Manuel International*. Berlin – New York, NY: Walter De Gruyter, str. 103–123, 2001.

Rao, Rajesh P. Probabilistic Analysis Of An Ancient Undeciphered Script. *Computer* 43 (4), str. 76-80, 2010.

Rao, Rajesh P., Yadav, N., Vahia, M. N., Joglekar, H., Adhikari, R. a I. Mahadevan. Entropic Evidence For Linguistic Structure In The Indus Script. *Science*, 324 (5931), str. 1165-1165, 2009.

Reddy, Sravana a Kevin Knight. What We Know About The Voynich Manuscript. In *Proceedings Of The 5th ACL-HLT Workshop On Language Technology For Cultural Heritage, Social Sciences, And Humanities*. Association For Computational Linguistics, 2011.

Riedel, Marko. *Probability Of Throwing Exactly V Distinct Sides On N Sided Dice By K Rolls*. Dostupné online: Https://Math.Stackexchange.Com/Q/2857744, Cit. 20.6.2018.

Rugg, Gordon. An Elegant Hoax? A Possible Solution To The Voynich Manuscript. *Cryptologia* 28 (1), str. 31-46, 2004.

Rukhin, A., Soto, J., Nechvatal, J., Smid, M. A Barker, E. *A Statistical Test Suite For Random And Pseudorandom Number Generators For Cryptographic Applications*. Booz-Allen And Hamilton Inc Mclean Va, 2001.

Searls, David B. The Language Of Genes. *Nature*, 420, str. 211–217, 2002.

Shahzad, Khuram, Jay E. Mittenthal a Gustavo Caetano-Anollés. The Organization Of Domains In Proteins Obeys Menzerath-Altmann's Law Of Language. *BMC Systems Biology* 9 (1), 44, 2015.

Schinner, Andreas. The Voynich Manuscript: Evidence Of The Hoax Hypothesis. *Cryptologia*, 31 (2), str. 95-107, 2007.

Schmidt, Michael a Hod Lipson. Distilling Free-Form Natural Laws From Experimental Data. *Science* 324 (5923), str. 81-85, 2009.

Sproat, Richard. A Statistical Comparison Of Written Language And Nonlinguistic Symbol Systems. *Language* 90 (2), str. 457-481, 2014.

Sproat, Richard. Ancient Symbols, Computational Linguistics, And The Reviewing Practices Of The General Science Journals. *Computational Linguistics* 36 (3), str. 585-594, 2010.

Stanford, Augustus L. *Foundations Of Biophysics*. New York, NY: Academic Press, 1975.

Stuttard, Dafydd a Marcus Pinto. *The Web Application Hacker's Handbook: Finding And Exploiting Security Flaws*. Indianapolis: Wiley, 2011.

The ENCODE Project Consortium. An Integrated Encyclopedia Of DNA Elements In The Human Genome. *Nature*, 489, str. 57–74, 2012.

The European Bioinformatics Institute (EMBL-EBI), 2014. Cit. 19. 10. 2014. Dostupné Z WWW: <Http://Www.Ebi.Ac.Uk/Ena/Data/Warehouse/Search?Query=Tax_Eq(9606)&Domain=Coding&Result=Coding_Release&Display=Fasta&Download=Zip>.

Tiltman, John. The Voynich Manuscript: The Most Mysterious Manuscript In The World. *NSA Technical Journal*. XII (3), str. 41-85, 1967.

Torgerson, Warren S. *Theory & Methods Of Scaling*. New York: Wiley, 1958.

Trifonov, Edward N. Codes Of Nucleotide Sequences. *Mathematical Biosciences*, 90 (1–2), str. 507–517, 1988.

Tsonis, Anastasios A., Elsner, James B., Panagiotis, A. Tsonis. Is DNA A Language? *Journal Of Theoretical Biology*, 184 (1), str. 25–29, 1997.

Twyman, Richard M. *Advanced Molecular Biology: A Concise Reference*. New York, NY – Abingdon: Taylor And Francis, 1998.

Twyman, Richard M. *Principles Of Proteomics*. Abingdon – New York, NY: Garland Science / BIOS Scientific Publishers, 2004.

Uniprot Consortium. Uniprot: The Universal Protein Knowledgebase. *Nucleic Acids Research*, 46 (5), 2018.

Valastyan, Julie S. a Susan Lindquist. Mechanisms Of Protein-Folding Diseases at a Glance. *Disease Models & Mechanisms*, 7 (1), str. 9-14, 2014.

Walker, Lary C. a Harry Levine. The Cerebral Proteopathies: Neurodegenerative Disorders Of Protein Conformation And Assembly. *Molecular Neurobiology*, 2, Vol. 21, Iss: 1-2, str. 83-95, 2000.

Wang, S., Li, W., Liu, S. a Xu, J. Raptorx-Property: A Web Server For Protein Structure Property Prediction. *Nucleic Acids Research*, 44 (W1), W430-W435, 2016b.

Wang, Sheng, Peng, Jian, Ma, Jianzhu a Jinbo Xu. Protein Secondary Structure Prediction Using Deep Convolutional Neural Fields. *Scientific Reports* Vol. 6, 2016a.

Washietl, Stefan, Hofacker, Ivo L. a Peter F. Stadler. Fast And Reliable Prediction Of Noncoding Rnas. *Proceedings Of The National Academy Of Sciences* 102 (7), str. 2454-2459, 2005.

Watson, James D. a Andrew Berry. DNA: *The Secret Of Life*. New York, NY: Alfred A. Knopf, 2003.

Weaver, Robert F. *Molecular Biology*. Boston, MA: Mcgraw-Hill, 2002.

Wilkins, John. *Mercury Or The Secret And Swift Messenger*. Londýn, 1641.

Xiong, Jin. *Essential Bioinformatics*. New York: Cambridge University Press, 2006.

Yandell, Mark D. a Majoros, William H. Genomics And Natural Language Processing. *Nature Reviews Genetics*, 3 (8), 601, 2002.

Yang, Yuedong, Gao, Jianzhao, Wang, Jihua, Heffernan, Rhys, Hanson, Jack, Paliwal, Kuldip a Yaoqi Zhou. Sixty-Five Years Of The Long March In Protein Secondary Structure Prediction: The Final Stretch. *Briefings In Bioinformatics*, Volume 19, Issue 3, 1 May 2018, str. 482–494, 2018.

Zhang, J., Rasmussen, E., Croft, W. B. *Visualization For Information Retrieval*. Berlin: Springer-Verlag Berlin And Heidelberg Gmbh & Co. K, 2007.

Zheng H. a H. Wu. Gene-Centric Association Analysis For The Correlation Between The Guanine-Cytosine Content Levels And Temperature Range Conditions Of Prokaryotic Species. *BMC Bioinformatics*, 12 (11), 2010.

Zhou, Huan-Xiang. Influence Of Crowded Cellular Environments On Protein Folding, Binding, And Oligomerization: Biological Consequences And Potentials Of Atomistic Modeling. *FEBS Letters* 587 (8), str. 1053-1061, 2013.

Zipf, George Kingsley. *Human Behavior And The Principle Of Least Effort: An Introduction To Human Ecology*. Cambridge, MA: Addison-Wesley Press, 1949.

Ziv, Jacob a Abraham Lempel. Compression Of Individual Sequences Via Variable-Rate Coding. In *IEEE Transactions On Information Theory*, 24 (5), 530, 1978.

Datová příloha

Programy

DomainsToStructs
Program pro konverzi anotací sekundárních struktur odpovídajících zadanému filtru do tabulky hodnot X, Y pro interpretaci v Menzerath-Altmannově zákonu.

GramPlotter
Zdrojové kódy nástroje vykreslujícího grafy křivek pro zadané sekvence pomocí metody MKM.

SsToAnonymous
Zdrojové kódy programu konvertujícího anotované sekvence proteinů do anonymních sekvencí.

UniprotToMA
Zdrojové kódy programu pro konverzi XML výstupu služby Uniprot na tabulku hodnot X, Y pro interpretaci v Menzerath-Altmannově zákonu.

Sekvence

DNA
mRNA sekvence.

GPG Single File
Jediný soubor 200x separátně zašifrovaný pomocí OpenGPG.

MonkeyTyping
Sekvence vytvořené nahodilými úhozy do klávesnice.

Náhodné
Náhodné a pseudonáhodné sekvence z různých zdrojů.

NatLang Bible 150
Bible ve 150 různých jazycích.

Repetice
Sekvence triviálních repetic.

Voynich
Transkripce Vojničova rukopisu.

Zdrojové kódy C
Zdrojové kódy projektu Open GPG.

Skripty

Extraktory
Skripty určené pro získávání dat z internetových zdrojů či binárních sekvencí.

MAL
Skripty a předzpracovaná data použitá při analýze Menzerath-Altmannova zákona na proteinech a doménách.

MKM
Skripty a předzpracovaná data použitá při analýze sekvencí metodou MKM.

Soubor README
Soubor s tímto popiskem.

Soubor Dizertace.pdf
Soubor s touto prací.

Příloha

Příloha 1: Seznam jazyků 150 Biblí

Aguacateco
Akawaio
Akukem
Alamblak
Ambai
Ampeeli-Wojokeso (Safeyoka)
Ashéninka Ucayali Del Sur (Ashéninka, South Ucayali)
Awiyaana
Bambam
Bariba (Baatonum)
Biangai
Bo-Ung (Mara-Gomu) (Bo-Ung)
Bola
Bolinao
Borei
Brezhoneg (Breton)
Bukiyip
Bwaidoka
Candoshi-Shapra
Caquinte
Central Bontok (Bontok, Central)
Central Tunebo (Tunebo, Central)
Cuiba
Dano
Dobu
English
Epena
Ewage-Notu
Faiwol
Fore
Français (French)
Gela
Gilaki
Gofa
Golin
Gwahatike
Halia
Hdi
Masbatenyo
Maya, Mopán
Mazatec, San Jerónimo Tecóatl
Minica Huitoto (Huitoto, Minica)
Misima-Paneati
Mixtec, Ocotepec
Mixtec, Pinotepa Nacional
Mongi (Kube)
Monkole (Mokole)
Mountain Koiali (Koiali, Mountain)
Naasioi
Nahuatl, Guerrero
Nahuatl, Huasteca Oriental
Nali
Naro
Nəfe (Kwamera)
Nobonob
Nukna
Otomi, Tenango
Palikúr
Paranan
Poqomchi'
Quechua, Cajamarca
Quechua, North Junín
Quechua, Northern Conchucos Ancash
Quichua, Northern Pastaza
Rote Rikou (Rikou)
Saisai (Rifao)
Saliba
Saniyo-Hiyewe
Sea Island Creole English
Secoya
Seimat
Sepik Iwam (Iwam, Sepik)
Sharanahua
Shuar
Siriano
Suau

Helong
Huitoto, Murui
Chatino, Western Highland
Chinantec, Tepetotutla
Chiquitano
Chontal, Tabasco
Chortí (Chorti)
Chuukese
Imbo Ungu
Inabaknon
Inoke-Yate
Inupiatun, Northwest Alaska (Northwest Alaska Eskimo)

Ivbie North-Okpela-Arhe
K'iche'
Kaapor
Kalam
Kamano-Kafe (Kamano)
Kanasi
Kandas
Kaninuwa
Kaqchikel, (Kaqchikel)
Kayabí
Keyagana
Kote
Kuna, Border
Kuot
Kwere
Lacandon
Lote
Maale
Makhuwa-Meetto
Mamara Sénoufo (Sénoufo, Mamara)
Mankanya
Manobo, Matigsalug
Manobo, Western Bukidnon
中国语文 (Chinese)
Sursurunga
Tagabawa
Tatuyo
Tawala
Tenharim
Terêna
Timbe
Tohono O'odham
Tok Pisin (Melanesian Pidgin)
Totonac, Xicotepec De Juárez
Triqui, Copala

Tsikimba
Tubetube (Bwanabwana) (Bwanabwana)
Tz'utujil
Tzotzil De Chenalhó (Tzotzil)
Ubir
Waima
Wala
Wantoat
Washkuk (Kwoma)
West Kewa
Wuvalu-Aua
Yagua
Yopno
Yucuna
Zapotec, Coatecas Altas
Zapotec, Choapan
Zapotec, Ocotlán
Zapotec, Ozolotepec
Zapotec, Sierra De Juárez
Zapotec, Tabaa
Zapotec, Texmelucan
Zapotec, Zoogocho
বাংলা (Bengali)
ಕನ್ನಡ (Kannada)
ไทย (Thai)