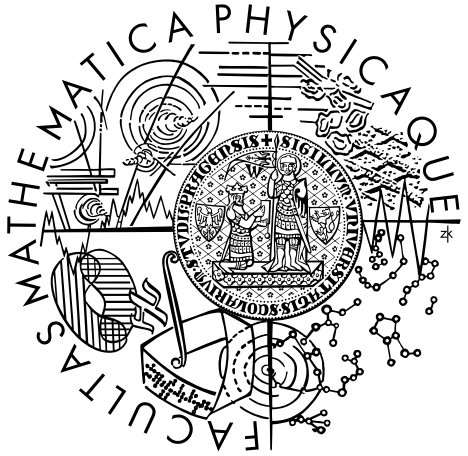
Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

BAKALÁŘSKÁ PRÁCE



Richard Eliáš

Vizualizace sekundární struktury RNA s využitím existujících struktur

Katedra softwarového inženýrství

Vedoucí bakalářské práce: RNDr. David Hoksza, Ph.D.

Studijní program: Informatika

Studijní obor: Obecná informatika

Praha 2016

Prohlašuji, že jsem tuto bakalářskou práci vypracoval(a) samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle §60 odst. 1 autorského zákona.

V ..... dne ..... Podpis autora

Název práce: Vizualizace sekundární struktury RNA s využitím existujících struktur

Autor: Richard Eliáš

Katedra: Katedra softwarového inženýrství

Vedoucí bakalářské práce: RNDr. David Hoksza, Ph.D., Katedra softwarového inženýrství

Abstrakt: Abstrakt .. TODO

Klíčová slova: TODO klíčová slova

Title: RNA secondary structure visualization using existing structures

Author: Richard Eliáš

Department: Department of Software Engineering

Supervisor: RNDr. David Hoksza, Ph.D., Department of Software Engineering

Abstract: RNA secondary structure data, both experimental and predicted, are becoming increasingly available which is reflected in the increased demand for tools enabling their analysis. The common first step in the analysis of RNA molecules is visual inspection of their secondary structure. In order to correctly lay out an RNA structure, the notion of optimal layout is required. However, optimal layout of RNA structure has never been formalized and is largely habitual. To tackle this problem we propose an algorithm capable of visualizing an RNA structure using a related structure with a well-defined layout. The algorithm first converts both structures into a tree representation and then uses tree-edit distance algorithm to find out the minimum number of tree edit operations to convert one structure into the other. We couple each tree edit operation with a layout modification operation which is then used to gradually transform the known layout into the target one. The optimality of tree edit distance algorithm causes that the common motives are retained and the regions which differ in both the structures are taken care of. Visual inspection and planarity evaluation reveals that the algorithm is able to give good layouts even for relatively distant structures while keeping the layout planar. The new method is well suited for situations when one needs to visualize a structure for with a homologous structure with a good visualization is already available. [ii]

Keywords: RNA secondary structure, visualization, homology

Poděkování.

Obsah

Úvod

Následuje několik ukázkových kapitol, které doporučují, jak by se měla bakalářská práce sázet. Primárně popisují použití TEXové šablony, ale obecné rady poslouží dobře i uživatelům jiných systémů.

1. Uvod do molekularnej genetiky a bioinformatiky

Na zaciatku prace strucne zoznamime citatela s vybranymi pojmami z bioinformatiky i genetiky.

1.1 Co je RNA

Nositelkami genetickej informacie bunky su molekuly nukleovych kyselin tvorene retazcami nukleotidov, ktore su zakladnymi stavebnymi jednotkami nukleovych kyselin. Vyskytuje sa niekolko variant nukleotidov (baz). U RNA su to adein (A), guanin (G), cytozin (C), uracyl (U), pri DNA sa namiesto uracylu vyskytuje tymin (T). Medzi jednotlivymi bazami existuju vazby na principe komplementarity. Vodikove vazby existuju medzi bazami A-U a C-G u RNA a podobne A-T a C-G u DNA. Strukturu nukleovych kyselin mozeme chapat podla stupna zjednodusenia

- Primarna struktura - je urcena poradim jednotlivych nukleotidov do polynukleotidoveho retazca
- Sekundarna struktura - je dana 2D priestorovym usporiadanim molekuly
- Terciarna struktura - 3D priestorove usporiadanie molekuly

DNA je dvojvlaknova molekula u ktorej spojenie medzi vlaknami sa realizuje na principe komplementarity. Naopak, RNA je iba jednovlaknova molekula. V snahe minimalizovat energiu molekuly, RNA komplementaritou vytvara v molekule bazove pary. Tie tvoria sekundarnu strukturu.

V praci budeme primarnou a sekundarnou strukturou mysliet prave primarnu a sekundarnu strukturu RNA, ak nebude povedane inak.

Az donedavna sa myslelo, ze funkcia RNA je iba pri tvorbe bielkovin (mRNA), alebo ako transporter aminokyselin (tRNA). Avsak existuje mnoho dalsich, od relativne malych molekul tvorenych desiatkami baz, ktore pomahaju pri expresii genov (miRNA, siRNA, tmRNA a dalsie), az po velke, tvorene tisickami nukleotidov (rRNA).

Definice 1. *Nech Σ je abeceda $\{A, C, G, U\}$. Potom slovo $W \in \Sigma^n$ nad touto abecedou je sekvencia nukleotidov (baz) RNA.*

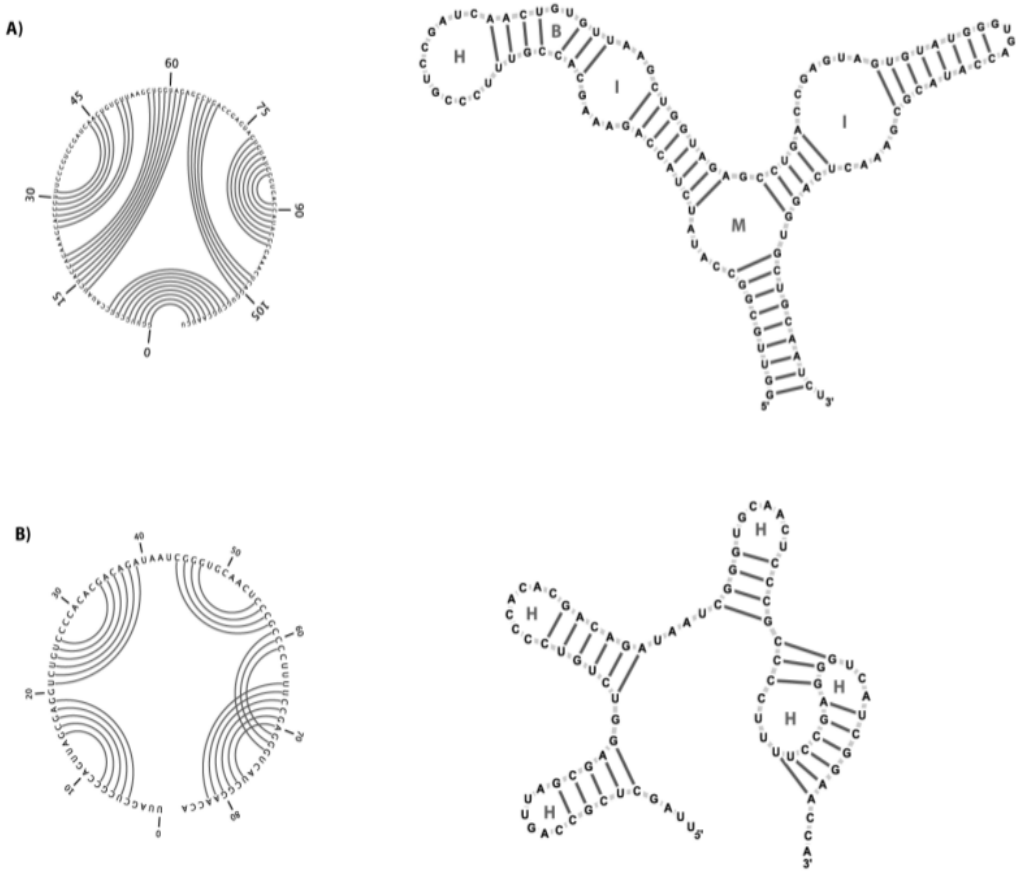
1.2 Sekundarna struktura rRNA + konzervovanost

Ako hlavny objekt zaujmu sme si spomedzi RNA vybrali ribozomalnu. Je to hlavne kvoli jej velkosti a konzervovanosti. Konzervovanostou myslime to, ze napriec celym spektrom organizmov sa sekundarna struktura rRNA velmi nemeni.

Definice 2. *Nech W je sekvencia podla definicie 1 dlzky n. Sekundarnou strukturou oznacime mnozinu $\mathbb{S}$ parov (i, j) takych, ze pre dva pary (i, j) a $(k, l) \in \mathbb{S}$ (BUNO $i \leq k$) plati jedno z nasledujucich:*

- $i = k \iff j = l$
- $i < j < k < l$, *cize par* (i, j) *predchadza par* (k, l)
- $i < k < l < j$, *cize par* (i, j) *obsahuje par* (k, l)

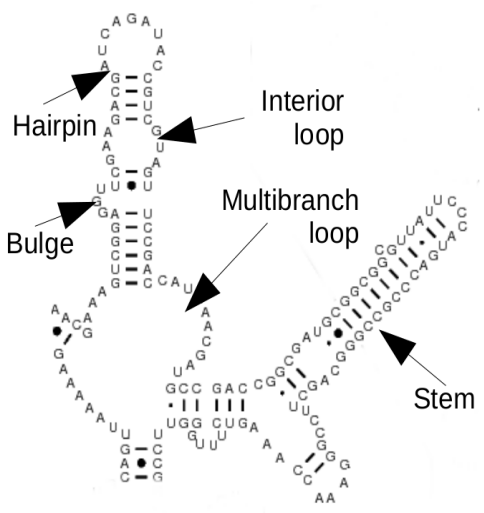
Prva podmienka zabezpecuje, ze nukleotid je najviac v jednom bazickom pare, druha a tretia hovoria o usporiadani parov, bud su na sebe nezavisle alebo na seba nadvazuju. Posledna podmienka zakazuje existenciu pseudouzlov (pseudoknots).



Obr. 1.1: Circular Feynman - kruhova reprezentacia sekundarnej struktury

1.2.1 Motivy

Na obrazkoch mozeme pozorovat motivy RNA molekul, stem/loop, s dalsim moznym delenim loopov na bulge, interior loop a multibranch loop. V dalsom rozpravani nam bude stacit rozdelenie na stem a loop.



Obr. 1.2: Strukturalne motivy v RNA

1.3 Reprezentacia sekundarnej struktury

Definicia 2 nam ponuka reprezentovat sekundarnu strukturu ako usporiadany les.

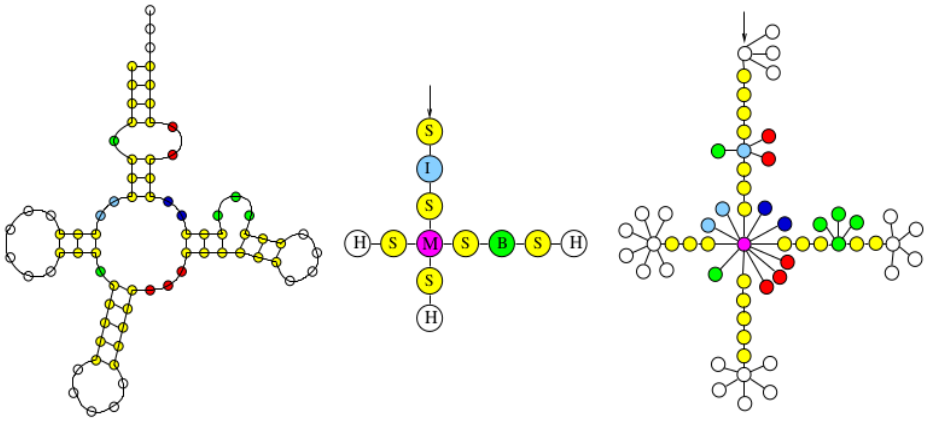


Figure 3: A secondary structure and its tree representations.

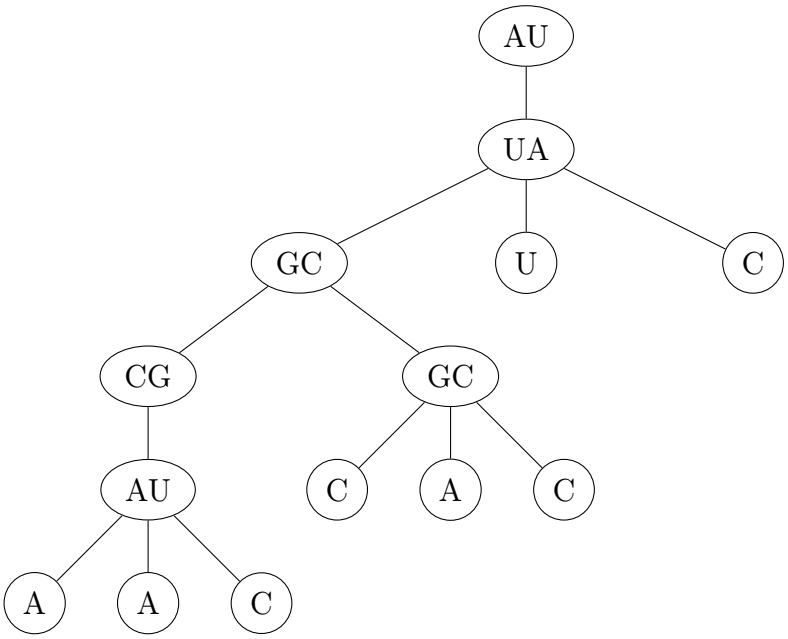
Obr. 1.3: Varianty reprezentacie vrcholov

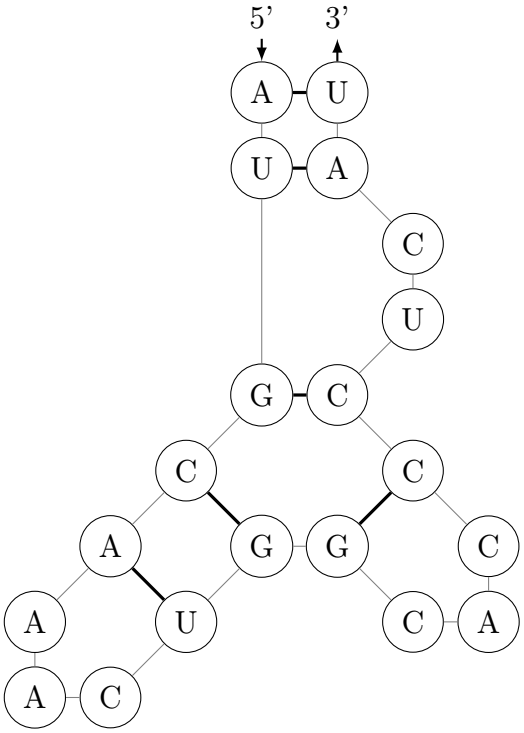
Definice 3. *Usporiadany zakoreneny strom je orientovany graf, v ktorom plati, ze hrany su orientovane vzdy v smere z predka na potomka. Okrem korena ma kazdy vrchol svojho predka. Existuje tu dalej usporiadanie medzi potomkami. Usporiadany les je usporiadana mnozina stromov.*

Kazdy vrchol moze reprezentovat napriklad motiv v strukture RNA, alebo nukleotid, bazovy par, ...

2. Uvod a motivace

V predchadzajucej kapitole sme definovali vybrane casti genetiky, bioinformatiky, teorie grafov.
Cielom prace je vytvorit program na vizualizaciu sekundarnej struktury RNA pomocou uz existujucej vizualizacie inej molekuly.


AU
UA
GC
U
C
CG
GC
AU
C
A
C
A
A
C

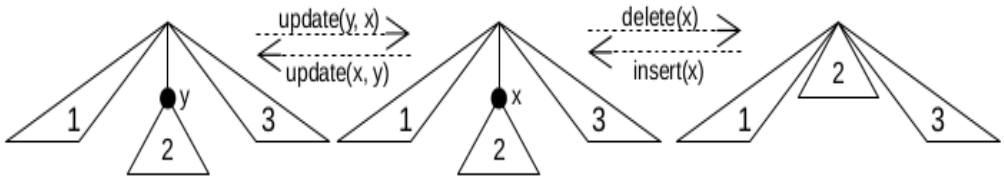

5’
3’
A
U
U
A
C
U
G
C
C
C
A
G
G
C
A
U
C
A
A
C

3. Tree-edit-distance algoritmus

Jadro aplikacie lezi v pouziti tree-edit-distance (TED) algoritmu, vdaka ktoremu dostaneme mapovanie medzi 2 RNA stromami. Mapovanie nam ukaze spolocne casti oboch RNA stromov. TED algoritmus je obdoba Levenstheinoveho string-edit-distance algoritmu. Problem u retazcov je specialnym pripadom TED-u, kedy stromy zdegenerovali na cesty (spojovy zoznam).

3.1 Hlavna myslienka TED-u

Zaklad TED algoritmu je v rekurzivnom vzorci 3.2 z Demaine a kol. (2009) a Pawlik a Augsten (2011). Vzdialenost medzi lesmi F a G, $\delta(F, G)$ je definovana ako minimalny pocet editacnych operacii, ktore z F urobia G. Pouzivame standardne editacne operacie - delete, insert, update.



Obr. 3.1: Ukazky TED operacii

Delete, zmazanie vrcholu, znamena pripojit k predkovi vsetkych jeho potomkov so zachovanim poradia medzi nimi. Insert, vlozenie vrcholu, je opacna operacia k delete, co znamena, ze vkladame vrchol medzi rodica nejakych jeho, po sebe nasledujucich potomkov. Update iba zmeni hodnotu vo vrchole stromu.

3.2 Znacenie

V tejto kapitole sa budeme riadit znacenim Pawlik a Augsten (2011). Teda, pouzivame definiciu stromu a lesa z 3. Ak F je les (strom), N_F oznacuje mnozinu jeho vrcholov a E_F mnozinu jeho hran. Plati dalej ze $E_F \subseteq N_F \times N_F$. $\emptyset$ oznacuje prazdny strom, resp. prazdny les. Podles lesa F je graf $\tilde{F}$ s vrcholmi $N_{\tilde{F}} \subseteq N_F$ a hranami $E_{\tilde{F}} \subseteq E_F \cap N_{\tilde{F}} \times N_{\tilde{F}}$. Obdobne to plati aj pre podstrom stromu T. F_v oznacuje podstrom F zakoreneny vo v, t.j. v strome ostavaju iba potomkovia v. $F - v$ budeme znacit les, ktory dostaneme zmazanim vrcholu v z F, spolu so vsetkymi hranami zasahujucimi do v. Podobne $F - F_v$ budeme znacit les, ktory dostaneme zmazanim podstromu F_v z F.

Definice 4 (Editacna vzdialenost)**.** *Nech F a G su dva lesy. Editacna vzdialenost, tree-edit-distance - $\delta(F, G)$, medzi F a G je rovna minimalnej cene, za ktoru les F transformujeme na G.*

Vo vzorci 3.2 pocitame editacnu vzdialenost $\delta(F, G)$, c_{del}, c_{ins}a c_{upd}su ceny zmazania, vlozenia a editacie vrcholu v strome a r_F a r_G su korene, bud obidva

najpravejsie alebo najlavejsie (tzn. vyberieme najpravejsi/najlavejsi strom lesa a jeho koren).

$$\begin{aligned}\delta(\emptyset,\emptyset) &= 0\\ \delta(F,\emptyset) &= \delta(F-r_F,\emptyset)+c_{del}(r_F)\\ \delta(\emptyset,G) &= \delta(\emptyset,G-r_G)+c_{ins}(r_G)\end{aligned} \tag{3.1a}$$

$$\delta(F,G) = \begin{cases}\delta(F-r_F,G)+c_{del}(r_F)\\ \delta(F,G-r_G)+c_{ins}(r_G)\\ \delta(F-F_{r_F},G-G_{r_G})+\\ \quad \delta(F_{r_F}-r_F,G_{r_G}-r_G)+c_{upd}(r_F,r_G)\end{cases} \tag{3.1b}$$

Obr. 3.2: Rekurzivny vzorec pre vypocet tree-edit-distance

3.3 Algoritmy dynamickeho programovania

Tai (1979) predstavil algoritmus s priestorovou a casovou zlozitostou $\mathcal{O}(m^3 \cdot n^3)$, Zhang a Shasha (1989) algoritmus nasledne vylepsili pozorovanim toho, ze nepotrebujeme vzdialenosti medzi vsetkymi parmi podlesov. Algoritmus mal casovu zlozitost $\mathcal{O}(m^2 \cdot n^2)$ a priestorovu $\mathcal{O}(m \cdot n)$. Klein (1998) dosiahol casovu zlozitost $\mathcal{O}(m^2 \cdot n \cdot \log n)$, avsak jeho riesenie potrebovalo rovnako vela pamete. Dulucq a Touzet (2003) ukazali, ze minimalny cas na beh algoritmu je $\mathcal{O}(m \cdot n \cdot \log m \cdot \log n)$. Demaine a kol. (2009) predviedli worst-case optimalny algoritmus pre tree-edit-distance. Jeho casova a priestorova zlozitost je $\mathcal{O}(m^2 \cdot n \cdot (1 + \log \frac{n}{m}))$ a $\mathcal{O}(m \cdot n)$. Pawlik a Augsten (2011) ukazali spojitost medzi efektivnostou predchadzajucich algoritmov a tvarom stromov. Zovseobecnili predchadzajuce pristupy a vytvorili algoritmus beziaci vo worst-case case $\mathcal{O}(m^3)$ a priestore $\mathcal{O}(m \cdot n)$. Ich algoritmus je teda efektivny pre vsetky tvary stromov a nikdy nespadne do worst-case, ak existuje lepsi smer vypoctu.

3.3.1 RTED: Robust Tree Edit Distance algoritmus

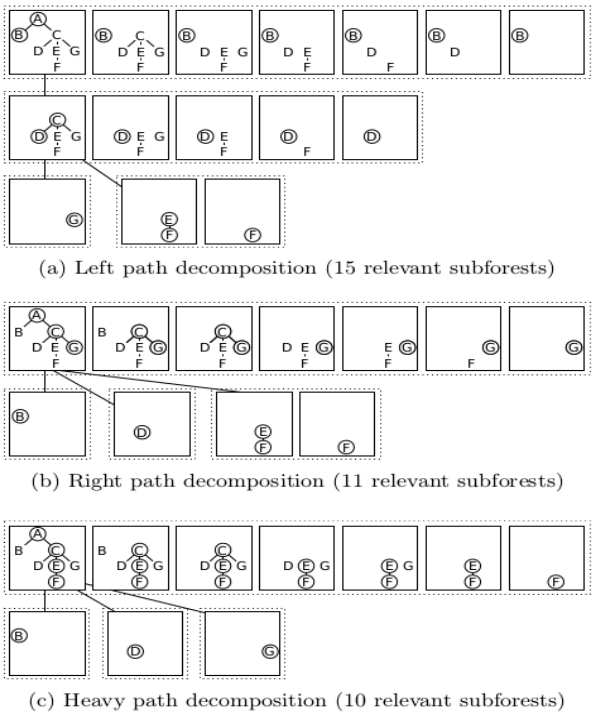
Dalej sa v nasej praci budeme venovat vyhradne algoritmu RTED od tvorcov Pawlik a Augsten (2011). Ich algoritmus rozdelime na 2 casti, rovnako pomenovany RTED a GTED.

RTED (Robust Tree Edit Distance) algoritmus bude pre nas algoritmus na vypocet optimalnej dekompozicnej strategie (viz definicia 5) a GTED (General Tree Edit Distance) algoritmus samotny vypocet rekurzie 3.2 s aplikovanim danej strategie.

Definice 5 (Dekompozicna strategia)**.** *Nech F a G su lesy. Dekompozicina strategia v rekurzii 3.2 priradi kazdej dvojici podstromov F_v a G_w lesov F a G jednu cestu γ_T z korena do listu, kde $T \in \{F, G\}$. LRH dekompozicna strategia vybera vzdy najlavejsi/najpravejsi/najtazsi (left/right/heavy) vrchol na ceste z korena do listu. Najtazsi vrchol je taky v ktoreho podstrome je najviac vrcholov.*

Definice 6. *Celkova dekompozicia lesa (full decomposition) F, $\mathcal{A}(F)$ je mnozina vsetkych podlesov F, ktore dostaneme rekurzivnym odstranenim najlavejsieho alebo najpravejsieho korenoveho vrcholu - $r_R(F)$ a $r_L(F)$ - z F a nasledne aj vsetkych jeho podlesov.*

$$\mathcal{A}(\emptyset) = \emptyset$$
$$\mathcal{A}(F) = F \cup \mathcal{A}(F - r_L(F)) \cup \mathcal{A}(F - r_R(F))$$



(a) Left path decomposition (15 relevant subforests)

(b) Right path decomposition (11 relevant subforests)

(c) Heavy path decomposition (10 relevant subforests)

Obr. 3.3: Celkova dekompozicia pomocou LRH strategii

Definice 7. *Relevant subtrees stromu F pre root-leaf cestu γ su definovane ako $F - \gamma$. Relevant subforests stromu F pre nejaku root-leaf cestu γ su definovane rekurzivne ako*

$$\mathcal{F}(\emptyset, \gamma) = \emptyset$$
$$\mathcal{F}(F, \gamma) = \{F\} \cup \begin{cases} \mathcal{F}(F - r_R(F), \gamma), & \text{ak } r_L(F) \in \gamma \\ \mathcal{F}(F - r_L(F), \gamma), & \text{v ostatnych pripadoch} \end{cases}$$

Kazdy krok rekurzie 3.2 je vypocitana v konstantnom case z inych podproblemov. Doba behu zavisi na pocte podproblemov ktore potrebujeme vyratat, takze dekompozicna strategia hraje velku ulohu v dobe behu algoritmu.

Ulohou RTEDu je najst strategiu s najnizsim poctom problemov, ktore potrebujeme vyratat, zatial co ulohou GTEDu je podla danej strategie prejst celou rekurziou 3.2 a vratit vzdialenost medzi stromami F a G.

GTED: General Tree Edit Distance algoritmus

Pozn. D^T budeme oznacovat transponovanu maticu, teda $D[F_v][G_w] = D^T[G_w][F_v]$, podobne S^T.

- Vstup: Stromy F a G, funkcia $S(F_v, G_w) = \gamma$ na vypocet dekompozicnej strategie.
- Vystup: matica D vzdialenosti medzi vsetkymi podstromami F a G.

Algorithm 1 General Tree Edit Distance for LRH strategies

```
1:  procedure GTED(F, G, D)
2:                                  ▷ Usporiadanie vrcholov je zlava doprava v postorder
3:                                  ▷ Plati, id(root(F)) = |F| − id(mostleft(F))
4:      γ ← S(F, G)
5:      if γ ∈ γ*(F) then
6:          for all F' ∈ F − γ do
7:              D ← D ∪ GTED(F', G, D)
8:          if γ = γ^L(F) then
9:              D ← D ∪ Δ^L(F, G, γ, D)
10:         else if γ = γ^R(F) then
11:             D ← D ∪ Δ^R(F, G, γ, D)
12:         else
13:             D ← D ∪ Δ^H(F, G, γ, D)
14:     else
15:         D ← D ∪ (GTED(G, F, S^T, D^T))^T
16:     return D
```

Definice 8 (Single path function)**.** *Oznacme D maticu vzdialenosti medzi stromami F_v a G_w, teda $D[F_v][G_w] = \delta(F_v, G_w)$. Potom funkcia $\Delta(F, G, \gamma_F, D)$ ktora pocita vzdialenosti medzi podstromami stromov F a G pre root-leaf cestu γ_F nazveme single-path funkciou.*

Nasledovat budu 2 algoritmy na vypocet single-path-function, algoritmus 2 je z dielne Zhang a Shasha (1989) a algoritmus 3 je od Demaine a kol. (2009).

Algorithm 2 Zhang & Shasha: Single path function

```
1: procedure Δ^L(F, G, γ, D)
2:     L_F ← id(mostleft(F))
3:     L_G ← id(mostleft(G))
4:     R_F ← id(root(F))
5:     R_G ← id(root(G))
6:     forestdistance ← pole indexovane podlesmi stromov F a G
7:     forestdistance[∅, ∅] := 0
8:     for i := L_F to R_F do
9:         forestdistance[F[L_F ... i], ∅] :=
10:            forestdistance[L_F ... i - 1] + C_del(F[i])
11:    for j := L_G to R_G do
12:        forestdistance[∅, G[L_G ... j]] :=
13:            forestdistance[∅, G[L_G ... j - 1]] + C_ins(G[j])
14:    for i := L_F to R_F do
15:        for j := L_G to R_G do
16:            if both subtrees F[L_F ... i] ∧ G[L_G ... j] are trees then
17:                C_min := min{
18:                    forestdistance(F[L_F ... i - 1], G[L_G ... j] + C_del(F[i]),
19:                    forestdistance(F[L_F ... i], G[L_G ... j - 1] + C_ins(G[j]),
20:                    forestdistance(F[L_F ... i - 1], G[L_G ... j - 1]+
21:                        C_upd(F[i], G[j])}
22:                forestdistance[F[L_F ... i], G[L_G ... j]] := C_min
23:                D[i, j] := C_min
24:            else
25:                C_min := min{
26:                    forestdistance(F[L_F ... i - 1], G[L_G ... j] + C_del(F[i]),
27:                    forestdistance(F[L_F ... i], G[L_G ... j - 1] + C_ins(G[j]),
28:                    forestdistance(F[L_F ... id(mostleft(F[i])) - 1],
29:                        G[L_G ... id(mostleft(G[j])) - 1] + D[i, j]}
30:                forestdistance[F[L_F ... i], G[L_G ... j]] := C_min
```

Algorithm 3 DMRW

4. Kreslenie molekuly

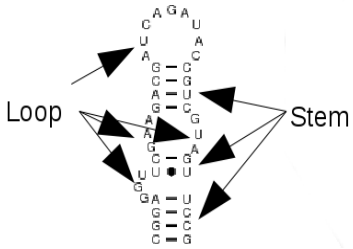
Po tom co ziskame a aplikujeme mapovanie medzi sablonovou a cielovou molekulou RNA, ziskame cielovu molekulu s ciastocnou vizualizaciou, ktorej zvysok treba dopocitat.

Po operaciach delete ostavaju v molekule prazdne diery, naopak po insertoch potrebujeme vypocitat, kam umiestnime bazovy par, resp. samotnu bazu, pripadne este potrebujeme pre nu urobit miesto. Update vrcholu v strome nerobi ziadne strukturne zmeny, zmeni sa iba nazov bazy na danom mieste.

Sekundarna struktura RNA obsahuje mnozstvo motivov popisanych na obrazku 1.2. Vo vseobecnosti ale sa kazdy z tychto motivov sklada zo stemu a loopu.

Stemom budeme dalej nazyvat cast RNA ktora zodpoveda vnutornemu vrcholu v strome. Loopom budeme oznacovat listy v RNA strome (lese), nezalezi ci je to bulge, interior loop, hairpin alebo multibranch loop, ako aj ukazuje obrazok 4.1.

Stem zacina vzdy v najvyssom vrchole stromu (v smere ku korenu), ktory je zaroven vnutornym vrcholom a nema ziadnych surodencov, ktory by boli rovnako vnutornymi vrcholmi. To znamena, ze do multibranch loop vchadza 1 stem (ten tu konci) a vychadza z nej niekolko novych stemov. Naopak pre bulge a interior loopy jeden stem vchadza do struktury ale pokracuje dalej.



Obr. 4.1: Stem a loop v molekule

5. Nápověda k sazbě

5.1 Úprava práce

Vlastní text bakalářské práce je uspořádaný hierarchicky do kapitol a podkapitol, každá kapitola začíná na nové straně. Text je zarovnán do bloku. Nový odstavec se obvykle odděluje malou vertikální mezerou a odsazením prvního řádku. Grafická úprava má být v celém textu jednotná.

Práce se tiskne na bílý papír formátu A4. Okraje musí ponechat dost místa na vazbu: doporučen je horní, dolní a pravý okraj 25 mm, levý okraj 40 mm. Číslují se všechny strany kromě obálky a informačních stran na začátku práce; první číslovaná strana bývá obvykle ta s obsahem.

Písmo se doporučuje dvanáctibodové (12 pt) se standardní vzdáleností mezi řádky (pokud píšete ve Wordu nebo podobném programu, odpovídá tomu řádkování 1,5; v TEXu není potřeba nic přepínat). Pro běžný text používejte vzpřímené patkové písmo. Text matematických vět se obvykle tiskne pro zdůraznění skloněným (slanted) písmem, není-li k dispozici, může být zastoupeno kurzívou.

Primárně je doporučován jednostranný tisk (příliš tenkou práci lze obtížně svázat). Delší práce je lepší tisknout oboustranně a přizpůsobit tomu velikosti okrajů: 40 mm má vždy *vnitřní* okraj. Rub titulního listu zůstává nepotištěný.

Zkratky použité v textu musí být vysvětleny vždy u prvního výskytu zkratky (v závorce nebo v poznámce pod čarou, jde-li o složitější vysvětlení pojmu či zkratky). Pokud je zkratek více, připojuje se seznam použitých zkratek, včetně jejich vysvětlení a/nebo odkazů na definici.

Delší převzatý text jiného autora je nutné vymezit uvozovkami nebo jinak vyznačit a řádně citovat.

5.2 Jednoduché příklady

Čísla v českém textu obvykle sázíme v matematickém režimu s desetinnou čárkou: $\pi \doteq 3{,}141\,592\,653\,589$. V matematických textech se považuje za přípustné používat desetinnou tečku (pro lepší odlišení od čárky v roli oddělovače). Numerické výsledky se uvádějí s přiměřeným počtem desetinných míst.

Mezi číslo a jednotku patří úzká mezera: šířka stránky A4 činí 210 mm, což si pamatuje pouze 5 % autorů. Pokud ale údaj slouží jako přívlastek, mezeru vynecháváme: 25mm okraj, 95% interval spolehlivosti.

Rozlišujeme různé druhy pomlček: červeno-černý (krátká pomlčka), strana 16–22 (střední), $45 - 44$ (matematické minus), a toto je — jak se asi dalo čekat — vložená věta ohraničená dlouhými pomlčkami.

V českém textu se používají „české“ uvozovky, nikoliv “anglické”.

Na některých místech je potřeba zabránit lámání řádku (v~TEXu značíme vlnovkou): u~předložek (neslabičnych, nebo obecně jednopísmenných), vrchol~v, před k~kroky, a~proto, … obecně kdekoliv, kde by při rozlomení čtenář „škobrtnul“.

5.3 Matematické vzorce a výrazy

Proměnné sázíme kurzívou (to TEX v matematickém módu dělá sám, ale nezapomínejte na to v okolním textu a také si matematický mód zapněte). Názvy funkcí sázíme vzpřímeně. Tedy například: $\mathrm{var}(X) = \mathsf{E}\, X^2 - \big(\mathsf{E}\, X\big)^2$.

Zlomky uvnitř odstavce (třeba $\frac{5}{7}$ nebo $\frac{x+y}{2}$) mohou být příliš stísněné, takže je lepší sázet jednoduché zlomky s lomítkem: $5/7$, $(x+y)/2$.

Nechť

$$\mathbb{X} = \begin{pmatrix} \boldsymbol{x}_1^\top \\ \vdots \\ \boldsymbol{x}_n^\top \end{pmatrix}.$$

Povšimněme si tečky za maticí. Byť je matematický text vysázen ve specifickém prostředí, stále je gramaticky součástí věty a tudíž je zapotřebí neopomenout patřičná interpunkční znaménka. Výrazy, na které chceme později odkazovat, je vhodné očíslovat:

$$\mathbb{X} = \begin{pmatrix} \boldsymbol{x}_1^\top \\ \vdots \\ \boldsymbol{x}_n^\top \end{pmatrix}. \tag{5.1}$$

Výraz (5.1) definuje matici $\mathbb{X}$. Pro lepší čitelnost a přehlednost textu je vhodné číslovat pouze ty výrazy, na které se autor někde v další části textu odkazuje. To jest, nečíslujte automaticky všechny výrazy vysázené některým z matematických prostředí.

Zarovnání vzorců do několika sloupečků:

$$\begin{aligned} S(t) &= \mathsf{P}(T > t), \qquad t > 0 \qquad \text{(zprava spojitá)}, \\ F(t) &= \mathsf{P}(T \le t), \qquad t > 0 \qquad \text{(zprava spojitá)}. \end{aligned}$$

Dva vzorce se spojovníkem:

$$\left.\begin{aligned} S(t) &= \mathsf{P}(T > t) \\ F(t) &= \mathsf{P}(T \le t) \end{aligned}\right\} \quad t > 0 \qquad \text{(zprava spojité)}. \tag{5.2}$$

Dva centrované nečíslované vzorce:

$$\boldsymbol{Y} = \mathbb{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon},$$

$$\mathbb{X} = \begin{pmatrix} 1 & \boldsymbol{x}_1^\top \\ \vdots & \vdots \\ 1 & \boldsymbol{x}_n^\top \end{pmatrix}.$$

Dva centrované číslované vzorce:

$$\boldsymbol{Y} = \mathbb{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \tag{5.3}$$

$$\mathbb{X} = \begin{pmatrix} 1 & \boldsymbol{x}_1^\top \\ \vdots & \vdots \\ 1 & \boldsymbol{x}_n^\top \end{pmatrix}. \tag{5.4}$$

Definice rozdělená na dva případy:

$$P_{r-j} = \begin{cases} 0, & \text{je-li } r-j \text{ liché,} \\ r!\,(-1)^{(r-j)/2}, & \text{je-li } r-j \text{ sudé.} \end{cases}$$

Všimněte si použití interpunkce v této konstrukci. Čárky a tečky se dávají na místa, kam podle jazykových pravidel patří.

$$\begin{aligned} x &= y_1 - y_2 + y_3 - y_5 + y_8 - \cdots = && \text{z (5.3)} \\ &= y' \circ y^* = && \text{podle (5.4)} \\ &= y(0)y' && \text{z Axiomu 1.} \end{aligned} \tag{5.5}$$

Dva zarovnané vzorce nečíslované:

$$\begin{aligned} L(\boldsymbol{\theta}) &= \prod_{i=1}^{n} f_i(y_i;\, \boldsymbol{\theta}), \\ \ell(\boldsymbol{\theta}) &= \log\bigl\{L(\boldsymbol{\theta})\bigr\} = \sum_{i=1}^{n} \log\bigl\{f_i(y_i;\, \boldsymbol{\theta})\bigr\}. \end{aligned}$$

Dva zarovnané vzorce, první číslovaný:

$$\begin{aligned} L(\boldsymbol{\theta}) &= \prod_{i=1}^{n} f_i(y_i;\, \boldsymbol{\theta}), \\ \ell(\boldsymbol{\theta}) &= \log\bigl\{L(\boldsymbol{\theta})\bigr\} = \sum_{i=1}^{n} \log\bigl\{f_i(y_i;\, \boldsymbol{\theta})\bigr\}. \end{aligned} \tag{5.6}$$

Vzorec na dva řádky, první řádek zarovnaný vlevo, druhý vpravo, nečíslovaný:

$$\begin{aligned} \ell(\mu, \sigma^2) = \log\bigl\{L(\mu, \sigma^2)\bigr\} &= \sum_{i=1}^{n} \log\bigl\{f_i(y_i;\, \mu, \sigma^2)\bigr\} = \\ &= -\frac{n}{2}\,\log(2\pi\sigma^2) \,-\, \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \mu)^2. \end{aligned}$$

Vzorec na dva řádky, zarovnaný na =, číslovaný uprostřed:

$$\begin{aligned} \ell(\mu, \sigma^2) = \log\bigl\{L(\mu, \sigma^2)\bigr\} &= \sum_{i=1}^{n} \log\bigl\{f(y_i;\, \mu, \sigma^2)\bigr\} = \\ &= -\frac{n}{2}\,\log(2\pi\sigma^2) \,-\, \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \mu)^2. \end{aligned} \tag{5.7}$$

5.4 Definice, věty, důkazy, ...

Konstrukce typu definice, věta, důkaz, příklad, ...je vhodné odlišit od okolního textu a případně též číslovat s možností použití křížových odkazů. Pro každý typ těchto konstrukcí je vhodné mít v souboru s makry (`makra.tex`) nadefinované jedno prostředí, které zajistí jak vizuální odlišení od okolního textu, tak automatické číslování s možností křížově odkazovat.

Definice 9. *Nechť náhodné veličiny $X_1, \ldots, X_n$ jsou definovány na témž pravděpodobnostním prostoru $(\Omega, \mathcal{A}, P)$. Pak vektor $\boldsymbol{X} = (X_1, \ldots, X_n)^\top$ nazveme* náhodným vektorem.

Definice 10 (náhodný vektor)**.** *Nechť náhodné veličiny $X_1, \ldots, X_n$ jsou definovány na témž pravděpodobnostním prostoru $(\Omega, \mathcal{A}, P)$. Pak vektor $\boldsymbol{X} = (X_1, \ldots, X_n)^\top$ nazveme* náhodným vektorem.

Definice 9 ukazuje použití prostředí pro sazbu definice bez titulku, definice 10 ukazuje použití prostředí pro sazbu definice s titulkem.

Věta 1. *Náhodný vektor $\boldsymbol{X}$ je měřitelné zobrazení prostoru $(\Omega, \mathcal{A}, P)$ do $(\mathbb{R}_n, \mathcal{B}_n)$.*

Lemma 2 (Anděl, 2007, str. 29)**.** *Náhodný vektor $\boldsymbol{X}$ je měřitelné zobrazení prostoru $(\Omega, \mathcal{A}, P)$ do $(\mathbb{R}_n, \mathcal{B}_n)$.*

Důkaz. Jednotlivé kroky důkazu jsou podrobně popsány v práci Anděl (2007, str. 29).

□

Věta 1 ukazuje použití prostředí pro sazbu matematické věty bez titulku, lemma 2 ukazuje použití prostředí pro sazbu matematické věty s titulkem. Lemmata byla zavedena v hlavním souboru tak, že sdílejí číslování s větami.

6. Odkazy na literaturu

Odkazy na literaturu vytváříme nejlépe pomocí příkazů `\citet`, `\citep` atp. (viz LaTeXový balíček natbib) a následného použití BibTeXu. V matematickém textu obvykle odkazujeme stylem „Jméno autora/autorů (rok vydání)“, resp. „Jméno autora/autorů [číslo odkazu]“. V českém/slovenském textu je potřeba se navíc vypořádat s nutností skloňovat jméno autora, respektive přechylovat jméno autorky. Je potřeba mít na paměti, že standardní příkazy `\citet`, `\citep` produkují referenci se jménem autora/autorů v prvním pádě a jména autorek jsou nepřechýlena.

Pokud nepoužíváme bibTeX, řídíme se normou ISO 690 a zvyklostmi oboru.

Jména časopisů lze uvádět zkráceně, ale pouze v kodifikované podobě.

6.1 Několik ukázek

Mezi nejvíce citované statistické články patří práce Kaplana a Meiera a Coxe (Kaplan a Meier, 1958; Cox, 1972). Student (1908) napsal článek o t-testu.

Prof. Anděl je autorem učebnice matematické statistiky (viz Anděl, 1998). Teorii odhadu se věnuje práce Lehmann a Casella (1998). V případě odkazů na specifickou informaci (definice, důkaz, ...) uvedenou v knize bývá užitečné uvést specificky číslo kapitoly, číslo věty atp. obsahující požadovanou informaci, např. viz Anděl (2007, Věta 4.22) nebo (viz Anděl, 2007, Věta 4.22).

Mnoho článků je výsledkem spolupráce celé řady osob. Při odkazování v textu na článek se třemi autory obvykle při prvním výskytu uvedeme plný seznam: Dempster, Laird a Rubin (1977) představili koncept EM algoritmu. Respektive: Koncept EM algoritmu byl představen v práci Dempstera, Lairdové a Rubina (Dempster, Laird a Rubin, 1977). Při každém dalším výskytu již používáme zkrácenou verzi: Dempster a kol. (1977) nabízejí též několik příkladů použití EM algoritmu. Respektive: Několik příkladů použití EM algoritmu lze nalézt též v práci Dempstera a kol. (Dempster a kol., 1977).

U článku s více než třemi autory odkazujeme vždy zkrácenou formou: První výsledky projektu ACCEPT jsou uvedeny v práci Genbergové a kol. (Genberg a kol., 2008). V textu *nenapíšeme*: První výsledky projektu ACCEPT jsou uvedeny v práci Genberg, Kulich, Kawichai, Modiba, Chingono, Kilonzo, Richter, Pettifor, Sweat a Celentano (2008).

7. Tabulky, obrázky, programy

Používání tabulek a grafů v odborném textu má některá společná pravidla a některá specifická. Tabulky a grafy neuvádíme přímo do textu, ale umístíme je buď na samostatné stránky nebo na vyhrazené místo v horní nebo dolní části běžných stránek. LaTeX se o umístění plovoucích grafů a tabulek postará automaticky.

Každý graf a tabulku očíslujeme a umístíme pod ně legendu. Legenda má popisovat obsah grafu či tabulky tak podrobně, aby jim čtenář rozuměl bez důkladného studování textu práce.

Na každou tabulku a graf musí být v textu odkaz pomocí jejich čísla. Na příslušném místě textu pak shrneme ty nejdůležitější závěry, které lze z tabulky či grafu učinit. Text by měl být čitelný a srozumitelný i bez prohlížení tabulek a grafů a tabulky a grafy by měly být srozumitelné i bez podrobné četby textu.

Na tabulky a grafy odkazujeme pokud možno nepřímo v průběhu běžného toku textu; místo *„Tabulka 7.1 ukazuje, že muži jsou v průměru o* 9,9 kg *těžší než ženy“* raději napíšeme *„Muži jsou o* 9,9 kg *těžší než ženy (viz Tabulka 7.1)“*.

7.1 Tabulky

U **tabulek** se doporučuje dodržovat následující pravidla:

- Vyhýbat se svislým linkám. Silnějšími vodorovnými linkami oddělit tabulku od okolního textu včetně legendy, slabšími vodorovnými linkami oddělovat záhlaví sloupců od těla tabulky a jednotlivé části tabulky mezi sebou. V LaTeXu tuto podobu tabulek implementuje balík `booktabs`. Chceme-li výrazněji oddělit některé sloupce od jiných, vložíme mezi ně větší mezeru.

- Neměnit typ, formát a význam obsahu políček v tomtéž sloupci (není dobré do téhož sloupce zapisovat tu průměr, onde procenta).

- Neopakovat tentýž obsah políček mnohokrát za sebou. Máme-li sloupec *Rozptyl*, který v prvních deseti řádcích obsahuje hodnotu 0,5 a v druhých deseti řádcích hodnotu 1,5, pak tento sloupec raději zrušíme a vyřešíme to jinak. Například můžeme tabulku rozdělit na dvě nebo do ní vložit popisné řádky, které informují o nějaké proměnné hodnotě opakující se v následujícím oddíle tabulky (např. *„Rozptyl* = 0,5“ a níže *„Rozptyl* = 1,5“).

Efekt	**Odhad**	**Směrod. chyba**[a]	**P-hodnota**
Abs. člen	−10,01	1,01	—
Pohlaví (muž)	9,89	5,98	0,098
Výška (cm)	0,78	0,12	$< 0{,}001$

Pozn: [a] Směrodatná chyba odhadu metodou Monte Carlo.

Tabuľka 7.1: Maximálně věrohodné odhady v modelu M.

- Čísla v tabulce zarovnávat na desetinnou čárku.
- V tabulce je někdy potřebné používat zkratky, které se jinde nevyskytují. Tyto zkratky můžeme vysvětlit v legendě nebo v poznámkách pod tabulkou. Poznámky pod tabulkou můžeme využít i k podrobnějšímu vysvětlení významu některých sloupců nebo hodnot.

7.2 Obrázky

Několik rad týkajících se obrázků a grafů.

- Graf by měl být vytvořen ve velikosti, v níž bude použit v práci. Zmenšení příliš velkého grafu vede ke špatné čitelnosti popisků.
- Osy grafu musí být řádně popsány ve stejném jazyce, v jakém je psána práce (absenci diakritiky lze tolerovat). Kreslíme-li graf hmotnosti proti výšce, nenecháme na nich popisky `ht` a `wt`, ale osy popíšeme *Výška [cm]* a *Hmotnost [kg]*. Kreslíme-li graf funkce $h(x)$, popíšeme osy x a $h(x)$. Každá osa musí mít jasně určenou škálu.
- Chceme-li na dvourozměrném grafu vyznačit velké množství bodů, dáme pozor, aby se neslily do jednolité černé tmy. Je-li bodů mnoho, zmenšíme velikost symbolu, kterým je vykreslujeme, anebo vybereme jen malou část bodů, kterou do grafu zaneseme. Grafy, které obsahují tisíce bodů, dělají problémy hlavně v elektronických dokumentech, protože výrazně zvětšují velikost souborů.
- Budeme-li práci tisknout černobíle, vyhneme se používání barev. Čáry rozlišujeme typem (plná, tečkovaná, čerchovaná,...), plochy dostatečně rozdílnými intensitami šedé nebo šrafováním. Význam jednotlivých typů čar a ploch vysvětlíme buď v textové legendě ke grafu anebo v grafické legendě, která je přímo součástí obrázku.
- Vyhýbejte se bitmapovým obrázkům o nízkém rozlišení a zejména JPEGům (zuby a kompresní artefakty nevypadají na papíře pěkně). Lepší je vytvářet obrázky vektorově a vložit do textu jako PDF.

7.3 Programy

Algoritmy, výpisy programů a popis interakce s programy je vhodné odlišit od ostatního textu. Jednou z možností je použití LaTeXového balíčku `fancyvrb` (fancy verbatim), pomocí něhož je v souboru `makra.tex` nadefinováno prostředí `code`. Pomocí něho lze vytvořit např. následující ukázky.

```
> mean(x)
[1] 158.90
> objekt$prumer
[1] 158.90
```

Menší písmo:

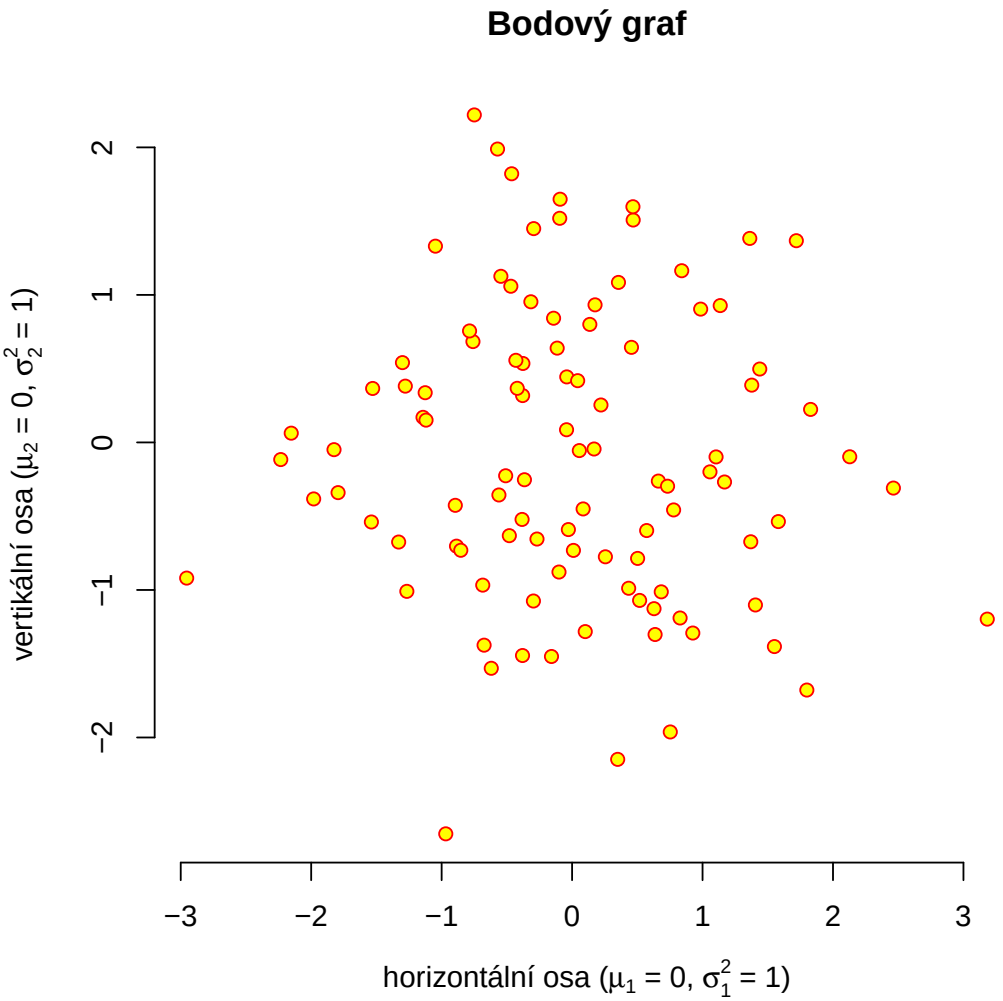
```
> mean(x)
[1] 158.90
> objekt$prumer
[1] 158.90
```

Bez rámečku:

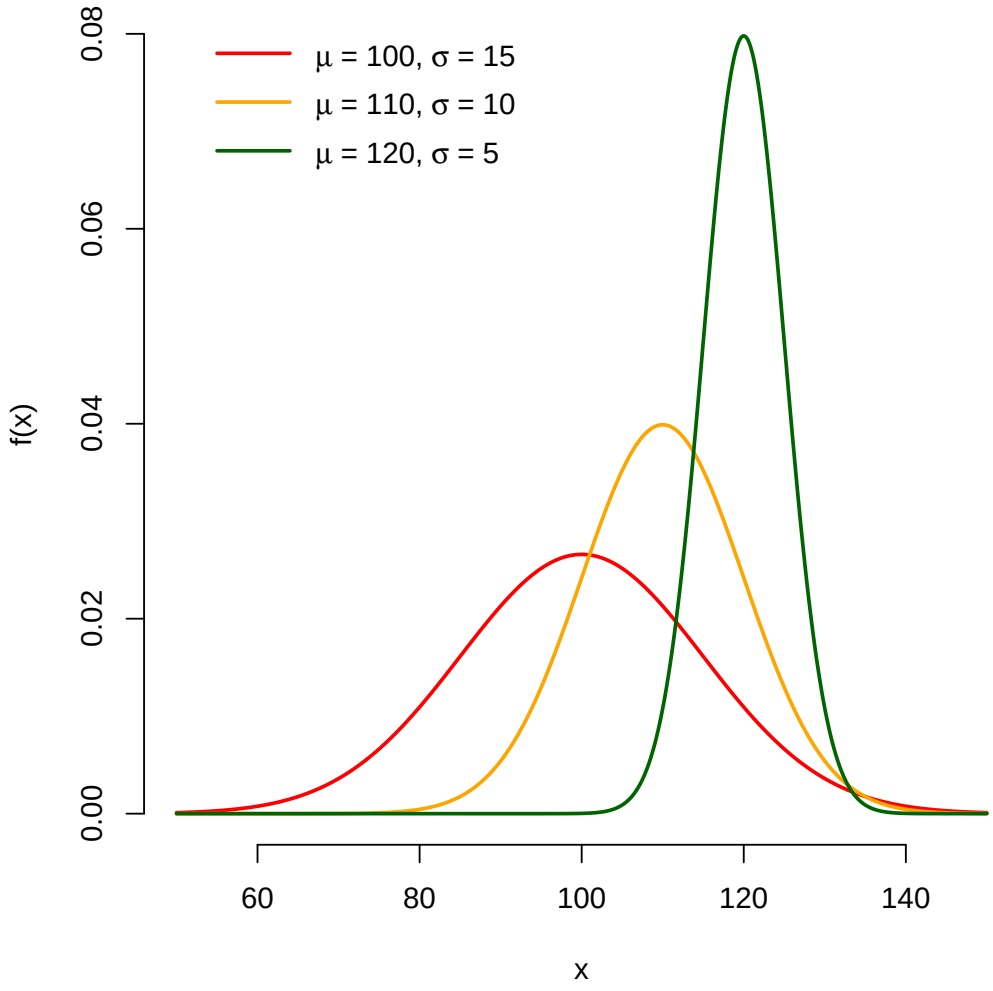
```
> mean(x)
[1] 158.90
> objekt$prumer
[1] 158.90
```

Užší rámeček:

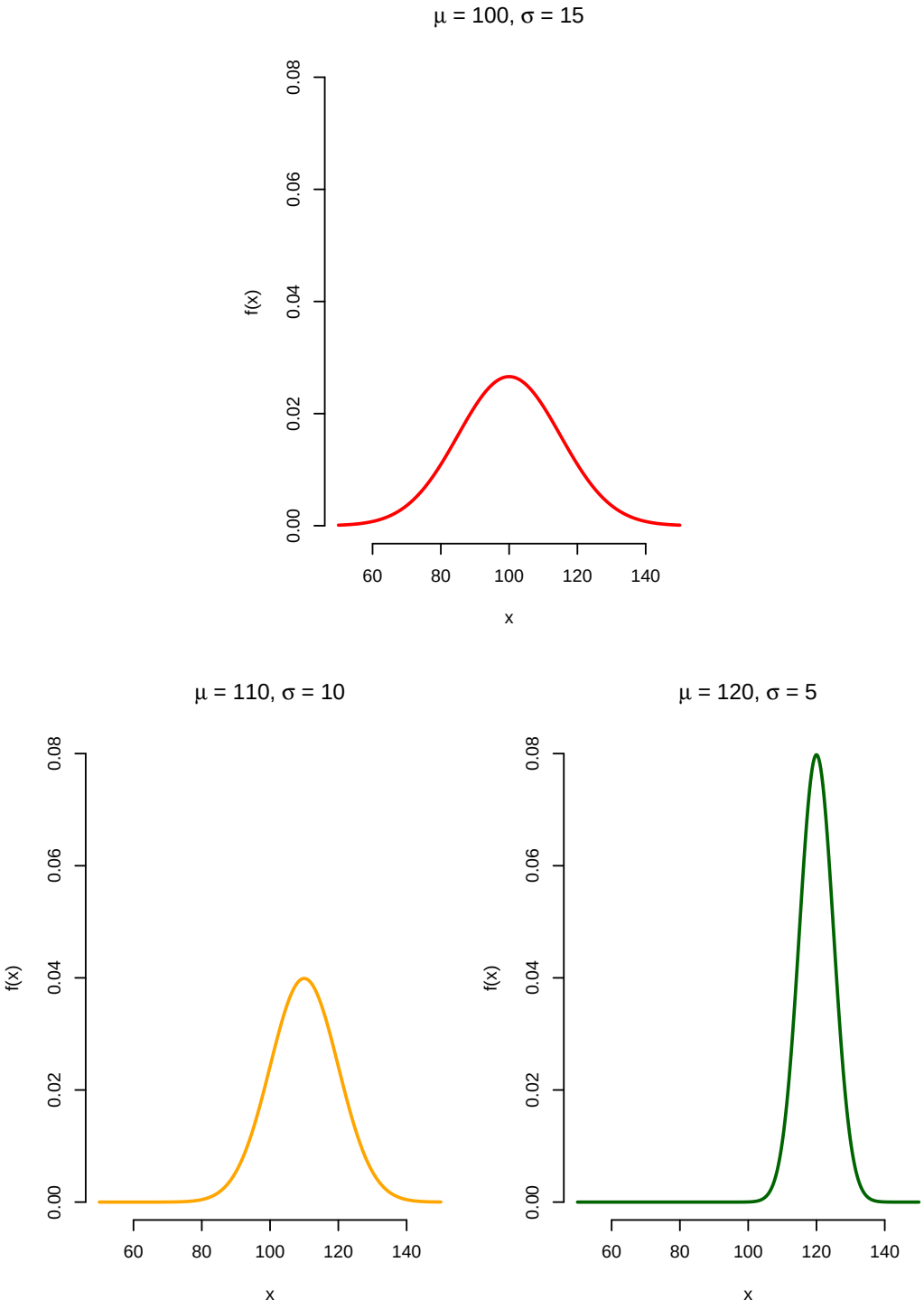
```
> mean(x)
[1] 158.90
> objekt$prumer
[1] 158.90
```



Obr. 7.1: Náhodný výběr z rozdělení $\mathcal{N}_2(\mathbf{0}, I)$.



Obr. 7.2: Hustoty několika normálních rozdělení.



Obr. 7.3: Hustoty několika normálních rozdělení.

Závěr

Seznam použité literatury

Anděl, J. (1998). *Statistické metody*. Druhé přepracované vydání. Matfyzpress, Praha. ISBN 80-85863-27-8.

Anděl, J. (2007). *Základy matematické statistiky*. Druhé opravené vydání. Matfyzpress, Praha. ISBN 80-7378-001-1.

Cox, D. R. (1972). Regression models and life-tables (with Discussion). *Journal of the Royal Statistical Society, Series B*, **34**(2), 187–220.

Demaine, E. D., Mozes, S., Rossman, B. a Weimann, O. (2009). An optimal decomposition algorithm for tree edit distance. *ACM Trans. Algorithms*, **6**(1), 2:1–2:19. ISSN 1549-6325. doi: 10.1145/1644015.1644017. URL `http://doi.acm.org/10.1145/1644015.1644017`.

Dempster, A. P., Laird, N. M. a Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, **39**(1), 1–38.

Dulucq, S. a Touzet, H. (2003). *Combinatorial Pattern Matching: 14th Annual Symposium, CPM 2003 Morelia, Michoacán, Mexico, June 25–27, 2003 Proceedings*, chapter Analysis of Tree Edit Distance Algorithms, pages 83–95. Springer Berlin Heidelberg, Berlin, Heidelberg. ISBN 978-3-540-44888-4. doi: 10.1007/3-540-44888-8˙7. URL `http://dx.doi.org/10.1007/3-540-44888-8_7`.

Genberg, B. L., Kulich, M., Kawichai, S., Modiba, P., Chingono, A., Kilonzo, G. P., Richter, L., Pettifor, A., Sweat, M. a Celentano, D. D. (2008). HIV risk behaviors in sub-Saharan Africa and Northern Thailand: Baseline behavioral data from project Accept. *Journal of Acquired Immune Deficiency Syndrome*, **49**, 309–319.

Kaplan, E. L. a Meier, P. (1958). Nonparametric estimation from incomplete observations. *Journal of the American Statistical Association*, **53**(282), 457–481.

Klein, P. N. (1998). Computing the edit-distance between unrooted ordered trees. In *Proceedings of the 6th Annual European Symposium on Algorithms*, ESA '98, pages 91–102, London, UK, UK, 1998. Springer-Verlag. ISBN 3-540-64848-8. URL `http://dl.acm.org/citation.cfm?id=647908.740125`.

Lehmann, E. L. a Casella, G. (1998). *Theory of Point Estimation*. Second Edition. Springer-Verlag, New York. ISBN 0-387-98502-6.

Pawlik, M. a Augsten, N. (2011). Rted: A robust algorithm for the tree edit distance. *Proc. VLDB Endow.*, **5**(4), 334–345. ISSN 2150-8097. doi: 10.14778/2095686.2095692. URL `http://dx.doi.org/10.14778/2095686.2095692`.

Student (1908). On the probable error of the mean. *Biometrika*, **6**, 1–25.

Tai, K.-C. (1979). The tree-to-tree correction problem. *J. ACM*, **26**(3), 422–433. ISSN 0004-5411. doi: 10.1145/322139.322143. URL http://doi.acm.org/10.1145/322139.322143.

Zhang, K. a Shasha, D. (1989). Simple fast algorithms for the editing distance between trees and related problems. *SIAM Journal on Computing*, **18**(6), 1245 – 1262.

Zoznam obrázkov

Zoznam tabuliek

Seznam použitých zkratek

Přílohy