

Západočeská univerzita v Plzni

Fakulta aplikovaných věd

Katedra kybernetiky

DIPLOMOVÁ PRÁCE

Plzeň, 2017

Bc. Adam VONÁŠEK

Západočeská univerzita v Plzni

Fakulta aplikovaných věd

Katedra kybernetiky

**NÁVRH ALGORITMU
PRO IDENTIFIKACI ŘÍDKÝCH
HAPLOTYPŮ U DÁRCŮ KOSTNÍ DŘENĚ**

Plzeň, 2017

Bc. Adam VONÁŠEK

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Adam VONÁŠEK**

Osobní číslo: **A14N0177P**

Studijní program: **N3918 Aplikované vědy a informatika**

Studijní obor: **Kybernetika a řídicí technika**

Název tématu: **Návrh algoritmu pro identifikaci řídkých haplotypů u dárců kostní dřeně**

Zadávací katedra: **Katedra kybernetiky**

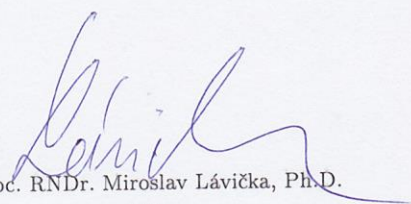
Z á s a d y p r o v y p r a c o v á n í :

1. Prostudujte problematiku řídkých haplotypů a genových vazeb (HLA-A/B/C/DRB1/DQB1).
2. Analyzujte možnosti využití dostupných HLA dat pro využití při identifikaci řídkých haplotypů, včetně možnosti aktualizace znalostí.
3. Navrhněte algoritmus pro identifikaci řídkých haplotypů.
4. Ověřte funkčnost algoritmu prototypovým řešením.

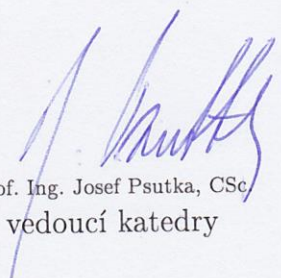
Rozsah grafických prací: dle potřeby
Rozsah kvalifikační práce: 40-50 stránek A4
Forma zpracování diplomové práce: tištěná
Seznam odborné literatury:
<http://allelefrequencies.net/>
<https://bioinformatics.bethematchclinical.org/>

Vedoucí diplomové práce: **Ing. Lucie Houdová, Ph.D.**
Nové technologie pro informační společnost

Datum zadání diplomové práce: **3. října 2016**
Termín odevzdání diplomové práce: **21. května 2017**


Doc. RNDr. Miroslav Lávička, Ph.D.
děkan




Prof. Ing. Josef Psutka, CSc.
vedoucí katedry

V Plzni dne 3. října 2016

PROHLÁŠENÍ

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

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

V Plzni dne 16.8.2017

.....

vlastnoruční podpis

ABSTRAKT

Práce se zabývá vývojem automatické metody identifikace řídkých haplotypů. Haplotypy s nízkou pravděpodobností výskytu je velmi obtížné získat základními metodami, tj. Clarkovým, EM a Bayesovým algoritmem. Nová metoda využívá haplotypy již získané k odvozování nových. Pro identifikaci haplotypu je potřeba najít nejlepší shodu varianty haplotypového páru s údaji v dostupné databázi. Často jsou haplotypy natolik vzácné, tj. řídké, že je potřeba přistoupit k analýze jednotlivých genových vazeb. Nedílnou součástí algoritmu je rozhodovací část, která z variant na základě dostupných parametrů vybírá tu nejvhodnější. Pomocí programu je možné stáhnout populační data z webu, případně stávající zaktualizovat. Taktéž je možné program používat pro porovnávání aktuálních výsledků s již dříve získanými.

KLÍČOVÁ SLOVA

Transplantace kostní dřeně, identifikace haplotypů, alela, řídký haplotyp, genotyp

ABSTRACT

The thesis deals with development of the automatic rare haplotype identification method. Haplotypes of low occurrence probability are very difficult to obtain by using the elementary methods, i.e. Clark algorithm, EM algorithm, Bayes algorithm. The new method uses haplotypes already obtained to derive new ones. It is necessary to find the best matching of genotype variant and data in the website database available, when doing the haplotype identification. Haplotypes can be so rare, so they need to be identified by single genetic linkages analysis. The decisive algorithm is part and parcel of the algorithm. It chooses the most suitable genotype variant from the possible options. The program can download a population data from a website and it can update the current data. The program is able to compare current results of the algorithm with results previously obtained as well.

KEYWORDS

Bone marrow transplantation, haplotype identification, allele, rare haplotype, genotype

Obsah

1.	Transplantace kostní dřeně a HLA systém	12
1.1.	HLA antigeny	12
1.2.	Polymorfismus HLA antigenů	14
1.3.	Nomenklatura.....	14
1.3.1.	Další označení HLA znaků	15
1.4.	Haplotyp.....	16
1.4.1.	Řídký haplotyp.....	17
1.5.	Shoda tkáňových znaků	18
1.6.	Hardy-Weinbergova rovnováha.....	19
1.7.	Metody identifikace haplotypů	19
1.7.1.	Clarkův algoritmus	20
1.7.2.	EM algoritmus	21
1.7.3.	Bayesův algoritmus.....	23
1.8.	Vazba	24
1.9.	Síla vazby.....	24
1.10.	Vazební nerovnováha	24
2.	Návrh metody pro identifikaci řídkých haplotypů.....	26
2.1.	Populace používané pro identifikaci.....	26
2.2.	Verifikační data.....	27
2.3.	Forma vstupních dat.....	28
2.3.1.	Forma dat dárců	28
2.3.2.	Forma populačních dat.....	29
2.4.	Načítání populačních dat	30
2.4.1.	Přístupy k načítání a aktualizaci populačních dat.....	32
2.4.2.	Načítání populačních dat z P2 a P3	32
2.5.	Identifikace haplotypů	33

2.5.1.	Informace o populacích	33
2.5.2.	Příprava dat	35
2.5.3.	Postup identifikace.....	38
2.5.4.	Vazby a kombinace vazeb související s haplotypem varianty.....	42
2.5.5.	Pravděpodobnost kombinace vazeb.....	43
2.5.6.	Hodnoticí funkce vazby	44
2.5.7.	Vyhodnocování polymorfismů	48
2.5.8.	Vážená pravděpodobnost výskytu haplotypového páru v populaci PX	49
2.5.9.	Váhy pravděpodobností kombinací	49
2.5.10.	Vážený součet hodnoticích funkcí	51
2.5.11.	Váhy hodnoticí funkce pro souhrnné populace.....	52
2.5.12.	Součet hodnoticích funkcí.....	52
2.5.13.	Genová vyrovnanost páru v P1	52
2.5.14.	Rozšířená hodnoticí funkce.....	53
2.5.15.	Rozhodovací algoritmus.....	53
2.5.16.	Příklady funkce rozhodovacího algoritmu	56
2.6.	Analýza neshod.....	72
2.7.	Uživatelské rozhraní	78
2.8.	Ukázka výstupu algoritmu	79
2.9.	Struktura programu	82
2.10.	Omezení programu	82
2.11.	Časová náročnost načítacího a identifikačního algoritmu	83
2.12.	Komplikace při implementaci metody pro stahování populačních dat.....	85
2.13.	Komplikace při implementaci identifikačního algoritmu.....	85
2.14.	Optimalizace	86
2.15.	Možnosti vylepšení a rozšíření algoritmu.....	86
2.16.	Původní návrhy implementace algoritmu	89

3.	Závěr	90
4.	Použitá literatura	92

Seznam obrázků

<i>Genetická organizace HLA systému na krátkém raménku 6. chromozomu [5].....</i>	<i>13</i>
<i>Princip Clarkova algoritmu.....</i>	<i>20</i>
<i>Genetická mapa Evropy [22].....</i>	<i>27</i>
<i>Schéma načítacího algoritmu</i>	<i>31</i>
<i>Schéma identifikačního algoritmu</i>	<i>38</i>

Seznam tabulek

<i>Nomenklatura platná od 1. dubna 2010. [2]</i>	15
<i>Postup odvozování haplotypů dle Clarkova algoritmu</i>	20
<i>Postup odvozování haplotypů pomocí EM algoritmu</i>	22
<i>Postup odvozování haplotypů dle Bayesova algoritmu s aplikací koalescenční teorie</i>	23
<i>Pozorované počty jedinců v populacích k červnu 2017</i>	33
<i>Nezávislé kombinace genů v populacích k červnu 2017</i>	34
<i>Polymorfní vazby genů v populacích k červnu 2017</i>	34
<i>Pořadí vyhledávání polymorfismů jinak kompletních haplotypů</i>	39
<i>Preference vyhledaných kombinací vazeb</i>	41
<i>Speciální označení vazeb nalezených aplikací</i>	43
<i>Váhy vazeb kombinací</i>	44
<i>Hodnocení kombinací vazeb</i>	46
<i>Váhy populací při výpočtu hodnoty polymorfismu</i>	48
<i>Váhy lokusů pro výpočet hodnoty polymorfismu</i>	48
<i>Váhy pravděpodobností kombinací</i>	51
<i>Váhy hodnoticí funkce pro různé souhrnné populace</i>	52
<i>Pseudokód rozhodovacího algoritmu</i>	55
<i>Shodnost algoritmického a expertního řešení ve validačních datech</i>	73
<i>Rozdíly algoritmického a expertního řešení ve validačních datech na konkrétních lokusech</i>	73
<i>Časová náročnost různých procesů aplikace k srpnu 2017</i>	84
<i>Parametry počítače, na němž byla aplikace testována</i>	84

1. Transplantace kostní dřeně a HLA systém

U vážně nemocného pacienta může dojít k poklesu krvevorbny natolik výraznému, že je nutné mu transplantovat celou krvevornou tkáň. Při transplantaci kostní dřeně jsou do těla vpraveny zdravé krvevorné buňky, které jsou schopné nahradit původní buňky pacienta. Samotná transplantace není obtížný zákrok. Problém však nastává při výběru vhodného dárce. Po transplantaci organismus pacienta spustí imunitní reakci, při které jsou buňky buď přijaty nebo odvrženy. Aby proces proběhl příznivě, je nutné zohlednit tkáňové HLA znaky pacienta a dárce. Z důvodu velkého množství kombinací znaků je tento problém velmi složitý.

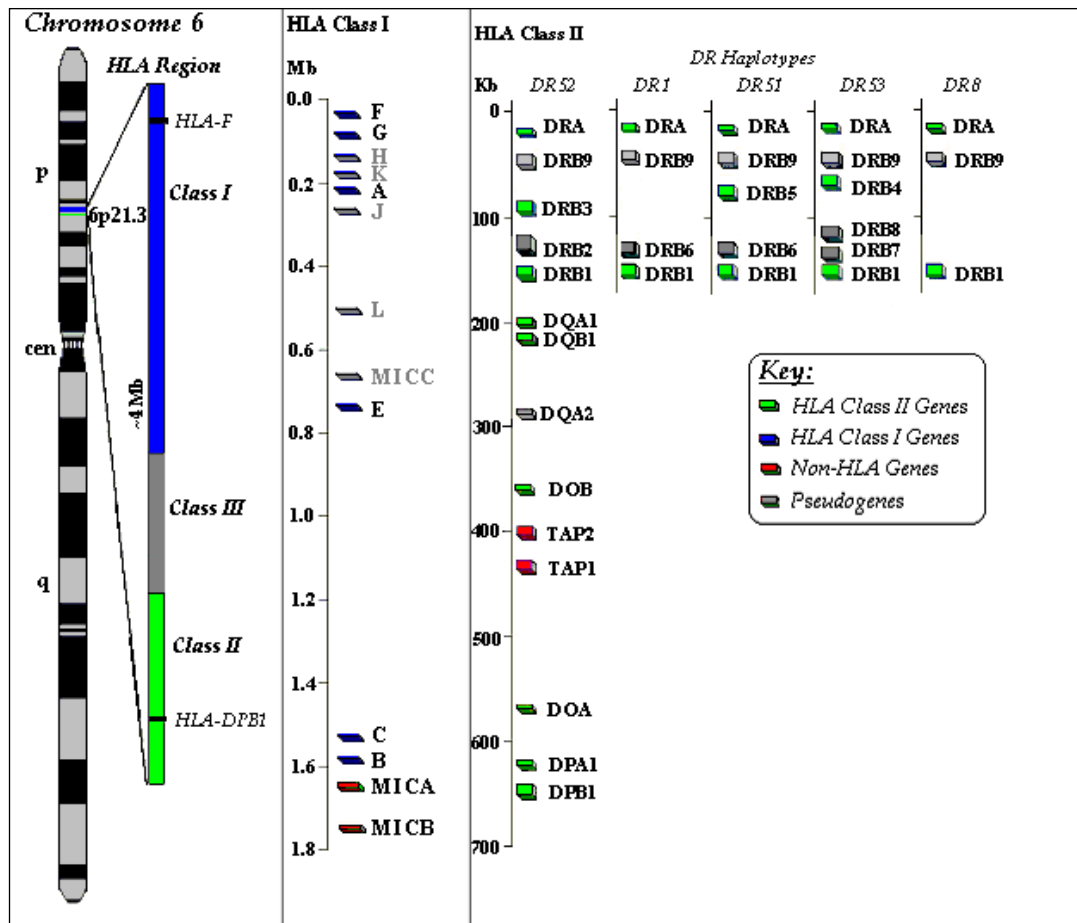
Pro jedince je potřeba určit genotyp, tj. soubor všech zděděných znaků. Proces určování znaků se nazývá HLA typizace. V souboru všech zděděných znaků jedince je poté hledán haplotyp, tj. soubor genů, které mají významnou roli v imunitním systému a dědí se jako jeden celek. Získat haplotypy lze odvozením z genotypových dat. K těmto účelům byly vyvinuty tři základní metody: Clarkův, EM a Bayesův algoritmus. Tyto metody buďto nemohou vždy identifikovat haplotypy řídké, nebo jsou implementačně náročné a vyžadují rozsáhlé předpoklady týkající se populace. Je proto vhodné vyvíjet jiné algoritmy. Tato práce se zabývá návrhem a implementací nové metody pro identifikaci řídkých haplotypů využívající informací o známých haplotypech a vazbách alel.

1.1. HLA antigeny

HLA znamená "Human Leucocyte Antigens" neboli lidské leukocytové antigeny. Tyto antigeny byly poprvé odhaleny na leukocytech, ale jinak se obecně nacházejí na povrchu všech jaderných buněk. [1] HLA jsou skupinou bílkovin, která má důležitou funkci v imunitním systému. Antigeny jsou polypeptidové produkty skupin genů, které se nazývají „Major Histocompatibility Complex“ neboli hlavní histokompatibilní komplex (MHC). HLA molekuly tvoří spolu s imunoglobuliny a receptory T-lymfocytů základní imunitní systém člověka. Jejich funkce v antigen-specifické imunitní odpovědi spočívá v tom, že T-lymfocyty dokáží rozpoznávat antigenní fragmenty a reagovat na ně pouze v rámci HLA molekul – pokud se tyto antigeny vážou na molekuly HLA systému. Komplex obsahuje více než 200 hustě umístěných vysoce polymorfních (tzn. variabilních) genů. Z hlediska strukturálního a funkčního je možné u HLA genů a molekul vymezit dvě základní

skupiny: HLA (anti)geny I. a II. třídy. [2]. I. třída zahrnuje genové lokusy (tzn. místa) *A*, *B*, *C*. II. třída pak zahrnuje lokusy *DRB1* a *DQB1*. [3] Jestliže bude v práci řeč o HLA antigenech bez zmínky o HLA systému, pro stručnost bude označení zobecněno a nazývat se budou geny.

MHC neboli hlavní histokompatibilní komplex je skupina 40–50 genů v lidském organismu, které jsou seřazeny v dlouhém úseku DNA na krátkém raménku 6. chromozomu. HLA komplex je tento soubor nazýván jen u lidí. MHC geny jsou uspořádány do oblastí, které kódují tři třídy MHC molekul. První dvě oblasti MHC zahrnují geny, které kódují HLA antigeny I. a II. třídy, III. třída pak obsahuje produkty, které mají souvislost s imunitními procesy. Pojmy HLA a MHC bývají chybně používány jako synonyma.



Obrázek 1: Genetická organizace HLA systému na krátkém raménku 6. chromozomu [5]

Izotopy v rámci jednotlivých tříd MHC:

- MHC I: *HLA-A*, *-B*, *-C*, klasické izotopy

HLA *-E,-F,-G*: neklasické izotopy mající v porovnání s klasickými geny omezenou buněčnou distribuci a nižší stupeň polymorfismu, z nich *HLA-E* a *HLA-G*: pomáhají k toleranci plodu v děloze

- MHC II: HLA *-DRB1, -DQB1, -DPB1*, klasické izotopy

LMP2, LMP7: bílkoviny štěpící proteiny pro HLA I. Třídy TAP: membránové transportéry

- MHC III: *C4, C2, TNF* (faktor nekrotizující tumory), ... [4]

6. chromozom je nositelem jednoho z nejdůležitějších genetických regionů v našem genomu, které bojují proti nemocem. Mutace v genech na 6. chromozomu jsou zodpovědné za vznik dětské Parkinsonovy nemoci, epilepsie a rakoviny. 6. chromozom lidí byl porovnáván se zvířecím, konkrétně myším a krysím. Vědci identifikovali oblasti (regiony) genů, které jsou přítomny pouze u omezených skupin zvířat. Mohou obsahovat důležité geny nebo regiony, které regulují geny v expresi. Geny exprimují, pokud vyjadřují svou informaci do bílkovinné struktury. [6]

1.2. Polymorfismus HLA antigenů

Antigeny jsou velmi polymorfní. To znamená, že v lokusu určitého genu na daném místě DNA, kde se uvedený gen nachází, je možné najít minimálně dvě různé varianty tohoto genu. Ty se nazývají alely. K tomu, aby měl gen možnost se genotypově projevit, musí jeho lokus obsahovat dvě alely – jednu od matky a jednu otcovu. MHC lokusy obecně jsou geneticky nejvariabilnější kódující lokusy u savců a totéž lze tvrdit i o lidských HLA lokusech. [7] Nejmenší polymorfismus vzniká na genech, které jsou nejbližší centromery 6. chromozomu. Centromera je místo, kde se dotýkají raménka chromozomu. Nejbližší jsou antigeny II. třídy: *DQB1, DRB1* (rovná pozice) a následně I. třídy v pořadí: *A, C, B*. Obecně antigeny I. třídy jsou vysoce polymorfní. [8]

1.3. Nomenklatura

Nomenklatura je univerzální názvosloví označující HLA znaky. Je nezbytná pro popis genotypových dat dárců a populačních dat. Specifické oblastí antigenů jsou odděleny dvojtečkou (např. *A*02:101*). Dále nomenklatura umožňuje pokrýt HLA alely, které mají shodnou sekvenci domén (např. *A*02:01:01G*) nebo také alely, které kódují identické vazebné domény HLA molekuly (např. *A*02:01P*). [9]

Nomenklatura	Význam
<i>HLA</i>	HLA region a předpona pro HLA gen
<i>HLA-DRB1</i>	Specifický HLA lokus, např. zde DRB1
<i>HLA-DRB1*13</i>	Skupina alel, které kódují DR13 antigen
<i>HLA-DRB1*13:01</i>	Specifická HLA alela
<i>HLA-DRB1*13:01N</i>	"Null" alela, tzn. která neexprimuje. To znamená, že alela dané skupiny může nabývat libovolné formy.
<i>HLA-DRB1*13:01:02</i>	Alela, která se liší synonymní mutací
<i>HLA-DRB1*13:01:01:02</i>	Alela s mutací mimo kódující region
<i>HLA-A*30:14L</i>	Alela kódující protein se signifikantně redukovanou nebo "nízkou" expresí na buňkách
<i>HLA-A*24:02:01:02L</i>	Alela kódující protein se signifikantně redukovanou nebo "nízkou" expresí na buňkách a kde se mutace nachází mimo kódující region
<i>HLA-A*44:02:01:02S</i>	Alela kódující protein
<i>HLA-A*03:01:01Q</i>	Alela s mutací ovlivňující expresi na povrchu buněk. Není ovšem dosud jednoznačně potvrzeno a její expresní status je "Questionable".

Tabulka 1: Nomenklatura platná od 1. dubna 2010. [2]

1.3.1. Další označení HLA znaků

V populačních datech jsou též často používány **P kódy**. Jde o HLA sekvence se oblastí spojující stejné antigeny. Analýza je prováděna na proteinové sekvenci a pro HLA alely I. třídy shodnost v oblastech spojujících antigeny je založena na identické proteinové sekvenci zakódované exony 2 a 3. U HLA alel II. třídy je shodnost založena na identické proteinové sekvenci zakódované exonem 2. HLA alely s nukleotidovými sekvencemi, které kódují stejné proteinové sekvence pro oblasti spojující peptidy (exony 2 a 3 pro HLA třídu I. a exon 2 pouze pro HLA alely II. třídy) budou označovány písmenem „P“ po 2 polích označení alel s nejnižší četností ve skupině. Označení skupiny bude obsahovat nejméně

4 číslice. Alely nesekvencované pro část analyzované oblasti jsou stále zahrnuty do analýzy, která je postavena na přítomnosti proteinové sekvence.

Příklad:

Skupina alel označená *A*02:03P* zahrnuje alely:

*A*02:03:01/A*02:03:02/A*02:03:03/A*02:03:04/A*02:03:05/A*02:03:06/A*02:03:07/A*02:03:08/A*02:253/A*02:264/A*02:370/A*02:480/A*02:505/A*02:557/A*02:684*

Velmi často jsou v populačních datech přítomny **G kódy**. Jsou to HLA alely s identickými nukleotidovými sekvencemi napříč exony kódujícími oblasti spojující peptidy.

Příklad:

Skupina alel označená *A*01:03:01G* zahrnuje alely:

01:03:01:01/01:03:01:02 [10]

1.4. Haplotyp

Předmětem transplantační imunologie jsou haplotypy. „Haplotyp je sada alel vázaných na jednom chromozomu, které mají tendenci dědit se společně.“ [11] Jde tedy o soubor alel přenášených na potomka jako celek. Jestliže lze vyzorovat asociaci některého haplotypu s určitou nemocí, je možné ji pak použít pro předpověď rizika vzniku této nemoci u konkrétního jedince. U člověka je pozorován diploidní genotyp, tedy dva haplotypy. Možné varianty genotypu jsou:

A1-B1-C1-DRB1-DQB1/ A2-B2-C2-DRB2-DQB2,
A1-B1-C1-DRB1-DQB2/A2-B2-C2-DRB2-DQB1,
A1-B1-C1-DRB2-DQB1/A2-B2-C2-DRB1-DQB2,
A1-B1-C2-DRB1-DQB1/A2-B2-C1-DRB2-DQB2,
A1-B2-C1-DRB1-DQB1/A2-B1-C2-DRB2-DQB2,
A2-B1-C1-DRB1-DQB1/A1-B2-C2-DRB2-DQB2,
A1-B1-C2-DRB2-DQB1/A2-B2-C1-DRB1-DQB2,
A1-B1-C2-DRB1-DQB2/A2-B2-C1-DRB2-DQB1,
A2-B1-C2-DRB1-DQB1/ A1-B2-C1-DRB2-DQB2,

A2-B1-C1-DRB2-DQB1/A1-B2-C2-DRB1-DQB2,
A2-B1-C1-DRB1-DQB2/A1-B2-C2-DRB2-DQB1,
A1-B2-C2-DRB1-DQB1/A2-B1-C1-DRB2-DQB2,
A1-B2-C1-DRB1-DQB2/A2-B1-C2-DRB2-DQB1,
A1-B1-C1-DRB2-DQB2/A2-B2-C2-DRB1-DQB1,
A1-B2-C1-DRB2-DQB1/A2-B1-C2-DRB1-DQB2,
A2-B2-C1-DRB1-DQB1/A1-B1-C2-DRB2-DQB2

Jednou z možností z identifikace haplotypu je odvodit ho z genotypu rodičů. To však neumožňuje získat kompletní soubor genů. Jiný způsob je použití genotypů jedinců, kteří mají jednoznačně rozluštitelný haplotyp. Tuto metodu však lze použít pouze u haplotypů tvořených malým počtem znaků. Všeobecně použitelné k tomuto účelu jsou tzv. metody „in silico“, tzn. počítačové. Pomocí nich lze odvodit haplotypy z genotypů nepříbuzných jedinců. Podrobně jsou tyto metody rozebrány v kapitole 1.7.

1.4.1. Řídký haplotyp

Práce se zabývá metodou identifikace řídkých haplotypů. [12] Proto je třeba tento pojem vysvětlit. Řídkým haplotypem je označován takový, jehož frekvence je v měřítku určité populace velmi nízká. Je potřeba posoudit, zda tyto nízké frekvence jsou spolehlivé. K tomu lze využít postup, který odhaduje minimální velikost vzorku populace, ze které pak vypočte spolehlivou frekvenci haplotypu. Velikost vzorku je většinou neměnná. Cílem je získat míru spolehlivosti haplotypových odhadů. Nutnou podmínkou spolehlivosti odhadu je přítomnost alespoň jednoho jedince ve vzorku, jenž je nositelem tohoto haplotypu. Jestliže frekvence i -tého haplotypu je p_i a velikost vzorku je n , potom pravděpodobnost, že jedinec nemá i -tý haplotyp, je:

$$\bar{P}_i = (1 - p_i)^2 \quad (1)$$

Pravděpodobnost Q , že aspoň jeden jedinec s i -tým haplotypem se nachází mezi n jedinci, je:

$$Q = 1 - \bar{P}_i = 1 - (1 - p_i)^{2n} \quad (2)$$

Pokud má být dosaženo jisté pravděpodobnosti Q , je možné ji zafixovat jako konstantu a výsledkem je:

$$\ln(1 - Q) = 2n \ln(1 - p_i) \quad (3)$$

$$p_i = 1 - e^{\frac{\ln(1-Q)}{2n}} \quad (4)$$

1.5. Shoda tkáňových znaků

Obtížnost transplantace krvetvorných buněk nespočívá v lékařské části procesu, nýbrž v následné imunitní odpovědi organismu. Bílé krvinky kontrolují shodu tkáňových znaků genotypu příjemce a transplantovaných buněk. V případě, že znaky ve velké míře odpovídají, krvinky buňky ignorují. Jestliže shoda není dostatečná, buňky se uchytí, ale protože se jejich tkáňové znaky neshodují, považují tělo svého nositele za cizí a začnou jej ničit. Protože různých kombinací HLA znaků existují tisíce, najít pro pacienta vhodného dárce je velice obtížné. [4]

Nejlepší vzájemnou shodu HLA znaků je možné nalézt u příbuzenstva, a to z důvodu dědičnosti znaků. Haplotypy generují shodu v HLA systému. Pokud není k dispozici jakýkoli vhodný rodinný dárce (zpravidla u starších osob), je nutné hledat dárce buněk v registru dárců krvetvorných buněk. Při vyhledávání je prioritní snahou zmenšit riziko imunitní reakce. To znamená najít dárce s co nejpodobnějším souborem tkáňových znaků.

Nejdůležitějšími tkáňovými znaky v transplantační imunologii jsou HLA antigeny I. a II. třídy *DRB1* a *DQB1*. [13] Současný požadavek na míru shody je 10/10, tedy v pěti HLA antigenech (konkrétně v HLA *-A*, *-B*, *-C*, *-DRB1* a *-DQB1*). [4] V akutní situaci je možné akceptovat u dárců shodu 9/10, ve výjimečných případech 8/10.

U příbuzenských dárců hledáme hlavně shodu v haplotypech, ze které pak vyplývá i shoda v HLA systému. Je-li možné jednoznačně oddělit haplotypy, pak postačuje sérologická úroveň molekulární typizace („low resolution“, viz níže). U nepříbuzného páru dárce-příjemce je hledána primárně shoda v jednotlivých alelách každého HLA genu, a pro hodnocení shody a uspokojivý výsledek je požadována výhradně typizace na vysoké úrovni rozlišení („high resolution“, viz níže). Typizační metody umožňují získat přesný HLA gen na úrovni sekvence nukleotidů. Takto přesně typizovat genotypy je však zbytečné. Proto se používají dvě úrovně typizace:

- **Nízké rozlišení (low resolution)** – Jeho výsledkem je HLA gen na úrovni alelických skupin, které sdílejí stejné základní polymorfni sekvence (např. A*02).
- **Vysoké rozlišení (high resolution)** – Jeho výsledkem je jednotlivá alela, dle HLA nomenklatury tedy alespoň první 4 číslice (např. A*02:01). Tato úroveň je nezbytná pro nepříbuzenskou transplantaci. [14]

1.6. Hardy-Weinbergova rovnováha

V populacích využívaných pro identifikaci haplotypů je předpokládána platnost Hardy-Weinbergova zákona. Ten říká, že v populaci, která je v genetické rovnováze, se frekvence výskytu genotypů odvíjí pouze od frekvence výskytu alely a frekvence genotypů je stálá z generace na generaci. Jde o předpoklad, který využívají identifikační algoritmy. Tento zákon má také své předpoklady:

- organismy jsou diploidní, tedy obsahují dvě sady chromozomů
- v populaci probíhá rozmnožování
- generace se nepřekrývají
- je brán ohled na jednotlivé lokusy
- populace je nekonečně četná [15]
- nedochází k vnějším vlivům, které mohou měnit frekvence genů a genotypů, tj. migrace, mutace, selekce, genetický drift, čili náhodná změna ve frekvencích alel v dané populaci. [16]

1.7. Metody identifikace haplotypů

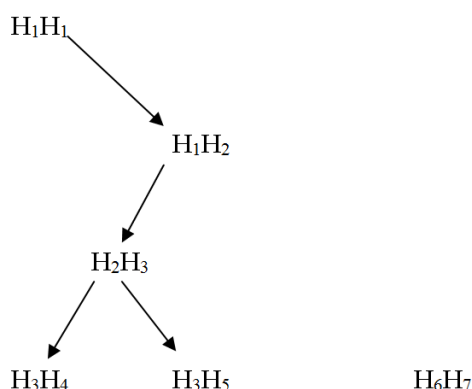
Pro účely identifikace haplotypů z genotypů nepříbuzných osob byly vyvinuty tři základní metody „in silico“:

- **Clarkův algoritmus** – rekonstruuje přímo haplotypy jedinců v souboru.
- **EM algoritmus** – odhaduje frekvence haplotypů v populaci. V dané populaci poté odvozuje haplotypy jedinců v souboru.
- **Bayesovské algoritmy** – určují pravděpodobnosti haplotypových párů daných genotypů.

Popišme si nyní podrobně každou z těchto metod.

1.7.1. Clarkův algoritmus

Jedná se o první algoritmus sloužící pro rekonstrukci haplotypů, které se vyznačují délkou větší než tři polymorfismy. Haplotypy se tímto algoritmem odvozují z genotypů získaných z populace. Při jeho použití se předpokládá, že v dostatečně velkém vzorku jedinců jsou genotypy zcela homozygotní, tzn. s oběma haplotypy shodnými nebo s nejvýše jednou heterozygotní pozicí, tzn. rozdílnou. Takové genotypy jsou jednoznačně rozluštitelné.



Obrázek 2: Princip Clarkova algoritmu

(1) Na počátku rozlušti homozygotní genotypy a s jednou heterozygotní pozicí. Haplotypy, které získáš tímto způsobem, považuj za známé.
(2) Tyto haplotypy se pokoušej přiřadit k zatím nerozluštěným genotypům. Pokud nějaký haplotyp odpovídá, potom jej k danému genotypu přiřaď a druhý haplotyp odvoď tak, aby haplotypový pár odpovídal genotypu.
(3) Tak je získán další známý haplotyp. V hledání pokračuj (krok (2)), dokud existují genotypy, které lze rozluštit.
(4) Odvozování ukonči, pokud již není možné žádné genotypy na základě známých haplotypů rozluštit nebo pokud jsou všechny genotypy už rozluštěné.

Tabulka 2: Postup odvozování haplotypů dle Clarkova algoritmu.

1.7.2. EM algoritmus

Tento iterační algoritmus vyhledává maximální důvěryhodné odhady haplotypových četností za předpokladu platnosti Hardy-Weinbergovy rovnováhy. Zkratka „EM“ pochází ze slov „expectation“ (tzn. očekávání) a „maximization“ (tzn. maximalizace). Nejlepších výsledků algoritmus dosahuje s velkými vzorky z populace a není závislý na míře rekombinace. Rekombinace je změna uspořádání haplotypu vznikající při překřížení odpovídajících částí chromatid analogických chromozomů během meiózy (tzn. buněčného dělení). Při rekombinaci potomek získá jednu alelu z maternálního a druhou alelu z paternálního chromosomu. Proces překřížení se nazývá crossing-over. [17] EM algoritmus je vhodné spouštět s různými počátečními podmínkami, ale na druhou stranu není závislý na pořadí vstupních dat jako Clarkův algoritmus. S rostoucím počtem znaků lze postřehnout exponenciální vzrůst nároků na výpočetní čas. Tento algoritmus je použitelný pro:

- nalezení haplotypů pro daný soubor
- nalezení nejčtetnějších haplotypů
- odhad haplotypových populačních četností
- stanovení nejpravděpodobnějšího haplotypového páru pro každého jedince v souboru.

Algoritmus je založen na předpokladu, že pokud jsou známé haplotypové páry pro dané genotypy, pak mohou posloužit pro odhad četnosti haplotypů, a naopak pokud jsou známé četnosti, pak z nich lze odvodit haplotypy, které odpovídají daným genotypům. [11]

(1) Nastav konvergenční kritérium $\varepsilon < 10^{-v}$ pro přesnost odhadu četností haplotypů.

Obvykle:

$$4 \leq v \leq 8. \quad (5)$$

Nastav maximální počet iterací N .

Obvykle:

$$25 \leq N \leq 100.$$

Nastav počáteční hodnotu věrohodnostní funkce, např. $L^{(-1)} = 1$.

(2) Začni iteraci s hodnotou $s = 0$.
(3) Nastav počáteční podmínky, konkrétně haplotypové četnosti $p_k^{(0)}$, $k=1, \dots, M$.
(4) Dokud $s \leq N$, opakuj následující kroky
(I) Počítej pravděpodobnosti všech přípustných kombinací haplotypů pro dané genotypy
$p^{(s)}(h_k, h_l) \begin{cases} (p_k^{(s)})^2 & k = l \\ 2p_k^{(s)} p_l^{(s)} & k \neq l \end{cases} \quad (6)$
(II) Odhadni pravděpodobnosti genotypů g_1, \dots, g_r jako součet pravděpodobností kombinací haplotypů kompatibilních s daným genotypem vypočtených v předchozím bodě
$p^{(s)}(g_i) = \sum_{k=1}^M \sum_{l=1}^M \hat{p}^{(s)}(h_k, h_l), \text{ kde } i=1, \dots, r \text{ a } k \leq l. \quad (7)$
(III) Spočítej hodnotu věrohodnostní funkce
$\ln L^{(s)} = \sum_{s=1}^r n_{g_i} \ln \hat{p}^{(s)}(g_i) \quad (8)$
a pokud $ L^{(s-1)} - L^{(s)} < \varepsilon$, pokračuj bodem 6.
(IV) Odhadni očekávaný počet haplotypů k , kde $k = 1, \dots, M$.
$\hat{E}^{(s)}(n_{h_k}) = \sum_{i=1}^r \hat{p}^{(s)}(h_k, h_l/g_i), \quad (9)$
kde
$\hat{p}^{(s)}\left(h_k, \frac{h_l}{g_i}\right) = \frac{2 \cdot p_k^{(s)} \cdot p_l^{(s)}}{\hat{p}^{(s)}(g_i)} \text{ pro } h_k \neq h_l \text{ i } h_k = h_l. \quad (10)$
(v) Aktualizuj haplotypové četnosti:
$p_k^{(s)} = n_{h_k}^{(s)} / 2n. \quad (11)$
(5) Ukonči běh algoritmu bez konvergence.
(6) Ukonči běh algoritmu, odhady haplotypových četností jsou $p_k^{(s)}$.

Tabulka 3: Postup odvozování haplotypů pomocí EM algoritmu

1.7.3. Bayesův algoritmus

Jedná se o třetí základní metodu identifikace haplotypů. Tato metoda pracuje s neznámými haplotypy jako s neznámou náhodnou veličinou. Jejím principem je určit podmíněné rozdělení pravděpodobnosti haplotypů v závislosti na pozorovaných genotypech. Mějme množinu neznámých haplotypových párů $H = (H_1, \dots, H_n)$ pro n osob, resp. n známých genotypů $G = (G_1, \dots, G_n)$. V algoritmu je využíván Gibbsův vzorkovač, tj. druh algoritmu Markovských řetězců Monte Carlo. Pomocí něj je získán řetězec odhadů haplotypových párů a odhad $P(H/G)$. Ten vyjadřuje pravděpodobnost, že haplotypový pár odpovídá genotypu vzhledem k celému souboru genotypů. Bayesův algoritmus je založen na koalescenční teorii. Ta říká, že následující vybraný haplotyp bude tím více podobný těm předchozím, čím více osob bylo sledováno a čím menší je mutační míra. Pokud je následující postup zopakován v dostatečné míře, je získán odhad $P(H/G)$. [18]

(1) Náhodně vyber jedince i ze všech osob s nejednoznačným genotypem. Tyto opakované výběry jsou prováděny rovnoměrně.
(2) Odvoď $H_i^{(t+1)}$ z podmíněné pravděpodobnosti $P(H_i/G, H_{-i}^{(t)})$, kde H_{-i} jsou všechny haplotypové páry mimo páru jedince i . Předpokládej, že tito ostatní nevybraní jedinci mají správně rekonstruované haplotypy.
(3) Proveď úpravu $H_j^{(t+1)} = H_j^{(t)}$ pro $j = 1, \dots, n, j \neq i$.

Tabulka 4: Postup odvozování haplotypů dle Bayesova algoritmu s aplikací koalescenční teorie.

Předmětem našeho zájmu by měla být metoda, která dokáže identifikovat jakýkoli haplotyp, tedy i řídký. V bakalářské práci již bylo dokázáno, že takovou vlastnost Clarkův ani EM algoritmus nemají. Bayesův algoritmus je implementačně obtížný a stojí na předpokladu koalescenční teorie. Identifikace řídkých haplotypů je obecně obtížná. Nová metoda by měla být postavena vzájemné vazební nerovnováze genů. Zabývejme se proto nyní touto problematikou.

1.8. Vazba

Pro pochopení vazební nerovnováhy je nejprve nutné vysvětlit pojem vazba. Vazba je vlastnost genů, která tvoří výjimku z Mendelových zákonů dědičnosti. Jedním z těchto zákonů je pravidlo o volné kombinovatelnosti vloh. Podle něho se distribuce znaků v samčích a samičích pohlavních buňkách řídí základními statistickými zákony, díky kterým je možné předpovídat zastoupení znaků u potomstva. [19]

1.9. Síla vazby

Sílu vazby lze vyhodnotit na základě vzdálenosti antigenů na krátkém raménku 6. chromozomu. Nejsilnější vazba vzniká mezi geny na lokusech *B* a *C*. Za druhou nejsilnější je možné považovat *B,DRB1*. Na tyto dvě pak navazuje lokus *A*, kde ale často dochází k rekombinaci pro jeho velkou vzdálenost od ostatních používaných HLA lokusů. Mohou vzniknout vazby *A,B,C*, dále *B,C,DRB1* a *A,B,DRB1*. Další silnou vazbou je *DRB1,DQB1*. Ta umožňuje vznik vazby *B,DRB1,DQB1*.

1.10. Vazební nerovnováha

V populační genetice vazební nerovnováha označuje nenáhodnou asociaci alel různých lokusů. Vyskytuje se tehdy, pokud mezi geny na dvou lokusech existuje vzájemná závislost. Určité kombinace alel na dvou či více lokusech se tedy vyskytují v populaci častěji či méně často, než by odpovídalo jejich náhodné kombinaci. Genové lokusy se dostávají do vazební nerovnováhy, pokud frekvence asociace jejich rozdílných alel je vyšší nebo nižší než očekávané hodnoty u nezávislých a náhodně asociovaných lokusů. Vazební nerovnováha je ovlivněna řadou faktorů, včetně míry rekombinace, míry mutace, genetického driftu, systému páření, populační struktury a genetické vazby. Výsledkem šablony vazební nerovnováhy v genomu je intenzivní signál složený z populačních genetických procesů. [20]

Navzdory svému názvu může vazební nerovnováha existovat mezi alelami na různých lokusech bez vzájemné genetické vazby a nezávisle na tom, zda frekvence alel jsou v rovnováze (nemění se v čase) či ne. Vazební rovnováha bývá též uváděna jako nerovnováha genetické fáze. Nicméně tento koncept platí i pro asexuální organismy, a proto záleží na přítomnosti gamet.

Vazební nerovnováha se využívá v evoluční biologii a v genetice pro lepší porozumění evolučním a demografickým událostem. Dále slouží pro zmapování genů, které souvisí s výskytem měřitelných znaků (výška, váha, hodnoty krevního tlaku) a dědičných chorob. Jako další příklad využití lze uvést pochopení společného vývoje propojených množin genů. [21]

Jestliže mezi lokusy existuje nenáhodná asociace, pak lze tohoto jevu využít při identifikaci řídkých haplotypů. Důležitá je pak především samotná existence dané asociace a teprve v druhé řadě pravděpodobnost výskytu této asociace. Tato asociace může být považována za genovou vazbu. Ze skladby genových vazeb nalezených pro konkrétní haplotypový pár je možné usoudit, zda tento pár je skutečnou variantou haplotypového páru daného genotypu. Tento princip umožňuje identifikovat také řídké haplotypy.

2. Návrh metody pro identifikaci řídkých haplotypů

Metoda pro identifikaci řídkých haplotypů je postavená na znalosti vazeb haplotypů příslušících určitému genotypu. Program nejprve vygeneruje všechny varianty haplotypového páru a ke každé prozkoumáním veškerých populačních dat najde vazby genů, které přísluší těmto haplotypům. Aplikace používá jako vstup data o dárcích a populační data. Pracuje se třemi souhrnnými populacemi, které mají při různou váhu a preferenci. Pomocí složitých funkcí vybírá nejvhodnější variantu haplotypového páru. Uživatel může zvolit rozsah používaných populací a soubor, ze kterého budou načítána data o dárcích.

2.1. Populace používané pro identifikaci

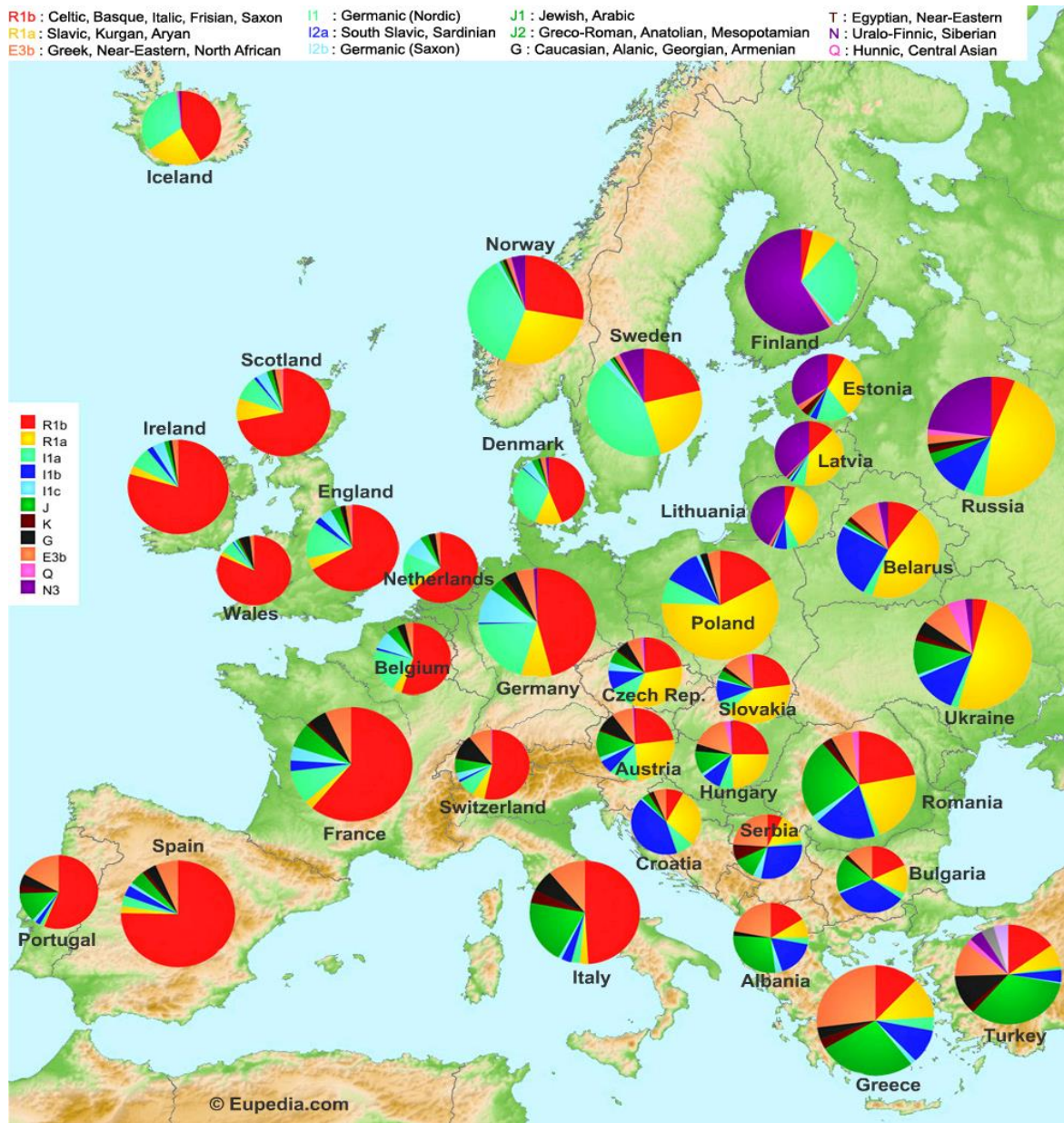
Program pracuje se třemi populacemi seřazenými vzestupně podle genetické vzdálenosti od populace české. Jsou to:

P1 – Souhrnná populace Poland DKMS, Poland a Germany DKMS – Austria minority. Tyto populace jsou geneticky nejbližší populaci české.

P2 – Souhrnná populace regionu Europe, státu Russia, populace USA Caucasian, USA NMDP Caucasian pop 2 a USA NMDP European Caucasian. Pod populace uvedené jako USA Caucasian neboli americkou europoidní populaci, patří evropští přistěhovalci. Tudíž je pravděpodobné, že genotyp jejich příslušníků se bude ve velké míře shodovat s genotypy příslušníků populace evropské. V databázi se do základní populace Europe z ruských populací řadí pouze Russian Tatars. V *P2* však budou zahrnuty všechny.

P3 – Souhrnná populace mimoevropských regionů, tj. Australia, South Africa, North-East Asia, South and Central America, South Asia, South-East Asia, Sub-Saharan Africa, Western Asia. Jde tedy o zbylé populace nezahrnuté v *P1* ani *P2*.

Pro zdůvodnění rozdělení na populace poslouží obrázek 3. Mapa ukazuje genetické složení evropských zemí podle haplotypových skupin. Tyto skupiny mají společného předka a jsou ukazatelem genetického složení populace.



Obrázek 3: Genetická mapa Evropy [22]

2.2. Verifikační data

Pro vývoj metody byly použity dva soubory genotypových dat. První soubor je od 74 dárců. Druhý soubor je od 100 dárců. K dispozici též byly expertní výsledky identifikace haplotypů z těchto dat. Expertní řešení prvního souboru pochází z roku 2015, druhé z roku 2017.

2.3. Forma vstupních dat

Jak již bylo výše uvedeno, vstupy souboru jsou: data dárců a populační data. Popišme si, jakou formu mají mít soubory obsahující tato data.

2.3.1. Forma dat dárců

Data o dárcích lze do programu načítat ze souboru ve formátu *txt*. Pokud jde jen o identifikaci, je nutné, aby existovaly soubory *nazev.txt* a *nazev_indexy.txt*. Text „*nazev*“ může být zastoupen libovolným jiným, ovšem pro oba soubory stejným. Soubor *nazev_indexy.txt* musí mít jeden sloupec a stejný počet řádků jako soubor *nazev.txt*. Soubor s „*_indexy*“ obsahuje indexy (resp. ID) dárců na každém řádku. Výsledků pro jeden index může být několik, ale v souboru indexů musí být na příslušných řádcích stejný index.

Jestliže uživatel žádá i porovnání s předešlým (resp. expertním) řešením, musejí existovat také soubory *nazev_spravne.txt* a *nazev_indexy_spravne.txt*. První soubor musí mít formát stejný jako soubor *nazev.txt* a druhý musí mít formát stejný jako soubor *nazev_indexy.txt*. Dále soubory *nazev_spravne.txt* a *nazev_indexy_spravne.txt* musejí mít stejný počet řádků. Indexy (resp. ID) musejí odpovídat řešením.

Příklad:

Názvy sady souborů mohou vypadat takto:

genotypy.txt

genotypy_indexy.txt

genotypy_spravne.txt

genotypy_indexy_spravne.txt

Forma řádku souborů *nazev.txt* a *nazev_spravne.txt* je následující:

$S_{iA1}:A_{iA1} S_{iA2}:A_{iA2} S_{iB1}:A_{iB1} S_{iB2}:A_{iB2} S_{iC1}:A_{iC1} S_{iC2}:A_{iC2} S_{iDRB1}:A_{iDRB1} S_{iDR2}:A_{iDRB2} S_{iDQB1}:A_{iDQB1}$
 $S_{DQB2}:A_{DQB2}$

kde:

S_{iX} je alelická skupina genu X dárce s indexem i

A_{ix} je konkrétní forma alely genu X dárce s indexem i

Jednotlivé alely jsou odděleny tabulátorem.

Příklad:

Jeden řádek souboru *nazev.txt*:

31:01 02:01 35:01 15:01:01G 04:01:01G 03:04 14:01 11:01 03:01 05:03

Příklad:

Jeden řádek souboru *nazev_indexy.txt*:

654123

Příklad:

Ukázka souborů pro více řešení připadajících na jeden index:

První dva řádky souboru *nazev.txt*:

31:01 02:01 35:01 15:01:01G 04:01:01G 03:04 14:01 11:01 03:01 05:03
01:01 31:01 35:01 15:01:01G 04:01:01G 03:04 14:01 11:01 05:03 03:01

První dva řádky souboru *nazev_indexy.txt*:

654123
654123

2.3.2. Forma populačních dat

Populační data lze do programu načíst ze souboru ve formátu *csv*. Název souboru má formát *tabulka_Px.csv*, kde x označuje index populace. Soubor má zpravidla velké množství řádků. Jeden řádek představuje informaci o vazbě určitých genů. Tuto vazbu lze označit v .

Forma řádku je následující:

„A*“ +	„B*“ +	„C*“ +	„DRBI*“	„DQBI*“	<i>ppst</i>	n_P	<i>nazev_P</i>
<i>Valela_A</i>	<i>Valela_B</i>	<i>Valela_C</i>	+ <i>Valela_{DRBI}</i>	+ <i>Valela_{DQBI}</i>			

kde:

$valela_x$ je údaj o alele lokusu X podmnožiny v , který může být uveden v libovolném rozlišení
ppst je pravděpodobnost podmnožiny v , u které musí být uveden znak „!“ na místě desetinné čárky

n_P je počet pozorovaných příslušníků populace P , v které se nachází podmnožina v lokusů určitého haplotypu

n_{zvez_P} je název populace P , v které se nachází podmnožina v lokusů určitého haplotypu.

Název populace P je pouze doplňující a přehledová informace. Při identifikaci nemá žádný smysl. Populace jsou totiž do souhrnných populací rozděleny už při načítání populačních dat. Pro úplnost se však populace uvádí. Pokud alela není ve vazbě definována, potom za znakem „*“ následuje označení *null*.

Příklad:

Jeden řádek populačních dat:

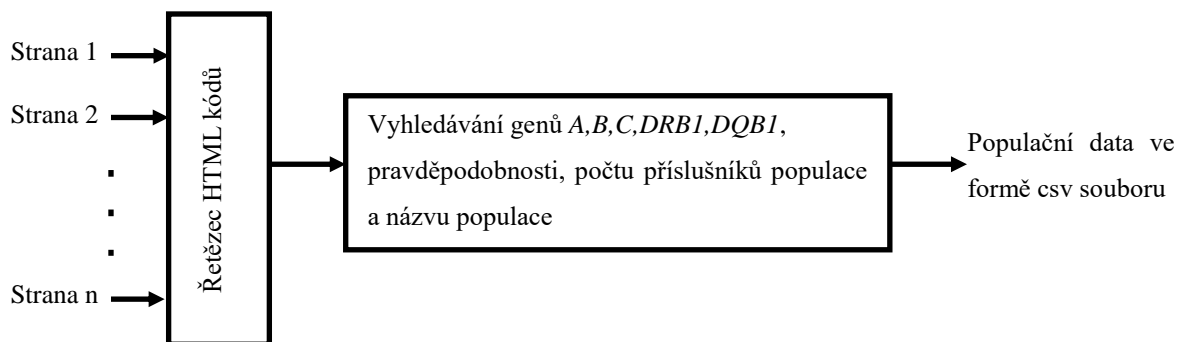
<i>A*01:01</i>	<i>B*07:02</i>	<i>C*07:01</i>	<i>DRB1*15:01</i>	<i>DQB1*null</i>	<i>0!0880</i>	<i>1698</i>	<i>Germany DKMS - Austria minority</i>
----------------	----------------	----------------	-------------------	------------------	---------------	-------------	--

2.4. Načítání populačních dat

Program je navržen tak, aby mohla být identifikace prováděna off-line. Předtím je ale nutno získat populační data. Protože je však z webu <http://allelefrequencies.net> nelze jednoduše stáhnout, je nutné je vyhledávat v HTML kódu. Webová aplikace databáze haplotypů využívá stránkování. Algoritmus je implementován v jazyce Java a vyvíjen v prostředí Eclipse SDK. HTML kódy stránek pro sledované populace řetězí program za sebe. K tomu je využíván cyklus while. Algoritmus ověřuje, zda počet znaků dané stránky dosahuje nejméně 797 000. To je experimentálně zjištěná hranice, která indikuje, že na stránce se nacházejí jakékoliv informace o haplotypech. V opačném případě je třeba cyklus ukončit, protože stránka prokazatelně žádné informace o haplotypech neobsahuje. Přístup k jednotlivým stránkám je řešen přes následující řetězcovou kostru, na kterou je nabalováno číslo stránky XYZ.

http://allelefrequencies.net/hla6003a.asp?page=XYZ&hla_selection=&hla_locus1=&hla_locus2=&hla_locus3=&hla_locus4=&hla_locus5=&hla_locus6=&hla_locus7=&hla_locus8=&hla_population=&hla_country=&hla_dataset=&hla_region=&hla_ethnic=&hla_study=&hla_sample_size=&hla_sample_size_pattern=equal&hla_sample_year=&hla_sample_year_pattern=equal&hla_loci=&hla_order=order_1

Po zřetězení HTML kódů následuje načítání haplotypů. To je realizováno průchodem dlouhého řetězce obsahujícího HTML všech stránek a porovnáváním. Jednotlivé haplotypy jsou nejprve ukládány do dvojrozměrného pole. Po skončení načítání je obsah tohoto pole uložen do souboru *tabulka_Px.csv*. Tento soubor je automaticky vytvořen. Důležité je, aby v momentě ukládání dat do souborů nebyly tyto soubory otevřeny. Pak okamžitě dochází k chybě, běh programu se přeruší a zápis do souboru není proveden. Při aktualizaci (opakovaném načítání) populačních dat jsou stávající soubory přepsány. Postup načítacího algoritmu je ukázán na obrázku 4.



Obrázek 4: Schéma načítacího algoritmu

Původní plán algoritmu počítal s možností tvorby rozdílového souboru. Do něj by se ukládaly pouze rozdíly mezi populačními daty uloženými na disku a novými daty z webu. Bylo by ho tedy možné použít k analýze změn v populačních datech. U rozdílového souboru se však objevuje problém, že vazby a haplotypy mohou v databázi nejen přibývat, ale i ubývat. Vytvořit generátor souboru, který by zaznamenal jak přírůstky, tak úbytky, je velmi složité. Vzhledem k malému využití takového souboru je na zvážení, zda se takovým problémem vůbec zabývat. Proto bylo v této práci od vývoje generátoru rozdílového souboru upuštěno.

2.4.1. Přístupy k načítání a aktualizaci populačních dat

K ukládání webové databáze haplotypů požadovaných populací lze přistupovat dvěma způsoby:

- a) Načtení kompletního souboru.
- b) Vytvoření aktualizovaného souboru na základě již dříve vytvořeného souboru s populačními daty

Bylo nutné posoudit, který způsob je rychlejší a který by v případě nejvyšší nutnosti mohl poskytnout populační data v co nejkratším čase. Aktualizovaného souboru se týkají stejné problémy jako rozdílového. Vzhledem k možnosti nárůstu ale i úbytku populačních dat, je implementace a výpočetní složitost metody pro aktualizaci populačních dat vysoká. Nejnáročnější činnost je načítání stran z webu. To je nutné jak při tvorbě kompletního, tak i aktualizovaného souboru, a to ve stejném rozsahu. V případném aktualizovaném souboru je navíc zahrnuta kontrola změn a opravy dat, čímž se čas přípravy prodlužuje. Zápis do *csv* souboru je pro každou případnou délku zanedbatelně krátký. Proto je vždy výhodnější načíst znovu celý soubor a nikoli opravovat stávající.

2.4.2. Načítání populačních dat z P2 a P3

Načítání těchto dat *P2* a *P3* se liší od načítání dat *P1* a je časově velmi náročné. Nejen z důvodu velkého objemu těchto dat, ale též kvůli načítání každé velikosti vzorku podřadné populace zvlášť. Rovněž problémem je nejednotnost dat, která musí být ošetřena mnoha podmínkami, což výpočetní nároky algoritmu opět zvyšuje. Data jsou často nekompletní a informace o alelách na některých genech vůbec neobsahují. Je však možné tento proces urychlit. Posloupnost některých řádků za sebou náleží stejné populaci. Vysvětleme si to na příkladu populace Germany DKMS – Turkey minority. Protože nejprve algoritmus zjišťuje název populace, lze ho porovnat s předchozím a v případě shody přiřadit novému haplotypu stejnou velikost vzorku, kterou získal předchozí haplotyp. Tím pádem už algoritmus nemusí vyhledávat velikost vzorku populace Germany DKMS – Turkey minority, protože z předchozího řádku už je známý. Na druhou stranu nebylo by efektivní ukládat do seznamu všechny nalezené populace s velikostí vzorků a pak u každého řádku v populačních datech porovnávat s uloženými názvy.

2.5. Identifikace haplotypů

Tato kapitola se zabývá částí algoritmu, která slouží k identifikaci haplotypů. Jejím cílem je vysvětlit postup této metody, uvést všechny použité parametry a poskytnout příklady funkce tohoto algoritmu.

2.5.1. Informace o populacích

Jedním z původních návrhů metody byl proces vyhledávání haplotypů s využitím síly vazeb. Vazby, které jsou jednoznačné, tzn. jeden konkrétní gen je závislý výhradně na jiném a naopak, lze považovat za velmi důvěryhodný důkaz existence haplotypu. Takových vazeb však existuje v nejlepším případě naprosté minimum, tudíž tuto metodu nelze k vyhledávání použít. Tento návrh též využíval polymorfismy ve smyslu závislosti jedné alelické skupiny genu na konkrétním genu. Využívání vazeb nezávislých a nezaručujících tak vysokou kvalitu nalezeného haplotypu však přineslo mnohem lepší výsledky než využívání jednoznačných vazeb a polymorfismů. Následující tabulky přinášejí zdůvodnění. Uvedme nejprve počty pozorovaných jedinců v populacích.

Populace	Počet jedinců
<i>P1</i>	22 551
<i>P2</i>	3 727 767
<i>P3</i>	2 728 353
celosvětově	6 478 671

Tabulka 5: Pozorované počty jedinců v populacích k červnu 2017

Další důležitý údaj je počet výskytů různých vazeb v populaci. Ten je obsahem tabulky 6. Mezi vazbami skládajícími se ze dvou genů má obecně nejsilnější zastoupení *B,DRB1*. Nejméně zastoupena je naopak kombinace *DRB1,DQB1*. Důvodem je, že populační data jsou nekompletní a většinou chybí informace o genu *DQB1*.

Vazba/ Populace	<i>B,C</i>	<i>B,DRBI</i>	<i>DRBI,DQB1</i>	<i>A,B,C</i>	<i>B,C,DRBI</i>	<i>A,B,DRBI</i>	<i>A,B,C,DRBI</i>	Haplotypy
<i>P1</i>	524	1331	9	2147	2524	5201	6737	13
<i>P2</i>	1138	2902	435	4218	4981	12551	12551	3930
<i>P3</i>	1197	2937	545	3705	4202	8720	8729	6754
celosvětově	1872	4412	794	16510	7734	16510	19595	10189

Tabulka 6: Nezávislé kombinace genů v populacích k červnu 2017

Postupným vývojem algoritmu a opětovným načítáním populačních dat se ukázalo, že jednoznačné vazby celosvětově ani v žádné z populací neexistují. Nelze tedy spolehlivě identifikovat haplotypy na základě jednoznačných vazeb, ačkoliv by šlo o velmi vlivný faktor.

Podobně situace vypadá v případě polymorfismů. Zde je uvažován polymorfismus se závislostí jednoho lokusu na alelické skupině na jiném lokusu, kde tato alelická skupina je polymorfní. Takovou závislost je možné nazvat vazba se striktním polymorfismem. Pokud by vazby byly vyhledávané celosvětově, vznikne takový polymorfismus u vazeb *pB-C* (polymorfismus na *B*) a *pB,DRBI*, který v běžně v souhrnných populacích neexistuje.

Vazba/ Populace	<i>B,pC</i>	<i>pB,C</i>	<i>B,pDRBI</i>	<i>pB,DRBI</i>	<i>DRBI,pDQB1</i>	<i>pDRBI,DQB1</i>
<i>P1</i>	2	1	0	0	0	0
<i>P2</i>	7	4	0	0	1	2
<i>P3</i>	4	1	1	0	3	0
celosvětově	9	6	0	3	2	0

Tabulka 7: Polymorfní vazby genů v populacích k červnu 2017

2.5.2. Příprava dat

Pro přípravu na spuštění algoritmu je nutné načíst všechny možné vazby B,C , $B,DRB1$, A,B,C , $B,C,DRB1$, $A,B,DRB1$ a též celé haplotypy $A,B,C,DRB1$. V populacích, kde je definován gen na lokusu $DQB1$, se k těmto vazbám přidají $DRB1,DQB1$ a tím pádem i haplotypy $A,B,C,DRB1,DQB1$. Během načítání probíhá při každém nalezeném případě vazby inkrementace pravděpodobnosti výskytu. Inkrementaci popisuje rovnice 12.

$$p_N = p_D + 0,01 \frac{f_N n_P}{n_{Vx}} \quad (12)$$

kde:

p_N je nová pravděpodobnost výskytu vazby

p_D je dosavadní pravděpodobnost výskytu vazby

f_N je nová nalezená frekvence vazby

n_P je počet jedinců v příslušné populaci

n_V je počet jedinců v souhrnné populaci PX , pro $x = 1, 2, 3$

Pokud je pro konkrétní typ vazby (např. B,C) součet pravděpodobností všech vazeb v jedné souhrnné populaci roven 1, je tak dokázáno, že všechny vazby v konkrétní populaci byly nalezeny. Jestliže se v populačních datech vyskytuje P kód, je použita každá alela v posloupnosti jako samostatný záznam. S nulovými alelami je pracováno jako s běžnými. V populačních datech se sice vyskytují G kódy, ale protože jsou záležitostí alel se synonymní mutací, není nutné se jimi v této práci zabývat. Algoritmus je totiž nastaven pouze na vysokou úroveň rozlišení.

Poté, co jsou předpřipraveny soubory, lze je použít k vyhledávání. Soubory s populačními daty jsou načteny do dvojrozměrného pole. Genotypové údaje o dárcích jsou pak načítány z textového souboru. Je vypočten celkový počet pozorovaných jedinců v P_x , pro $x = 1,2$ nebo 3 . Je též potřeba načíst údaje o genotypech dárců.

Mějme genotyp jednoho z dárců. Ten je tvořen dvěma haplotypy. Je potřeba najít nejvhodnější variantu haplotypového páru. Rozložme genotyp na všechny výše uvedené varianty.

Příklad:

Mějme genotyp:

Gen	A	B	C	DRBI	DQB1
1.	02:01	18:01	04:01	01:01	03:02
2.	68:01	35:03	07:36	04:03	05:01

Tyto geny je možné rozložit na následující varianty haplotypového páru:

Varianta	A	B	C	DRBI	DQB1
1	02:01	18:01	04:01	01:01	03:02
	68:01	35:03	07:36	04:03	05:01
2	02:01	18:01	04:01	01:01	05:01
	68:01	35:03	07:36	04:03	03:02
3	02:01	18:01	04:01	04:03	03:02
	68:01	35:03	07:36	01:01	05:01
4	02:01	18:01	07:36	01:01	03:02
	68:01	35:03	04:01	04:03	05:01
5	02:01	35:03	04:01	01:01	03:02
	68:01	18:01	07:36	04:03	05:01
6	68:01	18:01	04:01	01:01	03:02
	02:01	35:03	07:36	04:03	05:01
7	02:01	18:01	07:36	04:03	03:02
	68:01	35:03	04:01	01:01	05:01
8	02:01	18:01	07:36	01:01	05:01
	68:01	35:03	04:01	04:03	03:02
9	68:01	18:01	07:36	01:01	03:02
	02:01	35:03	04:01	04:03	05:01
10	68:01	18:01	04:01	04:03	03:02

	02:01	35:03	07:36	01:01	05:01
11	68:01	18:01	04:01	01:01	05:01
	02:01	35:03	07:36	04:03	03:02
12	02:01	35:03	07:36	01:01	03:02
	68:01	18:01	04:01	04:03	05:01
13	02:01	35:03	04:01	01:01	05:01
	68:01	18:01	07:36	04:03	03:02
14	02:01	18:01	04:01	04:03	05:01
	68:01	35:03	07:36	01:01	03:02
15	02:01	35:03	04:01	04:03	03:02
	68:01	18:01	07:36	01:01	05:01
16	68:01	35:03	04:01	01:01	03:02
	02:01	18:01	07:36	04:03	05:01

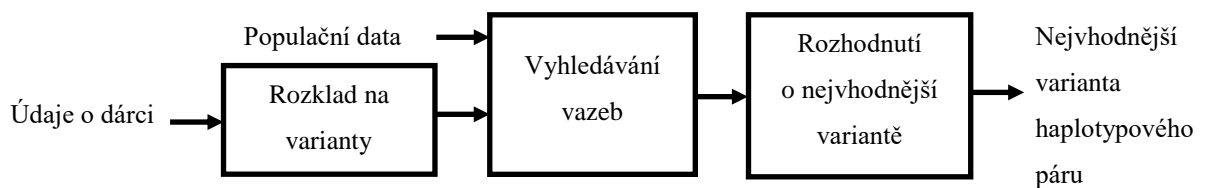
Nyní je potřeba z optimalizačních důvodů eliminovat takové varianty, které se v seznamu nacházejí vícekrát než jednou. Jde o případy, kdy oba potenciální haplotypy obsahují jeden nebo více homozygotních lokusů.

Příklad:

Gen	A	B	C	DRBI	DQBI
1.	02:01	18:01	04:01	01:01	03:02
2.	02:01	35:03	04:01	04:03	05:01

U zbylých případů je potřeba řešit, která varianta nejlépe odpovídá skutečnému haplotypu, který je obsažen v databázi některé populace. To lze posoudit na základě existence vazeb mezi geny na určitých lokusech. V ideálním případě existuje kompletní shoda s haplotypem v populaci včetně alel, a to v obou z haplotypové dvojice. Protože takto kvalitní důkaz existence haplotypu se však vyskytuje jen zřídka, je nutné postupně slevovat z požadavků, a přesto se snažit o identifikaci na co nejvyšší úrovni kvality. V takových

případech je nutné připouštět v některých lokusech existenci polymorfismů. Celý tento proces je konstruován jako prohledávání stavového prostoru. Se stoupající přípustností existence polymorfismu klesá kvalita nalezeného haplotypu. Protože informace o vazbách v *PI* je při rozhodování často nedostačující, algoritmus nezávisle na tom, jaké našel vazby v *PI*, prozkoumá všechny ostatní populace. Vhodnost haplotypového páru je pak určena na základě několika faktorů. Primárně jde o populaci, kvalitu nalezených vazeb a míru polymorfismu. Pro údaje o jednom dárci je schéma identifikačního algoritmu ukázáno na obrázku 5.



Obrázek 5: Schéma identifikačního algoritmu

2.5.3. Postup identifikace

Pro každý haplotyp každé varianty je definován seznam nalezených haplotypů a seznam nalezených kombinací vazeb. V populačních datech i datech dárců je využívána informace na vysoké úrovni rozlišení. Nejsou využívány geny typizované na nízké úrovni rozlišení, protože informace o nich je bezpředmětná. Pro jeden haplotyp a každou souhrnnou populaci postupně se používají následující kroky.

1) Jednoznačná shoda

Nejspolehlivější způsob nalezení haplotypu je porovnání shody s populačními daty. Pokud je nalezen haplotyp zcela shodný, proces pokračuje prohledáváním následující populace opět od bodu 1), protože výše hodnocenou vazbu už ve stejné populaci najít nelze. Haplotyp se tak ukládá do seznamu nalezených haplotypů. Tento proces je zařazen pro úplnost algoritmu. Nefunguje ale pro vyhledávání řídkých haplotypů. Většinou se totiž stává, že u určité varianty pouze jeden z haplotypů páru je nalezen kompletně a druhý neexistuje. Při rozhodování o nejvhodnější variantě pak dochází k výběru jiné.

2) Shoda s přípustností polymorfismu alel

Často nelze nalézt haplotyp přímou shodou alel. Proto je předpokládáno, že na některém genovém lokusu nebude nalezena shodná alela, ale tato alelická skupina bude existovat s polymorfismem. Přítomnost polymorfismu na alelické skupině některého lokusu

dává vyšší šanci, že haplotyp existuje, než když je na lokusu alela jiná a ojedinělá. Jsou uvažovány polymorfismy na jednom nebo dvou lokusech. Pokud je vyhledáván polymorfismus na dvou lokusech, musí se vyskytovat na obou současně. Varianty přítomnosti polymorfismů v nalezených haplotypech jsou znázorněny v tabulce 8. Nalezeným haplotypům se stejným počtem lokusů s polymorfismem je pak přidělena stejná váha nezávisle na tom, o které geny se jedná.

Pořadí	Pozice s polymorfismem
1.	<i>C</i>
2.	<i>A</i>
3.	<i>B</i>
4.	<i>DRB1</i>
5.	<i>DQB1</i>
6.	<i>A,B</i>
7.	<i>B,C</i>
8.	<i>C,DRB1</i>
9.	<i>DRB1,DQB1</i>
10.	<i>A,C</i>
11.	<i>B,DRB1</i>
12.	<i>C,DQB1</i>
13.	<i>A,DRB1</i>
14.	<i>B,DQB1</i>

Tabulka 8: Pořadí vyhledávání polymorfismů jinak kompletních haplotypů

3) Předpoklad existence haplotypu sestaveného z elementárních vazeb

Důležitou část procesu představuje identifikace haplotypu pomocí elementárních vazeb. Principem je vyhledání kombinací vazeb B,C , $B,DRB1$, $DRB1,DQB1$, A,B,C , $A,B,C,DRB1$, $A,B,DRB1,DQB1$ a $B,C,DRB1,DQB1$ ve stejné populaci. Jestliže existuje kombinace vazeb $A,B,C,DRB1$, $A,B,DRB1,DQB1$ a $B,C,DRB1,DQB1$ současně v populaci PI , je pravděpodobné, že haplotyp bude existovat. Takovému haplotypu je tedy třeba přiřadit velkou váhu. Tento postup již umožňuje identifikovat řídké haplotypy. Preference kombinací nalezených vazeb se nachází v tabulce 9. Jeden řádek představuje jednu kombinaci. Ať už je kompletně shodná kombinace vazeb nalezena nebo ne, přistupuje se k hledání polymorfismů ve stejné kombinaci. Každá z vazeb kombinace je zkoumána na existenci polymorfismu, jestliže neexistuje kompletní shoda. V kroku 3) se vyhodnocuje polymorfismus pouze na jednom lokusu. Pokud kombinace zahrnuje lokusy A , B , C , $DRB1$ i $DQB1$, je uložena do seznamu haplotypů, jinak je uložena do seznamu kombinací vazeb.

Pořadí	Použitá kombinace			
1.	$A,B,C,DRB1$	$A,B,DRB1,DQB1$	$B,C,DRB1,DQB1$	-
2.	$A,B,C,DRB1$	$A,B,DRB1,DQB1$	-	-
3.	$A,B,C,DRB1$	$B,C,DRB1,DQB1$	-	-
4.	$A,B,DRB1,DQB1$	$B,C,DRB1,DQB1$	-	-
5.	$A,B,DRB1$	$B,C,DRB1,DQB1$	-	-
6.	$B,C,DRB1$	$A,B,DRB1,DQB1$	-	-
7.	$B,DRB1,DQB1$	$A,B,C,DRB1$	-	-
8.	$DRB1,DQB1$	$A,B,C,DRB1$	-	-
9.	A,B,C	$A,B,DRB1$	$B,DRB1,DQB1$	$B,C,DRB1$
10.	A,B,C	$A,B,DRB1$	$DRB1,DQB1$	$B,C,DRB1$
11.	A,B,C	$A,B,DRB1$	$B,DRB1,DQB1$	-
12.	$A,B,DRB1$	$B,C,DRB1$	$B,DRB1,DQB1$	-
13.	A,B,C	$B,C,DRB1$	$B,DRB1,DQB1$	-

14.	<i>A,B,C</i>	<i>A,B,DRB1</i>	<i>DRB1,DQB1</i>	-
15.	<i>A,B,C</i>	<i>B,C,DRB1</i>	<i>DRB1,DQB1</i>	-
16.	<i>A,B,DRB1</i>	<i>B,C</i>	<i>DRB1,DQB1</i>	-
17.	<i>A,B,C</i>	<i>B,DRB1</i>	<i>DRB1,DQB1</i>	-
18.	<i>A,B,C,DRB1</i>	-	-	-
19.	<i>A,B,DRB1,DQB1</i>	-	-	-
20.	<i>B,C,DRB1,DQB1</i>	-	-	-
21.	<i>A,B,C</i>	<i>DRB1,DQB1</i>	-	-
22.	<i>A,B,DRB1</i>	<i>DRB1,DQB1</i>	-	-
23.	<i>A,B,C</i>	<i>A,B,DRB1</i>	-	-
24.	<i>B-C</i>	<i>B,DRB1</i>	<i>DRB1,DQB1</i>	-
25.	<i>A,B,C</i>	<i>A,B,DRB1</i>	<i>B,C,DRB1</i>	-
26.	<i>A,B,C</i>	-	-	-
27.	<i>A,B,DRB1</i>	-	-	-
28.	<i>B,C,DRB1</i>	-	-	-
29.	<i>B,C</i>	<i>B,DRB1</i>	-	-
30.	<i>B,C</i>	<i>DRB1,DQB1</i>	-	-
31.	<i>B,DRB1</i>	-	-	-
32.	<i>B,C</i>	-	-	-
33.	<i>DRB1,DQB1</i>	-	-	-

Tabulka 9: Preference vyhledaných kombinací vazeb

2.5.4. Vazby a kombinace vazeb související s haplotypem varianty

Každý haplotyp každé varianty se vyznačuje určitým počtem souvisejících vazeb, které bylo možné nalézt v jednotlivých populacích. Soubor určitých vazeb nalezených v téže populaci tvoří kombinaci. Jestliže bude v práci řeč o souvisejících vazbách, bude pojem „vazba“ zobecněn na „kombinace“. Součástí kombinace je informace o obsažených vazbách, pravděpodobnosti v dané populaci a indexu populace.

V aplikaci bylo zvoleno speciální označení těchto vazeb, které se liší od běžného označení lokusů genů doposud používaného v diplomové práci. Jestliže bude řeč o genech obecně, bude používáno běžné označení. Jestliže bude řeč o kombinacích získaných programem, bude používáno speciální označení. Ta jsou uvedena v tabulce 10. V kódu je též uvedeno, zda konkrétní vazba z kombinace byla nalezena kompletně nebo s polymorfismem. Ten je vyjádřen písmen p , za kterým následuje označení daného polymorfního genu.

Označení	Význam
<i>ABCDDq</i>	Kompletně shodný haplotyp
<i>ABCDDqpg_{1...pg_k}</i>	Shodný haplotyp s polymorfismem na genech g_1 až g_k
<i>ABCD</i>	Shoda na genech $A,B,C,DRBI$
<i>ABCDpg_k</i>	Shoda na genech $A,B,C,DRBI$ haplotyp s polymorfismem na genu g_k
<i>ABDD</i>	Shoda na genech $A,B,DRBI,DQB1$
<i>ABDDqpg_k</i>	Shoda na genech $A,B,DRBI,DQB1$ s polymorfismem na genu g_k
<i>BCDD</i>	Shoda na genech $B,C,DRBI,DQB1$
<i>BCDDqpg_k</i>	Shoda na genech $B,C,DRBI,DQB1$ s polymorfismem na genu g_k
<i>ABC</i>	Shoda na genech A,B,C
<i>ABCpg_k</i>	Shoda na genech A,B,C s polymorfismem na genu g_k
<i>ABD</i>	Shoda na genech $A,B,DRBI$
<i>ABDpg_k</i>	Shoda na genech $A,B,DRBI$ s polymorfismem na genu g_k

<i>BCD</i>	Shoda na genech <i>B,C,DRB1</i>
<i>BCDpg_k</i>	Shoda na genech <i>B,C,DRB1</i> s polymorfismem na genu <i>g_k</i>
<i>BDD</i>	Shoda na genech <i>B,DRB1,DQB1</i>
<i>BDDpg_k</i>	Shoda na genech <i>B,DRB1,DQB1</i> s polymorfismem na genu <i>g_k</i>
<i>BC</i>	Shoda na genech <i>B,C</i>
<i>BCpg_k</i>	Shoda na genech <i>B,C</i> s polymorfismem na genu <i>g_k</i>
<i>BD</i>	Shoda na genech <i>B,DRB1</i>
<i>BDpg_k</i>	Shoda na genech <i>B,DRB1</i> s polymorfismem na genu <i>g_k</i>
<i>DrDq</i>	Shoda na genech <i>B,DRB1</i>
<i>BDpg_k</i>	Shoda na genech <i>B,DRB1</i> s polymorfismem na genu <i>g_k</i>

Tabulka 10: Speciální označení vazeb nalezených aplikací

2.5.5. Pravděpodobnost kombinace vazeb

Každá kombinace je ohodnocena pravděpodobností. Pravděpodobnost kombinace *KMB* v populaci *PX* je vypočtena jako:

$$p_{KMB,PX} = \prod_{k=1}^K W_{V_{kp}} \sum_{y=1}^Y p_{V_{ky},PX} \quad (13)$$

kde:

K je počet vazeb v kombinaci

Y je počet variant alely v polymorfismu v populaci *PX*

V_{ky} je *y*-tá varianta polymorfismu *k*-té vazby kombinace *KMB*

W_{V_{ky}} je váha *k*-té vazby kombinace *KMB*

p_{V_{ky},PX} je pravděpodobnost vazby *V_{ky}* v populaci *PX*

Polymorfismus je v tomto vztahu zobrazen. Pokud $P = 1$, jde o alelu s jedinou variantou. Pokud $P > 1$, jde o polymorfismus. V tabulce 11 jsou uvedeny váhy vazeb v kombinaci. Byly zvoleny na základě experimentů na verifikačních datech.

Vazba	<i>ABCDDq</i>	<i>ABCD</i>	<i>ABDD</i>	<i>BCDD</i>	<i>ABC</i>	<i>ABD</i>	<i>BCD</i>	<i>BDD</i>	<i>BC</i>	<i>BD</i>	<i>DD</i>
Váha	80	60	50	30	20	20	10	10	1	1	1

Tabulka 11: Váhy vazeb kombinací

Vysvětlení souvisejících vazeb mohou podat následující příklady.

Příklady:

B,C shoda v *P2* s pravděpodobností 0,5 : *BC*, 0.5, 2

B,C shoda v *P1* s pravděpodobností 0,4 : *BC*, 0.4, 1

A,B,C shoda s polymorfismem na *B* v *P3* s pravděpodobností 0,2 : *ABCpB*, 0.2, 3

A,B,C,DRB1 s polymorfismem na *C* v *P1* s pravděpodobností 0,3 :

ABCDpC, 0.3, 1

Kompletně shodný haplotyp nalezený v *P2* s pravděpodobností 0,001 : *ABCDDq*, 0.001, 2

Kombinace *B,DRB1-B,C-DRB1,DQB1* nalezená v *P1* s pravděpodobností 0,005 :

BD-BC-DrDq,0.005,1

Polymorfismus je započítán pouze jako jedna vazba, jejíž pravděpodobnost je tvořena součtem pravděpodobností kombinací tvořených danou variantou. Tím je zvýšena váha této vazby.

2.5.6. Hodnoticí funkce vazby

Pro určování kvality nalezeného haplotypu byla zavedena hodnoticí funkce vazby. Hodnota $f_{G,HX}$ je přiřazena X-tému haplotypu v potenciálním haplotypovém páru *G* při rozhodovacím algoritmu. Jde o hashovací funkci, ve které je nejlepší nalezené kombinaci přiděleno desetinné číslo, a to pak označuje X-tý haplotyp v páru *G*. Způsob hodnocení je uveden v tabulce 12.

Kombinace vazeb				Populace		
				1	2	3
				Hodnocení		
<i>ABCDDq</i>	-	-	-	139,8	92	39
<i>ABCD</i>	<i>ABDD</i>	<i>BCDD</i>	-	139,4	90	37
<i>ABCD</i>	<i>ABDD</i>	-	-	139,2	89,2	36,2
<i>ABCD</i>	<i>BCDD</i>	-	-	139,19	89,19	36,19
<i>ABDD</i>	<i>BCDD</i>	-	-	139,175	89,175	36,175
<i>ABC</i>	<i>BCDD</i>	-	-	139	87	35
<i>ABD</i>	<i>BCDD</i>	-	-	139	87	35
<i>BCD</i>	<i>ABDD</i>	-	-	139	87	35
<i>BDD</i>	<i>ABCD</i>	-	-	139	87	35
<i>DrDq</i>	<i>ABCD</i>	-	-	138	86	34
<i>BD</i>	<i>ABDD</i>	-	-	137,9	85,9	33
<i>ABC</i>	<i>ABD</i>	<i>BDD</i>	<i>BCD</i>	137,9	85,9	33
<i>ABC</i>	<i>ABD</i>	<i>DrDq</i>	<i>BCD</i>	137,8	85,8	33
<i>ABC</i>	<i>ABD</i>	<i>BDD</i>	-	137	84	31
<i>ABD</i>	<i>BCD</i>	<i>BDD</i>	-	137	84	31
<i>ABC</i>	<i>BCD</i>	<i>BDD</i>	-	137	84	31
<i>ABC</i>	<i>ABD</i>	<i>DrDq</i>	-	136	83	30
<i>ABC</i>	<i>BCD</i>	<i>DrDq</i>	-	135	82	29
<i>ABD</i>	<i>BC</i>	<i>BDD</i>	-	134	81	28
<i>ABC</i>	<i>BDD</i>	-	-	134	81	28
<i>ABD</i>	<i>BC</i>	<i>DrDq</i>	-	133	80	27
<i>ABC</i>	<i>BD</i>	<i>DrDq</i>	-	133	80	27

<i>ABC</i>	<i>DrDq</i>	-	-	131	78	25
<i>ABCD</i>	-	-	-	128	75	22
<i>ABDD</i>	-	-	-	127,8	74,8	21,8
<i>BCDD</i>	-	-	-	127,5	74,5	21,5
<i>ABC</i>	<i>ABD</i>	<i>BCD</i>	-	127	74	21
<i>ABC</i>	<i>ABD</i>	-	-	125	72	19
<i>ABD</i>	<i>BDD</i>	-	-	124,5	71,5	18,5
<i>BCD</i>	<i>BDD</i>	-	-	124	71	18
<i>ABC</i>	<i>BD</i>	-	-	123,5	71	17
<i>ABD</i>	<i>BC</i>	-	-	123,5	71	17
<i>ABD</i>	<i>DrDq</i>	-	-	123,4	70,9	16,9
<i>BC</i>	<i>BDD</i>	-	-	122	69	16
<i>BCD</i>	<i>DrDq</i>	-	-	122	69	16
<i>ABC</i>	-	-	-	121,1	68,1	15,1
<i>ABD</i>	-	-	-	121	68	15
<i>BCD</i>	-	-	-	120	67	14
<i>BDD</i>	-	-	-	120	67	14
<i>BC</i>	<i>BD</i>	<i>DrDq</i>	-	119	66	12
<i>BC</i>	<i>BD</i>	-	-	114	61	8
<i>BD</i>	<i>DrDq</i>	-	-	114	61	8
<i>BC</i>	<i>DrDq</i>	-	-	114	61	8
<i>BC</i>	-	-	-	110	57	4
<i>BD</i>	-	-	-	110	57	4
<i>BC</i>	-	-	-	110	57	4

Tabulka 12: Hodnocení kombinací vazeb

Hodnoticí funkci byly přiděleny hodnoty podle kvality kombinace souvisejících vazeb. Konkrétní hodnoty jsou výsledkem experimentů na verifikačních datech. Vyhodnocování polymorfismů je popsáno v kapitole 2.5.7. Toto rozhodnutí bylo kromě teoretických znalostí založeno na subjektivním posudku. Nejvyšší hodnota je přiřazena kompletně shodným haplotypům z *P1*. U nejlepších kombinací *P1* byl zvolen malý rozestup, protože v celé populaci až na výjimky nebyla uvedena informace o genu *DQB1*. Nelze tedy v *P1* uvažovat výrazný rozdíl např. mezi *ABCD* a *ABCDDq*. Pro hodnocení se používají desetinná čísla, která umožní tento rozestup zmírnit.

Kompletně shodnému haplotypu z *P3* je přiřazena hodnota mnohem nižší z důvodu genetické vzdálenosti *P1* a *P3*. Mezi vazbami nalezenými v *P1*, *P2* a *P3* je dostatečný bodový rozestup, aby vzdálené populace nemohly ovlivňovat kombinace nalezené v *P1*. Kompletní haplotyp z *P3* je ohodnocen s odstupem níže než nejslabší vazba v *P2*. Jinak je trend poklesu obdobný jako u *P1* a *P2*. Obecně vazbám *ABC* a *ABD* je přidělována vyšší váha než vazbám *BCD* a *BDD* z důvodu genetické blízkosti na raménku 6. chromozomu. Větší význam těchto vazeb je zohledněn ve většině existujících kombinací. Byl kladen vyšší důraz na možnost sestavit haplotyp za pomoci slabších vazeb než nalezení jedné silnější, ale nedostačující vazby. Proto tedy *ABC-BD-DrDq* získalo vyšší hodnocení než vazba *ABCD*.

Funkce byla laděna tak, aby bylo dosaženo co nejlepší shody s expertními výsledky. Algoritmus je zaměřen na *P1*. Pokud by data o *P1* byla rozšířena i o informaci o *DQB1*, bylo by vhodné přidělit nejlepším kombinacím z *P1* ještě vyšší hodnocení, ovšem nesmělo by způsobit výsledný výběr nevyrovnaného páru (tzn. takového s jednou silnou a jednou slabou kombinací vazeb). Informace o *P3* jsou načítány, pokud je uživatel vyžaduje. Při rozhodování je však kladen největší důraz na *P1* a *P2* nejen díky vysokému hodnocení jejich kombinací, ale také proto, že v konečném rozhodnutí v drtivé většině případů vystačí informace o *P1* a *P2*, a na *P3* se nedostane. Tento poznatek plyne z používání algoritmu na verifikačních datech. Informace o *P3* se však hodí v případě, že daný genotyp obsahuje pouze vazby spadající do *P3*. Informace o kombinacích nalezených v *P3* jsou tedy důležité.

Hodnoticí funkce zajišťuje, že upřednostňovány jsou vždy varianty s oběma kompletně nalezenými haplotypy v *P1*. Pokud se taková varianta objeví a je jediná, je zřejmé, že jde o nejvhodnější variantu. Vyhledávání však musí nadále pokračovat v dalších populacích, protože v případě nutnosti pak lze využít údaje o vazbách nalezených v *P2* a *P3*. Stejně tak se neukončuje vyhledání jedné kombinace vazeb s různou účastí polymorfismů. Může se totiž stát, že příští nalezená kombinace vazeb bude lépe hodnocená než první

nalezená kvůli nižšímu počtu polymorfismů. Takový přístup sice značně prodlouží dobu trvání identifikace, ale zato ji učiní spolehlivější.

2.5.7. Vyhodnocování polymorfismů

Protože kombinace s polymorfismy lze považovat za slabší než „stejně“ kombinace bez nich, je při přítomnosti polymorfismů v kombinaci od hodnotící funkce kombinace odečtena následující oslabující hodnota:

$$v_{pol,HX} = sum(W_{L-pol})W_{pop} + 0,1d \quad (14)$$

kde:

$sum(W_{L-pol})$ je součet vah lokusů s polymorfismem vyskytujícím se u nejlepší kombinace pro haplotyp HX .

W_{pop} je váha populace pro výpočet hodnoty polymorfismu

d je rozdíl počtu všech polymorfismů v kombinaci a počtu různých lokusů s polymorfismem

Jestliže je nalezen kompletní haplotyp s polymorfismy na B nebo C , je výpočet prováděn jinak. V takovém případě je hledána ve stejném haplotypu a aktuálně zkoumané populaci kompletně shodná kombinace B,C . Jestliže existuje, je hodnotící funkci kombinace přičtena hodnota 5. Důvodem tohoto kroku je velká síla vzájemné vazby genů B a C .

Populace	$P1$	$P2$	$P3$
Váha	2,7	5	5,2

Tabulka 13: Váhy populací při výpočtu hodnoty polymorfismu

Lokus	A	B	C	$DRB1$	$DQB1$
Váha	0,85	1	0,9	0,8	0,8

Tabulka 14: Váhy lokusů pro výpočet hodnoty polymorfismu

Oslabující hodnotu ovlivňuje váha populace. U *PI* je nutné být ve vyhodnocování polymorfismů benevolentní. S dalšími populacemi však přítomnost polymorfismů musí být hodnocena mnohem přísněji.

2.5.8. Vážená pravděpodobnost výskytu haplotypového páru v populaci *PX*

Vážená pravděpodobnost se využívá v rozhodovacím algoritmu. Při vyhodnocování součtu pravděpodobností nejlepších kombinací dané populace je nutné každé kombinaci přiřadit váhu. Ač byla váha přidělována již jednotlivým vazbám, je potřeba vypočítat ji také pro celou kombinaci. To zdůrazní důležitost této kombinace. Mějme haplotypový pár *G*. Váženou pravděpodobnost výskytu *G* v populaci *PX* lze vypočítat pomocí následujícího vztahu:

$$p_{W,G} = 10^{-7}(p_{G, KMB1,PX}W_{KMB1} + p_{G, KMB2,PX}W_{KMB1}) \quad (15)$$

kde:

$p_{G,KMBY,PX}$ je pravděpodobnost výskytu nejlepší kombinace *Y*-tého haplotypu v páru *G* v populaci *PX*

W_{Y} je váha pravděpodobnosti kombinace *KMBY*

Koeficient 10^{-7} má normalizační význam. Konkrétním vahám se věnuje kapitola 2.5.9.

2.5.9. Váhy pravděpodobností kombinací

Na základě verifikačních dat byly kombinacím vazeb přiděleny váhy. Využívají se při výpočtu pravděpodobnosti kombinace vazeb. Jsou nastaveny tak, aby při porovnávání kvality haplotypového páru podle pravděpodobností bylo zvoleno co nejvhodnější řešení. Hodnota váhy není jakkoli závislá na indexu populace. Váhy slouží k výpočtu, který se vztahuje pouze k jedné souhrnné populaci. Tyto váhy se odlišují od vah vazeb v kombinaci, které byly uvedeny výše v tabulce 12.

Kombinace vazeb				Váha
<i>ABCDDq</i>	-	-	-	4000000
<i>ABCD</i>	<i>ABDD</i>	<i>BCDD</i>	-	3000000
<i>ABCD</i>	<i>ABDD</i>	-	-	2000000
<i>ABCD</i>	<i>BCDD</i>	-	-	2000000
<i>ABDD</i>	<i>BCDD</i>	-	-	2000000
<i>ABC</i>	<i>BCDD</i>	-	-	1900000
<i>ABD</i>	<i>BCDD</i>	-	-	1900000
<i>BCD</i>	<i>ABDD</i>	-	-	1900000
<i>BDD</i>	<i>ABCD</i>	-	-	1900000
<i>DrDq</i>	<i>ABCD</i>	-	-	1800000
<i>BD</i>	<i>ABDD</i>	-	-	1800000
<i>ABC</i>	<i>ABD</i>	<i>BDD</i>	<i>BCD</i>	1800000
<i>ABC</i>	<i>ABD</i>	<i>DrDq</i>	<i>BCD</i>	1500000
<i>ABC</i>	<i>ABD</i>	<i>BDD</i>	-	700000
<i>ABD</i>	<i>BCD</i>	<i>BDD</i>	-	700000
<i>ABC</i>	<i>BCD</i>	<i>BDD</i>	-	700000
<i>ABC</i>	<i>ABD</i>	<i>DrDq</i>	-	650000
<i>ABC</i>	<i>BCD</i>	<i>DrDq</i>	-	650000
<i>ABD</i>	<i>BCD</i>	<i>DrDq</i>	-	650000
<i>BC</i>	<i>ABD</i>	<i>BDD</i>	-	550000
<i>ABC</i>	<i>BDD</i>	-	-	550000
<i>ABD</i>	<i>BC</i>	<i>DrDq</i>	-	480000
<i>ABC</i>	<i>BD</i>	<i>DrDq</i>	-	480000
<i>ABC</i>	<i>DrDq</i>	-	-	350000
<i>ABCD</i>	-	-	-	1000
<i>ABDD</i>	-	-	-	1000

<i>BCDD</i>	-	-	-	1000
<i>ABD</i>	<i>DrDq</i>	-	-	60
<i>ABC</i>	<i>ABD</i>	<i>BCD</i>	-	200
<i>ABC</i>	<i>ABD</i>	-	-	100
<i>BC</i>	<i>BDD</i>	-	-	10
<i>DrDq</i>	<i>BCD</i>	-	-	10
<i>BC</i>	<i>BD</i>	<i>DrDq</i>	-	0,0001
<i>ABC</i>	-	-	-	0,01
<i>ABD</i>	-	-	-	0,01
<i>BCD</i>	-	-	-	0,001
<i>BDD</i>				0,001
<i>BC</i>	<i>BD</i>	-	-	0,00001
<i>BC</i>	<i>DrDq</i>	-	-	0,00001
<i>BC</i>	-	-	-	0,000001
<i>BD</i>	-	-	-	0,000001
<i>DD</i>	-	-	-	0,000001

Tabulka 15: Váhy pravděpodobností kombinací

2.5.10. Vážený součet hodnoticích funkcí

Vážený součet je dalším z faktorů rozhodovací funkce. V případě, že nelze vybrat nejvhodnější variantu pomocí znalostí o *P1*, je nutné kombinovat znalosti o populacích *P1* a *P2*, resp. o *P1*, *P2* i *P3*. Mějme genotyp *G*. Vážený součet hodnoticích funkcí genotypu *G* v množině souhrnných populací je možné vypočítat podle následujícího vztahu:

$$F_{G,W} = \sum_{z=1}^{z \in \{min,max\}} W_z (f_{G,KMB_1,PZ} + f_{G,KMB_2,PZ}) \quad (16)$$

kde:

min je index dolní hranice množiny souhrnných populací

max je index horní hranice množiny souhrnných populací

$f_{KMB_X,PZ}$ je hodnoticí funkce pro nejlepší kombinaci KMB X-tého haplotypu varianty genotypu G v populaci PZ

W_z je váha populace PZ

Konkrétním vahám se věnuje kapitola 2.5.11.

2.5.11. Váhy hodnoticí funkce pro souhrnné populace

Vážený součet hodnoticích funkcí pro variantu, který se používá pro rozhodování, vyžaduje vážení hodnocení kombinace koeficientem podle příslušné populace. S rostoucí genetickou vzdáleností od české populace se váha snižuje. Hodnoty vah byly zvoleny experimentálně. Uvedeny jsou v tabulce 16.

Populace	$P1$	$P2$	$P3$
Váha	1	0,4	0,05

Tabulka 16: Váhy hodnoticí funkce pro různé souhrnné populace

2.5.12. Součet hodnoticích funkcí

Součet hodnoticích funkcí je jeden z nejdůležitějších faktorů rozhodovacího algoritmu. Mějme haplotypový pár G . Součet hodnoticích funkcí páru G je vypočten jako:

$$F_G = f_{G,H1} + f_{G,H2} \quad (17)$$

kde:

$f_{G,HX}$ je hodnota hodnoticí funkce X-tého haplotypu páru G

2.5.13. Genová vyrovnanost páru v P1

Genová vyrovnanost je součástí rozšířené hodnoticí funkce, která je rozebírána v kapitole 2.5.14. Tato hodnota je postavena na heuristickém předpokladu, že větší zastoupení genů v populacích může vyjadřovat silnější kombinace vazeb. Mějme haplotypový pár G . Jako vyrovnanost páru G se označuje vlastnost, kdy celkový počet

různých vazebních genů v páru nalezených v PI je co nejvyšší a zároveň je mezi těmito počty na obou haplotypech co nejmenší rozdíl. Genová vyrovnanost páru v PI je vypočtena pomocí následujícího vztahu:

$$V_{G,PI} = 10(n_{G,H1,PI} + n_{G,H2,PI} - |n_{G,H1,PI} - n_{G,H2,PI}|) + F_G \quad (18)$$

kde:

$n_{G,HX,PI}$ je počet různých genů nacházejících se v souvisejících kombinacích X -tého haplotypu páru G nalezených v PI

Koeficient 10 při výpočtu genové vyrovnanosti byl určen experimentálně.

2.5.14. Rozšířená hodnoticí funkce

Rozšířená hodnoticí funkce je nejdůležitější faktor rozhodovacího algoritmu. Mějme haplotypový pár G . Rozšířenou hodnoticí funkci páru G vypočteme jako:

$$F_{R,G} = V_{p,PI} F_G \quad (19)$$

kde:

$V_{G,PI}$ je genová vyrovnanost páru v PI

2.5.15. Rozhodovací algoritmus

Rozhodovací algoritmus je část identifikačního algoritmu, která slouží k výběru nejvhodnější varianty haplotypového páru z nabízené množiny variant za pomoci dostupných parametrů.

Mějme soubor variant $x = 1, \dots, n$ vzniklých rozkladem genotypu G . Mějme dvojice haplotypů $H_{x,1}$ a $H_{x,2}$. Potom pro výběr haplotypového páru genotypu G lze použít následující algoritmus.

(1)	Vytvoř seznam potenciálních haplotypových párů.
(2)	Pro všechny haplotypové páry v seznamu vypočti rozšířenou hodnoticí funkci každý haplotyp urči nejlepší související kombinaci vazeb. Nejvyšší z hodnot ulož do proměnné <i>max</i> .
(3)	Projdi všechny potenciální haplotypové páry. Každý, který má hodnoticí funkci nižší než <i>max</i> odeber ze seznamu. Jestliže délka seznamu haplotypových párů je 1, ukonči běh algoritmu. Jinak pokračuj bodem (4).
(4)	Pro všechny haplotypové páry v seznamu vypočti součet hodnoticích funkcí. Nejvyšší z hodnot ulož do proměnné <i>max</i> .
(5)	Projdi všechny potenciální haplotypové páry. Každý, který má hodnotu součtu hodnoticích funkcí nižší než <i>max</i> , odeber ze seznamu. Jestliže délka seznamu haplotypových párů je 1, ukonči běh algoritmu. Jinak pokračuj bodem (6).
(6)	Pro všechny haplotypové páry v seznamu vypočti součet pravděpodobností nejlepších souvisejících kombinací daného haplotypu v <i>P1</i> . Nejvyšší z hodnot ulož do proměnné <i>max</i> .
(7)	Projdi všechny potenciální haplotypové páry. Každý, který má hodnotu vážené pravděpodobnosti nejlepších souvisejících kombinací v <i>P1</i> nižší než <i>max</i> , odeber ze seznamu. Jestliže délka seznamu haplotypových párů je 1, ukonči běh algoritmu. Jinak pokračuj bodem (8).
(8)	Pro všechny haplotypové páry v seznamu vypočti součet hodnoticích funkcí pro vazby z <i>P1</i> a <i>P2</i> . Nejvyšší z hodnot ulož do proměnné <i>max</i> .
(9)	Projdi všechny potenciální haplotypové páry. Každý, který má hodnotu součtu hodnoticích funkcí pro vazby z <i>P1</i> a <i>P2</i> nižší než <i>max</i> , odeber ze seznamu. Jestliže délka seznamu haplotypových párů je 1, ukonči běh algoritmu. Jinak pokračuj bodem (10).
(10)	Pro všechny haplotypové páry v seznamu vypočti součet hodnot hodnoticí funkce pouze pro <i>P2</i> a pro každý haplotyp urči nejlepší související kombinaci. Nejvyšší z hodnot součtu funkce ulož do proměnné <i>max</i> .
(11)	Projdi všechny potenciální haplotypové páry. Každý, který má hodnotu součtu hodnoticích funkcí pouze pro <i>P2</i> nižší než <i>max</i> , odeber ze seznamu. Jestliže délka seznamu haplotypových párů je 1, ukonči běh algoritmu. Jinak pokračuj bodem (12).
(12)	Pro všechny haplotypové páry v seznamu vypočti vážený součet pravděpodobností nejlepších souvisejících kombinací daného haplotypu v <i>P2</i> . Nejvyšší z hodnot ulož do proměnné <i>max</i> .

(13)	Projdi všechny potenciální haplotypové páry. Každý, který má hodnotu váženého součinu pravděpodobností nejlepších souvisejících kombinací v $P2$ nižší než $max * 0,98$, odeber ze seznamu. Jestliže délka seznamu haplotypových párů je 1, ukonči běh algoritmu. Jinak pokračuj bodem (14).
(14)	Pro všechny haplotypové páry v seznamu vypočti součet hodnot hodnoticí funkce pouze pro $P3$ a pro každý haplotyp urči nejlepší související kombinaci. Nejvyšší z hodnot ulož do proměnné max .
(15)	Projdi všechny potenciální haplotypové páry. Každý, který má hodnotu součtu hodnoticích funkcí pouze pro $P3$ nižší než max , odeber ze seznamu. Jestliže délka seznamu haplotypových párů je 1, ukonči běh algoritmu. Jinak pokračuj bodem (16).
(16)	Pro všechny haplotypové páry v seznamu vypočti vážený součet pravděpodobností nejlepších souvisejících kombinací daného haplotypu v $P3$. Nejvyšší z hodnot ulož do proměnné max .
(17)	Projdi všechny potenciální haplotypové páry. Každý, který má hodnotu váženého součinu pravděpodobností nejlepších souvisejících kombinací v $P3$ nižší, než max odeber ze seznamu. Jestliže délka seznamu haplotypových párů je 1, ukonči běh algoritmu. Jinak pokračuj bodem (18).
(18)	Pro všechny haplotypové páry v seznamu vypočti součet hodnot hodnoticí funkce pro vazby ze všech populací. Nejvyšší z hodnot ulož do proměnné max .
(19)	Projdi všechny potenciální haplotypové páry. Každý, který má hodnotu součtu hodnoticích funkcí pro vazby ze všech populací nižší než max , odeber ze seznamu.

Tabulka 17: Pseudokód rozhodovacího algoritmu

Tato posloupnost kroků se na základě dostupných verifikačních dat ukázala dostačující k tomu, aby algoritmus dokázal najít nejvhodnější haplotypový pár. Ačkoliv nejvhodnějších variant může existovat více, algoritmus směřuje k nalezení jedné. Rozhodovací algoritmus je poměrně komplikovaný. Aby se podařilo identifikovat jakýkoli haplotyp, včetně řídkého, je nutné pracovat se všemi informacemi, které byly získány při identifikaci, tzn. informace o souvisejících kombinacích a pravděpodobnostech. Vyrovnanost haplotypového páru lze považovat za nejdůležitější rozhodovací faktor vedle součtu hodnoticí funkce. Proto byl algoritmus navržen tak, aby prvotní rozhodovací pravidlo bylo postaveno na vyhodnocení rozšířené hodnoticí funkce. Z experimentů plyne, že je to

nejlepší nalezený způsob na vyhodnocení souladu kvality nalezených haplotypů a vyrovnanosti celého páru. Toleranční koeficient váženého součinu pravděpodobností nejlepších souvisejících kombinací v hodnotě 98 % kompenzuje nepřesnosti v nastavení vah kombinací.

2.5.16. Příklady funkce rozhodovacího algoritmu

Uvedme několik příkladů ukazujících funkci rozhodovacího algoritmu. Příklady jsou čerpány z verifikačních dat.

Příklad:

Určete nejvhodnější variantu haplotypového páru následujícího genotypu:

Gen	A	B	C	DRBI	DQBI
1.	01:01	08:01	06:02	03:01	03:03
2.	01:01	57:01	07:01	07:01	06:03

Řešení:

Provedeme rozklad na všechny varianty:

Číslo varianty	A	B	C	DRBI	DQBI	Nejlepší nalezené vazby	Hodnoty funkce
1	01:01	08:01	06:02	03:01	03:03	ABC-BCD- ABD,2.8792030144053362E- 11,1	127
	01:01	57:01	07:01	07:01	06:03	ABCD,3.754924393596737E- 5,1	128
2	01:01	08:01	06:02	03:01	06:03	ABC-BCD- ABD,2.8792030144053362E- 11,1	127
	01:01	57:01	07:01	07:01	03:03	BCD- ABDD,1.4945118185701558E -8,1	139,2

3	01:01	08:01	06:02	07:01	03:03	<i>ABC-ABD-DrDq,3.3155848987770844E-12,1</i>	136
	01:01	57:01	07:01	03:01	06:03	<i>ABC-BCD-ABD,9.188206405518817E-13,1</i>	127
4	01:01	08:01	07:01	03:01	03:03	<i>ABCD,0.05945102540907276,1</i>	128
	01:01	57:01	06:02	07:01	06:03	<i>ABCD,0.007748587113653497,1</i>	128
5	01:01	57:01	06:02	03:01	03:03	<i>ABCD,4.8539266551372444E-4,1</i>	128
	01:01	08:01	07:01	07:01	06:03	<i>ABCD,0.0017008553500953395,1</i>	128
6	01:01	08:01	07:01	07:01	03:03	<i>ABCD-DrDq,1.5084522638422592E-7,1</i>	138
	01:01	57:01	06:02	03:01	06:03	<i>ABCD,4.8539266551372444E-4,1</i>	128
7	01:01	08:01	07:01	03:01	06:03	<i>ABCD,0.05945102540907276,1</i>	128
	01:01	57:01	06:02	07:01	03:03	<i>ABCDDq,0.0070950290452751535,1</i>	139,8
8	01:01	57:01	07:01	03:01	03:03	<i>ABC-BCD-ABD,9.188206405518817E-13,1</i>	127
	01:01	08:01	06:02	07:01	06:03	<i>ABD-ABC,3.7384877526161015E-8,1</i>	125

Nejlépe hodnocená varianta kritériem rozšířené hodnotící funkce je č.7, a to díky kompletně shodnému haplotypu v *PI*. Proto lze tento pár označit za nejvhodnější.

Příklad:

Určete nejvhodnější variantu haplotypového páru následujícího genotypu:

Gen	A	B	C	DRBI	DQBI
1.	11:01	35:01	04:01	04:01	03:02
2.	11:01	39:01	04:01	15:01	06:02

Řešení:

Provedeme rozklad na všechny varianty:

Číslo varianty	A	B	C	DRBI	DQBI	Nejlepší nalezené vazby	Hodnoty funkce
1						<i>ABCD,1.6852693893840628E-4,1,</i> <i>ABCD-BCDD-</i> <i>ABDD,2.2197653177488736E-19,2,</i> <i>ABD-</i> <i>BCDDpDr,1.08097194107736E-9,3</i>	128
	11:01	35:01	04:01	04:01	03:02	<i>BD-ABC-</i> <i>DrDq,7.484775411962123E-11,1,</i> <i>ABC-ABD-</i> <i>BDD,5.0867708577756375E-20,2,</i> <i>BCD-</i> <i>ABDD,1.1858238612747921E-13,3</i>	133
2	11:01	35:01	04:01	04:01	06:02	<i>ABCD,1.6852693893840628E-4,1,</i> <i>ABCD-</i> <i>DrDq,1.54045957435345E-13,2,</i>	128

						<i>ABD-ABC,9.072760720962508E-9,3</i>	
	11:01	39:01	04:01	15:01	03:02	<i>ABC-BD,1.2787058357208925E-7,1,</i> <i>ABC-ABD-</i> <i>DrDq,1.6423596567346048E-20,2,</i> <i>ABD-BCD-</i> <i>DrDq,7.147783302806753E-19,3</i>	123,5
3	11:01	35:01	04:01	15:01	03:02	<i>ABCD,6.389459004035299E-4,1,</i> <i>ABCD-</i> <i>DrDq,1.1650872853834186E-12,2,</i> <i>ABCD-</i> <i>DrDq,1.339041880436581E-12,3</i>	128
	11:01	39:01	04:01	04:01	06:02	<i>ABC-BD,4.201670299302425E-8,1,</i> <i>BD-ABC-</i> <i>DrDq,2.0250550040037033E-19,2,</i> <i>ABDpDr-</i> <i>BC,2.5624390596714748E-11,3</i>	123,5
4	11:01	39:01	04:01	04:01	03:02	<i>ABC-BD,4.201670299302425E-8,1,</i> <i>BD-ABC-</i> <i>DrDq,2.7293110329858422E-17,2,</i> <i>ABDDpDr-</i> <i>BC,3.4063435861720426E-11,3</i>	123.5
	11:01	35:01	04:01	15:01	06:02	<i>ABCD-</i> <i>DrDq,3.7400052704210875E-7,1,</i> <i>ABCDDq,2.5014034899916576E-5,2,</i>	138

						<i>ABCDDq,2.520509626137087E-6,3</i>	
--	--	--	--	--	--	--------------------------------------	--

V tomto příkladu se poprvé vyskytuje nejednoznačnost. Rozšířenou hodnoticí funkcí je nejlépe hodnocená varianta č.4. Je však vhodné zamyslet se nad dalšími řešeními. U varianty č.1, která získala druhé nejlepší hodnocení rozšířenou hodnoticí funkcí, lze najít nejlepší kombinace: *ABCD* a *BD-ABC-DrDq*. U poslední to jsou: *ABC-BD* a *ABCD-DrDq*. Je tedy obtížné rozhodnout, která varianta je lepší. Z důvodu poměrně vysokého hodnocení kombinace *ABCD-DrDq* byla vybrána poslední varianta. Varianta č.1. má lépe hodnocené kombinace v *P2*. Algoritmus se však nejprve rozhoduje podle kvality kombinací z *P1* a tam je lépe hodnocena varianta č.4.

Příklad:

Určete nejvhodnější variantu haplotypového páru následujícího genotypu:

Gen	A	B	C	DRBI	DQBI
1.	03:01	18:01	12:03	09:01	03:03
2.	02:01	39:93	07:01	11:01	03:01

Řešení:

Číslo varianty	A	B	C	DRBI	DQBI	Nejllepší nalezené vazby	Hodnoty funkce
1	<i>03:01</i>	<i>18:01</i>	<i>12:03</i>	<i>09:01</i>	<i>03:03</i>	<i>ABC-BCD-</i> <i>ABD,7.1513351859155385E-12,1</i>	127
	<i>02:01</i>	<i>39:93</i>	<i>07:01</i>	<i>11:01</i>	<i>03:01</i>	<i>ABCpB-ABDpB-</i> <i>DrDq,4.463916274308829E-6,1</i>	130,5
2	<i>03:01</i>	<i>18:01</i>	<i>12:03</i>	<i>09:01</i>	<i>03:01</i>	<i>ABC-BCD-</i> <i>ABD,7.1513351859155385E-12,1</i>	127
	<i>02:01</i>	<i>39:93</i>	<i>07:01</i>	<i>11:01</i>	<i>03:03</i>	<i>ABDpB-</i> <i>ABCpB,0.050332887950969206,1</i>	119,5
3	<i>03:01</i>	<i>18:01</i>	<i>12:03</i>	<i>11:01</i>	<i>03:03</i>	<i>ABCD,4.9455101769322875E-5,1</i>	128

	02:01	39:93	07:01	09:01	03:01	ABDpB- ABCpB,0.003813446052030516,1	119,5
4	03:01	18:01	07:01	09:01	03:03	ABCD,1.185272050019955E-4,1	128
	02:01	39:93	12:03	11:01	03:01	ABCpB-ABDpB- DrDq,2.465866443605302E-5,1	130,5
5	03:01	39:93	12:03	09:01	03:03	ABCpB,0.5565372355993083,1	118,4
	02:01	18:01	07:01	11:01	03:01	ABCD- DrDq,1.6118476980092959E-7,1	138
6	02:01	18:01	12:03	09:01	03:03	ABC-BCD- ABD,4.8986047236481054E-12,1	127
	03:01	39:93	07:01	11:01	03:01	BCpB-ABDpB- DrDq,3.113141352108431E-9,1	127,7
7	03:01	18:01	07:01	11:01	03:03	ABCD,2.4635967362866395E-4,1	128
	02:01	39:93	12:03	09:01	03:01	ABDpB- ABCpB,0.02106546824885767,1	119,5
8	03:01	18:01	07:01	09:01	03:01	ABCD,1.185272050019955E-4,1	128
	02:01	39:93	12:03	11:01	03:03	ABDpB- ABCpB,0.2780387708487158,1	119,5
9	02:01	18:01	07:01	09:01	03:03	ABCD,3.9153917786350935E-5,1	128
	03:01	39:93	12:03	11:01	03:01	ABCDpB- DrDq,3.772017872521632E-5,1	135,3
10	02:01	18:01	12:03	11:01	03:03	ABCD,2.023995831670436E-4,1	128
	03:01	39:93	07:01	09:01	03:01	BDpB- BCpB,7.05012040589923E-8,1	107,3
11	02:01	18:01	12:03	09:01	03:01	ABC-BCD- ABD,4.8986047236481054E-12,1	127
	03:01	39:93	07:01	11:01	03:03	ABDpB,0.05495011307702541,1	118,3
12	03:01	39:93	07:01	09:01	03:03	BDpB- BCpB,7.05012040589923E-8,1	107,3
	02:01	18:01	12:03	11:01	03:01	ABCD- DrDq,1.795038651652198E-8,1	138
13	03:01	39:93	12:03	09:01	03:01	ABCpB,0.5565372355993083,1	118,3

	02:01	18:01	07:01	11:01	03:03	ABCD,0.0018174388718903817, 1	128
14	03:01	18:01	12:03	11:01	03:01	ABCD- DrDq,4.3860672936298055E-9,1	138
	02:01	39:93	07:01	09:01	03:03	ABDpB- ABCpB,0.003813446052030516,1	119,5
15	03:01	39:93	12:03	11:01	03:03	ABCDpB,0.42531387521617664, 1	125,3
	02:01	18:01	07:01	09:01	03:01	ABCD,3.9153917786350935E- 5,1	128
16	02:01	39:93	12:03	09:01	03:03	ABDpB- ABCpB,0.02106546824885767,1	119,5
	03:01	18:01	07:01	11:01	03:01	ABCD- DrDq,2.1849112999748476E-8,1	138

Na základě kvality a vyrovnanosti kombinací v *PI* sice je možné rozhodnout, že nejlepším řešením bude varianta č.9. Při prozkoumání databáze však lze zjistit, že druhý haplotyp má polymorfismus skládající se pouze ze tří alel. Naproti tomu druhé nejlépe hodnocené řešení, tj. varianta č.4 sice v druhém haplotypu obsahuje slabší kombinaci, ale zato se varianty jejího polymorfismu vyskytují mnohem častěji. Navíc vazba *ABD* s polymorfní alelou se nachází v populaci Israel Poland Jews. Ta má k *PI* blíže než ostatní světové populace. *ABC* je zde nalezeno se třemi variantami alely *B* a *ABD* se čtyřmi. Takovou informaci bohužel není algoritmus schopen zpracovat. Přehnaným vlivem polymorfismů by mohly slabé polymorfní varianty převážit varianty běžně lépe hodnocené bez polymorfismů. Na druhou stranu počet variant polymorfismu u obou variant je podobný. Obě řešení se liší na lokusu *A*. Při pochybnostech o správnosti řešení je vhodné prozkoumat variantu se stejným genem *A* na druhém haplotypu, druhou nejlépe hodnocenou variantu nebo takovou, která se též dostala do konečného výběru variant.

Příklad:

Určete nejvhodnější variantu haplotypového páru následujícího genotypu:

<i>Gen</i>	<i>A</i>	<i>B</i>	<i>C</i>	<i>DRB1</i>	<i>DQB1</i>
1.	68:02	40:01	04:01	15:01	06:03
2.	02:01	44:03	03:04	01:02	06:02

Řešení:

Číslo varianty	<i>A</i>	<i>B</i>	<i>C</i>	<i>DRB1</i>	<i>DQB1</i>	Nejlepší nalezené vazby	Hodnoty funkce
1	68:02	40:01	04:01	15:01	06:03	<i>BD-BC</i> ,5.311798649012601E-8,1, <i>BDD-BC</i> ,9.07426828944102E-13,2, <i>BCD-</i> <i>DrDq</i> ,1.1842415901558413E-11,3	114
	02:01	44:03	03:04	01:02	06:02	<i>ABC-BD</i> ,9.051941313552673E-9,1, <i>ABC-ABD-</i> <i>DrDqpDr</i> ,9.059700043330968E-20,2, <i>BC-ABD-</i> <i>DrDq</i> ,2.199255612588608E-17,3	123,5
2	68:02	40:01	04:01	15:01	06:02	<i>BD-BC-</i> <i>DrDq</i> ,3.109207670035313E-11,1, <i>BDD-BC</i> ,6.636604252790884E-12,2, <i>BCD-BDD</i> ,3.8926677071872085E-12,3	119

	02:01	44:03	03:04	01:02	06:03	ABC-BD,9.051941313552673E-9,1, ABD-ABC,8.220946634740041E-14,2, ABD-BC,3.227606087115044E-11,3	123,5
3	68:02	40:01	04:01	01:02	06:03	BCD,2.1980045230810162E-5,1, BD-BC,4.54772328262005E-13,2, BD-BC,7.002238584843758E-9,3	120
	02:01	44:03	03:04	15:01	06:02	ABC-ABD- DrDq,2.8475463372496226E-12,1, ABC-ABD-BDD- BCD,1.5645827236878098E-24,2, ABDD-BC,4.585349672786929E-13,3	136
4	68:02	40:01	03:04	15:01	06:03	BCD,0.0023946658684758986,1, BCD-BDD,5.3582169931242345E-12,2,ABD- BCDD,5.88127943611094E-11,3	120
	02:01	44:03	04:01	01:02	06:02	ABC-BD,5.756675280083369E-7,1, ABCD- DrDqpDr,2.2426570764252684E-13,2, ABC-ABD-DrDq- BCD,3.028608314343001E-22,3	123,5
5	68:02	44:03	04:01	15:01	06:03	BCD,9.359835927453327E-4,1, BCDD,1.9542214765559825E-7,2, BCD- DrDq,1.5593665285497305E-11,3	120

	02:01	40:01	03:04	01:02	06:02	ABC-BD,3.4786955870782297E-7,1, ABCD- DrDqpDr,2.2665613669685613E-13,2, ABC-ABD-DrDq- BCD,2.7063513418357514E-20,3	123,5
6	02:01	40:01	04:01	15:01	06:03	ABD-BC,1.502943561846042E-8,1, ABC-ABD- BDD,5.212090154789726E-19,2, ABC-ABD-DrDq- BCD,2.4710019168678738E-20,3	123,5
	68:02	44:03	03:04	01:02	06:02	BD-BC,1.7379727322021134E-8,1, BD-BC- DrDqpDr,3.4549464419917293E-18,2, BD-BC- DrDq,9.682216870227078E-17,3	114
7	68:02	40:01	03:04	01:02	06:03	BD-BC,7.026463819616371E-7,1, BCD,2.0567215198176813E-7,2, ABC-BD,7.902966673807336E-12,3	114
	02:01	44:03	04:01	15:01	06:02	ABCD- DrDq,4.931889001281514E-8,1, ABCD- BCDD,7.838821456926881E-14,2, BCDD- ABDD,3.35571757466578E-13,3	138
8	68:02	40:01	03:04	15:01	06:02	BCD- DrDq,1.4016934709716583E-6,1, BCDD,1.7898068998988681E-6,2, ABD-BCDD,8.931943434353942E-13,3	122
	02:01	44:03	04:01	01:02	06:03	ABC-BD,5.756675280083369E-7,1,	123,5

						<i>ABCD,2.0350303053229596E-7,2,</i> <i>ABC-BCD-</i> <i>ABD,4.444756023314168E-17,3</i>	
9	02:01	40:01	03:04	15:01	06:03	<i>ABCD,6.837763735532791E-4,1,</i> <i>ABCD-</i> <i>BDD,2.4512237078455083E-</i> <i>12,2,ABCD-</i> <i>DrDq,7.174943232552471E-11,3</i>	128
	68:02	44:03	04:01	01:02	06:02	<i>BCD,6.59401356924305E-5,1,</i> <i>BCD,8.274212365441025E-7,2,</i> <i>BCD-</i> <i>DrDq,2.3464174276288443E-13,3</i>	120
10	02:01	40:01	04:01	01:02	06:03	<i>ABDpDr-</i> <i>BCD,9.876613000018457E-6,1,</i> <i>ABD-ABC,4.094113038845053E-</i> <i>14,2,</i> <i>ABD-ABC,6.872346452297094E-</i> <i>11,3</i>	121,34
	68:02	44:03	03:04	15:01	06:02	<i>BD-BC-</i> <i>DrDq,7.529545469636365E-11,1,</i> <i>BCD-BDD,3.87492974582774E-</i> <i>13,2,BDD-</i> <i>BC,5.759700988121009E-12,3</i>	119
11	02:01	40:01	04:01	15:01	06:02	<i>BC-ABD-</i> <i>DrDq,8.797328285383245E-12,1,</i> <i>ABDD-BC,2.1745284090641724E-</i> <i>12,2,</i> <i>BCD-ABDD,2.09888166307017E-</i> <i>12,3</i>	133
	68:02	44:03	03:04	01:02	06:03	<i>BD-BC,1.7379727322021134E-8,1,</i> <i>BD-BC,3.135085067899899E-12,2,</i> <i>BD-BC,1.4209527045530645E-</i> <i>10,3</i>	114
12	68:02	44:03	03:04	15:01	06:03	<i>BD-BC,1.2863543930740127E-7,1,</i>	114

						<i>BCD-BDD,3.917680547352495E-14,2,BD-BC-DrDq,6.996762619050837E-16,3</i>	
	02:01	40:01	04:01	01:02	06:02	<i>ABDpDr-BCD,9.876613000018457E-6,1,ABC-ABD-DrDqpDr,4.511820563180106E-20,2,ABC-ABD-DrDq,4.6827419762296004E-17,3</i>	121,34
13	68:02	44:03	04:01	15:01	06:02	<i>BCD-DrDq,5.478685390554029E-7,1,BCDD,4.01122470045801E-7,2,BCDD,6.887158663120205E-7,3</i>	122
	02:01	40:01	03:04	01:02	06:03	<i>ABC-BD,3.4786955870782297E-7,1,ABCD,2.0567215198176813E-7,2,ABC-BCD-ABD,3.97181483351977E-15,3</i>	123,5
14	68:02	40:01	04:01	01:02	06:02	<i>BCD,2.1980045230810162E-5,1,BD-BC-DrDqpDr,5.011711017135553E-19,2,BD-BC-DrDq,4.771249059753444E-15,3</i>	120
	02:01	44:03	03:04	15:01	06:03	<i>ABD-ABC,4.864774049342139E-9,1,ABC-ABD-BDD-BCD,1.5818442406382675E-25,2,BC-ABD-DrDq,1.0942947484775467E-16,3</i>	125
15	68:02	44:03	04:01	01:02	06:03	<i>BCD,6.59401356924305E-5,1,BCD,8.274212365441025E-7,2,BCD,3.4435793315601025E-7,3</i>	120
	02:01	40:01	03:04	15:01	06:02	<i>ABCD-DrDq,4.002415915437579E-7,1,ABCDDq,3.146998422859562E-5,2,</i>	138

						<i>ABCDDq,4.310146084469275E-6,3</i>	
16	02:01	44:03	04:01	15:01	06:03	<i>ABCD,8.425684005143896E-5,1,</i> <i>ABCD-</i> <i>BCDD,3.8189815794326444E-14,2,</i> <i>ABC-ABD-DrDq-</i> <i>BCD,6.22996977288963E-21,3</i>	128
	68:02	40:01	03:04	01:02	06:02	<i>BD-BC,7.026463819616371E-7,1,</i> <i>BCD,2.0567215198176813E-7,2,</i> <i>ABC-BCD-</i> <i>DrDq,1.2641752360281165E-19,3</i>	114

Příklad ilustruje situaci, kdy o výsledku rozhoduje kvalita kombinací v *PI*. Zde by však vhodný výsledek bylo možné najít i mezi jinými variantami, které nemají tak vysoké hodnocení. Algoritmem bylo za nejvhodnější variantu zvoleno č.15. Toho varianta dosáhla především díky vazbě *BCD* v prvním haplotypu. Po přihlédnutí ke kombinacím z prvního haplotypu v dalších populacích, nenachází se zde jakákoli jiná vazba (pouze *BCD*). Pokud by tedy byla hledána varianta, která má shodnou kvalitu druhého haplotypu a zároveň z prvního lze najít i jiné vazby, vyhovovalo by č.7. Ta vybrána nebyla, protože kombinace *BD-BC* v *PI* je považována za slabší než vazba *BCD*.

Příklad:

Určete nejvhodnější variantu haplotypového páru následujícího genotypu:

Gen	A	B	C	DRB1	DQB1
1.	01:01	35:01	14:02	08:01	04:02
2.	66:01	51:01	15:05	14:01	05:03

Řešení:

Číslo varianty	A	B	C	DRB1	DQB1	Nejlepší nalezené vazby	Hodnoty funkce
1	01:01	35:01	14:02	08:01	04:02	<i>ABD,2.1980045230810162E-5,1,</i> <i>BC-ABD-BDD,1.872160883997543E-18,2,</i> <i>BC-ABD-BDD,1.550267326811986E-18,3</i>	121
	66:01	51:01	15:05	14:01	05:03	<i>ABD-BC,9.662447766969212E10,1,</i> <i>BDD,7.439680112939192E-6,2,</i> <i>BDD-BC,4.1108314977343905E-13,3</i>	123,5
2	01:01	35:01	14:02	08:01	05:03	<i>ABD,2.1980045230810162E-5,1,</i> <i>ABD-BC,5.100287484193648E-13,2,</i> <i>BC-ABD-DrDqpDr,2.4373488759687786E-19,3</i>	121
	66:01	51:01	15:05	14:01	04:02	<i>ABD-BC,9.662447766969212E-10,1,</i> <i>BD-BCpC-DrDq,1.8098641304195634E-15,2,</i> <i>BD-BC-DrDq,1.2329736403592162E-17,3</i>	123,5
3	01:01	35:01	14:02	14:01	04:02	<i>ABD,3.7732410979557444E-4,1,</i> <i>BC-ABD-DrDq,3.680293216183622E-18,2,</i> <i>BC-ABD-DrDq,1.0683124305715985E-17,3</i>	121
	66:01	51:01	15:05	08:01	05:03	<i>BD-BC,9.28140670881269E8,1,</i> <i>BCDpB,1.6222163109079444E-4,2,</i> <i>BD-BC-DrDqpDr,2.8365545790962806E-18,3</i>	114
4	01:01	35:01	15:05	08:01	04:02	<i>ABD,2.1980045230810162E-5,1,</i> <i>BC-ABD-BDD,5.629269068348452E-19,2,</i> <i>BC-ABD-BDD,4.557371807387354E-19,3</i>	121
	66:01	51:01	14:02	14:01	05:03	<i>ABD-BCD,3.278036556235765E-9,1,</i> <i>BCDD,2.316464758617831E-6,2,</i> <i>ABC-BCD-BDD,5.526374895037122E-20,3</i>	123,5
5	01:01	51:01	14:02	08:01	04:02	<i>ABCD,2.2895880448760587E-5,1,</i> <i>BCDD-ABDD,2.110856413345084E-12,2,</i> <i>ABD-BCDD,1.441402078882361E-13,3</i>	128

	66:01	35:01	15:05	14:01	05:03	<i>null,0.0,0, BCDD,1.9542214765559825E-7,2, BDD-BC,1.809061022025342E-12,3</i>	110
6	66:01	35:01	14:02	08:01	04:02	<i>null,0.0,0, BDD-BC,4.771373867636064E-12,2, BDD-BC,2.2638531039341995E-12,3</i>	110
	01:01	51:01	15:05	14:01	05:03	<i>ABD-BC,2.818213932032687E-9,1, ABDDpDr-BCpC,1.114895852140063E-7,2, BC-ABD-BDD,8.065456331517112E-18,3</i>	123,5
7	01:01	35:01	15:05	14:01	04:02	<i>ABD,3.7732410979557444E-4,1, ABD-BCD-DrDq,5.533007590693337E-19,2, BC-ABD-DrDq,3.140553160325317E-18,3</i>	121
	66:01	51:01	14:02	08:01	05:03	<i>BCD,0.0013365158086115915,1, BCD,1.4760517321380497E-5,2, ABC-BCD-DrDqpDr,2.5609291845925466E-20,3</i>	120
8	01:01	35:01	15:05	08:01	05:03	<i>ABD,2.1980045230810162E-5,1, ABD-BC,1.5335696210654153E-13,2, BC-ABD-DrDqpDr,7.165154589789351E-20,3</i>	121
	66:01	51:01	14:02	14:01	04:02	<i>ABD-BCD,3.278036556235765E-9,1, BCD-DrDq,3.4022274110119585E-12,2, ABC-BCD-DrDq,2.7639250821046496E-20,3</i>	123,5
9	66:01	35:01	15:05	08:01	04:02	<i>null,0.0,0, BDD-BC,1.4346708958718458E-12,2, BDD-BC,6.655123367111595E-13,3</i>	110
	01:01	51:01	14:02	14:01	05:03	<i>ABCD,1.6485033923107625E-5,1, ABCD-BCDD,4.590649379159469E-13,2, ABC-ABD-BDD-BCD,1.3641453085828572E-22,3</i>	128
10	66:01	35:01	14:02	14:01	04:02	<i>null,0.0,0, BD-BC-DrDq,6.845213666440053E-17,2, BD-BC-DrDq,1.2199272104529083E-16,3</i>	110

	01:01	51:01	15:05	08:01	05:03	ABD-BC,5.294507602123169E-9,1, ABD,1.4181104336351456E-6,2, BC-ABD-DrDqpDr,2.4814124861292816E-20,3	123,5
11	66:01	35:01	14:02	08:01	05:03	null,0.0,0, BD-BC,1.4008590842816601E-11,2, BD-BC-DrDqpDr,1.2984986488607763E-17,3	110
	01:01	51:01	15:05	14:01	04:02	ABD-BC,2.818213932032687E-9,1, BCpC-ABD-DrDq,8.003216624840142E-17,2, BC-ABD-DrDq,4.03380470501905E-18,3	123,5
12	01:01	51:01	15:05	08:01	04:02	ABD-BC,5.294507602123169E-9,1, BCDpB-ABDD,9.910893972130924E-11,2, BC-ABD-BDD,1.6685234648047195E-19,3	123,5
	66:01	35:01	14:02	14:01	05:03	null,0.0,0, BDD-BC,5.0274203307899844E-11,2, BDD-BC,6.153827936770345E-12,3	110
13	01:01	51:01	14:02	08:01	05:03	ABCD,2.2895880448760587E-5,1, ABC-BCD-ABD,1.9825911044818233E-16,2, ABC-ABD-DrDqpDr- BCD,6.743162838673955E-25,3	128
	66:01	35:01	15:05	14:01	04:02	null,0.0,0, BCD-DrDq,1.415602727137306E-13,2, BD-BC-DrDq,3.586260111290504E-17,3	110
14	01:01	35:01	14:02	14:01	05:03	ABD,3.7732410979557444E-4,1, ABDD-BC,4.2421418829557E-12,2, ABDD-BC,4.809956802507648E-13,3	121
	66:01	51:01	15:05	08:01	04:02	BD-BC,9.28140670881269E-8,1, BDD,6.154743272967838E-6,2, BDD-BC,7.972366630801165E-13,3	114
15	01:01	51:01	14:02	14:01	04:02	ABCD,1.6485033923107625E-5,1, ABCD-DrDq,1.4355424360615628E-13,2, ABC-ABD-DrDq-BCD,6.822547340053773E- 23,3	128

						<i>null,0.0,0,</i> <i>BD-BC,4.2121447893000825E-12,2,</i> <i>BD-BC-DrDqpDr,3.817239150887664E-18,3</i>	110
16	66:01	51:01	14:02	08:01	04:02	<i>BCD,0.0013365158086115915,1,</i> <i>BCDD,3.4550523023876033E-6,2,</i> <i>ABD-BCDD,1.2110541561955326E-13,3</i>	120
	01:01	35:01	15:05	14:01	05:03	<i>ABD,3.7732410979557444E-4,1,</i> <i>BCDD-ABDD,6.377699237652517E-13,2,</i> <i>ABDD-BC,1.413998808294429E-13,3</i>	121

Varianty jsou nejprve podrobeny výpočtu rozšířené hodnoticí funkce. Nejlépe jsou tímto kritériem hodnoceny varianty: 1, 2, 4 a 8. Tyto varianty tedy postupují do užšího výběru. Dalším kritériem je samostatný součet hodnoticí funkce. Protože ten je pro všechny zbylé varianty shodný, je možné přejít k dalšímu kroku. Tím je výpočet vážené pravděpodobnosti. Nejlepší pravděpodobnosti dosahují varianty 4 a 8. Z těchto dvou variant jsou dále vybírány ty, které splňují nejlepší vážený součet hodnotících funkcí pro *P1* a *P2*. Lepší hodnotou se vyznačuje varianta 4. Tu je tedy možné označit za nejvhodnější.

2.6. Analýza neshod

Jedním z cílů vývoje nové metody identifikace haplotypů byla snaha o minimalizaci počtu neshod výsledných variant algoritnického a expertního řešení. Tabulka 18 ukazuje počet haplotypových párů, které se shodují. Důvodem slabšího výsledku v prvním souboru je patrně stáří expertního řešení. Při vývoji algoritmu a nastavování hodnoticí funkce tak byla snaha o co nejlepší přizpůsobení se expertnímu řešení v druhém souboru. Kvůli subjektivnímu posuzování některých situací nebylo možné dosáhnout stoprocentní shody. Rovněž je téměř nemožné přenést všechny expertní úvahy do kódu programu.

Soubor	Celkem	Nalezeno shodně
1.	74	35
2.	100	76

Tabulka 18: Shodnost algoritmického a expertního řešení ve validačních datech

Tabulka 19 ukazuje počty neshod na konkrétních lokusech. Je uváděna vždy menší množina lokusů. Je vidět, že nejvíce neshod vzniká na lokusech *DRBI, DQBI*. Dále jsou to lokusy *B, C* a ty které zahrnují gen *A*. Neshoda na *B, C* je způsobena silnou vazbou těchto dvou lokusů. Neshoda na *A* je pak zapříčiněna jeho již zmíněnou vysokou mírou rekombinace.

Lokusy	A	B	C	<i>DRBI</i>	<i>DQBI</i>	<i>A, B</i>	<i>A, C</i>	<i>A, DRBI</i>	<i>A, DQBI</i>	<i>B, C</i>	<i>A, DQBI</i>
1.soubor	8	1	1	1	4	2	4	1	2	5	11
2.soubor	2	0	1	1	4	0	1	1	1	4	9

Tabulka 19: Rozdíly algoritmického a expertního řešení ve validačních datech na konkrétních lokusech

Největší neshoda spočívá ve způsobu vyhodnocování kvality haplotypového páru. V počáteční fázi vývoje algoritmu byl preferovaný vždy pár s vyšším součtem hodnotící funkce. Bohužel tento přístup skýtal vážný nedostatek v tom, že velmi vysoko hodnocený haplotyp vyrovnával případně nízké hodnocení druhého. Bylo tedy nutné snížit hodnocení kompletně nalezeného haplotypu v *PI*. Takový pár mohl být i přes svou vysokou hodnotu součtu funkce značně nevyrovnaný, a tedy i nevhodný. Ukažme si příklad takového expertního řešení.

Příklad:

1. haplotyp: nejlepší kombinace: *ABCD-DrDq* v *PI*
2. haplotyp: nejlepší kombinace: *BCpB* v *PI*.

Tento problém bylo možné vyřešit zavedením genové vyrovnanosti v *PI*. Algoritmus poté může variantu toto nespĺňující okamžitě vyřadit, pokud nedosahuje nejvyšší hodnoty rozšířené hodnotící funkce. Takto bylo vyřazeno i několik expertních řešení. Použití genové vyrovnanosti přineslo lepší výsledky než výpočet rozdílu hodnotících funkcí pro konkrétní pár. Je však možné, že není nastavena zcela ideálně. Při určování hodnotící funkce byl experty zvolen odlišný postup. To se týká především nekompletních haplotypů sestavených kombinací jinak nízko hodnocených vazeb. Takto sestavené haplotypy jsou algoritmem hodnocené výše než samostatné vazby sestávající ze 4 genů, tj. *ABCD*, *ABDD* a *BCDD*. Expertní řešení patrně preferuje druhou možnost.

K tomu se váže další možná příčina neshody, a to problém s vyhodnocováním polymorfismů. Je známo, že HLA antigeny I. třídy jsou velmi polymorfní. O antigenech II. třídy taková informace neexistuje. Tudíž je vhodné jejich polymorfismu přiřadit vyšší váhu. Dobrým řešením je snížit odečítanou hodnotu při nalezení polymorfismu na antigenu II. třídy. S vyhodnocováním polymorfismů nastává problém, pokud polymorfní alelická skupina obsahuje mnoho variant. V takovém případě může převážit silnější kombinaci s méně variantami. Může se proto stát, že varianta vybraná algoritmem není nejlepší. Příklad takové situace byl ukázán výše.

Neshodu konečné volby lze přisuzovat neaktuálnosti populačních dat, která byla použita při expertním řešení prvního souboru. Je možné vyzorovat, že populační data se mění velmi dynamicky. Je tedy pravděpodobné, že nová populační data přinášejí výsledky velmi odlišné. Expertní výsledky pro nový soubor jsou získané algoritmem založeným na populačním modelu, tedy jeho problém může být také neaktuálnost.

Dále varianty se stejně hodnocenými nejlepšími kombinacemi, ale různými příslušnými pravděpodobnostmi mohou způsobit nerozhodnou situaci. Přizpůsobení vah jednomu genotypu z expertního řešení může přinést rozdílný výsledek u jednoho nebo několika jiných. Jestliže pro oba haplotypy v algoritmickeém řešení jsou výsledné pravděpodobnosti nejlepších kombinací v *PI* vyšší než v expertní variantě, je rozhodnuto, že tento výsledek je správně. Může však nastat složitější situace. Uvedme příklad.

Příklad:

Výsledek	A	B	C	DRBI	DQBI	Nejlepší nalezené vazby
algoritmický výsledek:	02:30	08:01	07:01	11:01	03:01	<i>BCD-DrDq,1.5741353493388236E-8,1,</i> <i>BCDD,9.893407198827144E-7,2,</i> <i>BCDD,5.654644639702657E-7,3</i>
	32:01	51:01	01:02	13:01	06:03	<i>ABCD-DrDq,4.4221666129189395E-10,1,</i> <i>BCDD,4.330690546587752E-7,2,</i> <i>BDD-BC,1.1450347738142297E-11,3</i>
expertní výsledek:	02:30	08:01	07:01	13:01	06:03	<i>BCD-DrDq,1.9586221585577787E-8,1,</i> <i>ABCDpA-BCDD,1.0895151704424645E-10,2,</i> <i>BCDD,9.182878217493604E-7,3</i>
	32:01	51:01	01:02	11:01	03:01	<i>ABCD-DrDq,1.624469368011039E-10,1,</i> <i>BCDD,1.7975108358817492E-6,2,</i> <i>BCD-ABDD,4.1879251467074574E-14,3</i>

Ohodnocení v *PI* je u obou řešení shodné, včetně kvality vazeb, ale pravděpodobnosti se převyšují střídavě. Váhy kombinací při vyhodnocování součtu pravděpodobností rozhodly, že bude udělena přednost řešení, které se u hodnotnější kombinace (zde *ABCD-DrDq*) vyznačuje vyšší pravděpodobností. Algoritmus počítá s výše uvedenou tolerancí ve výši 98 % nejvyšší hodnoty vážené pravděpodobnosti. Ani tato tolerance však nepřináší zcela spolehlivé řešení.

Příčinou neshod s expertním řešením může být též příliš velký důraz na *PI*. Program se primárně snaží rozhodnout o nevhodnějším řešení na základě vazeb z *PI*. Pokud je výsledek u několika párů vyrovnaný, teprve tehdy přistupuje k dalším populacím. Snaha o zavedení včasějšího vlivu populace *P2* na výběr páru však měla za následek velký pokles shody s expertním řešením. Proto byla vrácen původní postup.

Paradoxně největší shodnost byla dosažena, pokud algoritmus využívá pouze *PI*. Shodných haplotypů tak vyšlo 78. Ovšem lze zjistit, že 21 z nich vzešlo ze dvou finálních variant. Z některých pak díky přítomnosti *P2* a *P3* byla vybrána jiná než expertní. Taková

neurčitost není žádoucí. Je nutné používat minimálně $P1$ a $P2$. Z hlediska experimentů na dostupných verifikačních datech se zdá zbytečné používat $P3$. Nicméně v případech, kdy u některého haplotypu je nalezena jakákoli kombinace až v $P3$, by tak došlo ke ztrátě mnoha informací.

Pro zdůvodnění řešení, které zvolil algoritmus odlišně od expertního řešení, je uvedeno několik příkladů. Jsou čerpány z obou výše uvedených souborů genotypů.

Příklad:

Výsledek	<i>A</i>	<i>B</i>	<i>C</i>	<i>DRBI</i>	<i>DQBI</i>	Nejlepší nalezené vazby
algoritmický výsledek:	03:01	37:01	12:03	03:01	02:01	<i>BD-ABC-DrDq</i> ,1.667167606368312E-12,1, <i>ABDD-BC</i> ,9.374256959900093E-13,2, <i>ABD-BDD</i> ,8.347231467939731E-13,3
	26:01	38:01	06:04	01:01	05:01	<i>ABD-DrDq</i> ,5.5329426674455995E-8,1, <i>ABDD</i> ,1.0039329914518908E-6,2, <i>ABDD</i> ,1.0257470349327966E-6,3
expertní výsledek:	03:01	37:01	06:04	01:01	05:01	<i>BD-DrDq</i> ,2.824407306564012E-8,1, <i>ABDD-BCpC</i> ,1.653339961410057E-11,2, <i>ABDD-BCpC</i> ,9.029253051074044E-12,3
	26:01	38:01	12:03	03:01	02:01	<i>ABCD-DrDq</i> ,1.5759704665286585E-8,1, <i>ABCDDq</i> ,7.504210469974973E-6,2, <i>ABC-BCD-DrDq</i> ,1.3260902286132957E-14,3

Algoritmický výsledek se vyznačuje větší vyšší hodnotou rozšířené hodnoticí funkce. To nedává expertnímu výsledku možnost postoupit do užšího výběru.

Příklad:

Výsledek	A	B	C	DRBI	DQBI	Nejlepší nalezené vazby
algoritmický výsledek:	01:01	08:01	07:01	03:01	04:02	ABCD,0.05945102540907276,1, ABCD-DrDq,2.9867902417734654E-10,2, ABCD-DrDqpDr,2.320157743885116E-10,3
	02:01	07:02	07:02	08:01	06:03	ABCD,0.0016060624805995303,1, ABCD-DrDqpDq,3.6542740708228E-12,2, ABCD-DrDqpDq,2.9041912042254336E-14,3
expertní výsledek:	01:01	08:01	07:01	03:01	06:03	ABCD,0.05945102540907276,1, ABCD,5.437220787990712E-4,2, ABCD-DrDqpDq,6.614877693741434E-12,3
	02:01	07:02	07:02	08:01	04:02	ABCD,0.0016060624805995303,1, ABCDDq,4.259236976317138E-5,2, ABCDDq,5.715389467565231E-7,3

Oba výsledky se vyznačují vzájemně velkou shodou v *PI*, včetně pravděpodobností. Rozhodnout tedy musí hodnoticí funkce pro *PI* a *P2*. Ač je výsledek těsný, vychází jako vhodnější algoritmický výsledek. Důvodem je vyšší součet hodnoticí funkce vazeb v *P2*. Ač expertní výsledek obsahuje kompletní haplotyp, algoritmický výsledek zase vyrovnanější kombinace. Je však třeba připustit, že zde záleží na subjektivním pohledu.

Příklad:

Výsledek	A	B	C	DRBI	DQBI	Nejlepší nalezené vazby
algoritmický výsledek:	24:02	40:01	03:04	11:01	05:02	<i>ABCD,1.5554778058622678E-4,1,</i> <i>ABCD-DrDq,1.99378068168603E-11,2,</i> <i>ABCD-DrDq,2.1991440414300105E-10,3</i>
	02:01	35:01	04:01	01:01	05:01	<i>ABCD-DrDq,2.7324574676343817E-7,1,</i> <i>ABCDDq,2.3373853917115417E-4,2,</i> <i>ABCDDq,2.9327143518452347E-6,3</i>
expertní výsledek:	02:01	40:01	03:04	11:01	05:02	<i>ABCD,0.0011240501529865637,1,</i> <i>ABCD-DrDq,4.9865095380888694E-11,2,</i> <i>ABCD-DrDq,6.963434343237266E-11,3</i>
	24:02	35:01	04:01	01:01	05:01	<i>ABCD-DrDq,7.882376737433474E-8,1,</i> <i>ABCDDq,2.581040429767751E-5,2,</i> <i>ABCDDq,4.190909314153997E-6,3</i>

Řešení se opět vyznačují shodnými kvalitami kombinací v *PI*. Po výpočtu vážené pravděpodobnosti v *P1*, kdy kombinaci *ABCD-DrDq* je přidělena vyšší váha, je možné tvrdit, že algoritmický výsledek je lepší variantou. Problém zde spočívá v tom, že pravděpodobnosti nejlepších kombinací v *PI* se převyšují střídavě a jsou vzájemně nepřilíživě vzdáleně hodnocené. Nechávat rozhodnout hůře hodnocenou kombinaci by však také nebylo správné. Jelikož se i expertní výsledek dostal do konečného výběru, lze ho považovat za jedno z nejvhodnějších řešení.

2.7. Uživatelské rozhraní

Program má implementované jednoduché uživatelské rozhraní. Je spustitelné rozeběhnutím třídy *UzivatelkaTrida*. Nejprve nabídne uživateli možnost vybrat si rozsah populací – minimum a maximum. Program je nastaven na opakované zadávání, dokud vstupy nejsou provedeny správně.

Dále umožní vybrat, zda chce uživatel provádět identifikaci z populačních dat na disku – stisk **p**, provádět identifikaci a zároveň porovnávat s již hotovým řešením – stisk **i** nebo načítat aktuální populační data z webu – stisk **n**.

Při výběru jedné z prvních dvou možností je uživatel dotázán na název textového souboru „navez“, který obsahuje data určená k identifikaci. Název souboru je nutné vepsat bez přípony „.txt“. Výstup se pak načítá do nového souboru *výsledek_identifikace.txt*. Velikost souboru pro 100 genotypů je přibližně 1,5 MB.

Program je nastaven na opakované dotazování na název souboru v případě neexistence toho požadovaného. Potom se objeví varovné hlášení a běh programu začne znovu od začátku. Po skončení načítání nebo identifikace začne dotazování znovu od procesu. Rozsah populací již změnit nelze. K tomu je nutno spustit program znovu.

2.8. Ukázka výstupu algoritmu

Popišme si textový výstup identifikačního algoritmu. Nejprve jsou vypsány všechny existující varianty haplotypového páru s příslušným pořadím. Do tohoto záznamu jsou zahrnuty nejlepší kombinace z každé souhrnné populace, ohodnocení a url adresa kompletního haplotypu. Uvedme ukázkou jedné varianty.

1.

A*31:01;B*35:01;C*04:01;DRB1*14:01;DQB1*03:01;128.0;ABCD,4.9455101769322875E-5,1,BCD-DrDq,3.43718210782493E-10,2,ABC-ABD-BDD-BCD,5.201201570847558E-22,3

http://allelefrequencies.net/hla6003a.asp?page=1&hla_selection=&hla_locus1=A*31%3A01&hla_locus2=B*35%3A01&hla_locus3=C*04:01&hla_locus4=DRB1*14:01&hla_locus5=&hla_locus6=&hla_locus7=&hla_locus8=DQB1*03:01&hla_population=&hla_country=&hla_dataset=&hla_region=&hla_ethnic=&hla_study=&hla_sample_size=&hla_sample_size_pattern=equal&hla_sample_year=&hla_sample_year_pattern=equal&hla_loci=&hla_order=order_1

A*02:01;B*15:01;C*03:04;DRB1*11:01;DQB1*05:03;128.0;ABCD,6.758755709281185E-4,1,ABCD-DrDq,4.870889975678512E-13,2,ABC-ABD-BDD-BCD,5.201201570847558E-22,3

http://allelefrequencies.net/hla6003a.asp?page=1&hla_selection=&hla_locus1=A*02%3A01&hla_locus2=B*15%3A01&hla_locus3=C*03:04&hla_locus4=DRB1*11:01&hla_locus5=&hla_locus6=&hla_locus7=&hla_locus8=DQB1*05:03&hla_population=&hla_country=&hla_dataset=&hla_region=&hla_ethnic=&hla_study=&hla_sample_size=&hla_sample_size_pattern=equal&hla_sample_year=&hla_sample_year_pattern=equal&hla_loci=&hla_order=order_1

Následuje popis výpočtu, který provádí rozhodovací algoritmus pro všechny zbývající potenciální varianty konkrétního páru.

Rozšířená hodnoticí funkce

varianta 2, výpočet: $(128.0 + 128.0) * (4 + 4 - 0 + 128.0 + 128.0) = 86016.0$

Ohodnocení

varianta 2, výpočet: $128.0 + 128.0 = 256.0$

Pokud byl na začátku zvolen režim porovnávání s expertním řešením, potom výpis dále obsahuje informaci o případné shodě či neshodě s algoritmickým řešením. Pokud nastala shoda, je vypsáno identifikační číslo genotypu, znak „;“ a poté „shoda;“.

893407;shoda;

V případě neshody jsou nejprve vypsány podrobnosti o algoritmickém a expertním řešení. Následuje identifikační číslo genotypu, znak „;“ a slovo „neshoda“.za symbolem „;“ dále vypsán přehled o algoritmickém a expertním řešení a nejlepších vazbách pro případný pohodlný export do tabulky. Pokud dochází k neshodě a potenciální řešení i expertní řešení jsou oboje pouze jediné, pak po slově „neshoda“ a znaku „;“ následuje seznam lokusů, kde k neshodě dochází (vždy menší z množin genů).

algoritmický výsledek:

A*24:02;B*40:01;C*03:04;DRB1*11:01;DQB1*05:02

http://allelefrequencies.net/hla6003a.asp?page=1&hla_selection=&hla_locus1=A*02%3A02&hla_locus2=B*40%3A01&hla_locus3=C*03:04&hla_locus4=DRB1*11:01&hla_locus5=&hla_locus6=&hla_locus7=&hla_locus8=DQB1*05:02&hla_population=&hla_country=&hla_dataset=&hla_region=&hla_ethnic=&hla_study=&hla_sample_size=&hla_sample_size_pattern=equal&hla_sample_year=&hla_sample_year_pattern=equal&hla_loci=&hla_order=order_1

A*02:01;B*35:01;C*04:01;DRB1*01:01;DQB1*05:01

http://allelefrequencies.net/hla6003a.asp?page=1&hla_selection=&hla_locus1=A*02%3A01&hla_locus2=B*35%3A01&hla_locus3=C*04:01&hla_locus4=DRB1*01:01&hla_locus5=&hla_locus6=&hla_locus7=&hla_locus8=DQB1*05:01&hla_population=&hla_country=&hla_dataset=&hla_region=&hla_ethnic=&hla_study=&hla_sample_size=&hla_sample_size_pattern=equal&hla_sample_year=&hla_sample_year_pattern=equal&hla_loci=&hla_order=order_1

H1: ABCD,1.5554778058622678E-4,1,null,0.0,0,null,0.0,0

H2: ABCD-DrDq,2.7324574676343817E-7,1,null,0.0,0,null,0.0,0

expertní výsledek:

A*02:01;B*40:01;C*03:04;DRB1*11:01;DQB1*05:02

http://allelefrequencies.net/hla6003a.asp?page=1&hla_selection=&hla_locus1=A*02%3A01&hla_locus2=B*40%3A01&hla_locus3=C*03:04&hla_locus4=DRB1*11:01&hla_locus5=&hla_locus6=&hla_locus7=&hla_locus8=DQB1*05:02&hla_population=&hla_country=&hla_dataset=&hla_region=&hla_ethnic=&hla_study=&hla_sample_size=&hla_sample_size_patern=equal&hla_sample_year=&hla_sample_year_pattern=equal&hla_loci=&hla_order=order_1

A*24:02;B*35:01;C*04:01;DRB1*01:01;DQB1*05:01

http://allelefrequencies.net/hla6003a.asp?page=1&hla_selection=&hla_locus1=A*24%3A02&hla_locus2=B*35%3A01&hla_locus3=C*04:01&hla_locus4=DRB1*01:01&hla_locus5=&hla_locus6=&hla_locus7=&hla_locus8=DQB1*05:01&hla_population=&hla_country=&hla_dataset=&hla_region=&hla_ethnic=&hla_study=&hla_sample_size=&hla_sample_size_patern=equal&hla_sample_year=&hla_sample_year_pattern=equal&hla_loci=&hla_order=order_1

H1: ABCD,0.0011240501529865637,1,null,0.0,0,null,0.0,0

H2: ABCD-DrDq,7.882376737433474E-8,1,null,0.0,0,null,0.0,0

896554;neshoda;A;alg. řešení:;A*24:02;B*40:01;C*03:04;DRB1*11:01;DQB1*05:02;A*02:01;B*35:01;C*04:01;DRB1*01:01;DQB1*05:01;exp. řešení:;A*02:01;B*40:01;C*03:04;DRB1*11:01;DQB1*05:02;A*24:02;B*35:01;C*04:01;DRB1*01:01;DQB1*05:01;alg:;H1:

ABCD,1.5554778058622678E-4 P1,null P2,null P3;H2: ABCD-DrDq,2.7324574676343817E-7 P1,null P2,null P3;exp:;H1: ABCD,0.0011240501529865637 P1,null P2,null P3;H2: ABCD-DrDq,7.882376737433474E-8 P1,null P2,null P3;

Pokud na straně algoritmického nebo předchozího výsledku je více řešení a v některém nastala shoda, potom je proveden odpovídající výpis.

893407;shoda jedné ze 2 variant

Na konci výsledku je vypsána doba identifikace tohoto genotypu.

Doba výpočtu genotypu s id 74 je:0m2s39set

Na konci celého souboru je proveden výpis s průměrnou dobou identifikace jednoho haplotypového páru, počtem všech genotypů a shodných s předchozím výsledkem a dobou identifikace párů v celém souboru.

Průměrná doba výpočtu jednoho genotypu: 0m7s94set

Počet všech genotypů: 100

Počet shodných s expertním výsledkem: 76

Doba výpočtu: 12m54s72set

2.9. Struktura programu

Program obsahuje několik tříd. Zde jsou popsány jednotlivé:

UzivatelkaTrida – generuje uživatelské rozhraní v pracovní ploše Javy. Obsahuje metody pro načítání populačních dat to souborů a pro identifikaci haplotypů. Metoda *main* obsahuje uživatelské rozhraní. Nejdůležitější metodou je *vyhledavaniVazeb*, která vyhledává zpracovává všechny vstupní informace. Z této metody je pak volána *rozhodovaciMetoda*. Populační data jsou načtena z HTML kódu do dvojrozměrného pole metodou *pridavat*. Metoda *vytvoreniCsvSouboru* provádí načtení populačních dat z dvojrozměrného pole do souboru ve formátu *csv*. Jsou zde definovány metody s předponou názvu „*identifikace_*“, které umožňují nalézt kombinaci vazeb z názvu dané metody. Třída dále obsahuje metodu *zmenaPopulace*, která změní populaci na jinou vyznačující se indexem v parametru. Třída má naimplementovány url kódy populací, které jsou pak rozdělené do *P1*, *P2* a *P3*. Dokáže vygenerovat url adresu pro libovolný haplotyp. K tomu slouží metoda *url_vazby*.

Vazba – třída zastupující jednotlivou vazbu nebo kombinaci vazeb. Obsahuje metodu pro přičtení pravděpodobnosti vazby.

Varianta – třída zastupující variantu genotypu. Obsahuje metodu pro stanovení hodnotící funkce a jejího součtu, výpočet genové vyrovnanosti a také pro výpis vazeb do souboru. Dále také obsahuje vyčíslení hodnotící funkce pro jednotlivé kombinace a váhy kombinací pro výpočet pravděpodobnosti výskytu kombinace.

VariantyPGenu – třída vytvořená za účelem zastoupení spojového seznamu spojových seznamů. Ten v Javě vytvořit nelze. Tento seznam byl vytvořen za účelem správného rozkladu P kódů do potřebného počtu záznamů v populačních datech.

TimeWatch – třída sloužící pro měření času.

2.10. Omezení programu

Algoritmus nevyužívá k vyhledávání populační data na nízké úrovni rozlišení. Z časových důvodů se nepodařilo implementovat načítání vazeb z textového souboru. Vazby jsou načítány ze souborů *tabulka_Px.csv*. Při neshodě algoritmického a předchozího řešení je možné vypsát lokusy, kde došlo k neshodě. Tato možnost je však omezena na řešení pouze

s jednou výslednou variantou u obou řešení. Algoritmus není přizpůsoben k tvorbě rozdílových souborů. Není také možné ukládat populační data do csv souboru po určitých úsecích. Do souboru se tedy data nenačítají po každé zpracované straně, ale až po zpracování všech populačních stran, protože by se data načítala vždy znova od prázdného souboru. S tím souvisí riziko výpadku připojení k internetu a ztráty do té doby zpracovaných dat. Není tedy výhodné používat pro tyto účely bezdrátové připojení. Získané výsledky se ukládají výhradě do textového souboru. Pokud by s nimi uživatel chtěl nadále pracovat (např. pro statistické účely), byla by potřeba úprava programu, která by vedla k uchování výsledků.

2.11. Časová náročnost načítacího a identifikačního algoritmu

Tabulka 20 ukazuje časy trván všech podstatných procesů algoritmu. Tyto časy byly měřeny 100 genotypech z druhého souboru verifikačních dat. Nejpomalejší částí algoritmu je vytvoření seznamů kombinací a vazeb pro všechny jednotlivé populace. Lze postřehnout, že ať už je vznesen požadavek na porovnávání s předchozím řešením, rychlost tím prakticky není ovlivněna. Pro úplnost jsou v tabulce 21 uvedeny parametry počítače, na kterém byly procesy prováděny. Uvedení časové náročnosti algoritmu zastupuje výpočet asymptotické složitosti. Z důvodu komplexnosti programu není možné ji vyčíslit. Bohužel neexistuje ani plugin vývojového prostředí Eclipse SDK, kterým by ji bylo možné získat. Kvůli dynamickému rozšiřování populačních dat by navíc takový výpočet neměl dlouhodobý význam.

Proces	Čas
Příprava všech tří populací	2m30s
Načtení dat <i>P1</i>	38m58s52set
Načtení dat <i>P2</i>	112m48s58set
Načtení dat <i>P3</i>	245m21s40set
Identifikace pouze v populaci <i>P1</i> pro 100 genotypů	0m28s45set
Identifikace pouze v populacích <i>P1</i> a <i>P2</i> pro 100 genotypů	3m50s27set
Identifikace v populacích <i>P1</i> , <i>P2</i> a <i>P3</i> pro 100 genotypů	12m4s65set
Průměrná doba identifikace páru jednoho dárce za přítomnosti všech populací	5s90set
Procesy při porovnávání s předchozím řešením	
Identifikace pouze v populaci <i>P1</i> pro 100 genotypů	38s
Identifikace pouze v populacích <i>P1</i> a <i>P2</i> pro 100 genotypů	5m29s
Identifikace v populacích <i>P1</i> , <i>P2</i> a <i>P3</i> pro 100 genotypů	12m3s7set
Průměrná doba identifikace páru jednoho dárce za přítomnosti všech populací	0m5s50set

Tabulka 20: Časová náročnost různých procesů aplikace k srpnu 2017

Procesor	Intel® Core™ i5-3230M CPU@ 2.6GHz
Nainstalovaná paměť	8 GB
Typ systému	64bit
Verze systému	Windows 10 Home

Tabulka 21: Parametry počítače, na němž byla aplikace testována

2.12. Komplikace při implementaci metody pro stahování populačních dat

Při tvorbě *csv* souboru nastává problém s formátem desetinných čísel. Taková čísla, která jsou větší než 1, se přepisují na data s římskými číslicemi. Např. 1.12 se mění na I.12. Aplikace pro identifikaci s takovými hodnotami nedokáže pracovat. Bylo nutné najít řešení. Při načítání čísel do souboru je desetinná tečka přepsána na znak „!“. Je to z důvodu, že právě tento znak není určující pro stanovení čísla datem a hodnota tak zůstává zachována.

Protože evropská populace stále obsahuje haplotypy z *P1*, bylo nutné již při jejím načítání nepřidávat do seznamu haplotypů takové, které jsou již zahrnuty v *P1*. Bohužel web *allelefreqencies.com* neobsahuje filtr, který by *P1* z regionu Europe eliminoval. Rovněž je nevýhodné, že filtr na webu neumí zahrnout do výběru více než jednu populaci. Tím by bylo možné načítání HTML kódů výrazně urychlit. S tím souvisí poznatek, že stahování stran je pomalé. To se netýká jen ukládání do dvojrozměrného pole, ale již samotného procesu nahrání HTML kódu do řetězce. Bohužel, aniž by byl použit jiný programovací jazyk, tento problém nelze vyřešit.

Bylo též nutné čelit problému s výběrem správných genů v souhrnných populacích. Je těžké vysegmentovat geny *A,B,C,DRB1 a DQB1*. Informace o haplotypu jsou velmi nejednotné a často se v souboru vyskytují i takové geny, které k výpočtu nejsou využívány, např. *DQA1*.

Algoritmus není schopen načíst více než 219 stran najednou, protože Java má omezené paměťové možnosti. Tento problém nastal u populací *P2* a *P3*. Jako řešení se ukázalo ukládání webu do dvou řetězcových proměnných a následné získání dat postupně z každé z nich. Algoritmická složitost sice o několik kroků stoupne, ale nenastává riziko, že algoritmus po několika hodinách skončí chybou.

2.13. Komplikace při implementaci identifikačního algoritmu

Při tvorbě kódu se vyskytl problém s překročením bytového limitu metody. Tuto komplikaci bylo možné přejít rozdělením programu na více metod. Velmi obtížným úkolem se stala konfigurace rozhodovacího algoritmu. Vyhledat vhodné pořadí kritérií představuje komplexní problém, jenž vyžaduje mnoho experimentů a porovnávání s prototypovým řešením. Při vývoji algoritmu bylo vyzkoušeno několik verzí s velmi odlišným přístupem.

K současné podobě program dospěl na základě neuspokojivých výsledků předchozích návrhů. Algoritmus s vyhodnocováním typu varianty haplotypového páru byl dlouhý, neefektivní a nepřehledný. Jako nejlepší řešení se ukázala hodnoticí funkce. Je však těžké nastavit funkci tak, aby výsledek byl odpovídající. To se týká především haplotypů získaných v *P1*.

2.14. Optimalizace

Problém algoritmu je, že vyhledávací a rozhodovací část jsou striktně odděleny. Je nutné procházet celý prostor vazeb ve všech populacích, protože informace o dalších populacích jsou nezbytné k určení nejvhodnějšího haplotypového páru. Jestliže haplotyp nemá jakékoli vazby v *P1* ani v *P2*, pak jsou při rozhodování okamžitě připraveny vazby v *P3* bez výpočtu a je možné ihned podat přesné vyhodnocení takových párů. Algoritmus tak sice může působit neoptimálně, ale zato je univerzální. Nebylo implementováno stahování vazeb ze souborů, čímž by se identifikační proces urychlil. Nicméně současná podoba nabízí výhodu v tom, že je možné nezávisle dodat populační data a ta používat. Ač bylo usilováno o optimalizaci, není vyloučeno, že některé výpočty v programu jsou nadbytečné. To se týká především části s načítáním populačních dat, kde cílem byla hlavně spolehlivost a funkčnost.

2.15. Možnosti vylepšení a rozšíření algoritmu

Algoritmus vyžaduje mnoho kroků vedoucích ke zvýšení jeho spolehlivosti. Protože jsou všechny metody tříd dostatečně okomentovány, není velký problém provádět dodatečné úpravy. Především je potřeba nadále se zabývat nastavením hodnoticí funkce. Je nutné odstraňovat neshody s expertními daty v závislosti na preferencích uživatele. Je třeba používat co nejrozmanitější vstupní data, která by dala podnět k případnému ladění algoritmu. Dalším důležitým krokem do budoucna je lepší nastavení vah kombinací vazeb.

Vhodným vylepšením do budoucna je nastavení včasější účasti *P2* v rozhodování. Bylo by vhodné v budoucnu více využít počet variant alely v polymorfismu. V současnosti je zastoupena pouze prostřednictvím pravděpodobnosti vazby s polymorfismem. Nabízí se cesta zvětšení rozestupů v hodnocení mezi kombinacemi a změna v hodnocení polymorfismů.

Dalším námětem je přidání dalších kroků rozhodovacího algoritmu. Pokud by byla použita data dárců, která jsou hodně orientovaná na *P3*, bylo by třeba rozhodovací algoritmus rozšířit. Protože haplotypy z dostupných populačních dat byly identifikovatelné pouze s využitím nanejvýš *P2*, není zřejmé, jak se algoritmus bude chovat v jiných případech.

Jednoznačnou kapitolou k následným úvahám je zlepšení rozhodování na základě pravděpodobnosti. S využitím velkého množství vstupních dat by bylo možné lépe nastavit váhy vazeb a kombinací. Porovnávání pravděpodobnosti má nastavenou určitou toleranci. Potom i slabší výsledek, než nejlepší není vyřazen. Nicméně tolerance vyžaduje ještě důkladnější nastavení. Ideálním řešením by bylo, pokud by algoritmus vypočetl pravděpodobnosti obou nejlepších kombinací pro všechny zbývající varianty. Dále by však neporovnával váženou pravděpodobnost obou kombinací, ale porovnával by pouze kombinace vzájemně si odpovídající. Pokud by se u obou variant střídavě převyšovaly, zanechal by obě. Toto řešení je však náročné na implementaci.

Možnosti vylepšení algoritmu se týkají také optimalizace. Je zřejmé, že algoritmus musí prohledávat všechny kombinace vazeb. Díky nepřítomnosti polymorfismů může existovat lépe hodnocená varianta než ta včasněji nalezená. Je ovšem třeba zvážit, zda by nebylo možné nalézt způsob minimalizace počtu prozkoumaných kombinací v jedné populaci.

Uživatelské rozhraní není nastaveno na uchování populačních dat zpracovaných do dvojrozměrného pole. Pokud je spuštěn nový proces, probíhá celé načítání znovu. Možností vylepšení by tak bylo ověřit, která populační data už načtena byla. To se bohužel při vývoji nepodařilo. Pokud je řeč o uživatelském rozhraní, bylo by vhodné rozšířit ho a též výpisy do textových souborů na různé jazykové verze. Byl by tak využitelný celosvětově. Vzhledem k tomu, že algoritmus obsahuje minimum textů, nešlo by o obtížný krok.

Algoritmus je orientován na oblasti světa. Pokud bude kdekoli ve světě přidána nová populace, stačí pouze načíst všechna populační data a nová populace bude automaticky zahrnuta v příslušné velké populaci. Je též možné přizpůsobit preference populací podle vlastních potřeb, např. nastavit australskou populaci jako *P1*. Takto by program mohl být použitelný pro libovolnou světovou populaci. K nastavení preferencí oblastí světa a přiřazení jich do populací slouží pole celých čísel s názvy *spodni_int* a *horni_int*. Jak už název napovídá, obsahují indexy url adres, které jsou definovány v hlavičce hlavní třídy. Změnou

indexů ve dvou výše uvedených polích je možné změnit zařazení dílčích populací do souhrnné populace. Jiný způsob je změna pořadí konců adres v poli *konce_url*. Začátek všech používaných url adres je univerzální. Je obsažen v řetězcové proměnné *url_univerzalni_zacatek*. Při testování algoritmu je výhodné zúžit rozsah populací na 1, čímž se proces značně urychlí. To se však týká hlavně vývoje kódu algoritmu. Pokud je testován program po případné změně, která se týká identifikace, je vhodné spouštět ho minimálně na dvou populacích.

Konkrétním příkladem populace, u které by bylo vhodné zvážit zařazení (ideálně) do *P2*, je Israel Poland Jews. V takovém případě by bylo nutné:

1. deklarovat konec url adresy první stránky populace Israel Poland Jews v hlavičce hlavní třídy.
2. umístit tento nový konec url do pole *konce_url* na pozici, která sousedí s koncem url populací patřících do *P2*.
3. při načítání populačních dat zahrnout tuto populaci do *P2* a odebrat ji z *P3*. K tomuto kroku se používá řádek obsahující kód:

```
if
((index_populace==2&&(populace.contains("Poland")||populace.contains("Austria
minority")))||((index_populace==3&&populace.contains("Caucasian")&&
populace.contains("USA"))||(index_populace==3&&(populace.contains("Russia
"))))){
```

Tento řádek eliminuje populace s názvem obsahujícím text „*Poland*“ nebo „*Austria minority*“ z *P2*. Dále eliminuje populace s názvem obsahujícím text „*Caucasian*“ nebo „*Russia*“ z *P3*. Pro Israel Poland Jews je nutné jej stejným způsobem upravit. Výsledná podoba tedy bude:

```
if
((index_populace==2&&(populace.contains("Poland")||populace.contains("Austria
minority")))||((index_populace==3&&(populace.contains("Caucasian")&&
populace.contains("USA"))||(index_populace==3&&(populace.contains("Russia
")||populace.contains("Israel Poland Jews"))))){
```

4. změnit hodnoty *spodni_int* a „*horni_int*“ podle rozsahu url souhrnných populací.

Jako poslední námět k vylepšení lze uvést propojení algoritmu s databází navrženou např. v jazyce SQL. Navzdory obtížné implementaci takového rozhraní by tak bylo možné dosáhnout vyšší efektivity programu. Identifikované haplotypy by bylo možné snadno uchovávat.

2.16. Původní návrhy implementace algoritmu

Původní návrh řešení byl velmi primitivní a plynul z neznalosti o počtu a charakteru vazeb existujících v populaci *P1*. Plánem bylo projít všechny kombinace *B,C* a *B,DRBI* v *P1*, které jsou vázané jednoznačně a na ně poté navazovat stejným způsobem ostatní geny až posléze sestavit haplotyp. Tento haplotyp by pak bylo možné přiřadit k jedné z variant vzniklých po rozkladu genotypu jedince. Protože haplotypů by v *P1* nebylo velké množství, nečinilo by potíže rychle určit přesnou formu haplotypů jedince. Po průchodu dat bylo dokázáno, že tato metoda není funkční, protože jednoznačných vazeb lze nalézt jen naprosté minimum, ať už v jakékoli výše uvedené populaci. Navíc tyto haplotypy neodpovídaly žádné kombinaci, kterou by bylo možné najít mezi dárci. Nelze tedy algoritmus stavět pouze na jednoznačných vazbách.

Druhý návrh byl rozsáhlejší a pokrýval všechny možnosti vyhledávání, včetně polymorfismů. V návrhu byl kladen důraz na nalezení haplotypu prvotně v *P1*. Při neúspěchu pokračovalo vyhledávání v populacích *P2* a *P3*. Smyslem tohoto návrhu byl průchod všech cest nejprve v jedné populaci a ukládání nalezených vazeb. Pokud haplotyp nešlo nalézt, pokračoval celý proces v jiné populaci. Při nalezení haplotypu cyklus skončil a na řadu se dostaly další geny. V opačném případě vyhledávání pokračovalo v dalších populacích. Výhodou bylo zaměření se na geneticky nejbližší populace pro domácí dárce. Nevýhodou pak byla neefektivita a několikanásobný průchod jedné cesty. Ukončení vyhledávání při nálezu haplotypu vedlo k nerelevantním výsledkům, které nekorespondovaly s ostatními výsledky.

3. Závěr

Práce se zabývala vývojem algoritmu pro identifikaci řídkých haplotypů. Zadáním bylo nejprve prostudovat problematiku řídkých haplotypů a genových vazeb, dále pak analyzovat možnosti využití dostupných HLA dat a možnost aktualizace dat. Hlavním bodem zadání bylo navrhnout a implementovat algoritmus pro identifikaci řídkých haplotypů. Ten pak měl být podroben validaci porovnáním s řešením expertního identifikačního algoritmu.

V programovacím jazyce Java se podařilo vytvořit aplikaci, která dokáže stáhnout populační data z databáze webu *allelefrequencies.com* a pomocí nich identifikovat jakýkoliv haplotyp na vysoké úrovni rozlišení alel. Je tedy připraven k využití v praxi. Populační data je možné kdykoli zaktualizovat. Kromě nich jsou vstupem genotypová data dárců, pro které se haplotypy identifikují. Algoritmus je rovněž schopen porovnávat aktuální řešení s předchozími výsledky. Program využívá mnoho vah a koeficientů, které byly voleny experimentálně s využitím teoretických znalostí.

Vývoj algoritmu se v první fázi zaměřoval na implementaci algoritmu pro získávání populačních dat z webu. Ta nejsou dostupná jiným způsobem než vyhledáváním dat v HTML kódu. V dalších fázích vývoje byly odstraňovány nedostatky načítacího algoritmu, které vyvstaly při aktualizaci populačních dat. Druhou fází byl vývoj identifikačního algoritmu. Bylo obtížné najít způsob, jak haplotypy spolehlivě a efektivně identifikovat. Algoritmus tak několikrát změnil svou podobu v závislosti na neshodách s expertním řešením, až nabyl současné formy. Následovala fáze ladění vah za účelem dosažení co nejhodnějšího výsledku. Dále bylo vytvořeno uživatelské prostředí a umožněno automaticky porovnávat aktuální výstup algoritmu s předchozími výsledky. Na konci bylo provedeno odstranění přebytečných výpočtů a celý program byl okomentován.

Současná podoba algoritmu je postavena na nejschůdnějším postupu identifikace haplotypů. Vyhledává nejlepší kombinace vazeb genů z každé populace u každé varianty haplotypového páru. Z variant ohodnocených speciální funkcí vyjadřující kvalitu páru pak vybírá tu nejvhodnější. Tento postup je sice výpočetně náročný, ale zaručuje, že rozhodnutí je jen málokdy chybné. Expertní postup založený na populačním modelu a vylučování variant již při identifikaci je sice efektivnější, ale je implementačně mnohem náročnější. Navíc může odstranit použitelnou variantu již na začátku výpočtu. Protože počet celosvětově nalezených alel se neustále zvětšuje, mohou mít expertní předpoklady omezenou platnost.

U každého haplotypu jsou kombinace ihned vyhledávány ve všech populacích. To může být pro některá data dárců neefektivní, avšak pro úplnost algoritmu je to nezbytné. Výsledky, které algoritmus dává, jsou odůvodnitelné a ve většině případů použitelné. Algoritmus identifikuje haplotypy v přijatelně krátké časové době. Není paměťově náročný. Je vybaven měřením a výpisem časů vlastních procesů. V aktuální verzi jsou však potřeba další opravy a vylepšení. Týkají se především nastavení hodnotící funkce a využití polymorfismů genů. Nevýhodou algoritmu je, že byl testován pouze na dvou množinách verifikačních dat. S tím souvisí také určitá neshoda výsledků algoritmu s expertním řešením. Ta byla způsobena mj. odlišným způsobem vyhodnocování kombinací vazeb a také tím, že expertní algoritmus nepracoval s aktuálními daty. Není možné tvrdit, že program je stoprocentně optimální. Neoptimálnost se však týká pouze některých výpočtů. Velkým problémem algoritmu je dlouhá doba načítání populačních dat. Bez použití jiného programovacího jazyka je tento nedostatek zřejmě neřešitelný. Výsledky nejsou trvale uchovávány, protože algoritmus nevyužívá databázových systémů. Výstupem je pouze textový soubor s výsledky a změřenou dobou trvání procesu.

V teoretické části práce je rozebrána transplantační imunologie, HLA systém, haplotypy a základní metody jejich identifikace. Dále je věnována kapitola genové vazbě a vazební nerovnováze. Plynule je navázáno praktickou částí, tzn. návrhem metody pro identifikaci řídkých haplotypů. Pro tuto metodu jsou nejprve definovány souhrnné populace. Následně je popsán způsob načítání populačních a dárcovských dat. Poté je uveden princip metody, hodnocení kombinací vazeb, váhy populací a výpočty parametrů. Podrobně je zde popsán rozhodovací algoritmus a uvedeno několik příkladů rozhodnutí. Následující kapitolou je analýza neshod výsledků algoritmu a expertních řešení. Tato část obsahuje podrobný rozbor všech příčin neshod a opět je všechno vysvětleno na příkladech. Poslední část se zabývá mj. časovou náročností programu a možnostmi vylepšení. Rovněž jsou zde stručně uvedeny komplikace při tvorbě algoritmu, struktura programu, a nakonec původní návrhy identifikačního algoritmu.

Podobný algoritmus, který by načítal populační data z databáze výše uvedeného webu a zároveň identifikoval všechny haplotypy, předtím vyvinut nebyl. To lze označit za největší přínos této práce. Program může být i přes své nedostatky použitelný v lékařské praxi při vyhledávání dárců krvetvorných buněk.

4. Použitá literatura

- [1] Lékařské slovníky: HLA antigen. [Online] [Citace: 6. 5. 2014.] <http://lekarske.slovníky.cz/lexikon-pojem/hla-antigeny-hla-system>.
- [2] Jindra, P. Kostní dřeň. HLA data dárců - pravidla reportování, záznam do databáze ČNRDD. [Online] [Citace: 21. 3. 2012.] <http://www.kostnidren.cz/registr/odbornici/specificka-cast.php..>
- [3] Systém HLA a prezentace antigenu. [Online] imunologie.lf2.cuni.cz/soubory_vyuka/cz_medici3_2.ppt.
- [4] Chvojková, M. Podpora tvorby aplikace pro určení kompatibility příznaků. *Bakalářská práce na Fakultě aplikovaných věd Západočeské univerzity na katedře kybernetiky, vedoucí bakalářské práce: Ing. Lucie Houdová.* 2012.
- [5] Pospíšilová Š., Dvořáková D., Mayer J. *Molekulární hematologie.* 2013. 978-80-7262-942-8.
- [6] ŠTÍPEK, S. *Stručná biochemie uchování a exprese genetické informace.*
- [7] JINDRA, P. Kostní dřeň. HLA data dárců - pravidla reportování, záznam do databáze ČNRDD. [Online] [Citace: 21. 3. 2012.] <http://www.kostnidren.cz/registr/odbornici/specificka-cast.php..>
- [8] MCCULLOUGH, Jeffrey J. *Transfusion medicine.* West Sussex : UK: Wiley-Blackwell, 2011. 14-443-3705-X.
- [9] Chvojková, M. Podpora tvorby aplikace pro určení kompatibility příznaků. *Bakalářská práce na Fakultě aplikovaných věd Západočeské univerzity na katedře kybernetiky, vedoucí bakalářské práce: Ing. Lucie Houdová.* 2012.
- [10] Nomenclature. [Online] <http://hla.alleles.org/alleles/index.html>.
- [11] Cvanová, M. Matematické metody hodnocení genových polymorfizmů v biomedicínském výzkumu. *Diplomová práce na Institutu biostatistiky a analýzy Masarykovy univerzity v Centru pro výzkum toxických látek v prostředí MU, vedoucí diplomové práce: doc. RNDr. Ladislav Dušek, Dr.* 2011.

- [12] Steiner, D. Doctoral Thesis: PROBABILISTIC MATCHING IN SEARCH FOR UNRELATED HEMATOPOIETIC STEM CELL DONORS. Praha : autor neznámý, 2013.
- [13] Švojgrová, M., Koza, V. a Hamplová, A. *Transplantace kostní dřeně: Průvodce Vaší léčbou*. Plzeň : F. S. Publishing,, 2006. Sv. 1. vyd. ISNB 80-903560-2-8..
- [14] Jindra, P. Kostní dřeň. HLA data dárců - pravidla reportování, záznam do databáze ČNRDD. [Online] [Citace: 21. 3. 2012.]. [Online] <http://www.kostnidren.cz/registr/odbornici/specificka-cast.php...>
- [15] ZARZĄDZANIE POPULACJAMI ZWIERZĄT. [Online] <http://kgohz.sggw.pl/wp-content/uploads/2014/02/Zarz%C4%85dzanie-pop.-r%C3%B3wnowaga-genetyczna-cz.-I.pdf>.
- [16] FLEGR, J. *Evoluční biologie*. Praha : Academia, 2005. Sv. 1. vydání. ISBN 80-200-1270-2.
- [17] Velký lékařský slovník. [Online] <http://lekarske.slovníky.cz/>.
- [18] *Diplomová práce: Matematické metody hodnocení genových polymorfizmů v biomedicinském výzkumu*. Cvanová, M. Brno : autor neznámý, 2011.
- [19] OTOVÁ, B.,. *Lékařská biologie a genetika I. díl*. místo neznámé : Praha : Karolinum, 2008. Sv. 1. vydání.
- [20] Haplotype-specific linkage disequilibrium patterns define the genetic topography of the human MHC. [Online] 3 2003. <http://hmg.oxfordjournals.org/content/12/6/647.full>.
- [21] Nature reviews genetics, Linkage disequilibrium — understanding the evolutionary past and mapping the medical future. [Online] <http://www.nature.com/nrg/journal/v9/n6/full/nrg2361.html?foxtrotcallback=true>.
- [22] Brilliant Maps. *The Genetic Map Of Europe*. [Online] 4. 21 2015. <http://brilliantmaps.com/the-genetic-map-of-europe/>.
- [23] Linkage disequilibrium and recombination . [Online] 2005. http://bio.classes.ucsc.edu/bio107/Class%20pdfs/W05_lecture15.pdf.
- [24] Analýza haplotypů. [Online] <http://ccy.zcu.cz/registr/CZ-HANA.html>.

[25] The Allele Frequencies in Worldwide Populations. [Online]
<http://allelefrequencies.net>.