Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra kybernetiky

Bakalářská práce

Metoda pro identifikaci MICA a MICB genů

Plzeň, 2021 Lucie Rottenbornová

Zadání

Prohlášení

Předkládám tímto k posouzení a obhajobě bakalářskou práci zpracovanou na závěr studia na Fakultě aplikovaných věd Západočeské univerzity v Plzni.

Prohlašuji, že jsem bakalářskou práci vypracovala samostatně a výhradně s použitím odborné literatury a pramenů, jejichž úplný seznam je její součástí.

V Plzni dne 10. srpna 2021

Lucie Rottenbornová

Poděkování

Chtěla bych poděkovat především Ing. Lucii Houdové, Ph.D. za rady, připomínky a hlavně trpělivost při zpracování praktické části práce. Dále bych chtěla poděkovat Ing. Kateřině Kratochvílové za její věcné připomínky k řešení praktické části. V neposlední řadě bych chtěla poděkovat Mgr. Monice Holubové, Ph.D. za pomoc v teoretické části práce.

Abstrakt

Bakalářská práce se zabývá identifikací alel MICA a MICB genů pro následné použití v rámci transplantce kostní dřeně. Cílem práce je seznámení se s významem genů MICA a MICB, porovnání sekvenačních metod, zejména Sangerovo sekvenování a Next-generation sekvenování (NGS), porozumění problematice identifikace alel a následný návrh metody pro automatickou identifikaci alel. Metoda byla vyvíjena na základě syntetických dat s vlastnostmi vycházejícími z reálných experimentů prováděných ve FN Plzeň. Kromě samotné metody realizované v jazyce Python za pomoci rozšíření Biopython je v práci popsán způsob získávání referenčních dat, vytváření syntetických dat a zhodnoceny výsledky ověřování metody.

Abstract

This bachelor thesis deals with the identification of alleles of MICA and MICB genes for subsequent use in bone marrow transplantation. The aim is to get acquainted with the importance of MICA and MICB genes in bone marrow transplant, comparison of sequencing methods, especially Sanger sequencing and Next-generation sequencing (NGS), understanding the problems of allele identification and subsequent design of method for automatic allele identification. The method was developed based on synthetic data with properties similar to real data from experiments performed at FN Plzeň. In addition to the method implemented in Python using the Biopython extension, the work describes the method of obtaining reference data, creating synthetic data and evaluating the results of verifying the method.

Obsah

Úvod

Transplantace kmenových krvetvorných buněk (dále jen TKB) je proces, kdy pacient obdrží zdravé kmenové buňky, aby nahradily ty poškozené. Před touto transplantací pacient podstoupí léčbu chemoterapií či terapii radiací. Poté jsou zdravé kmenové buňky dárce injikovány do krve příjemce a putují do kostní dřeně, kde začnou vyrábět nové červené a bílé krvinky a trombocyty (krevní destičky).[**1**]

Hlavní zdravotní komplikací je reakce štěpu proti hostiteli (Graft versus Host Disease, dále jen GvHD) při alogenní (nepříbuzenské) transplantaci kmenových buněk. Po této transplantaci mohou nastat dva stavy - mírná forma onemocnění, která netrvá dlouho, či závažná forma onemocnění, která může být i smrtelná. Akutní GvHD (aGvHD) se objeví v prvních 100 dnech po transplantaci. Příčinou jsou zralé T-lymfocyty dárce, které jsou zaměřené proti hlavním nebo minoritním HLA (Human Leucocyte Antigen) genům příjemce. Postižené jsou především játra, žaludek, střeva a kůže. Jako doprovod jsou puchýře či vyrážka, průjem, krev ve stolici, tmavá moč nebo nechutenství k jídlu. Chronické GvHD (cGvHD) se objevuje po více než 100 dnech od transplantace a postihuje hlavně kůži, ústa a oči. Hlavními symptomy jsou také bolest, ztuhlost kloubů, obtížné otvírání úst, kašel, potíže s dýcháním a změna struktury pokožky.[**1**] Faktory, které je důležité zohlednit, abychom snížili šanci na rozvoj komplikací, jsou věk, stádium léčeného onemocnění, časový interval mezi určením diagnózy a transplantací[**2**] a další. Věk se musí vzít v potaz také u dárce, dále také rozhoduje, jestli je dárce v příbuzenském vztahu k příjemci a odkud kmenové buňky pocházejí.[**1**] Základním kritériem výběru dárce je shoda HLA genů, avšak v posledních letech je řešen význam a vliv i jiných, tzv. non-HLA genů. Pokud by došlo k situaci, kdy je více dárců se shodnými HLA znaky, může shoda non-HLA genů pomoci při výběru vhodného dárce.

V práci se zaměřím na biologické funkci HLA a non-HLA genů (zejména MICA a MICB) při TKB, metodám čtení biologických sekvencí, porozumění genetického kódu a značení, možnostem ukládání biologických dat, hlavním problémům při identifikaci alel a samotnému návrhu řešení a jeho vyhodnocení na syntetických datech.

Část I. - Teoretické znalosti

1 Imunitní systém

Imunitu organismu rozdělujeme na imunitu přirozenou a imunitu adaptivní. **Přirozená** imunita (nazývaná také jako vrozená) je tvořena látkovou imunitou, která pracuje s protilátkami, a buněčnou imunitou. **Adaptivní** imunita (získaná) reaguje později po reakci imunity vrozené. Přirozená imunita u obratlovců zahrnuje **hlavní histokompatibilní systém** (MHC = Major histocompatibility complex). Jedná se o skupinů glykoproteinů na površích buněk obratlovců, které slouží jako kontrola buněk před parazity. Pokud je buňka infikována, na svém povrchu "ukáže" specifické molekuly, které buňky imunitního systému rozeznají a buňku zničí. U člověka se tento MHC komplex nazývá **Human Leukocyte Antigen** (HLA). [**3**]

1.1 HLA - Human Leukocyte Antigen

HLA je rozsáhlý komplex genů lidského MHC (Main Histocompatibility Complex - hlavní histokompatibilní komplex/systém) s lokalizací na chromozomu 6p21. Tyto proteiny na povrchu buněk jsou zodpovědné za regulaci imunitního systému. Proteiny zakódované určitými geny jsou důležitými faktory v transplantaci, a proto se začaly nazývat "antigeny". HLA geny jsou vysoce polymorfní, což znamená, že obsahují mnoho různých alel (variant genů), a to pomáhá vytvořit přizpůsobivý imunitní systém. Každý jedinec má unikátní sestavu HLA alel (výjimku tvoří jednovaječná dvojčata). [**1**]

HLA se, podle jejich funkce, dělí na více skupin. **HLA I. třídy** (HLA-A, B, C) se nacházejí téměř ve všech buňkách (kromě buněk pohlavních) a antigeny jsou "ukazovány" na povrchu buňky T-lymfocytům (druh bílých krvinek). **HLA II. třídy** (HLA-DP, DM, DR) nejsou přítomny ve všech buňkách, nachází se zejména v B-lymfocytech, dendrických buňkách, makrofázích a od nich odvozených imunitních buňkách. Na tyto antigeny reagují tzv. pomocné T-lymfocyty, které buňku nezabíjejí, ale pouze vytvoří imunologický poplach upozorněním

B-lymfocytům, které začnou vyrábět protilátky. **HLA III. třídy** (složky komplementu HLA-C2, C4, faktor B) se nepodílí na imunitní prezentaci antigenů. Pouze několik proteinů se podílí na funci imunity a většina z nich slouží pouze ke komunikaci buněk na jejich povrchu. [**1**]

1.1.1 HLA geny a jejich význam při transplantaci

Pro pochopení role HLA při transplantaci je třeba si nejdříve popsat samotné typy transplantace podle typu dárce. **Autologní** transplantace využívá pacientových kmenových buněk odebraných v bezpříznakovém období nebo s minimální aktivitou léčené nemoci. **Syngenní** transplantaci lze podstoupit, pokud má pacient jednovaječné dvojče. Jednovaječná dvojčata mají stejnou unikátní sestavu HLA alel. **Alogenní** transplantace je převod kmenových buněk mezi jedinci stejného živočišného druhu. Pokud dochází k transplantaci mezi příbuznými členy rodiny, volí se nejčastěji sourozenci.[**4**] Pokud dochází k nepříbuzenské transplantaci využívá se buněk nepříbuzného dárce dostupného v národních (Český národní registr dárců dřeně [**5**] a Český národní registr dárců krvetvorných buněk [**6**]) či mezinárodních registrech dárců kmenových buněk (kostní dřeně).

Produkty HLA genů - antigeny - jsou zodpovědné za imunitní odpověď organismu typu hostitel proti štěpu (HvGR) a štěp proti hostiteli (GvHR). Štěp proti leukémii (GVL) je také spojován s mírou HLA neshod, a tedy můžeme snížit pravděpodobnost vrácení onemocnění po transplantaci. V současnosti v transplantaci rozhoduje shoda HLA alel mezi dárcem a příjemcem v 5 klasických HLA lokusech : HLA-A, -B, -C, -DRB1, -DQB1. Ideální je tedy shoda 10/10 (popř. 12/12 pokud je řešena i shoda -DPB1 genu), což snižuje riziko vážné akutní reakce GvHD, a je spojena s lepším přežíváním ve srovnání s transplantacemi od HLA neshodných dárců. [**1**]

1.2 Genový polymorfismus

Genový polymorfismus chápeme jako označení pro dvě a více alel (variant) genu, který reprezentuje znak stéjného charakteru. Zároveň se ale samotná varianta alely musí vyskytovat alespoň v 1 % populace. Pokud je její výskyt menší, označujeme tuto odlišnost jako tzv. mutaci. Genový polymorfismus se častěji objevuje v nekódujících oblastech (introny) než v oblastech kódujících (exony) - více viz kapitola 2 Genetický kód. Polymorfismus u non-HLA zahrnuje **jedno-nukleový polymorfismus** (SNPs = Single nucleotide polymorphisms), kdy se liší pouze jeden nukleotid v celé DNA sekvenci, **opakování dvojic** (TRs = Tandem Repeats), kdy se opakují dvě a více (různých) dvojice bází za sebou (časté pro introny), a **variace počtu kopií** (CNPs = Copy Number Polymorphisms), kdy se opakuje stejná kopie vícekrát za sebou.[**7**; **8**]

1.3 Non-HLA geny

GvHD je jednou z hlavních příčin úmrtí pacienta, i přes to, že proběhne alogenní TKB s HLA shodou 10/10 (popř. 12/12). To nám ukazuje, že i non-HLA geny a jejich **polymorfismus** hrajou roli ve výběru dárce k příjemci.[**1**]

Vědci se v posledních letech snaží identifikovat takové non-HLA geny, které mají schopnost významně ovlivnit riziko GvHD a dalších komplikací po transplantaci, a ty by poté mohly být začleněny do klasické strategie výběru dárce. Zaměřují se na skupiny genů: pro cytokiny, pro chemokiny a jejich receptory, pro molekuly vrozené imunity, pro adhezivní molekuly, pro vedlejší histokompatibilní antigeny, **pro receptory NK buněk** či geny ovlivňující metabolismus léků.[**1**]

1.4 NK buňky, NKG2D receptor a jeho ligandy

NK buňky (Natural Killer Cells) jsou součástí přirozeného imunitního systému. Velmi rychle reagují na virové infekce (jejich působnost zde trvá až 3 dny) či tvorbu nádoru. Tvoří kolem 5 - 10 % všech lymfocytů v lidském oběhu. Klasické imunitní buňky s pomalejší reakcí detekují MHC přítomný na povrchu infikované buňky a spustí uvolňování cytokinů, což způsobí smrt infikované buňky.

To, že NK buňky mohou reagovat rychleji, je způsobeno jejich schopností rozpoznávat a zabíjet napadené buňky i s nepřítomností protilátek a MHC. Tuto kontrolní činnost vykonávají díky svým inhibičním a aktivačním receptorům na povrchu buněčné membrány. Při normálních podmínkách jsou receptory NK buněk v inhibičním stavu. NK buňky mají dva hlavní typy receptorů, které kontrolují rovnováhu mezi jejich aktivací a inhibicí. První identifikuje MHC I molekuly a zahrnuje "killer lectin-like" receptor (KLR) vytvořený kombinací CD94 buď s NKG2A nebo NKG2C a "killer" imunoglobulinový receptor (KIR), druhý identifikuje non-MHC I molekuly a zahrnuje NKG2D a přirozený cytotoxický receptor (NCR).[**9**]

Vzhledem k zaměření práce je vhodné se více zaměřit na NKG2D receptor (Natural Killer Group 2, member D) a jeho ligandy MICA a MICB.

NKG2D je kódován KLRK1 genem v NK-genovém komplexu (NKC), který se u člověka nachází na 12. chromosomu. NKG2D rozpoznává proteiny, které se objevují na povrchu infikovaných buněk. Exprese těchto proteinů závisí na různých typech stresu. Jedna z nejvýznamějších stresových situací je poškození DNA. Genotoxický stres, zastavená replikace DNA, špatně regulované buněčné dělení, virová replikace nebo některé virové produkty aktivují ATM a ATR kinázy. Tyto kinázy iniciují reakci na poškození DNA, která se podílí na regulaci ligandu NKG2D. Odpověď na poškození DNA se tak podílí na upozornění imunitního systému na přítomnost potenciálně nebezpečných buněk. [**10**]

MICA a **MICB** (MHC class I polypeptide-related sequence A/B) geny jsou lokalizovány na krátkém raménku 6. chromosomu (6p21). Klasifikují se jako non-HLA geny a jsou součástí skupiny 7 genů MIC (MICA - MICG). MICA a MICB jsou jako jediný ze skupiny MIC pravými geny, které mají bílkovinné produkty, zatímco ostatní MICC-MICG jsou pseudogeny. Oba geny jsou vysoce polymorfní. Jejich polymorfismus a stupeň exprese jsou spojovány s autoimunitními nemocemi, infekcemi a rakovinou. I přes jejich podobnost k HLA, nepředstavují peptidy a nejsou na povrchu leukocytů, ale na endoteliálních buňkách (vnitřních povrch krevních a lymfatických cév a srdce), fibroblastech (vazivové buňky), epiteliálních (vnitřní/vnější povrch organismu) a nádorových buňkách.[**11**]

Sekvence (posloupnosti znaků nukleových bází) MICA a MICB jsou z přibližně 91 % totožné. Exon 1 (o exonech více viz kapitola 2 Genetický kód) představuje vedoucí

peptid, exony 2 - 4 kódují syntézu extracelulárních domén a exon 5 kóduje transmembránovou oblast (TM) genu, exon 6 kóduje cytoplazmatický ocas. TM kóduje opakovaný polymorfismus.[**11**]

Dle [**12**] je pro MICA gen aktuálně popsáno 109 alel a stále se pracuje na popisu dalších. Bylo prokázáno, že polymorfismus MICA (MICA-129 a MICA A5.1) ovlivňuje signalizaci NKG2D a polymorfismus NKG2D ovlivňuje cytotoxickou reakci (zabíjení buněk). Při TKB MICA shoda mezi dárcem a příjemcem ukázala snížení aGvHD a cGvHD a zvýšila přežívání pacientů.[**11**] Pro MICB gen je u člověka je známo až 47 alel.[**12**] V praktické části při analýze referenčních dat z IPD databáze (více viz kapitola 4.6 Immuno Poylmorphism Database IPD) bylo zjištěno 388 záznamů alel genu MICA a 236 záznamů alel genu MICB.

MICA gen kóduje 383 bílkovin (proteinů), v databázi IPD je 1152 bps. Vzhledem k tomu, že jediné rozdíly mezi MICA alelami jsou synonymní substituce, daných 100 alel kóduje pouze 79 různých bílkovin. Dvě MICA alely jsou "null" (prázdné) alely, které nemají žádný bílkovinný produkt.[**13**] Kombinace mimobuněčných a TM typů usnadňuje identifikaci MICA alel a snižuje počet možných nejednoznačných typizací u heterozygotních jedinců.[**11**]

2 Genetický kód

Genetický kód odpovídá souboru pravidel, které používají živé buňky pro přenos genetické informace v podobě DNA (Deoxyribonukleová kyselina, dvoušroubovice) či mRNA (messenger "poslíček" ribonukleová kyselina, jednovláknová) na hlavní strukturu bílkovin, což je pořadí aminokyselin v řetězci (DNA).[**14**]

Tyto řetězce jsou tvořeny **nukleovými bázemi** (dále jen báze), které tvoří komplementární páry (značení "bp" = base pair) typu Guanin - Cytosin (G - C) a Adenin - Thymin /-Uracil (A - T /U).[**14**]

Veškerá genetická informace organismu se nazývá **genom** a je zapsána v podobě DNA (s výjimkou nebuněčných organismů - RNA). Každá část představující nějakou funkci se nazývá **gen**. Pro kopírování DNA řetězce se využívá schopnosti replikace DNA, kdy se podle DNA tvoří mRNA, která přenese genetickou informaci k ribozómům, na kterých probíhá translace - sestavení hlavní struktury bílkovin. Pořadí aminokyselin je stanoveno připojením tRNA (transferová RNA) ke každému kodonu (triplet bází, více kapitole 2.2 Kodon) s odpovídajícím antikodonem, který nese dané aminokyseliny.[**14**]

2.1 Části genomu

Lidský genom je z nějvětší části (kolem 75 %) tvořen intergenními oblastmi, zbylá část jsou introny a exony, z čehož exony tvoří nejmenší zastoupení.[**15**]

Exony jsou části genu, které označují jak DNA sekvenci v genu, tak odpovídající RNA sekvenci v transkripci, poté, co jsou odstraněny introny. Tyto oblasti kódují bílkovinné produkty. Všechny exony v genomu se nazývají exom. **Introny** jsou nekódující oblasti RNA transkripce a před DNA translací jsou tyto oblasti odstraněny. I když introny nekódují žádné bílkoviny, jsou nedílnou součástí regulace genové exprese. **Intergenní oblasti** se nachází mezi jednotlivými geny. Obsahují důležité části jako jsou promotery (specifická RNA sekvence bází, na kterou se navazuje RNA-polymeráza), nebo enhancery (sekvence DNA, která váže aktivační faktory). [**15**]

2.2 Kodon

Kodon jsou tři za sebou jdoucí báze v mRNA, **antikodon** je komplementární (tedy druhá báze z dané dvojice) trojice bází tRNA (transferová RNA) ke kodonu. Každý kodon kóduje nějakou aminokyselinu (viz tabulka v obrázku 2). Máme 64 (4^3) různých možných kombinací bází v kodonu. Genetický kód je ale degenerovaný (redundantní), jedna aminokyselina může odpovídat více kodonům. Tato degenerace způsobuje nemožnost zrekonstruování genu podle dané bílkoviny.[**16**]

Iniciační kodon, též jako "Start kodon", je rozeznáván ribozomem na mRNA jako počátek genu a od tohoto místa probíhá syntéza bílkovin neboli translace (sestavení hlavní struktury bílkovin podle mRNA). Ve většině případů se jedná o kodon typu ATG, což je aminokyselian methioninu. V 90 % případů je rozeznán první ATG kodon. **Terminační kodon** neboli "Stop kodon", pokud je rozeznán ribozomem, ukončuje translaci aktuální bílkoviny. Obykle jde o kodon typu TAA, TAG nebo TGA a nepatří k nim tRNA s antikodonem. Pokud je tedy tento kodon rozeznán ribozomem, naváže se na bázi A release faktor (= bílkovina podobná tRNA) a místo další aminokyseliny je vložena molekula vody. Tímto dojde k terminaci translace, uvolní se vzniklý polypeptid (= spojení 10 a více aminokyselin) a ribozom se rozpadne na dvě podjednotky.[**16**]

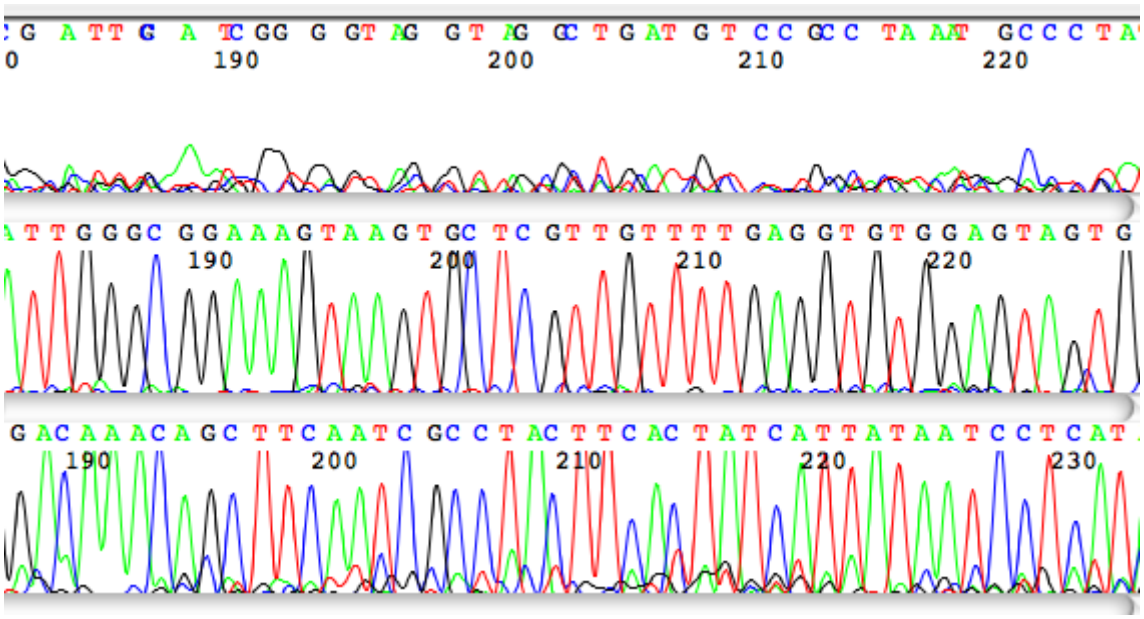
		Druhý nukleotid					
		U	C	A	G		
První nukleotid	U	UUU fenyalanin	UCU serin	UAU tyrosin	UGU cystein	U	Třetí nukleotid
		UUC fenyalanin	UCC serin	UAC tyrosin	UGC cystein	C	
		UUA leucin	UCA serin	UAA stop kodon	UGA stop kodon	A	
		UUG leucin	UCG serin	UAG stop kodon	UGG tryptofan	G	
	C	CUU leucin	CCU prolin	CAU histidin	CGU arginin	U	
		CUC leucin	CCC prolin	CAC histidin	CGC arginin	C	
		CUA leucin	CCA prolin	CAA glutamin	CGA arginin	A	
		CUG leucin	CCG prolin	CAG glutamin	CGG arginin	G	
	A	AUU isoleucin	ACU threonin	AAU asparagin	AGU serin	U	
		AUC isoleucin	ACC threonin	AAC asparagin	AGC serin	C	
		AUA isoleucin	ACA threonin	AAA lysin	AGA arginin	A	
		AUG methionin	ACG threonin	AAG lysin	AGG arginin	G	
	G	GUU valin	GCU alanin	GAU kyselina asparagová	GGU glycin	U	
		GUC valin	GCC alanin	GAC kyselina asparagová	GGC glycin	C	
		GUA valin	GCA alanin	GAA kyselina glutamová	GGA glycin	A	
		GUG valin	GCG alanin	GAG kyselina glutamová	GGG glycin	G	

Obrázek 1: Kodony a které bílkoviny představují (RNA báze) [**17**]

3 Metody sekvenování

3.1 Sangerova metoda sekvenování (1977)

Sangerovo sekvenování zahrnujeme do sekvenování první generace. Tuto metodu lze použít pro sekvenování krátké sekvence jednovlákonové DNA. K počátku DNA se naváže komplementární **primer**, který je 15 - 25 bp dlouhý. Od tohoto místa dochází k syntéze DNA pomocí čtyř deoxyribonukleotidů dNTP (dATP, dGTP, dCTP, dTTP) či čtyř dideoxynulkeotidů ddNTP (ddATP, ddGTP, ddCTP, ddTTP). Tento ddNTP se začlení do DNA, ale jelikož nemá OH skupinu (OH = hydroxyl, chemická skupina obsahující jeden atom vodíku a jeden atom kyslíku) jako dNTP, zastaví se syntéza. Malé množství ddNTP zaručí náhodnost připojení k DNA řetězci.[**18**]



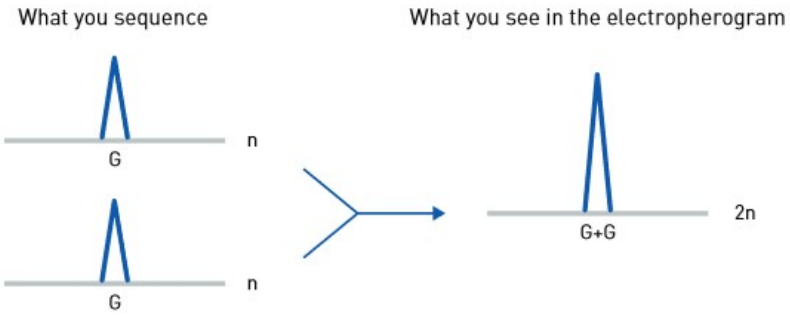
Obrázek 2: Chromatogramy ze Sangerova sekvenování různé kvality. Horní sekvence je strojově přečtena ale její kvalita způsobí nepřesné informace o sekvenci. Prostřední sekvence je jasně čitelná, toto je optimální kvalita signálu. Spodní sekvence má v celé délce rušivý signál, musíme být tedy opatrní, ale kvalita je stále akceptovatelná. [**19**]

Dnes se používá fluorescenčně značených ddNTP, kdy každý typ má svoji barvu, a díky tomu může reakce probíhat v jedné zkumavce. Jednotlivé fragmenty jsou poté seřazeny kapilární elektroforézou (separace látek na základě jejich rozdílné pohyblivosti v elektrickém poli).[**18**]

3.1.1 Homozygocie a heterozygocie

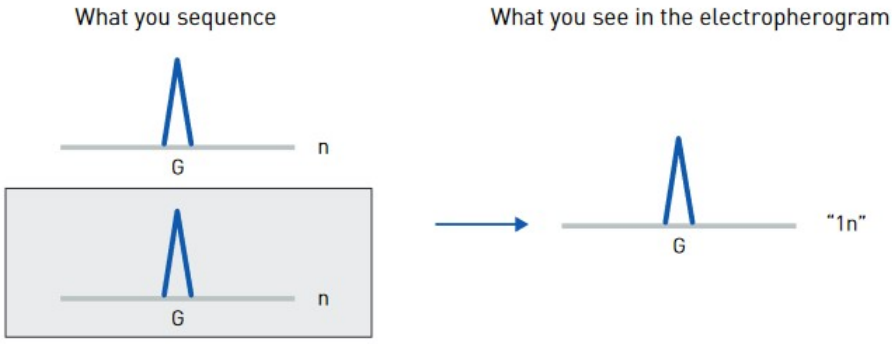
Genetická informace uložená v DNA obsahuje zjednodušeně soubor pokynů, které se buď projeví nebo neprojeví jako znaky jedince. Protože máme páry stejných chomozomů, každý znak je tak tvořen dvěma alelami. Pokud jsou tyto alely (pro jeden daný gen) stejného typu, je daný jedinec homozygot (pro daný gen). Pokud jsou alely rozdílné, jedná se o heterozygotního jedince.

DNA "basecalling" programy analyzují fluorescentní signály ze Sangerovo sekvenování a odhalují tak typ vzorku a jeho kvalitu (viz Phred skóre), což ukazuje spolehlivost "basecallu". V klasickém sekvenování založeném na PCR jsou u homologního genu (geny odvozené ze společného genu) obě kopie alely sekvenovány současně. Ve srovnání s hypotetickým signálem n z jedné alely, je sledovaný signál ve skutečnosti součtem signálů z obou alel, tedy $2n$. [**20**]



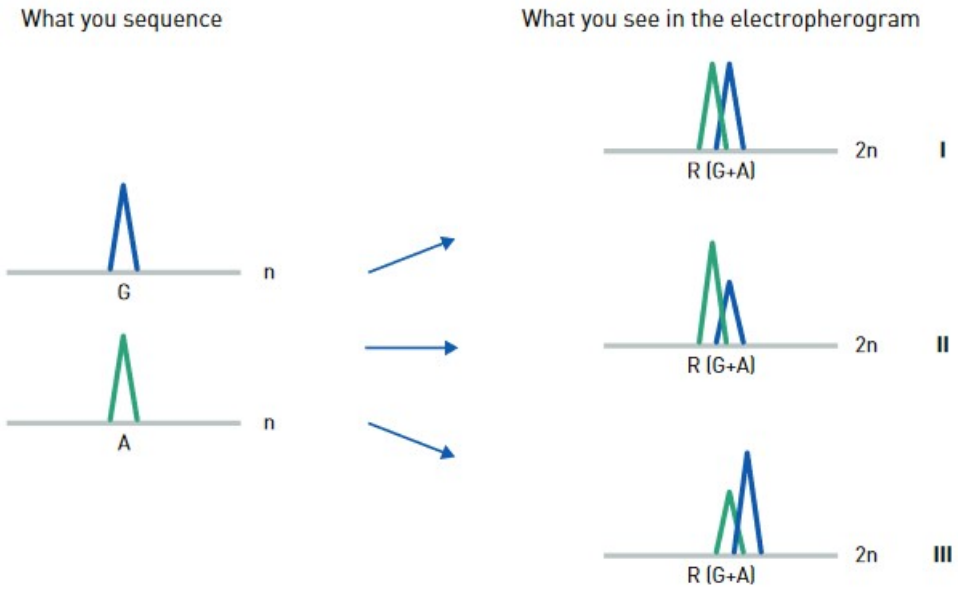
Obrázek 3: **Homozygot**. Výsledný signál je součtem signálů z jednotlivých alel. Vlevo jsou signály z jednotlivých alel, vpravo výsledný signál - jejich součet. [**20**]

Jeden peak (hrot, signál) v chromatogramu, který reprezentuje homozygotní bázi, je součet dvou identických signálů bází, kde každý je přibližně polovina fluorescentního signálu v relativních fluorescentních jednotkách (RFU) výšky výsledného signálu. Ztráta signálu jedné alely tedy představuje snížení signálu na přibližně polovinu.



Obrázek 4: **Homozygot** - ztráta alely. Výsledný pík je přibližně polovina očekávaného signálu. [**20**]

V případě heterozygota výsledné peaky páru se pohybují ve stejné nebo podobné poloze jako smíšená báze. Síla signálu každé báze je přibližně polovina síly výsledného signálu u homozygota. V ideálním případě jsou oba signály stejně vysoké, ale nemusí tomu tak být vždy, tedy jeden signál bude výškou přesahovat druhý.



Obrázek 5: **Heterozygot**. Vlevo jsou signály z jednotlivých (rozdílných) alel, vpravo možné výsledné výstupy. Výstup I je ideální případ. Výstupy II a III jsou realita. [**20**]

Tato nerovnováha výšek komplikuje stanovení jejich poměrů.[**20**]

Base calling je proces přiřazování báze podle maxim z chromatogramu (graf intenzity světla daného fluorescenčně označeného konce při sekvenování).[**21**]

3.2 Sekvenování nové generace

Metody sekvenování nové generace (Next-generation sequencing, dále jen NGS) umožňují levnější, rychlejší a rozsáhlejší čtení dat, jelikož využívá fragmentace genomu - tedy rozdělení do menších úseků dlouhých několik set bází. Protože se ale genom fragmentuje, je nutné správně určit lokalitu těchto úseků v genomu. Ve většině případů se používá referenční genom stejného/příbuzného druhu - dříve osekvenovaný genom - který již máme zmapovaný. Fragmenty se namnoží **PCR** reakcí, která funguje na principu replikace nukleových kyselin, a tyto kopie jsou dále paralelně sekvenovány. Paralelizací sekvenování dosahujeme čtení až milionů vláken DNA najednou. Jsme tak schopni osekvenovat celý genom najednou.[**22**]

Celý tento proces je možno redukovat tím, že se sekvenují jen jeho určité části. Hovoří se tak o "redukované reprezentaci genomu". Této reprezentace se docílí například použitím restrikčních enzymů, které naštěpí DNA sekvence na přesně daném místě a sekvenuje se pouze oblast do určité vzdálenosti od místa štěpení, můžou se také sekvenovat pouze části, které jsou překládány do RNA, či se použije "sequence capture", kdy se přesně určí sekvence, které se mají sekvenovat, například konkrétní gen.[**22**]

PCR (Polymerase Chain Reaction) slouží k vytvoření až milionů kopií daného fragmentu DNA, který může být až 10 000 nukleotidů dlouhý. Vzorový úsek DNA musí být ohraničený tvz. primery. Primer je řetězec dlouhý několik bází (resp. aminokyselin) a slouží jako počáteční místo replikace.[**23**]

3.3 Rozdíl mezi Sangerovo sekvenací a NGS

Jeden z hlavních rozdílů je, jaké množství fragmentů jsou metody schopné sekvenovat. Sangerovo sekvenování dokáže sekvenovat pouze jeden DNA fragment, zatímco NGS metody jsou schopné sekvenovat miliony fragmentů najednou. Sangerovo sekvenace je efektivní pro "variant screening studies", když je počet vzorků malý a zkoumáme varianty jediného genu.[**24**]

Část II. - Identifikace a analýza variant genu

4 Získávání dat

4.1 Čtení biologické sekvence

Identifikace (predikce, hledání) genu je proces hledání určitých oblastí DNA, které odpovídají danému genu. Díky této identifikaci můžeme lépe pochopit nejen lidský genom.

Čtecí rámec (RF; Reading Frame) je způsob rozdělení sekvence nukleotidů v DNA nebo RNA na po sobě jdoucí trojice bází, které se nepřekrývají. Pokud tyto trojice kódují aminokyseliny nebo slouží jako stop signály během translace, označují se jako kodony.[**25**] Více o čtecím rámci v příloze B ORF - Otevřený čtecí rámec.

Jednořetězcová nukleová kyselina má jeden fosforylový konec, zvaný **5'-end**, a jeden hydroxylový konec, zvaný **3'-end**. Tyto konce definují směr čtení sekvence. V daném směru můžeme sekvenci číst ve třech čtecích rámcích, každý z nich začínající rozdílnou bází z jedné trojice. V DNA dvoušroubovici, kde jsou jednotlivé řetězce k sobě komplementární, odpovídá směr druhého řetězce 5'→3' směru 3'→5'. Pokud bychom z jednoho směru čtení chtěli vytvořit druhý, děláme tzv. **reverzní komplement**.[**25**]

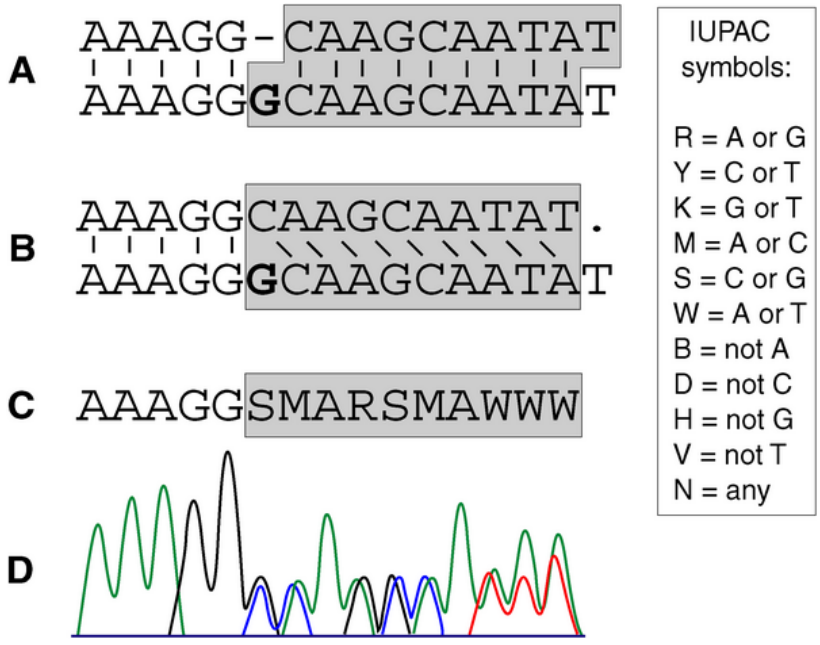
4.2 Hledání homologie

Většina dnešních programů pro lokalizaci genu pracují až s 95% úspěšností. Problémem ale zůstává špatná indetifikace hranic mezi introny a exony a tím vznikne identifikace falešného RF pro daný gen. Identifikaci můžeme posílit tím, že budeme využívat hledání homologie k testování, zda je řada kodonů skutečný exon nebo pouze náhodná sekvence. V tomto případě se prohledávají DNA databáze a zjišťuje se, zda je daná sekvence indentická nebo podobná jakémukoli genu, který byl již sekvenován. Samozřejmě pokud je testovaná sekvence součástí genu, který byl již sekvenován, bude nalezena identická shoda. To ale není důvod hledání

homologie. Záměr je zjistit, zda je nová testovaná sekvence podobná jakémukoli známému genu. Pokud se najdou podobnosti, mohou být sekvence homologní - geny mohou být evolučně příbuzné.[**26**]

4.3 Sestavení sekvence

Pro jednotlivá čtení se určí jejich směr (F/R - Foward/Reverse - Dopředné/Zpětné) a seřadí se tak, aby sekvence maximálně vzájemně odpovídala. Takto sestavená sekvence se nazývá **kontig** (z angl. contig). Po případných ručních opravách kontigu se sestaví sekvence z nejčastěji se vyskytujících nukleotidových bází v jednotlivých pozicích. Tyto sekvence se nazývají **konsenzuální sekvence**. Kontig je sestaven vždy ze čtení jednoho jedince a jeho výsledkem by měla být reálně existující sekvence DNA. Porovnávání více sekvencí se nazývá *alignment* (dále jako zarovnávání, porovnávání apod.). [**27**]



Obrázek 6: A: zarovnané sekvence; B: nezarovnané sekvence; C: vytvořená konsenzuální sekvence dle uvedené legendy; D: vzhled chromatogramu. [**28**]

V ideálním případě, pokud je kontig sestaven pouze z bází A, G, T a C (signály bází se v chromatogramu nepřekrávají), jedná se o homozygotní genotyp, pokud je však kontig sestaven i ze znaků, které označují více možných bází (v chromatogramu se signály bází překrývají), jedná se o genotyp heterozygota. V reálných datech se můžeme setkat s chybou či mutací.

4.4 Formát dat FASTQ, FASTA a další

FASTQ je textový formát pro ukládání nejčastěji nukleotidové sekvence či skóre kvality sekvence. Písmena sekvence i skóre kvality jsou kvůli stručnosti zakódovány jedním znakem ASCII. Formát běžně využívá 4 řádky pro jednu sekvenci: první řádek začíná znakem "@" a za ním je zapsán identifikátor sekvence a volitelný popis; druhý řádek obsahuje čistou sekvenci zapsanou písmeny (G, A, T, C); třetí řádek obsahuje pouze znak "+" a za ním může být zapsán identifikátor sekvence či jiný popis; čtvrtý řádek kóduje znakem ASCII hodnoty kvality sekvence na druhém řádku (na druhém a čtvrtém řádku je tedy stejný počet znaků).[**29**]

```
@SRR014849.1 EIXKN4201CFU84 length=93
GGGGGGGGGGGGGGGCTTTTTTTGTTTGGAACCGAAAGG
GTTTTGAATTTCAAACCCTTTTCGGTTTCCAACCTTCCAA
AGCAATGCCAATA
+SRR014849.1 EIXKN4201CFU84 length=93
3+&$#""""""""""""7F@71,'";C?,B;?6B;:EA1EA
1EA5'9B:?:#9EA0D@2EA5':>5?:%A;A8A;?9B;D@
/=<?7=9<2A8==
```

@title and optional description
sequence line(s)
+optional repeat of title line
quality line(s)

Obrázek 7: Příklad FASTQ souboru. [**29**]

FASTA je textový formát, který reprezentuje nukleotidové sekvence nebo proteinové sekvence pomocí jednopísmenných kódů. Je umožněno, aby před sekvencemi byly názvy sekvencí či komentáře. Informace mohou být uloženy jako textový soubor, pokud je formátovaný jako *fasta* soubor.[**30**] Formát FASTQ slouží tedy k uložení fragmentů sekvence před mapováním. FASTA slouží k uložení referenčního genomu, na který budou mapovány fragmenty sekvence. V práci se bude pracovat se soubory ve formátu FASTA (dále jako *fasta*).

SAM formát (**S**equence **A**lignment **M**ap) je textový formát sloužící k uložení nukleotidových sekvencí generovaných pomocí NGS a zahrnuje nyní i nezmapované sekvence. Tento formát podporuje krátké i dlouhé čtecí sekvence (až 128 Mbp). **BAM** formát (**B**inary **A**lignment **M**ap) je binární ekvivalent formátu SAM. **CRAM** je komprimovaný sloupcový formát souboru pro ukládání sekvencí seřazených podle referenční sekvence. **VCF** (Variant Calling Format) je textový

formát který obsahuje meta-informační řádky, záhlaví a datové řádky, z nichž každý obsahuje informace o poloze genomu. Formát je schopný také obsahovat informace o genotypu na vzorcích pro každou pozici.[**31**]

Variant calling je proces, kterým identifikujeme varianty sekvencí genů ze sekvenačních dat. Nejdříve se sekvenováním genomu nebo exomu vytvoří soubory typu FASTQ (NGS) či ab1 (Sangerovo sekvenování). Poté se zarovnají sekvence s referenčním genomem a vytvoří se soubory typu BAM nebo CRAM. V posledním kroku se určí, kde se zarovnané kusy sekvence DNA (reads = ready) liší od referenčního genomu (exomu) a zapíšou se do souboru typu VCF. [**32**]

4.4.1 Kvalita dat FASTQ

Hodnota kvality Q je celočíselné mapování p, tedy pravděpodobnost, že odpovídající báze z *Base calling* je nesprávná. Používají se dva různé vzorce - následující dvě varianty jsou pro metodu Sanger **"Skóre kvality Phred"**: [**33**]

$$Q_{Sanger} = -10 \log_{10} P$$

$$P = 10^{\frac{-Q}{10}}$$

Vzorce nám tedy prozrazují následující tabulku:

Skóre kvality Phred Q	Pravděpodobnost nesprávné báze	Přesnost určení báze
10	1 z 10	90%
20	1 z 100	99%
30	1 z 1 000	99.9%
40	1 z 10 000	99.99%
50	1 z 100 000	99.999%
60	1 z 1 000 000	99.9999%

Tabulka 1: Hodnoty přesnosti podle skóre kvality Phred [**33**]

Skóre kvality Phred se používá pro hodnocení kvality sekvence, rozpoznávání a odstraňování sekvence nízké kvality (ořezávání konce) a stanovení přesných

konsenzuálních sekvencí (charakteristické sekvence bází RNA/DNA, které jsou společné většímu počtu genů příbuzných funkcí nebo vykazující určitou evoluční homologii; viz kapitola 4.3 Sestavení sekvence). Míra podobnosti dvou konsenzuálních sekvencí je mírou jejich evoluční příbuznosti. To, že se určité báze vyskytují ve stejných pozicích, většinou odpovídá jejich biologickému významu.[**34**]

V začátcích bylo skóre kvality Phred používáno hlavně v programu Phrap, který sestavoval sekvence. Phrap se hojně používal v projektu "Lidský genom" a je nyní jeden z nejpoužívanějším programem v biotechnologickém průmyslu. Před Phred a Phrap musely být manuálně zkoumány nesrovnalosti mezi překrývajícími se fragmenty DNA a to často zahrnovalo ruční stanovení sekvence nejvyšší kvality a ruční úpravy chyb. Phrap, který využíval skóre kvality Phred, efektivně automatizoval hledání konsenzuálních sekvencí nejvyšší kvality, což ve většině případů eliminuje nutnost ručních úprav. Tato automatizace zredukovala i chybovost v sestavování sekvencí oproti manuálním úpravám.[**29**]

Skóre kvality se nejčastěji ukládají se sekvencemi ve formátu FASTQ. Před kompresí představují data o kvalitě přibližně polovinu požadovaného úložného místa, díky kompresi těchto dat se zmenší nároky na paměť a urychlí se analýza a přenos dat.[**29**]

Formát *fastq/fasta* pro Sanger data umí skóre kvality Phred zakódovat od hodnoty 0 do 93 pomocí ASCII od 33 do 126. V nezpracovaných datech ale skóre kvality Phred zřídka překračuje hodnotu 60, vyšší hodnoty jsou možné v sestavách sekvencí či genových mapách.[**29**] Výstupní formát dat ze Sangerovo sekvenování je *ab1* (chromatogram), které lze přeformátovat na soubor *fastq* či *fasta*.

4.5 Dostupné datové zdroje

S rostoucím počtem bioinformatických dat je potřeba tato data ukládat, sdílet a organizovat. V důsledku toho počet online databází každoročně rapidně roste.[**35**]

Kolaborace mezinárodní databáze nukleových sekvencí (INSDC; International Nucleotide Sequence Database Collaboration) zahrnuje spolupráci DNA Databanky Japonska (DDBJ, DNA Databank of Japan [**36**]), Evropského institutu pro bioinformatiku Evropské laboratoře molekulární biologie (EMBL-EBI, European Molecular Biology Laboratory's European Bioinformatics Institute [**37**]) a Národního centra pro biotechnologické informace (NCBI, National Center for Biotechnology Information [**38**]). Tato trojice od roku 1988[**39**] zachycuje, uchovává a předkládá trvalý vědecký záznam pro sekvenování nukleových kyselin a související informace. Díky této spolupráci rozsah aktivit sekvenování enormně vzrostl a INSDC vytvořila mandát, který dominuje v obsahu biotechnologických dat. Zahrnuje také úložiště a služby pro nezpracovaná data, detaily experimentálního návrhu, funkční anotace či informace o projektu Lidský genom (Human Genome Project). [**35**]

Existují také specializované databáze jako například WormBase, která uchovává data hlístic, či RiceWiki, která obsahuje data genů rýže.[**39**]

Projekt NCBI zahrnuje databázi GenBank. Do projektu EMBL-EBI spadá Databáze imunopolymorfismů (IPD, Immuno Polymorphism Database [**40**]), která bude sloužit jako zdroj bioinformací v této práci.

4.5.1 Immuno Polymorphism Database IPD

Databáze imunopolymorfismů (dále jen IPD [**40**]) byla vytvořena v roce 2003 za účelem vytvoření centralizovaného systému pro studium polymorfismu v genech imunitního systému. Tento projekt byl založen Skupinou HLA pro informatiku Výzkumného ústavu Anthonyho Nolana (HLA informatics Group of the Anthony Nolan
Research Institute) v kolaboraci s Evropským institutem bioinformatiky (European Bioinformatics Institute).

Sekce **IPD-IMGT/HLA** [**41**] poskytuje specializovanou databázi sekvencí lidského hlavního histokompatibilního komplexu (MHC) a obsahuje oficiální sekvence pojmenované Výborem pro nomenklaturu WHO pro faktory systému HLA (WHO Nomenclature Committee For Factors of the HLA System). Tato databáze je součástí mezinárodního ImMunoGeneTics projektu (IMGT).

IPD dále obsahuje sekce KIR (Killer-cell immunoglobulin-like receptor) či MHC (Hlavní histokompatibilní komplex) pro různé typy živočichů - např. pes, ryba, prase či myš. [**41**]

Na stránkách IPD najdeme seznam oficiálních referenčních sekvencí genů HLA. Mezi nimi najdeme i non-HLA geny MICA a MICB (ostatní ze skupiny MIC nejsou zmíněny, jelikož se jedná o pseudogeny bez známé funkcionality).

5 Metoda pro identifikaci alel

5.1 Problémy alignmentu a identifikace

Nejdůležitější při identifikaci genů je způsob přístupu k datům - sekvencím. Každá báze a její pozice má svůj biologický význam. Nelze tedy s biologickými sekvencemi zacházet jen jako s řetězci znaků.

5.1.1 Hledání slov

Na obrázku 8 vidíme dva příklady zarovnaných sekvencí. Každé zarovnání má 5 pozic, na kterých se báze liší, chybovost/odlišnost je tedy stejná pro oba příklady. Z evolučního hlediska ale dává zarovnání vlevo větší smysl, a to z důvodu potřeby menšího počtu změn *subsekvencí* (kratší úsek sekvence) pro vytvoření shodné sekvence. V levé horní sekvenci stačí udělat pouze dvě (subsekvenční) změny (GGT a TG) ale v pravé horní sekvenci musíme udělat změn pět (C, G, G, T a C). Proto by měly algoritmy pracovat na principu hledání *slov*, tedy *k-mers* (k značí délku slova). Základním předpokladem pro dvě příbuzné sekvence je alespoň jedno společné slovo.[**42**]

Obrázek 8: Sekvence se stejným počtem chyb.

5.1.2 Různá váha chyb

Dalším problémem při identifikaci alel je záměna bází v zarovnání. Chyba může být způsobená sekvenátorem, který chybně vyhodnotí výslednou bázi, nebo došlo k mutaci genu. Pokud došlo k mutaci, mělo by se k této odlišnosti přistupovat jinak. Jinými slovy: (obrázek 8) v sekvenci vpravo může záměna na druhé pozici C - A mít jinou biologickou složitost než záměna na osmé pozici T - A, i když se

jedná o podobnou záměnu (stejnou chybu). Je to z důvodu chemické struktury aminokyselin, které jsou zakódovány jednotlivými kodony. Každá záměna aminokyseliny má své skóre. [**42**] Používají se dvě skórové matice: BLOSUM či PAM - více v příloze C Problémy identifikace.

5.1.3 Odlišnost alel

Dalším z problémů je také míra odlišnosti jednotlivých alel genu. Alely se mohou lišit například pouze v jednom exonu, a v tom to exonu se mohou lišit pouze jednou bází. V kombinaci s chybovostí přístrojů to značně ztěžuje přesnou identifikaci alely.

5.1.4 Heterozygocie

V neposlední řadě ztěžují identifikaci heterozygotní geny. Abychom mohli určit, o které dvě alely se jedná, musíme heterozygotní sekvenci přepsat na kombinace sekvencí s homozygotními znaky. Počet kombinací je závislý na počtu heterozygotních znaků. Počet kombinací roste funkcí 2^n, kde n je počet heterozygotních znaků.

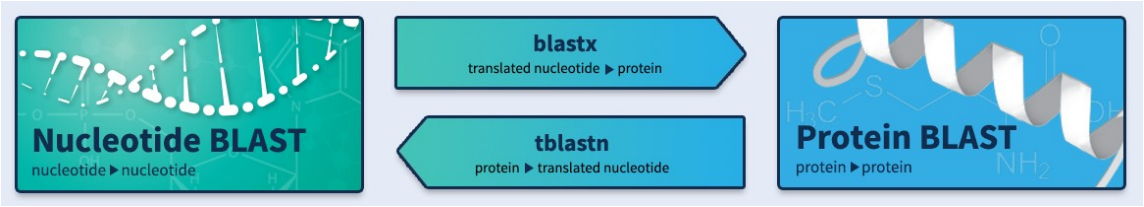
5.1.5 Využívané nástroje

Programy, které na těchto principech fungují, jsou **FASTA**[**43**] (FAST Alignment) a **BLAST**[**44**] (Basic Local Alignment Search Tool). Hlavním rozdílem mezi FASTA a BLAST je to, že BLAST se ve většině případů používá při hledání neuzavřeného lokálně optimálního zarovnání sekvence, zatímco FASTA se využívá spíše při hledání podobností mezi méně podobnými sekvencemi.

BLAST je jeden z nejpoužívanějších algoritmů/programů používaný pro srovnávání primárních sekvenčních informací. Je založen na základě FASTA. Algoritmus dokáže srovnat dotazovanou (zadanou) sekvenci se sekvencemi v databázi a zároveň rozpozná podobné sekvence v rámci hranice podobnosti, to vše za účelem hledání podobnosti až možnosti homologie. Cílem je tedy najít mezi souvisejícími sekvencemi, které nemají vysoké skóre, segmenty s vysokým skórem. Existence takových segmentů

nad danou prahovou hodnotou naznačuje párovou podobnost bez šance náhodnosti, což pomáhá rozlišit příbuzné sekvence od nepříbuzných sekvencí v databázi (či dvou zadaných sekvencí). BLAST je populární díky své schopnosti rychle identifikovat regiony s lokálními podobnostmi mezi dvěma sekvencemi. Program využívá skórové matice BLOSUM.[**42**]

BLAST je volně dostupný na stránkách NCBI [**44**] ve variantách **BLAST-N**, který porovnává nukleotidové sekvence, **BLAST-P**, který porovnává proteinové sekvence, **BLAST-X**, který porovnává nukleotidovou sekvenci s proteinovou sekvencí a **tBLAST-N**, který porovnává proteinovou sekvenci s nukleotidovou sekvencí.



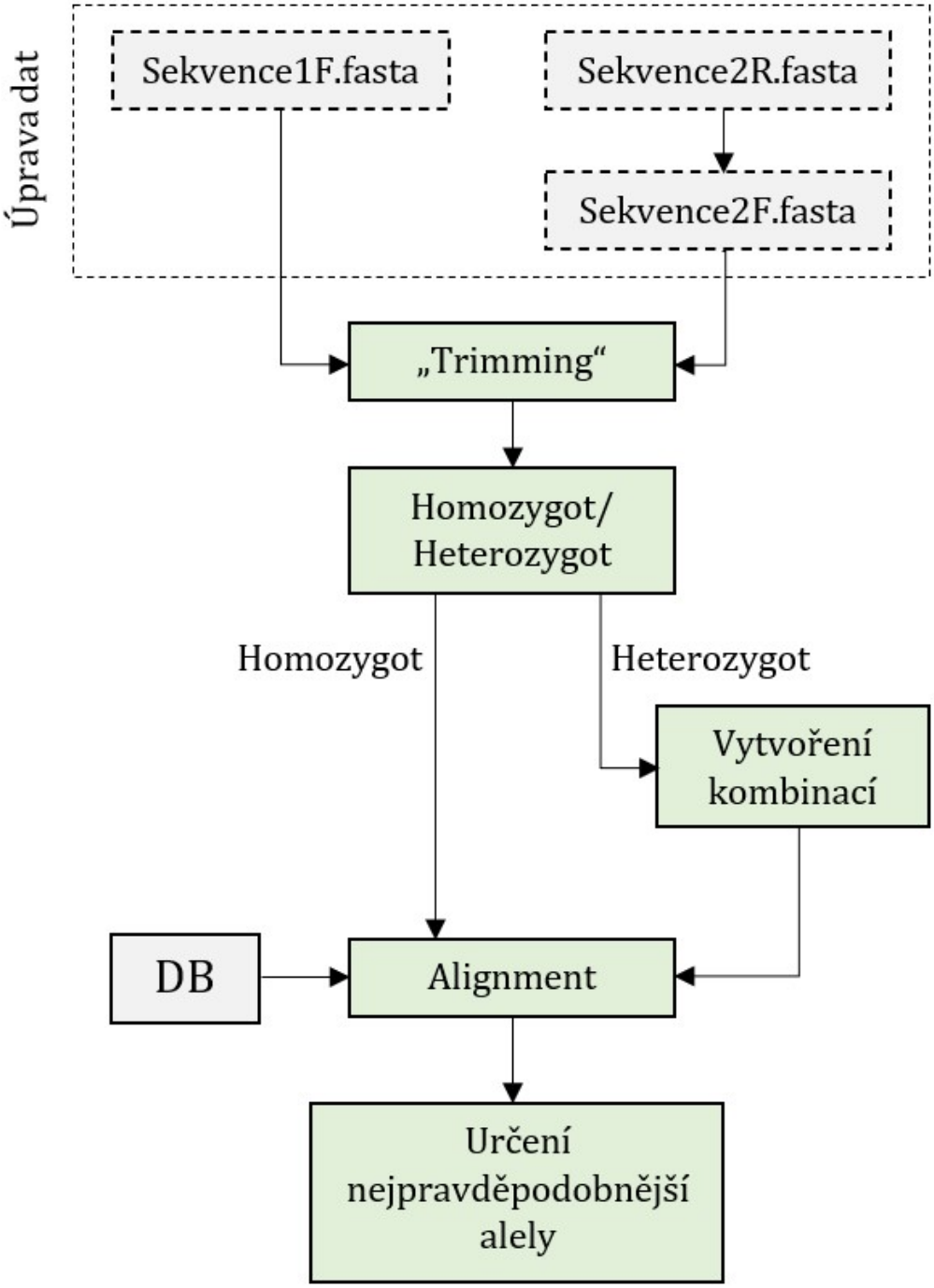
Obrázek 9: NCBI Web BLAST [**44**]

5.2 Metoda pro identifikaci

Návrh metodiky vycházel z požadavku využití metody offline a z formátu dat získávaných ve FN Plzeň v rámci projektu Ministerstva zdravotnictví ČR - Agentury pro zdravotnický výzkum č. NV18-03-00277.

Prvním krokem je správná **úprava vstupních dat**. Pro jeden exon máme 2 vstupní soubory (2 sekvence) - dopředné čtení (foward) 3'→5' a zpětné čtení (reverse) 5'→3' (více kapitola 4 Čtení biologické sekvence). To znamená nutnost úpravy zpětného čtení na jeho *reverzní komplement*. Upravené soubory poté vstoupí do fáze **trimming**, ve které najdeme nejpodobnější (nemusí být nutně uplně stejná) subsekvenci mezi dopřednou sekvencí a reverzním komplementem zpětné sekvence, a tím určíme nejpravděpodobnější polohu exonu a dle zvolených parametrů ořízneme dopřednou sekvenci. Z této sekvence vyhodnotíme, zda-li se jedná o **homozygotní** či **heterozygotní** sekvenci. Pokud se jedná o homozygota, můžeme ihned začít **alignment** (zarovnávání) s referenčními exony z databáze (DB), pokud se ale jedná o heterozygota, je nutné vytvořit všechny možné kombinace sekvencí a až poté můžeme přejít k **alignmentu** a určení nejpravděpodobnější alely (dvojice alel v případě heterozygotní sekvence). Nejpravděpodobnější alela (dvojice alel) se určí nejvyšší hodnotou podobnosti k referenčním alelám.

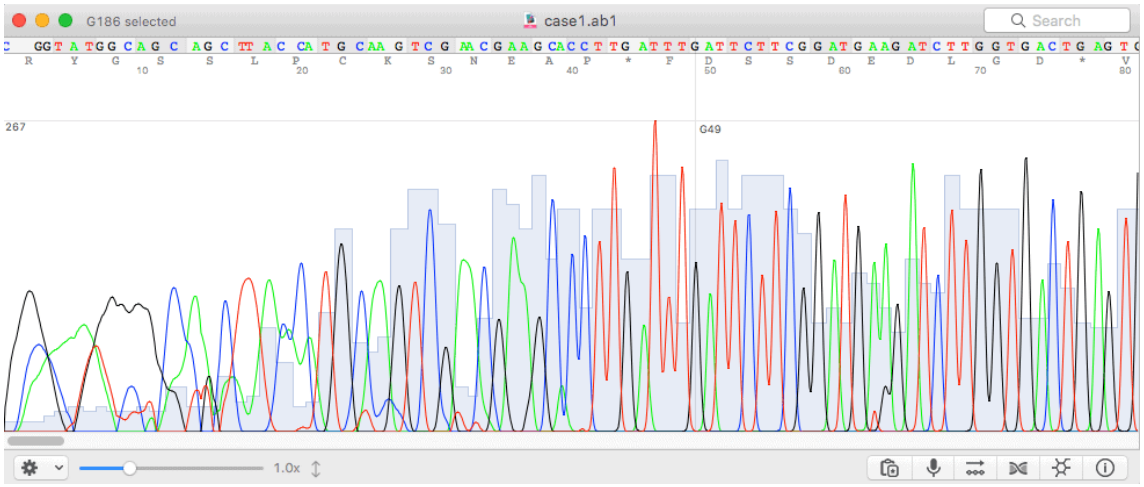
Na obrázku 10 je zobrazený postup řešení identifikace.



Obrázek 10: Postup identifikační metody. *Sekvence1F* značí dopřednou sekvenci, *Sekvence2R* značí zpětnou sekvenci, *Sekvence2F* značí reverzní komplement k sekvenci *Sekvence2R*.

5.2.1 Reálná data

Metoda je navrhnuta pro soubory, které mají podobné vlastnosti jako reálná data získávaná z FN Plzeň. Pro MICA gen se získávají data pro exony 2, 3 a 4, pro MICB se získávají data pro exony 2, 3, 4 a 5. Pro každý exon existují dva soubory - dopředné a zpětné čtení. Pro gen MICB jsou exony 4 a 5 v jednom souboru. Data obsahují sekvence intronů a daný exon na neznámém místě. Pro MICB exony 4 a 5 soubory vypadají data *intron - exon 4 - intron - exon 5 - intron*.



Obrázek 11: Data ve formátu *ab1*. V levé části vidíme nekvalitní data. Nízká kvalita dat se vyskytuje nejčasteji v částech s introny. [**49**]

Jak již bylo zmíněno, přesné určení alely ze vstupních dat může selhat na jejich chybovosti. Reálná data jsou získaná Sangerovo sekvenováním ve FN Plzeň. Chybovost této metody může při správných podmínkách dosahovat až hodnoty 0.001 % [**44**]. Reálná data jsou ve formátu *ab1*, tedy data, která vyjdou přímo ze sekvenátoru. Biopython obsahuje funkci, která umí přečíst *ab1* soubory a zapsat je do souboru ve formátu *fasta*.

5.3 Implementace metody

Program byl napsán a spuštěn na počítači s operačním systémem Windows 10, s 8 GB operační paměti a s CPU Intel Core i7 8th gen. 1.9 GHz. Rozhodla jsem se použít pouze možnosti jazyka Python, a to ze dvou důvodů. Zaprvé, program bude spustitelný na více operačních systémech, a zadruhé, z důvodu nedostatku znalostí s operačním systémem Linux, pro který je tvořena většina bioinformatických softwarů pro práci se sekvencemi. Řešení spadá do kategorie dynamického programování. Program není paměťově náročný, odhad maximálního využití operační paměti je 2 GB, maximálního zatížení procesoru je přibližně 30 %. Podmínka návrhu metody je použití offline nástrojů.

Biopython [**46**] je sada volně dostupných nástrojů pro biologické výpočty napsaných v programovacím jazyce Python mezinárodním týmem vývojářů. Obsahuje třídy představující biologické sekvence a anotace sekvencí a je schopen pracovat s různými formáty souborů, které se v bioinformatice používají (např. formát *fasta*, *ab1*). Umožňuje také programové prostředky pro přístup k online databázím biologických informací, jako je např. databáze NCBI[**58**]. Další moduly rozšiřují schopnosti Biopythonu o zarovnávání sekvencí (alignment), překládání sekvencí (nukleotidy - proteiny), populační genetika a další. Licence Biopythonu je kompatibilní téměř se všemi bioinformatickými software licencemi, je tedy možné Biopython používat v řadě projektů.

Požadavky pro běh Biopythonu ve verzi 1.79 jsou Python ve verzi 3.6, 3.7 či 3.8. (použitá verze 3.9.6) a NumPy (Numerical Python). Možné rozšiřující moduly jsou NCBI Standalone BLAST (online nástroj), pairwise2, ClustalW či EMBOSS.

Jako rozšuřující knihovny jsem použila SeqIO (součástí Biopythonu), difflib a NumPy.

Knihovna **SeqIO** umí přečíst soubor ve formátu *fasta*, vytvořit objekt typu *Seq* a rozdělit informace v souboru do příslušných "vlastností" sekvence - ID/název sekvence a seq jako samotnou sekvenci posloupnosti znaků (A, G, T, C, R, Y, K, M, S, W, B, D, H, V, N - viz obrázek 6 v kapitole 4.3 Sestavení sekvence). Výhoda této knihovny je, že biologické sekvence se chovají jako *Seq* objekty, což umožňuje použití spousty biologických metod/operací, které obyčejný typ *String*

neumožňuje. Sekvenci typu Seq je možné uchovávat také jako MutableSeq objekt, který umožňuje editaci sekvence (např. přidávání či odebírání znaků), což nám klasický *Seq* objekt neumožňuje. Ale nemožnost editace není brána jako nevýhoda, spíše jako jistota, že se prací se sekvencí neprovede nechtěná editace. S objektem typu Seq můžeme provádět klasické biologické operace jako je počet znaků *len(sekvence)*, vytvoření komplementární sekvence *sekvence.complement()*, vytvoření reverzního komplementu *sekvence.reverse_complement()* či překladu do proteinové sekvence *sekvence.translate()*.

Funkce, která se bude využívat pro všechna porovnávání v metodě, je **SequenceMatcher** z knihovny **difflib**[**47**]. Používá se pro porovnání dvojice sekvencí znaků (slova, biologické sekvence), pokud jsou znaky sekvence *hashovatelné*. Myšlenkou je nalezení největší kontinuální společné subsekvence, která neobsahuje chyby. V principu se jedná o lepší verzi algoritmu **Gestalt Pattern Matching**. Algoritmus nevrací výsledek s minimálními úpravami sekvence (jako například Levenshteinova vzdálenost), ale má tendenci vracet shody, které lidsky "vypadají dobře". Vzhledem k tomu, že *SequenceMatcher* určuje skóre shody dvou sekvencí na základě podobností delších subsekvencí, vybrala jsem tuto funkci pro všechna porovnávání, která v metodě probíhají. *SequenceMatcher* umí funkcí "*.ratio()*", která vrací číslo typu *float* v intervalu [0; 1], popsat podobnost dvou sekvencí. Pokud vrátí číslo větší než 0.6 včetně, sekvence jsou podobné, pokud vrátí číslo 1.0, sekvence jsou stejné. Aby *SequenceMatcher* uměl pracovat i s chybami v sekvencích, je potřeba zadat pro '*junk*' parametr hodnotu *None*. Složitost této funkce je v nejhorším případě kvadratická, předpokládaná složitost je lineární.[**47**] Volání funkce vypadá následovně:

SequenceMatcher(None, sekvence1, sekvence2).ratio().

5.3.1 Úprava dat

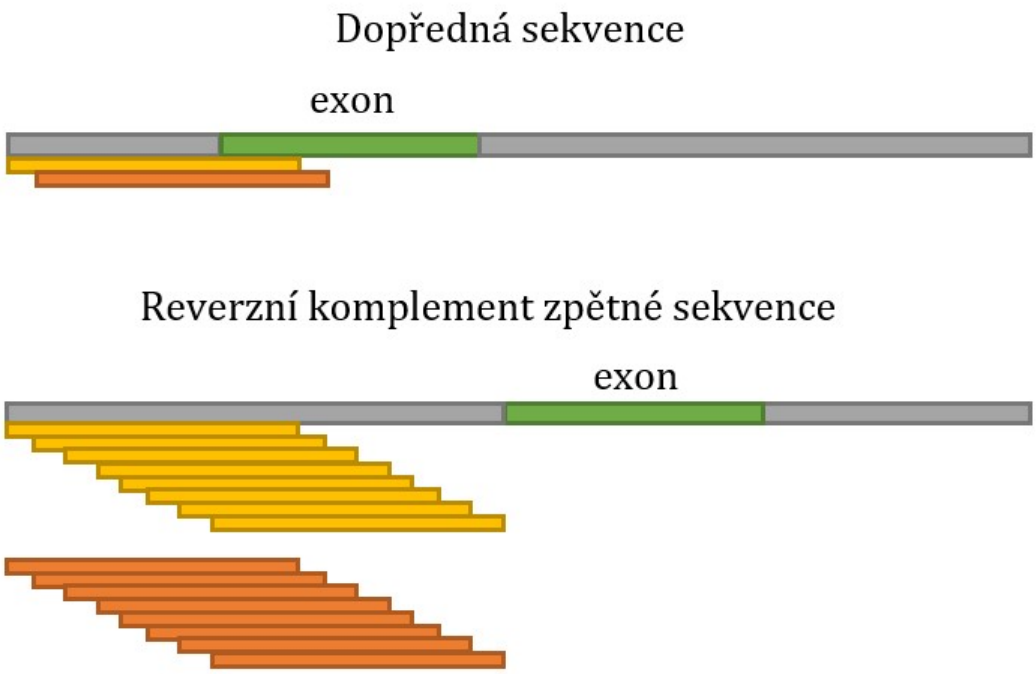
Prvním krokem je úprava vstupních dat. V knihovně *SeqIO* je funkce *SeqIO.parse()*, do které vložíme dva parametry - název souboru a jaký typ souboru to je, aby věděla jak soubor číst. Funkce data v souboru uloží jako objekt *Seq* a k jeho atributům, jako je *id* či samotná sekvence, lze přistupovat jednotlivě - např. *Seq.id* nám vrátí *id* sekvence, *Seq.seq* nám vrátí sekvenci znaků a *Seq.description* nám vrátí popis sekvence (často se v souboru nachází hned za *id* rozdělené pouze mezerou).

```
>HLA:HLA01013 MICA*001 1152 bp
ATGGGGCTGGGCCCGGTCTTCCTGCTTCTGG
```

Obrázek 12: Příklad formátování *fasta* souboru.První řádek: ">HLA:HLA01013" je *id* sekvence; "MICA*001 1152 bp" je popis sekvence (*description*); druhý řádek obsahuje samotnou sekvenci.

5.3.2 Trimming

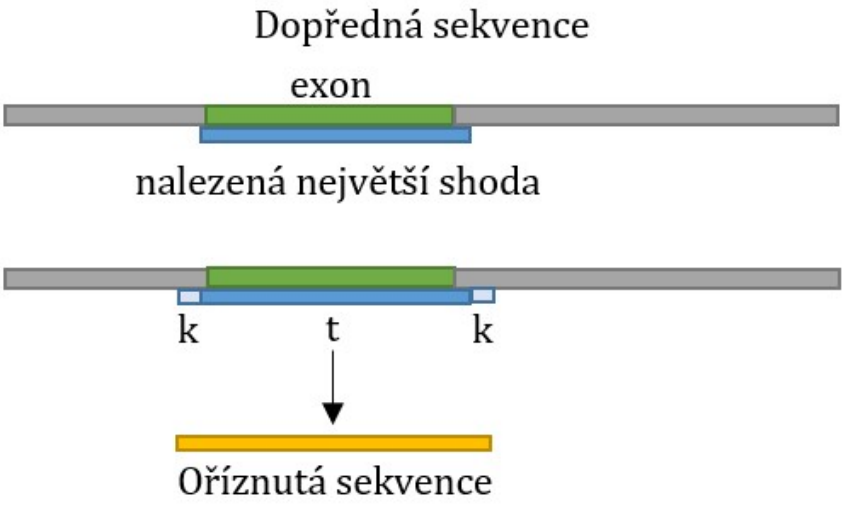
Metoda pro **trimming** (oříznutí) nám ořízne dopřednou sekvenci za pomoci porovnávání s reverzním komplementem zpětné sekvence. Metodu jsem pojmenovala *trim()* a vkládají se do ní 4 parametry: *trim(sekvence1F.seq, sekvence2F.seq, t, k)*, kde t je přibližná délka hledaného exonu a k je počet kroků, které se přidají na začátek a konec oříznutí (hodnota by měla být 3 až 5 kroků). Z referenčních dat z IPD byly zjištěny maximální délky typů exonů, doporučuji zvolit hodnotu tohoto parametru o 3 až 5 větší, a to z důvodu možných chyb v přečtené sekvenci (readu). Moje zvolené parametry pro MICA: exon 2 = 260, exon 3 = 290, exon 4 = 285; MICB: exon 2 = 260, exon 3 = 290, exon 4 = 285, exon 5 = 135 (vychází se z maximálních délek exonů, které jsou zmíněné v kapitole 7.7 Referenční data). Exony nejsou v sekvencích umístěny na stejném či podobném místě.



Obrázek 13: Vizualizace metody pro trimming.

Pro metodu předpokládám, že okolí exonu (šedé úseky na obrázku 15) není z většiny tvořeno znakem N, který mají některé programy tendenci vkládat, pokud čtení nemá silné signály.

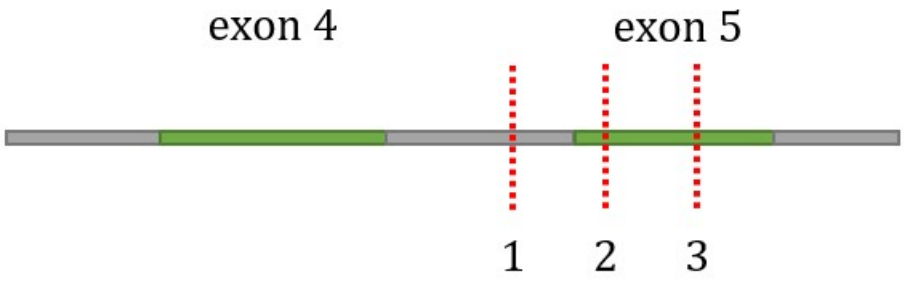
Z dopředné sekvence se vezme subsekvence délky t a porovnává se (funkcí *SequenceMatcher*) se stejně dlouhými subsekvencemi z reverzního komplementu zpětné sekvence, které se "posouvají" od nultého indexu až do indexu "*délka reverzního komplementu - 1 - t*". Hodnoty podobností se ukládají do seznamu, ve kterém se pak najde největší hodnota a ta se uloží. Z dopředné sekvence se vybere nová subsekvence posunutá o 1 krok a postup se opakuje. Z uložených hodnot se poté najde ta největší a zjistí se její index. Tento index představuje přibližné umístění začátku exonu, který hledáme (odchylka je závislá na rozdílu mezi zvoleným parametrem a délkou hledaného exonu, kterou ovšem neznáme přesně. Proto doporučuji zvolit hodnotu t zvolit maximálně o 5 větší, než je maximální délka referenčního exonu daného typu.



Obrázek 14: Vizualizace metody pro trimming.

Index pravděpodobného začátku použijeme k oříznutí dopředné sekvence a to tak, že začátek oříznutí bude 3 až 5 kroků před indexem začátku a konec oříznutí bude od indexu začátku vzdálen délkou $t + k$ kroků. Tyto kroky zvýší pravděpodobnost, že neořízneme části exonu na začátku či na konci, jelikož se ve čtení můžou vyskytovat chyby, které představují např. báze navíc, které by mohly prodloužit sekvenci exonu, nebo index začátku bude zvolen o jeden krok dříve nebo déle (experimentálně zjištěno).

Jediný problém ořezávání vstupních dat je u genu MICB, který má exony 4 a 5 v jedné sekvenci rozdělené sekvencí intronu. V tomto případě pro ořezávání sekvence exonu 4 vkládám k porovnávání dopředně a zpetné sekvence necelou (useklou) dopřednou sekvenci. Podmínkou pro správný ořez je neoříznutí exonu 4, nezáleží na tom, pokud se ořez udělá před nebo během exonu 5. Místo, kde se sekvence ořízne, se volí s ohledem na vzhled vstupních dat - přibližně jak daleko od sebe jsou exony 4 a 5. SequenceMatcher nalezne nejdelší společnou subsekvenci, což bude v našem případě exon 4. Pro oříznutí sekvence s exonom 5 vkládám dopřednou sekvenci ořízlou od indexu konce exonu 4. Předpokladem je pořadí exonů: exon 4 - exon 5 a exon 5 je kratší než exon 4.



Obrázek 15: Vizualizace správného oříznutí (1, 2, 3) dopředné sekvence genu MICB s exony 4 a 5. Šedé části značí introny.

Tato metoda umí oříznout sekvence jak homozygotní tak heterozygotní bez nutnosti znalosti této informace.

5.3.3 Homozygot / Heterozygot

V této části se v ořízlé sekvenci spočítají heterozygotní znaky. Pokud je tento počet nulový, sekvence obsahuje pouze homozygotní znaky a alignment přejde do homozygotního módu. Pokud je počet heterozygotních znaků větší než nula, alignment probíhá v heterozygotním módu. V ideálním případě se heterozygotní znaky objevují pouze v částech exonů, nikoli intronů, aby se při tvoření kombinací pro heterozygotní sekvence netvořilo zbytečně mnoho kombinací čtení sekvence.

5.3.4 Vytvoření kombinací heterozygotní sekvence

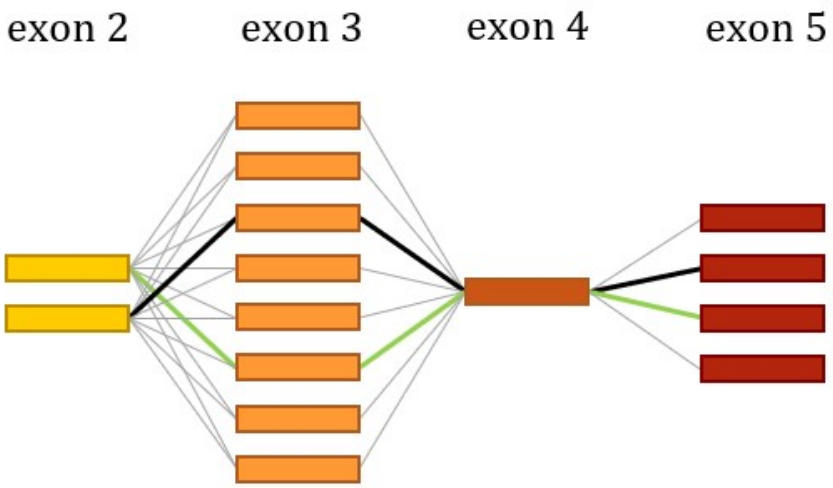
Pokud se jedná o heterozygotní sekvenci, musíme vytvořit všechny možné kombinace čtení dané sekvence. Počet těchto kombinací je 2^n, kde n je počet heterozygotních znaků. Každá varianta čtení představuje jednu možnou alelu, ale protože se jedná o heterozygotní sekvenci, nachází se v ní zakodované 2 alely, tedy přítomnost heterozygotních znaků, které představují pouze znaky dvou bází. Pokud se v sekvenci nachází znak, který představuje 3 znaky bází či chybný znak, nahradí se na daném místě tento znak písmenem **N** ve všech kombinacích. Legendu heterozygotních znaků a jaké báze představují najdete v kapitole 4.3 Sestavení sekvence na obrázku 6.

	A	K	G	M	T	T	Y
1	A	G	G	A	T	T	C
2	A	G	G	A	T	T	T
3	A	G	G	C	T	T	C
4	A	G	G	C	T	T	T
5	A	T	G	A	T	T	C
6	A	T	G	A	T	T	T
7	A	T	G	C	T	T	C
8	A	T	G	C	T	T	T

Obrázek 16: Možné kombinace heterozygotní sekvence. Šedý řádek značí heterozygotní sekvenci, řádek 1 - 8 zobrazují všechna možná čtení této sekvence. Sekvence obsahuje 3 heterozygotní znaky (K, M, Y), kombinací je tedy $2^3 = 8$.

Pokud heterozygotní znak **K** přeložíme do jedné alely jako **G**, druhá alela musí obsahovat na stejném místě znak **T**. Pokud budeme kombinace tvořit v podobě vzoru binárních kombinací (viz obrázek 16) víme, že čtení jsou k sobě komplementární (ve smyslu heterozygotních znaků) v 1. a 8. řádku, 2. a 7. řádku atd. Pro každý typ exonu bude jiný počet kombinací (závislý na počtu heterozygotních znaků). Pro každou kombinaci se dle typu exonu spočítá podobnost ke všem referenčním alelám. Pokud je podobnost menší než 0.6 (nebo jiná stanovená hodnota), dojde

k vyřazení dané kombinace ze stavového prostoru řešených kombinací (přiřadí se podobnost 0.0). Z těchto dat o podobnostech se vytvoří všechny možné kombinace kombinací typů exonů. Tyto kombinace k sobě budou komplementární (ve smyslu heterozygotních znaků) opět první a poslední, druhý a předposlední atd. Počet těchto kombinací je součinem počtu kombinací čtení jednotlivých typů exonů. Na obrázku 17 je tedy počet kombinací čtení $2 \cdot 8 \cdot 1 \cdot 4 = 64$ kombinací, a tedy 32 párů kombinací.



Obrázek 17: MICB - kombinace exonů. Černě vyznačená cesta je jedna z možností kombinace typů exonů. Zelená cesta je komplementární kombinace k černé. Barevné obdélníky představují data o podobnostech kombinací typů exonů k referenčním alelám.

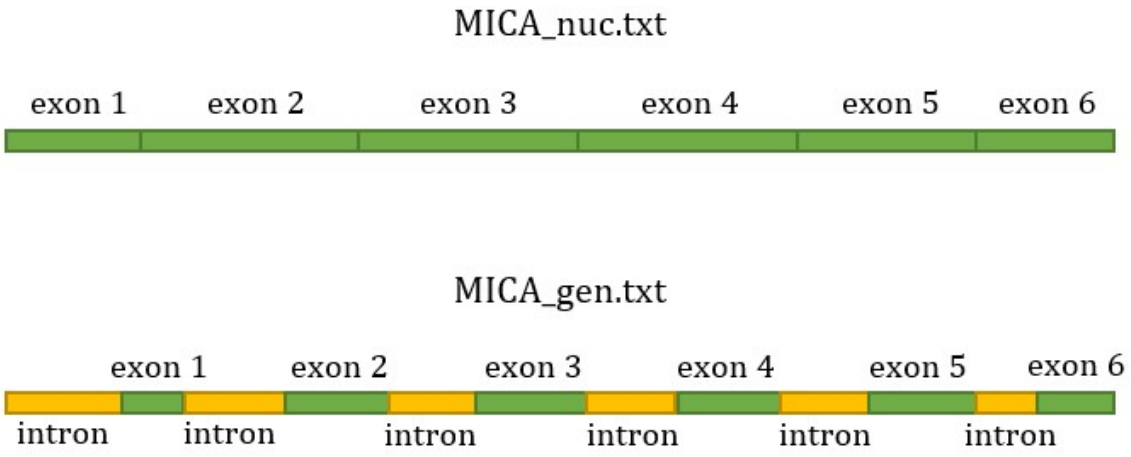
Předpoklad je, že v každé kombinaci exonů má nejvyšší hodnotu podobnosti pouze jedna alela (skupina alel). Pro každou kombinaci exonů se určí nejpravděpodobnější alela (najde se index s nejvyšší hodnotou a tento index odpovídá indexům referenčních alel) a poté se tyto alely sprárují dle komplementarity. Pokud je podobnost nejvyšší pro skupinu alel, najde se index pro první alelu z této skupiny (kromě skupiny 0).

Může se stát, že heterozygotní znaky jsou pouze v jediném exonu a ostatní exony jsou tvořeny pouze homozygotními znaky. Kombinace se vytvoří pouze pro exony s heterozygotními znaky a pro ostatní exony se pracuje s původní sekvencí.

Jedním z problémů, který může nastat je, že heterozygotní sekvence bude mít více kombinací, které "dávají smysl" - tedy více než dvě párové kombinace budou odpovídat různým alelám (se stejnou pravděpodobností). Tento fakt může ovlivnit finální vyhodnocení a jako výsledek určí více párů alel (skupin alel).

5.3.5 Referenční data, DB

Data, která jsou zpracovávána jsou exony 2, 3, 4 a 5 (exon 5 pouze u genu MICB), což vychází z definice reálných experimentů. Referenční data byla stažena z databáze IPD-IMGT/HLA. Pracuji se soubory *MICA_gen.txt*, *MICA_nuc.txt*, *MICB_gen.txt* a *MICB_nuc.txt* (formátované jako *fasta* soubory). V souborech *_gen* se nachází celé sekvence alel s introny i exony, v souborech *_nuc* se nachází sekvence alel pouze s exony. Každá alela v souboru má své *id* označení, popis a sekvenci, kde *id* a popis jsou na jednom řádku a na následujícím řádku začíná daná sekvence. Příklad *id* a popisu: HLA:HLA26812 MICA*008:12 1156 bp. V tomto případě se jedná o MICA gen, alelu 008:12. "1156 bp" označuje délku sekvence. Jako identifikátor sekvence používám pouze *id*, podle kterého můžu poté určit přesný název alely.

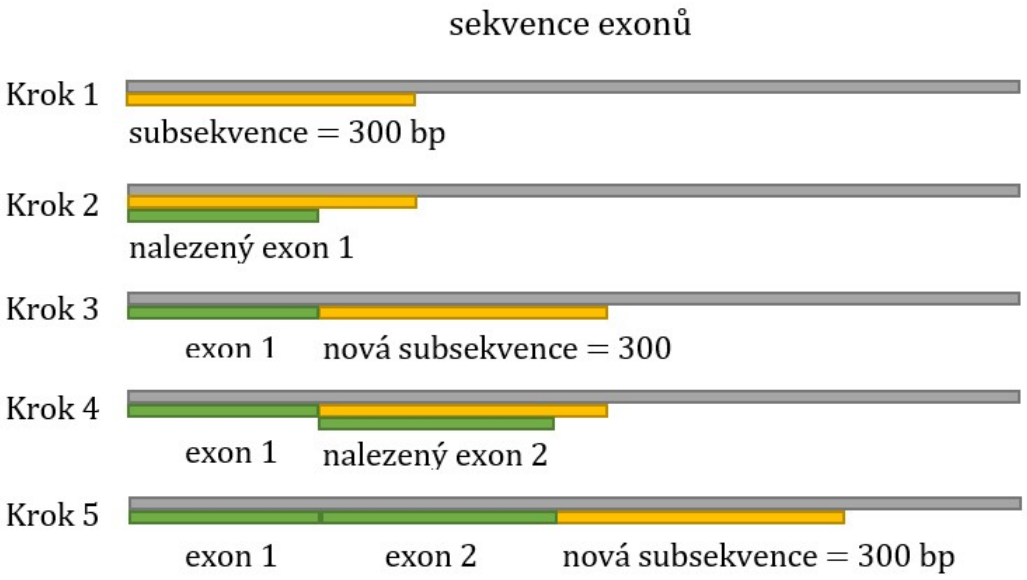


Obrázek 18: Vizualizace obsahu referenčních souborů genu MICA.

Navržená metoda pracuje pouze se samotnými exony. Problém je, že v případě souboru *_nuc* exony jsou poskládané do jedné dlouhé sekvence a nejsou přesně oddělené. Vytvořila jsem tedy *script*, který přečte soubory *_nuc* a *_gen*, porovná alely díky jejich *id* identifikátoru a vytvoří čtyři seznamy, které mají stejné pořadí alel, dva seznamy pro *id* identifikátory a dva seznamy pro samotné sekvence alel. Předpokladem pro tento postup je, že jsou exony poskládané v pořadí jako na obrázku 19. Za exonem 6 může být další sekvence (exon či intron), které nás ale nezajímají.

Při zjišťování počtu alel v souborech jsem zjistila, že v souborech *_nuc* je více alel než v souborech *_gen*. Přesněji MICA alel v *_nuc* je 388, v souboru *_gen* je 324 alel. MICB alel v souboru *_nuc* je 236, v souboru *_gen* je 202 alel. Rozhodla jsem se pracovat pouze s alelami, které mají zastoupení v obou souborech.

Délka jednotlivých exonů by se měla pohybovat v rozmezí 10 - 300 nukleotidů [**48**]. Jednotlivé exony získávám porovnáváním sekvence alel ze souboru *_nuc* se sekvencí se stejným *id* ze souboru *_gen*. Ze souboru *_nuc* vezmu subsekvenci dlouhou 300 bp (dle předpokladu délky exonu) a tuto subsekvenci porovnávám se sekvencí ze souboru *_gen*. Hledám vždy nejdelší společnou subsekvenci od počátku subsekvence ze souboru *_nuc*. Po nalezení nejdelší společné subsekvence, tedy exonu, zvolím ze souboru *_nuc* novou subsekvenci dlouhou 300 bp, která bude začínat na indexu hodnoty délky nalezeného (součtu nalezených) exonu. Princip se opakuje až do konce sekvence ze souboru *_nuc*. Algoritmus je zobrazen na obrázku 19.



Obrázek 19: **Algoritmus vybírání subsekvence a hledání exonů.** Krok 1: vytvoření subsekvence od nultého indexu; Krok 2: porovnání subsekvence se sekvencí ze souboru *_gen* a nalezení nejdelší společné subsekvence = exonu; Krok 3: vytvoření nové subsekvence od indexu s hodnotou délky posledního nalezeného exonu (nebo součtu délek exonů); Krok 4: porovnání nové subsekvence *_gen* sekvencí a nalezení nejdelší společné subsekvence, tedy dalšího exonu; Krok 5: algoritmus se opakuje až do konce sekvence exonů.

Nejdelší společná subsekvence se hledá způsobem porovnávání jednotlivých znaků jdoucích za sebou a pokud se znaky rovají, do hledané společné subsekvence se tento znak přidá. Algoritmus končí při prvním odlišném znaku. Předpokladem je, že jsou exony v obou souborech zaznamenány stejně.

V sekvenci se nachází exony 1, 2, 3, 4, 5 a 6. Když se dostáváme na konec sekvence, musíme délku subsekvence zmenšit abychom nepřekročili velikost sekvence, tímto nemusíme dostat přesná data o exonu 6 a dále. To nám ale v tomto případě nevadí. Pro MICA refereční data potřebujeme exony 2, 3 a 4, pro MICB potřebujeme exony 2, 3, 4 a 5. Referenční MICB exony 4 a 5 jsou v datech v jednom souboru, v referenčních jsou ale uchovávány jednotlivě.

```
>HLA:HLA01013
ATGGGGCTGGGCCCGGTCTTCCTGCTTCTGGCTGGCATCTTCCCTTTTGCACCTCCGGGA
>HLA:HLA01013
AGCCCCACAGTCTTCGTTATAACCTCACGGTGCTGTCCTGGGATGGATCTGTGCAGTCAG
>HLA:HLA01013
CTTGCATTCCCTCCAGGAGATTAGGGTCTGTGAGATCCATGAAGACAACAGCACCAGGAG
>HLA:HLA01013
TGCCCCCCATGGTGAATGTCACCCGCAGCGAGGCCTCAGAGGGCAACATTACCGTGACAT
>HLA:HLA01013
GAAAGTGCTGGTGCTTCAGAGTCATTGGCAGACATTCCATGTTTCTGCTGTTGCTGCTGC
>HLA:HLA01013
TCAGCCTCTG
>HLA:HLA01014
ATGGGGCTGGGCCCGGTCTTCCTGCTTCTGGCTGGCATCTTCCCTTTTGCACCTCCGGGA
>HLA:HLA01014
AGCCCCACAGTCTTCGTTATAACCTCACGGTGCTGTCCGGGGATGGATCTGTGCAGTCAG
>HLA:HLA01014
```

Obrázek 20: Výsledný textový soubor ve formátu fasta referenčních exonů. Řádek 1 - *id* alely; řádek 2 - sekvence exonu 1; řádek 3 - *id* alely, řádek 4 - sekvence exonu 2; řádek 5 - *id* alely, řádek 6 - sekvence exonu 3; řádek 7 - *id* alely, řádek 8 - sekvence exonu 4; řádek 9 - *id* alely; řádek 10 - sekvence exonu 5; řádek 11 - *id* alely; řádek 12 - neúplná sekvence exonu 6; řádek 13 - *id* nové alely; řádek 14 - sekvence exonu 1; atd.

Referenční data (nalezené exony) jsem se rozhodla uchovávat v textovém souboru ve formátu *fasta*, kde je pro každou alelu 12 řádků - 6 jako *id* identifikátory alely a 6 pro sekvence jednotlivých exonů. Jako první je zapsán identifikátor alely, hned za ním je nalezený exon.

Z obrázku 20 můžeme zpozorovat podobnost exonů stejného typu. Jednotlivé alely se mohou lišit pouze v jednom exonu a exony se mohou lišit i pouze v jedné bázi (= SNP = Single Nucleotid Polymorphism). Tato přílišná podobnost ztěžuje přesné

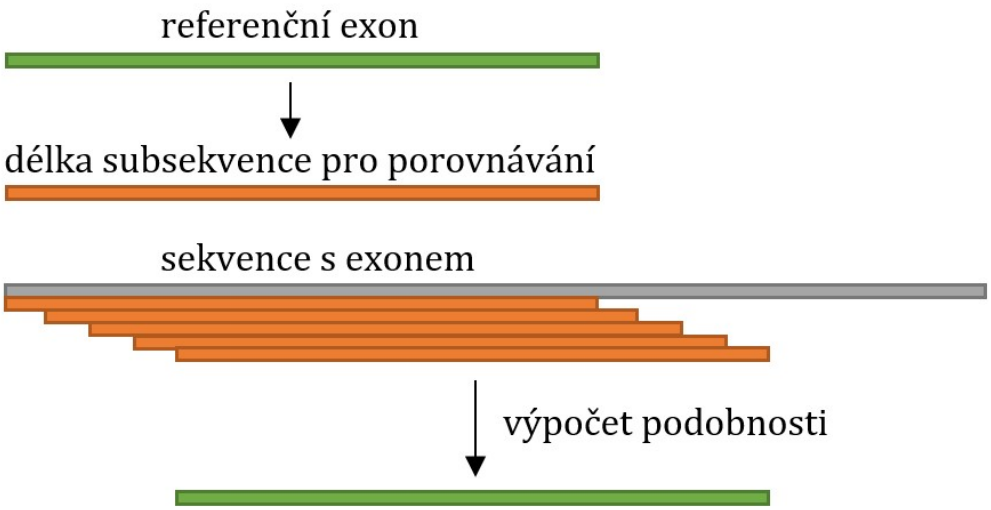
určení alel vstupních dat, protože vstupní data mohou obsahovat nepředvídatelné chyby nebo při tvoření kombinací heterozygotní sekvence může dojít k podobnosti více kombinací. Při zjišťování podobností referenčních alel bylo zjištěna totožnost několika alel v rámci řešených exonů. Pro MICA alely bylo vytvořeno 38 skupin. Skupina 0 obsahuje jedinečné záznamy alel, skupiny 1 - 37 zahrnují alely, které jsou totožné ve všech exonech. Pro MICB bylo vytvořeno 21 skupin stejného rozdělení. Skupiny naleznete v přílohách E a F. Alely, které jsou v referenčních exonech totožné se liší v jiných exonech, které ovšem do identifikace nezahrnujeme. Proto při vyhodnocení, o kterou alelu (alely) se jedná, algoritmus vrací buď název alely ze skupiny 0 nebo číslo skupiny 1 - 37/1 - 20.

Po analýze rozdělených exonů genu MICA byly zjištěny délky exonů. Délka exonu 2 dosahuje až 257 bp, pro exon 3 je to 288 bp a pro exon 4 je to 280 bp. Pro MICB exon 2 je délka 257 bp, pro exon 3 je délka 288 bp, pro exon 4 je délka 280 bp a pro exon 5 je to 131 bp. Délky se mohou lišit o jednu bázi.

Vzhledem k časové náročnosti programu je lepší nejdříve referenční data zpracovat, zapsat je do textového souboru ve formátu *fasta* a s tímto souborem poté pracovat při *alignmentu*. Zpracování referenčních dat pro MICA trvalo 30 minut, pro MICB přibližně 20 minut.

5.3.6 Alignment a určení alely

Alignment probíhá mezi vstupní sekvencí s jedním exonem a mezi všemi referenčními exony stejného typu (např. exon 2). Princip alignmentu je následující: z právě porovnávaného referenčního exonu vyčteme jeho délku a tuto délku použijeme na vytváření subsekvencí ze vstupní sekvence. Pro každou subsekvenci se spočítá podobnost s referenčním exonem a uloží se do seznamu. Z tohoto seznamu se vyjme nejvyšší hodnota podobnosti a vloží se do matice podobností. Takto se vytvoří matice podobností pro všechny exony všech referenčních alel. Poté se vytvoří všechny možné kombinace typů exonů (viz obrázek 17 v kapitole 5.3.4 Vytvoření kombinací heterozygotní sekvence) a sečtou se hodnoty podobností pro stejné alely. Nejvyšší hodnota součtu určí typ alely. Předpoklad je, že nejvyšší hodnota se v matici vykytuje pouze pro jednu alelu (skupinu alel).



Obrázek 21: Princip výpočtu podobností. Oranžová barva představuje vytvořenou subsekvenci k porovnávání, zelená barva představuje referenční exon z databáze.

Porovnávací funkcí je *SequenceMatcher*. Samotný alignment trvá od 20 vteřin do 2 minut (závislé na počtu sekvencí/kombinací sekvencí, pro které se počítá podobnost k referenčním alelám).

6 Validace a verifikace metody

6.1 Syntetická data

Pro účely validace a verifikace metody bylo nutné vytvořit syntetická data, aby bylo možné hodnotit správnost výsledků. Reálná data mohou být zatížena různými chybami (viz kapitola 5.1 Problémy alignmentu a identifikace) a navíc v tomto případě nemusíme vědět, jaká je reálná varianta genu.

Syntetická data byla vytvořena jednoduchým scriptem. Okolo exonů byly vloženy náhodné různě dlouhé sekvence znaků A, T, G, C. Takto bylo pro každou testovací alelu vytvořeno 6 souborů - 3x foward a 3x reverse.

Data byla nejdříve vyzkoušena s nezměněnými exony, poté se do exonů zanesla (manuálně) chyba jako např. změna báze, báze navíc či chybějící báze.

Homozygotní syntetická data byla vytvořena z referenčních exonů, ke kterým byly z obou stran "přilepeny" náhodné sekvence homozygotních znaků (A, T, G, C), neboli "pseudo-introny". Tyto pseudo-introny simulují reálné introny kolem exonů. Byly vyzkoušeny sekvence intronů z referenčních dat. Na výslednou funkčnost metody nemá záměna za pseudo-introny žádný vliv na identifikaci. V případě MICB exonů 4 a 5 byla data vytvořena způsobem: *intron - exon 4 - intron - exon 5 - intron.*

Heterozygotní syntetická data byla vytvořena zkombinováním dvou alel. Podmínkou byla stejná delka exonů stejného typu. Konsenzuální sekvence se vytvořila porovnáváním znaků na stejném místě a následným určením, o který typ kombinace se jedná (viz obrázek 6 v kapitole 4.3 Sestavení sekvence).

Pro gen MICA bylo vytvořeno 324 čistých (bez chyb) homozygotních syntetických dat (pro každou alelu), 129 čistých heterozygotních syntetických dat a 3 sady 93 syntetických unikátních homozygotních dat s chybou. Syntetická homozygotní data genu MICA s chybou byla tvořena manuálně. Pro gen MICB bylo vytvořeno 202 čistých homozygotních syntetických dat (pro každou alelu) a 93 čistých heterozygotních syntetických dat.

6.2 Výsledky pro syntetická data

Syntetická homozygotní data, která jsou vytvořena z referenčních alel ze stejné skupiny podobnosti, jsou totožná. Některá syntetická heterozygotní data byla vytvořena ze dvou alel ze dvou stejných skupin podobnosti (neplatí pro skupinu 0), což způsobilo totožnost některých syntetických heterozygotních dat. Pro tato data metoda určí vždy stejné výsledky.

Úspěšnost identifikace záleží na vlastnostech dat jako je např. ze kterých dvou alel je tvořena heterozygotní sekvence nebo jestli data obsahují chyby či mutace a jakého typu (báze navíc, chybějící báze, záměna báze apod.).

6.2.1 Čistá syntetická data genu MICA a MICB

Pro všechna **čistá homozygotní syntetická data MICA** (324 dat) **a MICB** (202 dat) byla úspěšnost identifikace alely či určité skupiny alel **100 %**.

Pro **čistá heterozygotní data genu MICA** bylo vytvořeno 129 syntetických dat. Ze 129 syntetických dat bylo pro společnou minimální hodnotu podobnosti 95 % správné řešení zahrnuto ve 126 výsledcích, což je úspěšnost **97,7 %**. Z těchto 126 výsledků se správně určenými dvojicemi bylo 88 výsledků jednoznačných - tedy pouze jedna kombinace alel (ta správná) měla nejvyšší (100%) pravděpodobnost. To je **69,8%** úspěšnost jednoznačné identifikace alely. Pro zbylých 38 výsledků byly určeny 2 kombinace různých alel, které měly stejnou pravděpodobnost (100%). To je způsobené tím, že více kombinací čtení heterozygotní sekvence může "dávat smysl". Nesprávné určení alely může být způsobeno rozdílností délek referenčních exonů a chybná báze navíc způsobila větší podobnost k jiné alele. Při porovnání špatně určených referenčních alel a správných referenčních alel byla zjištěna podobnost těchto alel větší než 90 %.

Unikátních typů čistých heterozygotních syntetických dat genu MICA je 69, tedy jedinečných výsledků oproti ostatním datům. Úspěšnost identifikace unikátních dat byla **95,7 %** (66 dat). Počet jednoznačně určených dvojic alel (skupin alel) bylo 60, což je **87%** úspěšnost jednoznačnosti výsledku. Počet správně určených alel z jednoznačných výsledků bylo 57. 3 nesprávně učené dvojice alel

obsahovaly vždy jednu správně určenou alelu s pravděpodobností 100 %. Po analýze podobností špatně určených alel se správnými alelami byla zjištěna podobnost těchto dvou referenčních alel vyšší než 90 %.

Zbylých 9 typů souborů, mělo ve výsledcích dva páry alel (skupiny alel) se stejnou nejvyšší pravděpodobností (100%).

Pro **čistá heterozygotní data genu MICB** bylo vytvořeno 93 syntetických dat. Z těchto 93 syntetických dat bylo pro společnou minimální hodnotu podobnosti 95 % správné řešení zahrnuto ve všech výsledcích, což je úspěšnost **100 %**. Z těchto 93 výsledků se správně určenými dvojicemi bylo 64 výsledků jednoznačných - tedy pouze jedna kombinace alel (ta správná) měla nejvyšší (100%) pravděpodobnost. To je **68,8%** úspěšnost jednoznačné identifikace alely. Pro zbylých 29 výsledků byly určeny 2 kombinace různých alel, které měly stejnou pravděpodobnost (100%). To je způsobené tím, že více kombinací čtení heterozygotní sekvence může "dávat smysl".

Unikátních typů čistých heterozygotních syntetických dat genu MICB je 16, tedy jedinečných výsledků oproti ostatním datům. Vzhledem k malému počtu unikátních typů dat tato statistika nemusí být přesná. Počet jednoznačně určených dvojic alel (skupin alel) bylo 14, což je **87,5%** úspěšnost jednoznačnosti výsledku. Zbylé 2 typy souborů, mělo ve výsledcích dva páry alel (skupiny alel) se stejnou nejvyšší pravděpodobností (100%).

Chybná identifikace alel mohla být způsobena špatně zvoleným místem pro *trimming*, kdy se mohlo oříznout pár znaků bází z exonu a tak alignment k referenčním exonům neproběhl správně v kombinaci s přílišnou podobností referenčních exonů alel.

6.2.2 Homozygotní syntetická data s chybou genu MICA

Syntetická data s chybou byla vytvořena manuálně pouze pro gen MICA, vzhledem k malému počtu unikátních dat genu MICB. Metodika identifikace je pro oba geny stejná. Pro simulaci chybných dat byly vytvořeny 3 sady 93 unikátních syntetických dat. Všechny chyby byly vytvořeny v dopředných sekvencí syntetických dat jednoho náhodného typu exonu. Syntetická data s chybou mají zastoupení téměř ve všech skupinách alel, ze skupiny 0 bylo vybráno 59 alel.

První sada 93 chybných syntetických dat je založena na **přebývající bázi**. Do sekvence exonu byl na náhodné místo vložen náhodný znak báze. Úspěšná identifikace proběhla u 79 dat, což odpovídá **84,9%** úspěšnosti identifikace.

Druhá sada 93 chybných syntetických dat je založena na **záměně báze**. V sekvenci exonu byla na náhodném místě vytvořena záměna znaku báze za náhodný výběr ze zbylých tří znaků bází. Úspěšná identifikace alely proběhla u 72 dat, což odpovídá **77,4%** úspěšnosti identifikace.

Třetí sada 93 chybných syntetických dat je založena na **chybějící bázi**. V sekvenci exonu byl na náhodném místě odebrán znak báze. Úspěšná identifikace alely proběhla pro 72 dat, což odpovídá **77,4%** úspěšnosti. Stejný výsledek úspěšnosti s předchozí sadou dat je zcela náhodný. Data, která nebyla správně identifikována se v obou sadách dat liší.

Nesprávné určení alely může být způsobeno rozdílností délek referenčních exonů, špatně zvolené místo pro *trimming* nebo chybná báze způsobila větší podobnost k jiné alele. Při porovnání některých špatně určených referenčních alel a správných referenčních alel byla zjištěna podobnost těchto alel větší než 90 %. Hodnoty pravděpodobností se pohybovaly mezi 66,7 až 98 %. Data, která se blížila k hodnotě pravděpodobnosti 66,7% byla špatně oříznuta při *trimmingu* a neproběhl tedy správný alingment k referenčním exonům pro chybný exon. I přes špatně ořízlý exon proběhla u několika dat úspěšná identifikace. Funkce *SequenceMatcher* má tendenci ohodnotit delší společnou subsekvenci vyšší hodnotou podobnosti. I přes to, že jsou chyby stejného charakteru, může hodnotu podobnosti ovlivnit umístění chyby a zda je chybný znak stejný jako alespoň jeden z okolních znaků.

7 Závěr

Práce se zabývá problematikou identifikace alel a následným návrhem metody pro automatickou identifikaci alel non-HLA genů MICA a MICB. V posledních letech se zkoumá význam non-HLA genů při TKB. Pokud by nastala situace, kdy je více dárců se shodnými HLA znaky, může shoda non-HLA genů pomoci při výberu vhodného dárce. Výběr správného dárce může značně snížit riziko GvHD.

První část práce rozebírá imunitu, HLA a non-HLA geny a jejich význam při TKB, Natural Killer buňky (NK), funkci NKG2D receptoru a jeho ligandy MICA a MICB, byl vysvětlen genový polymorfismus a vysvětlen genetický kód. Dále byly popsány metody získávání biologických sekvencí Sangerovo sekvenováním a Next-generation sekvenování (NGS) a byl zmíněn hlavní rozdíl mezi těmito metodami.

Ve druhé části práce byl vysvětlen způsob sestavení výsledné biologické sekvence a byly zmíněné formáty dat, které se používají k uchovávání biologických sekvencí a jejich vlastností (kvalita). Byla zmíněná databáze imunopolymorfismů (IPD; Imunno Polymorphism Database), ze které se v praktické části čerpají referenční data. Dále byly rozebrány hlavní problémy identifikace a alignmentu biologických sekvencí a byl zmíněn nástroj BLAST, který se zmíněnými problémy umí pracovat. Poté byla navržena metoda pro identifikaci alel s využitím pouze možností jazyka Python a jeho rozšíření Biopython, které bylo navrženo pro práci s biologickými sekvencemi.

Metoda identifikace byla navržena pro syntetická data s vlastnostmi reálných dat získávaných z FN Plzeň. Pro identifikaci se pro gen MICA využívají exony 2, 3 a 4, pro gen MICB exony 2, 3, 4 a 5. Byla popsána prvnotní úprava vstupních dat, získávání exonů ze vstupních dat (*trimming*), rozdělění přístupu identifikace podle skutečnosti, zda se jedná o homozygotní či heterozygotní sekvence, a následný průběh *alignmentu* s referenčními alelami z vytvořené databáze, jejíž způsob získávání byl také popsán.

Při analýze referenčních dat byla zjištěna totožnost všech exonů několika alel, které se používají pro identifikaci. Alely byly rozděleny do několika skupin: pro MICA bylo vytvořeno 38 skupin (0 - 37) a pro MICB 21 (0 - 20) skupin. Skupiny 0 obsahují unikátní alely. V poslední fázi se dle zvolených kritérií určuje nejpravděpodobnější

alela (skupina alel) pro homozygotní sekvenci nebo nejpravděpodobnější páry alel (skupin alel) pro heterozygotní sekvenci.

Na závěr byla provedena validace a verifikace metody identifikace na syntetických datech. Pro MICA bylo vytvořeno 324 čistých homozygotních syntetických dat, pro MICB bylo vytvořeno 202 čistých homozygotních syntetických dat (pro každou referenční alelu). Pro tato data byla úspěšnost identifikace alely (skupiny alel) 100 %. Dále bylo vytvořeno 129 čistých heterozygotních syntetických dat pro MICA a 93 čistých heterozygotních syntetických dat pro MICB. Některá čistá syntetická data byla vytvořena ze stejných skupin alel. Metoda v těchto případech vracela stejné výsledky. Úspěšnost jednoznačné i nejednozančné identifikace MICA 129 čistých syntetických heterozygotních dat byla 97,7 %, pro MICB byla úspěšnost jednoznačné i nejednoznačné identifikace 93 čistých syntetických heterozygotních dat 100 %. Pro gen MICA byly dále vytvořeny 3 sady 93 unikátních homozygotních syntetických dat s chybami - přebývající báze, záměna báze a chybějící báze. Pro data s přebývjící bází byla úspěšnost identifikace 84,9 %, pro data se záměnou báze byla úspěšnost identifikace 77,4 % a pro data s chybějící bází byla úspěšnost identifikace 77,4 %. Bylo zjištěno, že chybovost identifikace byla způsobena např. špatným ořízntutím dopředné sekvence ve fázi *trimming*, náhodná chyba způsobila vyšší podobnost k jiné alele, z důvodu SNP (Single nucleotide polymorphism) a celková podobnost referenčních alel.

Pro vyšší úspěšnost identifikace chybných dat, stejně tak možné praktické uplatnění, je třeba zohlednit kvalitu vstupních dat, tj. zaměřit se na zohlednění informace o kvalitě dat dostupné přímo ze sekvenace a oříznutí dat nekvalitních. Tímto bychom docílili správného oříznutí sekvence pro následný alignment. Jednotlivé kroky lze v budoucnu vylepšovat a zpřesňovat, jako např. zamyslet se nad možným využitím skórových matic BLOSUM či PAM nebo využít znalosti četnosti výskytu jednotlivých alel v dané populaci.

Reference

[1] Ambrůzová, Zuzana. Vybrané imunogenetické aspekty transplantace kmenových krvetvorných buněk [online]. Olomouc, 2014. [cit. 2020-10-16] Dostupné z: https://theses.cz/id/3eclk4/Z._Ambrzov__Vybran_imunogenetick_aspekty_TKB.pdf. Disertační práce. Lékařská fakulta Univerzita Palackého v Olomouci.

[2] Gratwohl A., et al. Risk assessment for patients with chronic myeloid leukaemia before allogeneic blood or marrow transplantation. Chronic Leukemia Working Party of the European Group for Blood and Marrow Transplantation. Lancet. 1998 Oct 3;352(9134):1087-92. [cit. 2020-10-16] doi: 10.1016/s0140-6736(98)03030-x.

[3] Hořejší, Václav a Jiřina Bartůnková. Základy imunologie. 3. vydání. Praha: Triton, 2008.[cit. 2020-10-17] 280 s.ISBN 80-7254-686-4.

[4] Typy, dárci a indikace k transplantacím. Linkos [online]. [cit. 2020-10-17] Dostupné z: https://www.linkos.cz/pacient-a-rodina/lecba/jak-se-lecit/transplantace-krvetvornych-bunek/typy-darci-a-indikace-k-transplantacim/

[5] Český národní registr dárců kostní dřeně [online]. [cit. 2020-10-20]. Dostupné z: https://registr.kostnidren.cz/

[6] Český registr dárců krvetvorných buněk [online]. [cit. 2020-10-20]. Dostupné z: https://www.darujzivot.cz/

[7] Shiel Jr. et al. Medical Definition of Copy number polymorphism [online]. [cit. 2020-11-8]. Dostupné také z: https://www.medicinenet.com/copy_number_polymorphism/definition.htm

[8] Takami, A. The role of non-HLA gene polymorphisms in graft-versus-host disease. Int J Hematol 98, 309–318 (2013).[cit. 2020-11-6] https://doi.org/10.1007/s12185-013-1416-7

[9] Perera Molligoda Arachchige, et al. (2021-03-24). "Human NK cells: From development to effector functions". Innate Immunity: 17534259211001512. [cit. 2020-12-10]. doi:10.1177/17534259211001512. ISSN 1753-4259

[10] Gasser, Stephan et al. "The DNA damage pathway regulates innate immune system ligands of the NKG2D receptor." Nature vol. 436,7054 (2005): 1186-90. [cit. 2020-12-11]. doi:10.1038/nature03884

[11] Klussmeier, Anja et al. High-Throughput MICA/B Genotyping of Over Two Million Samples: Workflow and Allele Frequencies [online]. 2020. [cit. 2021-1-9]. Dostupné z: doi:10.3389/fimmu.2020.00314

[12] Bahram, S., Spies, T. Nucleotide sequence of a human MHC class IMICB cDNA. Immunogenetics 43, 230–233 (1996). [cit. 2021-1-10]. https://doi.org/10.1007/BF00587305

[13] Zain Ahmed, Medhat Askar MICA (MHC class I polypeptide-related sequence A) Atlas Genet Cytogenet Oncol Haematol. [cit. 2021-1-16]. 2015;19(5):340-348.

[14] King, Robert C. et al. A Dictionary of Genetics, Seventh Edition. [s.l.]: Oxford University Press, 2006. [cit. 2021-1-22].

[15] Roth, Stephanie Clare (1 July 2019). "What is genomic medicine?". Journal of the Medical Library Association. University Library System, University of Pittsburgh. 107 (3). [cit. 2021-1-22]. doi:10.5195/jmla.2019.604

[16] Turanov AA, et al. (January 2009). "Genetic code supports targeted insertion of two amino acids by one codon". Science. 323 (5911): 259–61. [cit. 2021-1-22]. doi:10.1126/science.1164748

[17] Studium biochemie [online]. [cit. 2021-1-23]. Dostupné z: http://www.studiumbiochemie.cz/translace.html

[18] Sanger F; Coulson AR (May 1975). "A rapid method for determining sequences in DNA by primed synthesis with DNA polymerase". J. Mol. Biol. 94 (3): 441–8. [cit. 2021-2-2]. doi:10.1016/0022-2836(75)90213-2

[19] Matematická biologie - Sangerovo sekvenování. [cit. 2021-2-4]. Dostupné z: https://portal.matematickabiologie.cz/index.php?pg=analyza-genomickych-a-proteomickych-dat–analyza-sekvenci-dna–sekvenovani-genomu–sangerovo-sekvenovani

[20] Thermofisher - Detection and Quantification of Sequence Variants from Sanger Sequencing Traces. [cit. 2021-2-4]. Dostupné z: https://tools.thermofisher.com/content/sfs/brochures/seq-quantification-app-note.pdf

[21] Wick, Ryan R.; et al. (2019-06-24). "Performance of neural network basecalling tools for Oxford Nanopore sequencing". Genome Biology. Springer Science and Business Media LLC. 20 (1): 129. [cit. 2021-2-4]. doi:10.1186/s13059-019-1727-y. ISSN 1474-760X. PMC 6591954. PMID 31234903.

[22] Behjati S, et al. (December 2013). "What is next generation sequencing?". Archives of Disease in Childhood. Education and Practice Edition. 98 (6): 236–8. [cit. 2021-2-15]. doi:10.1136/archdischild-2013-304340

[23] Enners, Edward; Porta, Angela R. (2012). "Determining Annealing Temperatures for Polymerase Chain Reaction". The American Biology Teacher. 74 (4): 256–260. [cit. 2021-2-15]. doi:10.1525/abt.2012.74.4.9

[24] Key differences between next-generation sequencing and Sanger sequencing [online]. [cit. 2021-2-15]. Dostupné z: https://www.illumina.com/science/technology/next-generation-sequencing/ngs-vs-sanger-sequencing.html

[25] Sieber, Patricia et al. (March 2018). "The Definition of Open Reading Frame Revisited". Trends in Genetics. 34 (3): 167–170. [cit. 2021-3-4]. doi:10.1016/j.tig.2017.12.009

[26] Koonin EV (2005). "Orthologs, paralogs, and evolutionary genomics". Annual Review of Genetics. 39: 309–38. [cit. 2021-3-4]. doi:10.1146/annurev.genet.39.073003.114725

[27] Matematická biologie [online]. [cit. 2021-3-4]. Dostupné z: https://portal.matematickabiologie.cz/index.php? pg=analyza-genomickych-a-proteomickych-dat–analyza-sekvenci-dna–sekvenovani-genomu–sestaveni-sekvence

[28] Dmitriev DA, et al. (2008) Decoding of Superimposed Traces Produced by Direct Sequencing of Heterozygous Indels. PLOS Computational Biology 4(7): e1000113. [cit. 2021-3-10]. https://doi.org/10.1371/journal.pcbi.1000113

[29] Cock, P. J. A. et al. (2009). "The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants". Nucleic Acids Research. 38 (6): 1767–1771. [cit. 2021-3-21]. doi:10.1093/nar/gkp1137

[30] Cock PJ, et al. (April 2010). "The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants". Nucleic Acids Research. 38 (6): 1767–71. [cit. 2021-3-21]. doi:10.1093/nar/gkp1137

[31] Variant Calling Workflow [online]. [cit. 2021-3-21]. Dostupné z: https://datacarpentry.org/wrangling-genomics/04-variant_calling/index.html

[32] Variant identification and analysis [online]. [cit. 2021-3-21]. Dostupné z: https://www.ebi.ac.uk/training/online/courses/human-genetic-variation-introduction/variant-identification-and-analysis/

[33] Ewing B et al. (1998). "Base-calling of automated sequencer traces using phred. II. Error probabilities". Genome Research. 8 (3): 186–194. [cit. 2021-3-22]. doi:10.1101/gr.8.3.186

[34] Konsenzuální sekvence. [cit. 2021-3-22]. Dostupné z: http://lekarske.slovniky.cz/lexikon-pojem/konsenzualni-sekvence-consensus-sequence

[35] Karsch-Mizrachi, Ilene et al. "The International Nucleotide Sequence Database Collaboration." Nucleic acids research vol. 40,Database issue (2012): D33-7. [cit. 2021-4-2]. doi:10.1093/nar/gkr1006

[36] DNA Databank of Japan, DDBJ [online]. [cit. 2021-4-2]. Dostupné z: https://www.ddbj.nig.ac.jp/index-e.html

[37] EMBL - EBI [online]. [cit. 2021-4-2]. Dostupné z: https://www.ebi.ac.uk/

[38] NCBI [online]. [cit. 2021-4-2]. Dostupné z: https://www.ncbi.nlm.nih.gov/

[39] Zou, Dong. et al. Biological Databases for Human Research [online]. [cit. 2021-4-2]. Dostupné z: doi:https://doi.org/10.1016/j.gpb.2015.01.006

[40] Immuno Polymorphism Database IPD [online]. [cit. 2021-4-2]. Dostupné z: https://www.ebi.ac.uk/ipd/

[41] IPD-IMGT/HLA [online]. [cit. 2021-4-2]. Dostupné z: https://www.ebi.ac.uk/ipd/imgt/hla/

[42] Edwards, Rob. RobEdwards; EdwardsLab [online]. [cit. 2021-4-19]. Dostupné z: https://www.youtube.com/c/RobEdwardsVideos/videos

[43] FASTA [online]. [cit. 2021-5-2]. Dostupné z: https://www.ebi.ac.uk/Tools/sss/fasta/

[44] NCBI Web BLAST. [cit. 2021-5-2]. Dostupné také z: https://blast.ncbi.nlm.nih.gov/Blast.cgi

[45] Victoria Wang, X., Blades, N., Ding, J. et al. Estimation of sequencing error rates in short reads. BMC Bioinformatics 13, 185 (2012). [cit. 2021-6-12]. https://doi.org/10.1186/1471-2105-13-185

[46] Cock PA, et al. (2009) Biopython: freely available Python tools for computational molecular biology and bioinformatics. Bioinformatics, 25, 1422-1423 [cit. 2021-7-12].

[47] Python, difflib [online]. [cit. 2021-7-2]. Dostupné z: https://docs.python.org/3/library/difflib.html

[48] (2008) Exon. In: Encyclopedia of Genetics, Genomics, Proteomics and Informatics. Springer, Dordrecht. [cit. 2021-7-26]. https://doi.org/10.1007/978-1-4020-6754-9_5694

[49] How to identify bacteria using a single Sanger sequence. [cit. 2021-8-1]. Dostupné také z: https://help.ezbiocloud.net/tutorial-how-to-identify-using-single-sanger-sequencing/

[50] BLOSUM [online]. [cit. 2021-4-19]. Dostupné z: https://en.wikipedia.org/wiki/BLOSUM#/media/File:BLOSUM62.png

[51] Tyrosin. Wikipedie [online]. [cit. 2021-4-19]. Dostupné z: https://cs.wikipedia.org/wiki/Tyrosin#/media/Soubor:L-Tyrosin_-_L-Tyrosine.svg

[52] Fenylalanin. Wikipedie [online]. [cit. 2021-4-19]. Dostupné z: https://cs.wikipedia.org/wiki/Fenylalanin#/media/Soubor:L-Phenylalanin_-_L-Phenylalanine.svg

[53] Kyselina asparagová. Wikipedie [online]. [cit. 2021-4-19]. Dostupné z: https://cs.wikipedia.org/wiki/Kyselina_asparagov%C3%A1#/media/Soubor:L-aspartic-acid.png

[54] How To Install difflib in Python Using CMD? [online]. [cit. 2021-8-2]. Dostupné také z: https://wholeblogs.com/install-difflib-in-python-using-cmd/

Slovník pojmů a zkratek

A, *C*, *G*, *T*, *U*	Adenin, Cytosin, Guanin, Thymin, Uracil; znaky nukleových bází
Alela	Konkrétní forma genu
Alignment	Zarovnání více sekvencí
Antikodon	Komplementární trojice bází ke kodonu
Base − calling	Proces přiřazování báze
bp	Base-pair, dvojice komplementárních bází
DB	Databáze
DNA	Deoxyribonukleová kyselina
Exon	Kódující oblast genu
GvHD	Graft versus Host Disease, reakce štěpu proti hostiteli
Heterozygot	Jedinec se dvěma různými alelami daného genu
Heterozygotní znaky	R (A/G), Y (C/T), K (G/T), M (A/C), S (C/G), W (A/T), N (A/G/T/C)
HLA	Human Leukocyte Antigen
Homozygot	Jedinec se dvěma stejnými alelami daného genu
Homozygotní znaky	A, G, C, T
Intron	Nekódující oblast genu
IPD	Immuno Polymorphism Database
k − mers	Slova, kontinuální řetězec znaků
Kodon	Trojice za sebou jdoucích bází
Konsenzuální sekvence	Sestavení sekvence z kontigu podle frekvence bází
Kontig	Zarovnaná čtení sekvence na základě nejvyšší podobnosti

MHC	Major Histocompatibility Complex, hlavní histokompatibilní systém
MICA/B	MHC class I polypeptide-related sequence A/B, non-HLA geny
mRNA	Messenger ribonukleová kyselina
NGS	Next-generation sequencing, sekvenování nové generace
NK buňky	Natural Killer Cells
NKG2D	Natural killer Group 2, member D
ORF	Open Reading Frame, otevřený čtecí rámec
PCR	Polymerase Chain Reaction, proces replikace DNA
Primer	Řetězec nukleové kyseliny dlouhý několik bází, slouží jako počáteční místo replikace DNA či RNA
RF	Reading Frame, čtecí rámec
RNA	Ribonukleová kyselina
Sekvence	Posloupnost/řetězec znaků nukleových bází
SNP	Single nucleotide polymorphism, polymorfismus jednoho nukleotidu
Syntetické	Umělě vytvořené
TKB	Transplantace krvetvorných buněk
Trimming	Proces oříznutí sekvence
tRNA	Transferová ribonukleová kyselina

Přílohy

A Uživatelská příručka

A.1 Požadavky a specifikace

Program byl napsán a spuštěn na počítači s operačním systémem Windows 10, s 8 GB operační paměti a s CPU Intel Core i7 8th gen. 1.9 GHz. Rozhodla jsem se použít pouze možnosti jazyka Python, a to ze dvou důvodů. Zaprvé, program bude spustitelný na více operačních systémech, a zadruhé, z důvodu nedostatku znalostí s operačním systémem Linux, pro který je tvořena většina bioinformatických softwarů pro práci se sekvencemi. Řešení spadá do kategorie dynamického programování. Program není paměťově náročný, odhad maximálního využití operační paměti je 2 GB, maximálního zatížení procesoru je přibližně 30 %. Podmínka návrhu metody je použití offline nástrojů. Všechny použité nástroje a programy jsou open-source.

A.2 Potřebné nástroje

Program je potřeba spustit v prostředí Jupyter Notebook ve verzi 5.7.4.

Dále je nutné mít:

- Python verze min. 3.6 (použitá verze 3.9.6)
- Biopython verze 1.79
  - CMD příkaz: *pip install biopython*
- NumPy
- Python knihovna *difflib*
  - CMD příkaz: *pip install difflib*
  - Knihovna *difflib.py* se nachází ve hlavní složce programu a měla by být vložena do složky *../Python39/Lib*. V případě problémů viz [**54**].

A.3 Adresářová struktura

Ve složce *Program* se nachází dvě složky - *Software* a *Data*. Ve složce *Data* naleznete všechna data, která byla použita pro validaci a verifikaci metody identifikace. Adresářová struktura dat je následující:

- *Data / Test / MICA / Homozygot /*
  - *Clear* - obsahuje čistá homozygotní syntetická data genu MICA
  - *Error change* - obsahuje homozygotní syntetická data s chybou záměny báze genu MICA
  - *Error minus* - obsahuje homozygotní syntetická data s chybou chybějící báze genu MICA
  - *Error plus* - obsahuje homozygotní syntetická data s chybou přebývající báze genu MICA
- *Data / Test / MICA / Heterozygot* - obsahuje čistá heterozygotní syntetická data
- *Data / Test / MICB / Homozygot* - obsahuje čistá homozygotní syntetická data genu MICB
- *Data / Test / MICB / Heterozygot* - obsahuje čistá heterozygotní syntetická data genu MICB
- *Data / Reference* - obsahuje všechny soubory s referenčními daty
  - *MICA_gen.txt* - celé sekvence alel s introny i exony genu MICA
  - *MICA_nuc.txt* - sekvence alel pouze exonů genu MICA
  - *MICA_exons.txt* - rozdělené referenční exony genu MICA
  - *MICA_alleles.txt* - seznam *id* alel a jejich popisu genu MICA
  - *MICB_gen.txt* - celé sekvence alel s introny i exony genu MICB
  - *MICB_nuc.txt* - sekvence alel pouze exonů genu MICB
  - *MICB_exons.txt* - rozdělené referenční exony genu MICB
  - *MICB_alleles.txt* - seznam *id* alel a jejich popisu genu MICB

Ve složkách s čistými syntetickými daty jsou textové soubory seznamů čísel syntetických dat a správného *id* alel/y.

Ve složce *Software* jsou následující programy:

- Reference_data.ipynb
  - Vytvoří referenční data dle vložených souborů *_nuc.txt* a *_gen.txt*
- Synthetic_data.ipynb
  - Vytvoří syntetická (testovací) data dle vložené reference
- Alignment_MICA.ipynb
  - Pro vstupní soubory genu MICA určí nejpodobnější alelu/skupinu alel
- Alignment_MICB.ipynb
  - Pro vstupní soubory genu MICB určí nejpodobnější alelu/skupinu alel

A.4 Spuštění

Programy se spouští v prostředí Jupyter Notebook. V tomto prostředí je snadné upravit některé mezní hodnoty např. při alignmentu či trimmingu.

A.4.1 Reference_data

Reference_data vytvoří referenční data pro zadaný gen (MICA/MICB) z referenčních souborů *_nuc.txt* a *_gen.txt*. Algoritmus nalezne jednotlivé exony (podle principu v kapitole 5.3.5 Referenční data, DB) a zapíše je do souboru požadovaném formátu.

A.4.2 Synthetic_data

Program *Synthetic_data* vytvoří z referenčních dat syntetická data. Více v kapitole 6.1 Syntetická data. Program zahrnuje pět metod:

- *exonRead(exon_file)* načte referenční data ze souboru *exon_file* a uloží *id* a exony 1 až 5 do dvourozměrného pole, které metoda vrací.
- *homozygotMICA(exons)* vytvoří syntetická homozygotní data pro gen MICA z pseudo-intronů a referenčních exonů (parametr *exons*). Okolo referenčních exonů se "přilepí" pseudo-introny. Vytvoří takto šest sekvencí, pro každý exon dvě čtení - dopředné a zpětné - a data zapíše do šesti textových souborů ve formátu *fasta*. Lze zvolit počet syntetických dat, které chceme vytvořit, počet je omezený pouze počtem referenčních alel.
- *homozygotMICB(exons)* vytvoří syntetická homozygotní data pro gen MICA z pseudo-intronů a referenčních exonů (parametr *exons*). Okolo referenčních exonů se "přilepí" pseudo-introny. Pro exon 4 a 5 se vytvoří sekvence *intron - exon 4 - intron - exon 5 - intron*. Vytvoří takto šest sekvencí, pro každý typ exonu dvě čtení - dopředné a zpětné - a data zapíše do šesti textových souborů ve formátu *fasta*. Lze zvolit počet syntetických dat, které chceme vytvořit, počet je omezený pouze počtem referenčních alel.

- *heterozygotMICA(exons)* vytvoří heterozygotní data pro gen MICA. V programu lze upravit krok, s jakým se budou porovnávat dvě referenční alely z načteného dvourozměrného pole *exons*. Podmínka pro vytvoření heterozygotní sekvence je stejná délka totožných typů exonů. Tato podmínka může zmenšit počet vytvořených souborů, než byl stanovený počet, protože ne všechny alely mají stejně dlouhé typy exonů. Algoritmus porovná znaky na stejných pozicích ve dvou sekvencích exonů stejného typu a vytvoří novou heterozygotní sekvenci.
- *heterozygotMICB(exons)* vytvoří heterozygotní data pro gen MICB. V programu lze upravit krok, s jakým se budou porovnávat dvě referenční alely z načteného dvourozměrného pole *exons*. Podmínka pro vytvoření heterozygotní sekvence je stejná délka totožných typů exonů. Tato podmínka může zmenšit počet vytvořených souborů, než byl stanovený počet, protože ne všechny alely mají stejně dlouhé typy exonů. Algoritmus porovná znaky na stejných pozicích ve dvou sekvencích exonů stejného typu a vytvoří novou heterozygotní sekvenci.

A.4.3 Alignment_MICA/B

Hlavní program metody identifikace. Program obsahuje 4 metody:

- *trim(seq1, seq2, t, k)* ořízne dopřednou sekvenci (*seq1*) za pomoci nalezení podobnosti ve zpětné sekvenci (*seq2*). Parametr t představuje nejdelší očekávanou délku daného typu exonu, parametr k představuje o kolik kroků navíc se dopředná sekvence ořízne. Pro gen MICA tato metoda vrací pouze ořízlou sekvenci, pro gen MICB metoda vrací ořízlou sekvenci a navíc index začátku oříznutí sekvence, jelikož se tato hodnota používá pro správné oříznutí sekvence, které obsahuje exon 4 i exon 5. Pro exon 4 genu MICB se jako dopředná sekvence ořízlá dopředná sekvence - začátek index 0, konec 2/3 délky dopředné sekvence (zvolené podle charakteristik syntetických dat, tato hodnota lze upravit při optimalizaci s jinými daty).

- *combinations(exon, ex_comb)* vytvoří všechny možné kombinace vložené sekvence (parametr *exon*). Vkládá se také počet kombinací *ex_comb*, které se mají vytvořit (tato hodnota se počítá v metodě *Identifiaction_MICA/B*).

- *Alignment(exons, exon, a)* spočítá podobnost sekvence *exon* k referenčním datům *exons*. Parametr *a* představuje typ exonu.

- *Identification_MICA/B(vstupní soubory)* je hlavní metoda, ve které se využívají výše zmíněné metody. Program načte *vstupní soubory* (v pořadí: *exon2 F, exon2 R, exon3 F, exon3 R, exon4 F, exon4 R*) a referenční data, ořízne vstupní sekvence metodou *trim(seq1, seq2, t, k)*, zjistí, zda se jedná o homozygontní nebo heterozygotní data. Pokud se jedná o homozygotní data, přejde se rovnou k samotnému alignmentu k referenčním datům. Pokud se jedná o heterozygotní data, vytvoří se metodou *combinations(exon, ex_comb)* všechny možné kombinace, které projdou jednotlivě metodou *Alignment(exons, exon, a)*. Poté se vytvoří všechny možné kombinace kombinací exonů. Z každé kombinace se zjistí nejvyšší hodnota podobnosti k referenční alele a zjistí se její index. Kombinace se sprárují dle pravidla heterozygotní komplementarity. Předpoklad pro úspěšnou identifikaci je výskyt nejvyšší hodnoty pouze pro jednu alelu (skupinu alel).

  Parametr *vstupní soubory* představuje 6 souborů - 3 s dopřednou sekvencí a 3 se zpětnou sekvencí pro každý exon. Pokud bychom chtěli v metodě upravit minimální podobnost identifikovaných alel, změníme parametr *b* označený v kódu komentářem # MINIMALNI PODOBNOST.

A.5 Výpis programu pro identifikaci

Program *Identification_MICA/B(vstupní soubory)* vypisuje, ve které části programu se zrovna nachází - *Trimming*, *Alignment* apod., a navíc pro který exon se tato část děje.

Pro homozygotní data program na konci vypíše pravděpodobnost identifikace alely (hodnota dle podobnosti k referenční alele), ID a název alely nebo skupinu alel. V případě určení skupiny alel, vypíše se seznam všech alel v dané skupině.

```
Pravděpodobnost: 100.00%
Skupina 6, která zahrnuje tyto alely:
HLA:HLA02061 MICB*005:02:01 1152 bp
HLA:HLA02747 MICB*005:02:02 1152 bp
HLA:HLA02748 MICB*005:02:03 1152 bp
HLA:HLA02746 MICB*005:02:04 1152 bp
HLA:HLA27301 MICB*005:02:05 1152 bp
HLA:HLA27302 MICB*005:02:06 1152 bp
HLA:HLA27307 MICB*005:02:07 1152 bp
HLA:HLA27308 MICB*005:02:08 1152 bp
HLA:HLA27312 MICB*005:02:09 1152 bp
```

Obrázek 22: Výsledky, které program vypíše pro homozygotní data. Výsledný pár č.2 má největší pravděpodobnost. Pravděpodobnost (podobnost) se vypíše pro každou alelu zvlášť. Příklad pro syntetickou alelu test_mb91 ... .txt. Seznam alel není na obrázku úplný.

Pro heterozygotní data program na konci vypíše pořadí kombinace (vzhledem k výsledkům), dvojici názvů alel nebo čísla skupin alel, a dvojici pravděpodobností pro dané alely (skupiny alel) v pořadí odpovídající pořadí názvů alel (skupin alel). Pro heterozygotní data se v rámci úspory místa a zpřehlední při výpisu vyhodnocení nevypisují při určení skupiny alel alely v této skupině.

```
Výsledný pár č.1
Pravděpodobnost: 97.44% -- 100.00%
Alely: HLA:HLA31093 MICA*008:20 1156 bp -- HLA:HLA31093 MICA*008:20 1156 bp


Výsledný pár č.2
Pravděpodobnost: 100.00% -- 100.00%
Alely: HLA:HLA31093 MICA*008:20 1156 bp -- 7


Výsledný pár č.3
Pravděpodobnost: 100.00% -- 97.44%
Alely: 14 -- HLA:HLA01334 MICA*019:01:01 1155 bp
```

Obrázek 23: Výsledky, které program vypíše pro heterozygotní data. Výsledný pár č.2 má největší pravděpodobnost pro obě alely (skupiny alel). Pravděpodobnost (podobnost) se vypíše pro každou alelu (skupinu alel) zvlášť. Příklad pro syntetickou alelu test_ta50 ... .txt. Seznam dvojic alel (skupin alel) není na obrázků úplný.

B ORF - Otevřený čtecí rámec

ORF (Open Reading Frame) je část čtecího rámce, který je možno podrobit translaci. Tento nepřetržitý úsek kodonů začíná Start kodonem a končí Stop kodonem. Jedna z alternativních definicí ORF říká, že ORF je sekvence jejíž délka je dělitelná třemi a je ohraničená Stop kodony. Start kodon v ORF naznačuje začátek translace. Za ORF Stop kodonem je umístěno místo konce transkripce. Pokud by transkripce skončila před Stop kodonem, byla by během translace vyrobena nekompletní bílkovina. ORF se dá aplikovat pouze na "sestřiženou" mRNA, která neobsauhuje **introny**, ale pouze **exony**, jelikož introny mohou obsahovat Stop kodony a/nebo mohou způsobit posuny mezi čtecími rámci. Dlouhé ORF se často používá spolu s dalšími důkazy k počáteční identifikaci kandidátů oblastí kódujících bílkoviny nebo funkční oblasti kódujících RNA v DNA sekvenci.[**25**]

B.1 ORF hledání

Jeden z přístupů může být jednoduché hledání ORF, které začíná Start kodonem a končí Stop kodonem. Komplikuje to sice 6 různých čtecích možností ORF, ale to již není takovou překážkou pro dnešní výpočetní techniku. Klíčem k úspěchu této metody je frekvence výskytů Stop kodonů v sekvenci DNA. V případě, že má DNA náhodnou sekvenci a výskyt CG bází tvoří 50 %, pak se každý ze tří možných Stop kodonů (TAA, TAG, TGA) vyskytne přibližně jednou za 64 bp. Pokud je výskyt GC bází větší než 50 %, budou se Stop kodony kvůli AT bázím vyskytovat pravděpodobně každých 100 - 200 bp. Z těchto důvodů by náhodná DNA sekvence neměla mít ORF delších než 50 kodonů, navíc pokud bereme v potaz i počáteční Start kodon. Většina genů je ale delší než 50 kodonů, u lidí je to zhruba 450 kodonů. Použití jednoduchého ORF hledání pro lidskou DNA je méně efektivní, a to částečně z důvodu velkých intergenních částí mezi skutečnými geny. Tato skutečnost zvyšuje pravděpodobnost nálezů falešných ORF. [**25**]

C Problémy identifikace

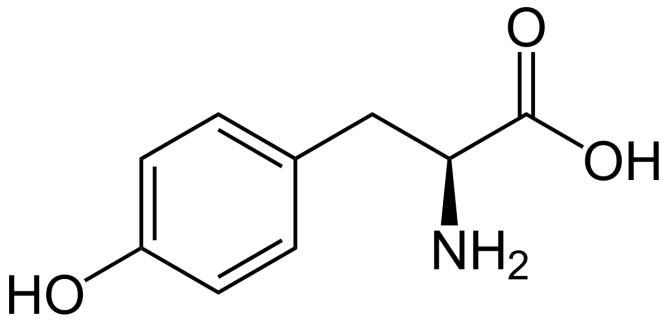
BLOSUM (BLOcks SUbstitution Matrix) je substituční matice používaná pro srovnávání sekvencí proteinů. BLOSUM matice určuje skóre shody mezi sekvencemi. Tato matice byla vytvořena prohledáním BLOCKS databáze za účelem najít stejné či velmi podobné regiony proteinových *rodin* a spočítáním relativní frekvence aminokyselin a pravděpodobnosti jejich substituce. [**42**]

	Ala	Arg	Asn	Asp	Cys	Gln	Glu	Gly	His	Ile	Leu	Lys	Met	Phe	Pro	Ser	Thr	Trp	Tyr	Val
Ala	4																			
Arg	−1	5																		
Asn	−2	0	6																	
Asp	−2	−2	1	6																
Cys	0	−3	−3	−3	9															
Gln	−1	1	0	0	−3	5														
Glu	−1	0	0	2	−4	2	5													
Gly	0	−2	0	−1	−3	−2	−2	6												
His	−2	0	1	−1	−3	0	0	−2	8											
Ile	−1	−3	−3	−3	−1	−3	−3	−4	−3	4										
Leu	−1	−2	−3	−4	−1	−2	−3	−4	−3	2	4									
Lys	−1	2	0	−1	−3	1	1	−2	−1	−3	−2	5								
Met	−1	−1	−2	−3	−1	0	−2	−3	−2	1	2	−1	5							
Phe	−2	−3	−3	−3	−2	−3	−3	−3	−1	0	0	−3	0	6						
Pro	−1	−2	−2	−1	−3	−1	−1	−2	−2	−3	−3	−1	−2	−4	7					
Ser	1	−1	1	0	−1	0	0	0	−1	−2	−2	0	−1	−2	−1	4				
Thr	0	−1	0	−1	−1	−1	−1	−2	−2	−1	−1	−1	−1	−2	−1	1	5			
Trp	−3	−3	−4	−4	−2	−2	−3	−2	−2	−3	−2	−3	−1	1	−4	−3	−2	11		
Tyr	−2	−2	−2	−3	−2	−1	−2	−3	2	−1	−1	−2	−1	3	−3	−2	−2	2	7	
Val	0	−3	−3	−3	−1	−2	−2	−3	−3	3	1	−2	1	−1	−2	−2	0	−3	−1	4

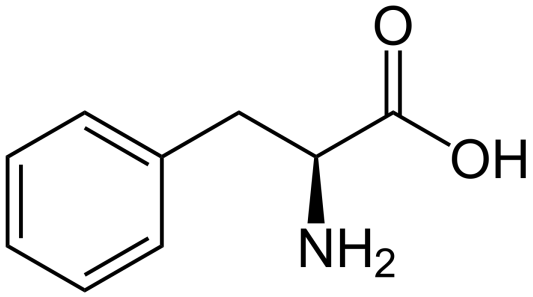
Obrázek 24: Substituční matice BLOSUM. [**50**]

Z BLOSUM matice můžeme vidět jednotlivá skóre záměn aminokyselin. Pro záměnu Tyrosin - Fenylalanin máme skóre +3, pro záměnu Tyrosin - kyselina asparagová máme skóre -3. Velikost hodnot se určuje podle složitosti struktury. Na *diagonále* matice najdeme skóre nalezení stejné aminokyseliny na daném místě. To, že jsou tyto hodnoty větší pro některé aminokyseliny, je způsobeno opět složitostí jejich struktury. Aminokyseliny s jednodušší strukturou mají menší velikost hodnot skóre v matici. [**42**]

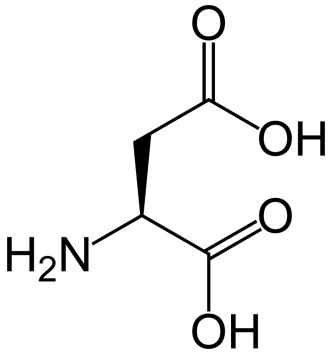
Hdnoty skóre této matice odpovídají tedy strukturální složitosti změny. Tedy záměna Tyrosinu za Fenylanin je mnohem jednodušší a nedělá takový problém, jako záměna Tyrosinu za kyselinu asparagovou. V obou případech se jedná pouze o změnu jedné báze, ale strukturálně je tato změna velmi odlišná. [**42**]



Obrázek 25: Tyrosin; kodony: TAT, TAC. [**51**]



Obrázek 26: Fenylalanin; kodony: TTT, TTC. [**52**]

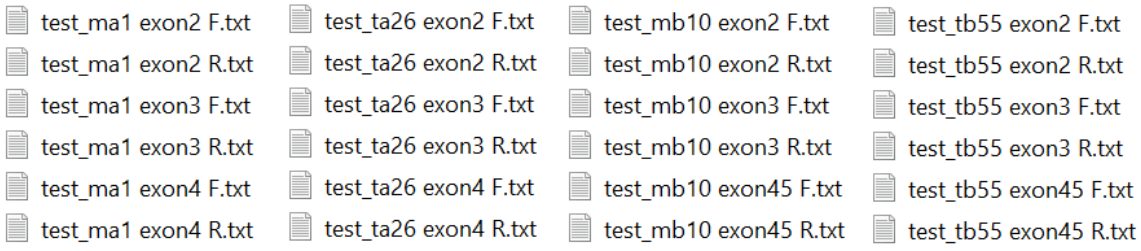


Obrázek 27: Kyselina asparagová; kodony: GAT, GAC. [**53**]

Tabulku všech aminokyselin a jejich kódujících kodonů je v kapitole 2.2 Kodon na obrázku 1.

D Syntetická data

Na obrázku 28 jsou příklady názvů souborů syntetických dat. Šestý znak názvu rozlišuje homozygotní m a heterozygotní t data, další znak rozlišuje gen MICA "a" nebo MICB "b", následuje číslo syntetické alely a typ exonu. Pro každý exon jsou dvě čtení - dopředné F a zpětné R. Pro gen MICB máme exony 4 a 5 v jednom souboru, typ exonu je označen jako exon45. Pro každou syntetickou alelu máme 6 souborů.



Obrázek 28: Nomenklatura souborů syntetických dat. Sloupec 1: MICA homozygot, sloupec 2: MICA heterozygot, sloupec 3: MICB homozygot, sloupec 4: MICB heterozygot.

E Skupiny alel MICA

Tabulky rozdělují referenční alely genu MICA do 38 skupin podle jejich podobnosti. Kritérium skupin 1 - 37 je maximální podobnost exonů 2, 3 a 4 (tedy podobnost s hodnotou 3.0 = 100 %). Ve skupině 0 jsou referenční alely, které jsou vzhledem k exonům 2, 3 a 4 jedinečné.

Skupina	Pořadí: ID alely	Název alely
0	1: HLA:HLA01013	MICA*001
	17: HLA:HLA03752	MICA*002:03
	18: HLA:HLA31203	MICA*002:05
	19: HLA:HLA31164	MICA*002:05
	20: HLA:HLA31346	MICA*002:06
	22: HLA:HLA31750	MICA*002:09
	34: HLA:HLA30886	MICA*004:02
	35: HLA:HLA31294	MICA*004:03
	36: HLA:HLA01017	MICA*006
	46: HLA:HLA02147	MICA*007:03
	49: HLA:HLA31215	MICA*007:08
	50: HLA:HLA31068	MICA*007:09
	74: HLA:HLA26250	MICA*008:10
	77: HLA:HLA30889	MICA*008:14
	78: HLA:HLA31331	MICA*008:15
	79: HLA:HLA31379	MICA*008:16
	80: HLA:HLA31214	MICA*008:17
	81: HLA:HLA31172	MICA*008:18
	82: HLA:HLA31069	MICA*008:19
	83: HLA:HLA31093	MICA*008:20
	84: HLA:HLA31045	MICA*008:21
	85: HLA:HLA31087	MICA*008:22
	88: HLA:HLA31771	MICA*008:24
	103: HLA:HLA30941	MICA*009:04
	104: HLA:HLA30920	MICA*009:05
	116: HLA:HLA31751	MICA*010:04

	120: HLA:HLA26921	MICA*011:01:04
	126: HLA:HLA01401	MICA*012:02
	127: HLA:HLA03110	MICA*012:03
	128: HLA:HLA30937	MICA*012:06
	129: HLA:HLA30974	MICA*012:07
	130: HLA:HLA01026	MICA*015:01
	131: HLA:HLA31509	MICA*015:02
	136: HLA:HLA31333	MICA*017:02
	137: HLA:HLA31267	MICA*017:03
	138: HLA:HLA31126	MICA*017:04
	151: HLA:HLA31278	MICA*018:03
	156: HLA:HLA13376	MICA*019:02
	159: HLA:HLA01391	MICA*024
	163: HLA:HLA31201	MICA*027:02
	166: HLA:HLA31144	MICA*029:03
	167: HLA:HLA01366	MICA*033
	168: HLA:HLA01368	MICA*043
	185: HLA:HLA04350	MICA*059
	186: HLA:HLA04853	MICA*060
	191: HLA:HLA31239	MICA*068:02
	195: HLA:HLA08959	MICA*074
	196: HLA:HLA10268	MICA*077
	202: HLA:HLA26447	MICA*094
	204: HLA:HLA26582	MICA*096N
	206: HLA:HLA26565	MICA*098:01
	207: HLA:HLA26449	MICA*098:02
	215: HLA:HLA26899	MICA*105Q
	216: HLA:HLA26908	MICA*106
	217: HLA:HLA26923	MICA*107N
	218: HLA:HLA26932	MICA*108
	227: HLA:HLA31003	MICA*113
	228: HLA:HLA31335	MICA*114
	229: HLA:HLA31345	MICA*115

231: HLA:HLA31004	MICA*117
232: HLA:HLA30925	MICA*118
235: HLA:HLA31375	MICA*120
236: HLA:HLA30956	MICA*121
237: HLA:HLA30975	MICA*122
238: HLA:HLA31007	MICA*123
241: HLA:HLA30922	MICA*125
242: HLA:HLA30907	MICA*126
243: HLA:HLA31378	MICA*127
244: HLA:HLA31363	MICA*128
245: HLA:HLA30940	MICA*129
248: HLA:HLA30906	MICA*131
249: HLA:HLA30955	MICA*132
250: HLA:HLA31350	MICA*133
251: HLA:HLA31362	MICA*134
252: HLA:HLA31402	MICA*135
256: HLA:HLA31306	MICA*138
257: HLA:HLA30957	MICA*139
258: HLA:HLA31332	MICA*140
259: HLA:HLA31349	MICA*141
262: HLA:HLA31380	MICA*143
263: HLA:HLA31361	MICA*144
264: HLA:HLA31388	MICA*145
265: HLA:HLA31401	MICA*146
266: HLA:HLA31296	MICA*147
267: HLA:HLA31266	MICA*148
269: HLA:HLA31184	MICA*150
273: HLA:HLA31295	MICA*152
274: HLA:HLA31277	MICA*153
275: HLA:HLA31249	MICA*154
276: HLA:HLA31050	MICA*155
277: HLA:HLA31023	MICA*156
278: HLA:HLA31143	MICA*157
279: HLA:HLA31106	MICA*158

280: HLA:HLA31049	MICA*159
281: HLA:HLA31024	MICA*159
282: HLA:HLA31092	MICA*161
283: HLA:HLA31048	MICA*162
284: HLA:HLA31171	MICA*163
285: HLA:HLA31252	MICA*163
286: HLA:HLA31217	MICA*165
290: HLA:HLA31110	MICA*167
293: HLA:HLA31265	MICA*169
294: HLA:HLA31090	MICA*170
295: HLA:HLA31071	MICA*171
296: HLA:HLA31127	MICA*172
297: HLA:HLA31029	MICA*173
298: HLA:HLA31072	MICA*174
299: HLA:HLA31046	MICA*175
300: HLA:HLA31187	MICA*176
301: HLA:HLA31202	MICA*177
302: HLA:HLA31254	MICA*178
303: HLA:HLA31218	MICA*179
304: HLA:HLA31150	MICA*180
305: HLA:HLA31163	MICA*181
308: HLA:HLA31392	MICA*184
309: HLA:HLA31310	MICA*185
310: HLA:HLA30958	MICA*186
311: HLA:HLA30972	MICA*187
312: HLA:HLA31334	MICA*188
313: HLA:HLA31309	MICA*189
314: HLA:HLA31391	MICA*190
315: HLA:HLA30924	MICA*191
316: HLA:HLA31589	MICA*192
317: HLA:HLA31508	MICA*193
318: HLA:HLA31491	MICA*194
319: HLA:HLA31830	MICA*195N
320: HLA:HLA31821	MICA*196

	321: HLA:HLA31745	MICA*197
	322: HLA:HLA31741	MICA*198
	323: HLA:HLA31832	MICA*199
	324: HLA:HLA31472	MICA*200
1	2: HLA:HLA01014	MICA*002:01:01
	3: HLA:HLA26233	MICA*002:01:02
	4: HLA:HLA26254	MICA*002:01:03
	5: HLA:HLA26318	MICA*002:01:04
	6: HLA:HLA26333	MICA*002:01:05
	7: HLA:HLA26266	MICA*002:01:06
	8: HLA:HLA26215	MICA*002:01:07
	9: HLA:HLA26896	MICA*002:01:08
	10: HLA:HLA26937	MICA*002:01:09
	11: HLA:HLA27206	MICA*002:01:10
	12: HLA:HLA30989	MICA*002:01:11
	13: HLA:HLA30921	MICA*002:01:12
	15: HLA:HLA31186	MICA*002:01:14
	16: HLA:HLA31400	MICA*002:01:15
	21: HLA:HLA31441	MICA*002:08
	197: HLA:HLA26343	MICA*089
	198: HLA:HLA26351	MICA*090
	199: HLA:HLA26237	MICA*091
	200: HLA:HLA26530	MICA*092
	201: HLA:HLA26514	MICA*093
2	14: HLA:HLA31276	MICA*002:01:13Q
	189: HLA:HLA07420	MICA*068:01:01
	190: HLA:HLA26911	MICA*068:01:02
3	192: HLA:HLA08898	MICA*072:01:01
	193: HLA:HLA26912	MICA*072:01:02
	194: HLA:HLA26913	MICA*072:01:03
4	23: HLA:HLA01015	MICA*004:01:01
	24: HLA:HLA26574	MICA*004:01:02
	25: HLA:HLA26521	MICA*004:01:03
	26: HLA:HLA26914	MICA*004:01:04

	27: HLA:HLA26928	MICA*004:01:05
	28: HLA:HLA26939	MICA*004:01:06
	29: HLA:HLA27243	MICA*004:01:07
	30: HLA:HLA27233	MICA*004:01:08
	31: HLA:HLA28790	MICA*004:01:09
	32: HLA:HLA30973	MICA*004:01:10
	33: HLA:HLA31802	MICA*004:01:11
	268: HLA:HLA31182	MICA*149
	306: HLA:HLA31219	MICA*182
5	101: HLA:HLA26938	MICA*009:03:01
	102: HLA:HLA31206	MICA*009:03:02
6	233: HLA:HLA30903	MICA*119:01:01
	234: HLA:HLA31006	MICA*119:01:02
7	89: HLA:HLA01020	MICA*009:01:01
	90: HLA:HLA26898	MICA*009:01:02
	91: HLA:HLA26900	MICA*009:01:03
	92: HLA:HLA26901	MICA*009:01:04
	93: HLA:HLA26902	MICA*009:01:04
	94: HLA:HLA26972	MICA*009:01:06
	95: HLA:HLA31185	MICA*009:01:07
	173: HLA:HLA01498	MICA*049:01:01
	174: HLA:HLA26903	MICA*049:01:01
	175: HLA:HLA31366	MICA*049:02
	219: HLA:HLA26935	MICA*109
	230: HLA:HLA30971	MICA*116
8	37: HLA:HLA01018	MICA*007:01:01
	38: HLA:HLA26294	MICA*007:01:02
	39: HLA:HLA26904	MICA*007:01:03
	40: HLA:HLA26936	MICA*007:01:04
	41: HLA:HLA27204	MICA*007:01:05
	42: HLA:HLA27205	MICA*007:01:06
	43: HLA:HLA30990	MICA*007:01:07
	44: HLA:HLA31105	MICA*007:01:08
	45: HLA:HLA31264	MICA*007:01:09

	210: HLA:HLA26871	MICA*100
9	47: HLA:HLA08961	MICA*007:06:01
	48: HLA:HLA26927	MICA*007:06:02
10	63: HLA:HLA01363	MICA*008:02
	76: HLA:HLA27319	MICA*008:13
11	86: HLA:HLA31292	MICA*008:23:01
	87: HLA:HLA31281	MICA*008:23:02
12	96: HLA:HLA01397	MICA*009:02:01
	97: HLA:HLA17992	MICA*009:02:02
	98: HLA:HLA26385	MICA*009:02:03
	99: HLA:HLA30993	MICA*009:02:04
	100: HLA:HLA31425	MICA*009:02:05
	253: HLA:HLA31348	MICA*136
13	51: HLA:HLA01019	MICA*008:01:01
	52: HLA:HLA06643	MICA*008:01:02
	53: HLA:HLA26248	MICA*008:01:03
	54: HLA:HLA26219	MICA*008:01:04
	55: HLA:HLA26382	MICA*008:01:05
	56: HLA:HLA26232	MICA*008:01:06
	57: HLA:HLA26265	MICA*008:01:07
	58: HLA:HLA27245	MICA*008:01:08
	59: HLA:HLA27265	MICA*008:01:09
	60: HLA:HLA26364	MICA*008:01:10
	61: HLA:HLA30959	MICA*008:01:10
	62: HLA:HLA31492	MICA*008:01:12
	64: HLA:HLA02488	MICA*008:04:01
	65: HLA:HLA26436	MICA*008:04:02
	66: HLA:HLA26443	MICA*008:04:03
	67: HLA:HLA26940	MICA*008:04:04
	68: HLA:HLA27228	MICA*008:04:05
	69: HLA:HLA31504	MICA*008:04:06
	71: HLA:HLA26330	MICA*008:06
	72: HLA:HLA26290	MICA*008:08
	73: HLA:HLA26245	MICA*008:09

	75: HLA:HLA26572	MICA*008:11
	160: HLA:HLA01371	MICA*027:01:01
	161: HLA:HLA26892	MICA*027:01:01
	162: HLA:HLA31181	MICA*027:01:03
	212: HLA:HLA26890	MICA*102
	213: HLA:HLA26891	MICA*103
	214: HLA:HLA26893	MICA*104
14	152: HLA:HLA01334	MICA*019:01:01
	153: HLA:HLA26435	MICA*019:01:02
	154: HLA:HLA31250	MICA*019:01:03
	155: HLA:HLA31025	MICA*019:01:04
15	105: HLA:HLA01021	MICA*010:01:01
	106: HLA:HLA26544	MICA*010:01:02
	107: HLA:HLA26424	MICA*010:01:03
	108: HLA:HLA26517	MICA*010:01:04
	109: HLA:HLA26405	MICA*010:01:05
	110: HLA:HLA26929	MICA*010:01:06
	111: HLA:HLA31376	MICA*010:01:07
	112: HLA:HLA31320	MICA*010:01:08
	113: HLA:HLA06451	MICA*010:02:01
	114: HLA:HLA26934	MICA*010:02:02
	115: HLA:HLA26304	MICA*010:02:02
16	208: HLA:HLA26508	MICA*099:01:01
	209: HLA:HLA26438	MICA*099:01:02
17	117: HLA:HLA01022	MICA*011:01:01
	118: HLA:HLA26918	MICA*011:01:02
	119: HLA:HLA26919	MICA*011:01:03
	121: HLA:HLA26933	MICA*011:01:05
	122: HLA:HLA26920	MICA*011:01:06
18	123: HLA:HLA01023	MICA*012:01:01
	124: HLA:HLA26297	MICA*012:01:02
	125: HLA:HLA26973	MICA*012:01:03
19	132: HLA:HLA01027	MICA*016:01:01
	133: HLA:HLA31280	MICA*016:01:02

	134: HLA:HLA31374	MICA*016:01:03
20	135: HLA:HLA01335	MICA*017:01
	203: HLA:HLA26431	MICA*095
	307: HLA:HLA31293	MICA*183
21	139: HLA:HLA01336	MICA*018:01:01
	140: HLA:HLA26895	MICA*018:01:02
	141: HLA:HLA26905	MICA*018:01:03
	142: HLA:HLA26906	MICA*018:01:04
	143: HLA:HLA26907	MICA*018:01:04
	144: HLA:HLA26909	MICA*018:01:06
	145: HLA:HLA26930	MICA*018:01:07
	146: HLA:HLA31168	MICA*018:01:08
	147: HLA:HLA31476	MICA*018:01:09
	148: HLA:HLA31775	MICA*018:01:10
	149: HLA:HLA31819	MICA*018:01:11
	150: HLA:HLA31809	MICA*018:01:12
22	157: HLA:HLA01394	MICA*022:01:01
	158: HLA:HLA26894	MICA*022:01:02
23	164: HLA:HLA01365	MICA*029:01:01
	165: HLA:HLA26910	MICA*029:01:02
24	169: HLA:HLA01369	MICA*045:01:01
	170: HLA:HLA26931	MICA*045:01:02
	171: HLA:HLA31073	MICA*045:01:03
25	172: HLA:HLA01410	MICA*047
	211: HLA:HLA26922	MICA*101
26	176: HLA:HLA02334	MICA*053:01:01
	177: HLA:HLA26519	MICA*053:01:02
	178: HLA:HLA26529	MICA*053:01:03
	179: HLA:HLA26570	MICA*053:01:04
	180: HLA:HLA26512	MICA*053:01:05
27	181: HLA:HLA03484	MICA*057:01:01
	182: HLA:HLA26583	MICA*057:01:02
	183: HLA:HLA26439	MICA*057:01:03
	184: HLA:HLA26578	MICA*057:01:04

	205: HLA:HLA26459	MICA*097
28	187: HLA:HLA05377	MICA*062:01:01
	188: HLA:HLA26941	MICA*062:01:02
29	220: HLA:HLA26888	MICA*111:01
	221: HLA:HLA26889	MICA*111:02
30	222: HLA:HLA07097	MICA*112:01:01
	223: HLA:HLA26924	MICA*112:01:02
	224: HLA:HLA26925	MICA*112:01:03
	225: HLA:HLA26926	MICA*112:01:04
	226: HLA:HLA26897	MICA*112:01:05
31	239: HLA:HLA30939	MICA*124:01:01
	240: HLA:HLA31466	MICA*124:01:02
32	246: HLA:HLA30887	MICA*130:01
	247: HLA:HLA31801	MICA*130:02
33	254: HLA:HLA31321	MICA*137:01:01
	255: HLA:HLA31263	MICA*137:01:02
34	260: HLA:HLA31307	MICA*142:01:01
	261: HLA:HLA31784	MICA*142:01:02
35	270: HLA:HLA31240	MICA*151:01:01
	271: HLA:HLA31251	MICA*151:01:02
	272: HLA:HLA31725	MICA*151:01:03
36	287: HLA:HLA31205	MICA*166:01:01
	288: HLA:HLA31149	MICA*166:01:02
	289: HLA:HLA31662	MICA*166:01:03
37	291: HLA:HLA31130	MICA*168:01:01
	292: HLA:HLA31440	MICA*168:01:02

Tabulka 2: Rozdělení alel genu MICA

F Skupiny alel MICB

Tabulky rozdělují referenční alely genu MICB do 21 skupin podle jejich podobnosti. Kritérium skupin 1 - 20 je maximální podobnost exonů 2, 3, 4 a 5 (tedy podobnost s hodnotou 4.0 = 100 %). Ve skupině 0 jsou referenční alely, které jsou vzhledem k exonům 2, 3, 4 a 5 jedinečné.

Skupina	Pořadí: ID alely	Název alely
0	196: HLA:HLA26135	MICB*034
	197: HLA:HLA26211	MICB*035
	198: HLA:HLA26634	MICB*036
	199: HLA:HLA26628	MICB*037
	201: HLA:HLA28931	MICB*041N
	202: HLA:HLA30842	MICB*043N
1	1: HLA:HLA02059	MICB*002:01:01
	2: HLA:HLA02749	MICB*002:01:02
	3: HLA:HLA26091	MICB*002:01:03
	4: HLA:HLA26141	MICB*002:01:04
	5: HLA:HLA26199	MICB*002:01:05
	6: HLA:HLA26167	MICB*002:01:06
	7: HLA:HLA26151	MICB*002:01:07
	8: HLA:HLA26136	MICB*002:01:08
	9: HLA:HLA26754	MICB*002:01:09
	10: HLA:HLA26729	MICB*002:01:10
	11: HLA:HLA26621	MICB*002:01:11
	12: HLA:HLA26688	MICB*002:01:12
	13: HLA:HLA26615	MICB*002:01:13
	14: HLA:HLA26639	MICB*002:01:14
	15: HLA:HLA26645	MICB*002:01:15
	16: HLA:HLA26701	MICB*002:01:16
	17: HLA:HLA27300	MICB*002:01:17
	18: HLA:HLA27305	MICB*002:01:18
	19: HLA:HLA28236	MICB*002:01:19
	20: HLA:HLA28237	MICB*002:01:20

	21: HLA:HLA28457	MICB*002:01:21
	22: HLA:HLA28464	MICB*002:01:22
	23: HLA:HLA28940	MICB*002:01:23
	24: HLA:HLA28945	MICB*002:01:24
	25: HLA:HLA28946	MICB*002:01:25
	26: HLA:HLA28977	MICB*002:01:26
	27: HLA:HLA28981	MICB*002:01:27
	28: HLA:HLA28983	MICB*002:01:28
2	29: HLA:HLA18243	MICB*002:02:01
	30: HLA:HLA26195	MICB*002:02:02
3	31: HLA:HLA02060	MICB*003:01:01
	32: HLA:HLA27310	MICB*003:01:02
	33: HLA:HLA27317	MICB*003:01:03
	34: HLA:HLA28932	MICB*003:01:04
	35: HLA:HLA28935	MICB*003:01:05
	36: HLA:HLA28937	MICB*003:01:06
	37: HLA:HLA28941	MICB*003:01:07
	38: HLA:HLA28961	MICB*003:01:08
	39: HLA:HLA28963	MICB*003:01:09
	40: HLA:HLA28970	MICB*003:01:10
	41: HLA:HLA28972	MICB*003:01:11
	42: HLA:HLA29309	MICB*003:01:12
4	43: HLA:HLA02064	MICB*004:01:01
	44: HLA:HLA02750	MICB*004:01:02
	45: HLA:HLA27296	MICB*004:01:03
	46: HLA:HLA27297	MICB*004:01:04
	47: HLA:HLA27299	MICB*004:01:05
	48: HLA:HLA27309	MICB*004:01:06
	49: HLA:HLA28235	MICB*004:01:07
	50: HLA:HLA28455	MICB*004:01:08
	51: HLA:HLA28456	MICB*004:01:09
	52: HLA:HLA28458	MICB*004:01:10
	53: HLA:HLA28459	MICB*004:01:11
	54: HLA:HLA28463	MICB*004:01:12

	55: HLA:HLA28690	MICB*004:01:13
	56: HLA:HLA28693	MICB*004:01:14
	57: HLA:HLA28938	MICB*004:01:15
	58: HLA:HLA28939	MICB*004:01:16
	59: HLA:HLA28942	MICB*004:01:17
	60: HLA:HLA28951	MICB*004:01:18
	61: HLA:HLA28952	MICB*004:01:19
	62: HLA:HLA28954	MICB*004:01:20
	63: HLA:HLA28957	MICB*004:01:21
	64: HLA:HLA28960	MICB*004:01:22
	65: HLA:HLA28971	MICB*004:01:23
	66: HLA:HLA29304	MICB*004:01:24
	67: HLA:HLA29305	MICB*004:01:25
	68: HLA:HLA29311	MICB*004:01:26
	69: HLA:HLA29312	MICB*004:01:27
	189: HLA:HLA08424	MICB*028
5	70: HLA:HLA02062	MICB*005:01:01
	71: HLA:HLA26086	MICB*005:01:02
	72: HLA:HLA28984	MICB*005:01:03
6	73: HLA:HLA02061	MICB*005:02:01
	74: HLA:HLA02747	MICB*005:02:02
	75: HLA:HLA02748	MICB*005:02:03
	76: HLA:HLA02746	MICB*005:02:04
	77: HLA:HLA27301	MICB*005:02:05
	78: HLA:HLA27302	MICB*005:02:06
	79: HLA:HLA27307	MICB*005:02:07
	80: HLA:HLA27308	MICB*005:02:08
	81: HLA:HLA27312	MICB*005:02:09
	82: HLA:HLA27313	MICB*005:02:10
	83: HLA:HLA27315	MICB*005:02:11
	84: HLA:HLA28461	MICB*005:02:12
	85: HLA:HLA28688	MICB*005:02:13
	86: HLA:HLA28692	MICB*005:02:14
	87: HLA:HLA28685	MICB*005:02:15

	88: HLA:HLA28686	MICB*005:02:16
	89: HLA:HLA28934	MICB*005:02:17
	90: HLA:HLA28943	MICB*005:02:18
	91: HLA:HLA28944	MICB*005:02:19
	92: HLA:HLA28947	MICB*005:02:20
	93: HLA:HLA28948	MICB*005:02:21
	94: HLA:HLA28953	MICB*005:02:22
	95: HLA:HLA28955	MICB*005:02:23
	96: HLA:HLA28956	MICB*005:02:24
	97: HLA:HLA28974	MICB*005:02:25
	98: HLA:HLA28975	MICB*005:02:26
	99: HLA:HLA28976	MICB*005:02:27
	100: HLA:HLA28978	MICB*005:02:28
	101: HLA:HLA28985	MICB*005:02:29
	102: HLA:HLA29308	MICB*005:02:30
	103: HLA:HLA29310	MICB*005:02:31
	104: HLA:HLA30349	MICB*005:02:32
	105: HLA:HLA30348	MICB*005:02:33
	106: HLA:HLA30350	MICB*005:02:34
7	107: HLA:HLA02067	MICB*005:03:01
	108: HLA:HLA26161	MICB*005:03:02
	109: HLA:HLA26128	MICB*005:03:03
	110: HLA:HLA26201	MICB*005:03:04
	111: HLA:HLA26142	MICB*005:03:05
8	112: HLA:HLA07364	MICB*005:06:01
	113: HLA:HLA26093	MICB*005:06:02
	114: HLA:HLA28986	MICB*005:06:03
9	115: HLA:HLA08420	MICB*005:07:01
	116: HLA:HLA26144	MICB*005:07:02
10	117: HLA:HLA02065	MICB*008:01:01
	118: HLA:HLA26647	MICB*008:01:02
	119: HLA:HLA26625	MICB*008:01:03
	120: HLA:HLA26721	MICB*008:01:04
	121: HLA:HLA26665	MICB*008:01:05

	122: HLA:HLA26629	MICB*008:01:06
	123: HLA:HLA26599	MICB*008:01:07
	124: HLA:HLA26604	MICB*008:01:08
	125: HLA:HLA26709	MICB*008:01:09
	126: HLA:HLA26696	MICB*008:01:10
	127: HLA:HLA26734	MICB*008:01:11
11	128: HLA:HLA02066	MICB*009:01:01N
	129: HLA:HLA28933	MICB*009:01:02N
	130: HLA:HLA28936	MICB*009:01:03N
	131: HLA:HLA28958	MICB*009:01:04N
	132: HLA:HLA28959	MICB*009:01:05N
	133: HLA:HLA28962	MICB*009:01:06N
	134: HLA:HLA28964	MICB*009:01:07N
	135: HLA:HLA28965	MICB*009:01:08N
	136: HLA:HLA28966	MICB*009:01:09N
	137: HLA:HLA28967	MICB*009:01:10N
	138: HLA:HLA28968	MICB*009:01:11N
	139: HLA:HLA28969	MICB*009:01:12N
	140: HLA:HLA29302	MICB*009:01:13N
	141: HLA:HLA29303	MICB*009:01:14N
	142: HLA:HLA29306	MICB*009:01:15N
12	143: HLA:HLA02074	MICB*013:01:01
	144: HLA:HLA26691	MICB*013:01:02
	145: HLA:HLA26646	MICB*013:01:03
	146: HLA:HLA26619	MICB*013:01:04
	147: HLA:HLA26672	MICB*013:01:05
	148: HLA:HLA28973	MICB*013:01:06
13	149: HLA:HLA02073	MICB*014:01:01
	150: HLA:HLA26685	MICB*014:01:02
	151: HLA:HLA26662	MICB*014:01:03
	152: HLA:HLA26605	MICB*014:01:04
	153: HLA:HLA26668	MICB*014:01:05
	154: HLA:HLA26650	MICB*014:01:06
	155: HLA:HLA28460	MICB*014:01:07

	156: HLA:HLA28982	MICB*014:01:08
14	157: HLA:HLA02190	MICB*018:01:01
	158: HLA:HLA26637	MICB*018:01:02
	159: HLA:HLA27304	MICB*018:01:03
15	160: HLA:HLA02191	MICB*019:01:01
	161: HLA:HLA26598	MICB*019:01:02
	162: HLA:HLA26686	MICB*019:01:03
	163: HLA:HLA26707	MICB*019:01:04
	164: HLA:HLA26748	MICB*019:01:05
	165: HLA:HLA26742	MICB*019:01:06
	166: HLA:HLA26630	MICB*019:01:07
	167: HLA:HLA26699	MICB*019:01:08
	200: HLA:HLA26766	MICB*038
16	168: HLA:HLA02192	MICB*020:01:01
	169: HLA:HLA26718	MICB*020:01:02
	170: HLA:HLA27298	MICB*020:01:03
17	171: HLA:HLA02186	MICB*021:01:01N
	172: HLA:HLA27311	MICB*021:01:02N
	173: HLA:HLA27316	MICB*021:01:03N
	174: HLA:HLA28687	MICB*021:01:04N
	175: HLA:HLA28689	MICB*021:01:05N
	176: HLA:HLA28691	MICB*021:01:06N
	177: HLA:HLA28980	MICB*021:01:07N
18	178: HLA:HLA07084	MICB*024:01:01
	179: HLA:HLA26705	MICB*024:01:02
	180: HLA:HLA26723	MICB*024:01:03
	181: HLA:HLA28949	MICB*024:01:04
	182: HLA:HLA28979	MICB*024:01:05
	183: HLA:HLA29307	MICB*024:01:06
19	184: HLA:HLA07085	MICB*025:01:01
	185: HLA:HLA26762	MICB*025:01:02
	186: HLA:HLA27303	MICB*025:01:03
	187: HLA:HLA27314	MICB*025:01:04
	188: HLA:HLA28462	MICB*025:01:05

20	190: HLA:HLA13777	MICB*031:01:01
	191: HLA:HLA26753	MICB*031:01:02
	192: HLA:HLA26689	MICB*031:01:03
	193: HLA:HLA26698	MICB*031:01:04
	194: HLA:HLA26617	MICB*031:01:05
	195: HLA:HLA27306	MICB*031:01:06

Tabulka 3: Rozdělení alel genu MICB