

Slovenská technická univerzita v Bratislave  
Fakulta informatiky a informačných technológií

FIIT-5208-72264

Bc. Martin Nemček

# Redukcia cytometrických dát na základe hustoty a predikcia stavu pacienta z klinických dát

Diplomová práca

Študijný program: Informačné systémy

Študijný odbor: Informačné systémy

Miesto vypracovania: Ústav informatiky, informačných systémov a softvérového inžinierstva,  
FIIT STU, Bratislava

Supervisor: doc. RNDr. Mária Lucká, PhD.

Máj 2018



# Návrh zadania diplomovej práce

Revízia č.: 1<sup>1</sup>

## Študent:

**Meno, priezvisko, tituly:** Martin Nemček, Bc.  
**Študijný program:** Informačné systémy  
**Kontakt:** mrtm.nemcek@gmail.com

## Výskumník:

**Meno, priezvisko, tituly:** Mária Lucká, doc. RNDr. PhD.

## Projekt:

**Názov:** Redukcia cytometrických dát na základe hustoty a predikcia stavu pacienta z klinických dát  
**Názov v angličtine:** Density based downsampling of cytometry data and clinical outcome prediction using clinical data  
**Miesto vypracovania:** Ústav informatiky, informačných systémov a softvérového inžinierstva, FIIT STU, Bratislava  
**Oblasť problematiky:** Bioinformatika

## Text návrhu zadania<sup>2</sup>

Hmotnostnou cytometriou je možné získať veľké množstvo cytometrických dát. Manuálne spracovanie týchto dát je neefektívne, neškálovateľné a subjektívne a preto na odstránenie problémov pri manuálnom spracovaní bolo vyvinutých viacero softvérových nástrojov. Tieto nástroje využívajú nedeterministické prístupy, aby boli použiteľné na veľké dátové množiny, čo znemožňuje reprodukovateľnosť výsledkov. V cytometrických dátach je možné identifikovať bunkové populácie, ktoré podľa hustoty môžu byť redundantné alebo vzácne. Na identifikovanie vzácných bunkových populácií je nutné vykonať redukciu dát na základe hustoty, ktorej výsledkom je podmnožina dát s vyrovnanou hustotou v priestore. S dostupnosťou cytometrických a klinických dát pacientov je potenciál predikovať klinický výsledok pacienta, ako aj nájsť závislosti medzi klinickými a cytometrickými dátami.

Analizujte existujúce algoritmy na spracovanie cytometrických dát a redukciu na základe hustoty na identifikáciu vzácných bunkových populácií a metódy na predikciu klinického výsledku pacienta s využitím klinických dát. Navrhnite a implementujte algoritmus na redukciu dát na základe hustoty, ktorý bude plne deterministický a zároveň použiteľný na veľkých dátových množinách. S použitím klinických dát navrhnite model na predikciu klinického stavu pacienta. Výsledky svojho riešenia overte a porovnajte s existujúcimi metódami.

<sup>1</sup> Vytlačiť obojstranne na jeden list papiera

<sup>2</sup> 150-200 slov (1200-1700 znakov), ktoré opisujú výskumný problém v kontexte súčasného stavu vrátane motivácie a smerov riešenia

### Literatúra<sup>3</sup>

- Robert V. Bruggner, Bernd Bodenmiller, David L. Dill, Robert J. Tibshirani, and Garry P. Nolan. Automated identification of stratifying signatures in cellular subpopulations. Proceedings of the National Academy of Sciences, 111(26):E2770-E2777, 2014. doi: 10.1073/pnas.1408792111
- Peng Qiu. Toward deterministic and semiautomated spade analysis. Cytometry. Part A : the journal of the International Society for Analytical Cytology, 91:281-289, Mar 2017. doi: 10.1002/cyto.a.23068

Vyššie je uvedený návrh diplomového projektu, ktorý vypracoval(a) Bc. Martin Nemček, konzultoval(a) a osvojil(a) si ho doc. RNDr. Mária Lucká, PhD. a súhlasí, že bude takýto projekt viesť.

V Bratislave dňa 8.2.2018

Podpis študenta

Podpis výskumníka

### Vyjadrenie garanta predmetov Diplomový projekt I, II, III

Návrh zadania schválený: áno / nie<sup>4</sup>

Dňa: ..... 12. 2. 2018 .....

Podpis garanta predmetov

<sup>3</sup> 2 vedecké zdroje, každý v samostatnej rubrike a s údajmi zodpovedajúcimi bibliografickým odkazom podľa normy STN ISO 690, ktoré sa viažu k téme zadania a preukazujú výskumnú povahu problému a jeho aktuálnosť (uvedte všetky potrebné údaje na identifikáciu zdroja, pričom uprednostnite vedecké príspevky v časopisoch a medzinárodných konferenciách)

<sup>4</sup> Nehodiace sa prečiarknite

## Návrh zadania diplomovej práce

*Finálna verzia do diplomovej práce<sup>1</sup>*

### Študent:

**Meno, priezvisko, tituly:** Martin Nemček, Bc.  
**Študijný program:** Informačné systémy  
**Kontakt:** mrt.nemcek@gmail.com

### Výskumník:

**Meno, priezvisko, tituly:** Mária Lucká, doc. RNDr. PhD.

### Projekt:

**Názov:** Určenie rozsahu zmeny odozvy na liek podľa variácií v genetickom kontexte  
**Názov v angličtine:** Determining the scale of alteration of drug response based on genetic variations  
**Miesto vypracovania:** Ústav informatiky, informačných systémov a softvérového inžinierstva, FIIT STU, Bratislava  
**Oblasť problematiky:** Bioinformatika

### Text návrhu zadania<sup>2</sup>

Bežne dostupné lieky sú vyrábané a testované na určitej vzorke ľudí a aj preto z pravidla dosahujú horšie výsledky v praxi ako pri testovaní a nie sú účinné na veľkú podskupinu populácie pacientov. Niektorí pacienti zažijú pri užívaní liekov škodlivú reakciu na liek, pričom niektoré lieky sú účinné len po určitú dobu. Na odozvu na lieky vplýva niekoľko faktorov zakódovaných v DNA sekvencii pacienta. Identifikácia a pochopenie týchto faktorov sú kľúčové kroky potrebné na zavedenie personalizovanej medicíny, optimalizácií výberu liekov, veľkosti dávok a dĺžky liečby.

Analyzujte spôsoby určovania a predikcie odozvy na lieky u pacienta. Identifikujte, ktoré gény a ich genetické variácie v genetickom kontexte pacienta vplývajú na zmenu odozvy na užitia liekov. Navrhnite efektívny prístup, ako tieto gény a variácie identifikovať v osekvenovanej sekvencii DNA. Z identifikovaných variácií v genetickom kontexte určte zmenu odozvy na liek u pacienta. Navrhnuté riešenie implementujte, vhodne overte a výsledky porovnajte s existujúcimi metódami riešenia problému.

<sup>1</sup> Vytlačiť obojstranne na jeden list papiera

<sup>2</sup> 150-200 slov (1200-1700 znakov), ktoré opisujú výskumný problém v kontexte súčasného stavu vrátane motivácie a smerov riešenia



### Literatúra<sup>3</sup>

- Z. Pulijz and H. Vikalo, "Decoding Genetic Variations : Communications- Inspired Haplotype Assembly," IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 13, no. 3, pp. 518-530, May/June 2016.
- G. J. Hua, C. Y. Tang, C. L. Hung and H. Zheng, "Drug resistance gene identification algorithm for next-generation sequencing data," 2014 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Belfast, 2014, pp. 17-21. doi: 10.1109/BIBM.2014.6999381

Vyššie je uvedený návrh diplomového projektu, ktorý vypracoval(a) Bc. Martin Nemček, konzultoval(a) a osvojil(a) si ho doc. RNDr. Mária Lucká, PhD. a súhlasí, že bude takýto projekt viesť v prípade, že bude pridelený tomuto študentovi.

V Bratislave dňa 12.1.2017


  
Podpis študenta

  
Podpis výskumníka

### Vyjadrenie garanta predmetov Diplomový projekt I, II, III

Návrh zadania schválený: áno / nie<sup>4</sup>

Dňa: .....13.12.2017.....

  
Podpis garanta predmetov

<sup>3</sup> 2 vedecké zdroje, každý v samostatnej rubrike a s údajmi zodpovedajúcimi bibliografickým odkazom podľa normy STN ISO 690, ktoré sa viažu k téme zadania a preukazujú výskumnú povahu problému a jeho aktuálnosť (uvedte všetky potrebné údaje na identifikáciu zdroja, pričom uprednostnite vedecké príspevky v časopisoch a medzinárodných konferenciách)

<sup>4</sup> Nehodiace sa prečiarknite

## Zadanie diplomovej práce

*Meno študenta:* **Bc. Martin Nemček**

*Študijný program:* Informačné systémy

*Študijný odbor:* Informačné systémy

*Názov práce:* **Určenie rozsahu zmeny odozvy na liek podľa variácií v genetickom kontexte**

Samostatnou výskumnou a vývojovou činnosťou v rámci predmetov Diplomový projekt I, II, III vypracujte diplomovú prácu na tému, vyjadrenú vyššie uvedeným názvom tak, aby ste dosiahli tieto ciele:

*Všeobecný cieľ:*

Vypracovaním diplomovej práce preukážte, ako ste si osvojili metódy a postupy riešenia relatívne rozsiahlych projektov, schopnosť samostatne a tvorivo riešiť zložité úlohy aj výskumného charakteru v súlade so súčasnými metódami a postupmi študovaného odboru využívanými v príslušnej oblasti a schopnosť samostatne, tvorivo a kriticky pristupovať k analýze možných riešení a k tvorbe modelov.

*Špecifický cieľ:*

Vytvorte riešenie zodpovedajúce návrhu textu zadania, ktorý je prílohou tohto zadania. Návrh bližšie opisuje tému vyjadrenú názvom. Tento opis je záväzný, má však rámcový charakter, aby vznikol dostatočný priestor pre Vašu tvorivosť.

Riadte sa pokynmi Vášho vedúceho.

Pokiaľ v priebehu riešenia, opierajúc sa o hlbšie poznanie súčasného stavu v príslušnej oblasti, alebo o priebežné výsledky Vášho riešenia, alebo o iné závažné skutočnosti, dospejete spoločne s Vaším vedúcim k presvedčeniu, že niečo v texte zadania a/alebo v názve by sa malo zmeniť, navrhnete zmenu. Zmena je spravidla možná len pri dosiahnutí kontrolného bodu.

*Miesto vypracovania:* Ústav informatiky, informačných systémov a softvérového inžinierstva, FIIT STU Bratislava

*Vedúci práce:* **doc. RNDr. Mária Lucká, PhD.**

*Termíny odovzdania:*

Podľa harmonogramu štúdia platného pre semester, v ktorom máte príslušný predmet (Diplomový projekt I, II, III) absolvovať podľa Vášho študijného plánu

*Predmety odovzdania:*

V každom predmete dokument podľa pokynov na [www.fiit.stuba.sk](http://www.fiit.stuba.sk) v časti:  
home > Informácie o > štúdiu > organizácia štúdia > diplomový projekt.

V Bratislave dňa 13. 2. 2017

SLOVENSKÁ TECHNICKÁ UNIVERZITA  
V BRATISLAVE  
Fakulta informatiky a informačných technológií  
Ilkovičova 2, 842 16 Bratislava 4  
1

prof. Ing. Pavol Návrat, PhD.  
riaditeľ Ústavu informatiky, informačných systémov  
a softvérového inžinierstva





## **ČESTNÉ PREHLÁSENIE**

Čestne prehlasujem, že diplomovú prácu s názvom: Redukcia cytometrických dát na základe hustoty a predikcia stavu pacienta z klinických dát som vypracoval samostatne, na základe konzultácií a štúdia odbornej literatúry. Neporušil som autorský zákon a zoznam použitej literatúry som uviedol na príslušnom mieste.

.....

Bc. Martin Nemček



## **POĎAKOVANIE**

Chcem sa poďakovať svojej vedúcej doc. RNDr. Márie Luckej, PhD. za odbornú pomoc, cenné rady, ochotu a nasmerovania ma pri tvorbe tejto práce. Taktiež sa chcem poďakovať RNDr. Jane Jakubíkovej, PhD. a RNDr. Dane Cholujovej, PhD. zo Slovenskej akadémie vied za cenné rady z domény cytometrie, nové poznatky, konzultácie, overenie výsledkov z biologického hľadiska a užitočné rady k výskumu. Okrem toho sa chcem poďakovať svojmu kolegovi Bc. Oliveri Moravčíkovi a priateľovi Ing. Richardovi Arpášovi za hodnotné debaty a cenné rady ohľadne výskumu.



# Anotácia

Slovenská technická univerzita v Bratislave

FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLOGIÍ

Študijný program: Informačné systémy

Autor: Bc. Martin Nemček

Diplomová práca: Redukcia cytometrických dát na základe hustoty a predikcia stavu pacienta z klinických dát

Vedúci práce: doc. RNDr. Mária Lucká, PhD.

Máj 2018

Identifikácia bunkových populácií je jeden z prvých a veľmi dôležitých krokov v analýze cytometrických dát. Bunkové populácie sú identifikované zhľukovaním, ktoré na veľkých dátových množinách cytometrických dát je často neefektívne. Z dôvodu nerovnomernej reprezentácie bunkových populácií v dátach majú zhľukovacie algoritmy problémy s identifikovaním vzácných bunkových populácií. Nami navrhnuté riešenie na redukciiu dát a rovnomernú reprezentáciu bunkových populácií je založené na nami navrhnutej váhovenej hustote a rieši viacero problémov existujúcich riešení. Navrhli sme efektívny prístup výpočtu hustoty buniek využitím delenia priestoru a stromovej indexovej štruktúry a následnej redukcie dát na základe hustoty aplikovaním nami navrhnutého iteratívneho prístupu. Navrhnuté riešenie dosiahlo niekoľkonásobné zrýchlenie v porovnaníach s existujúcimi riešeniami. Identifikované bunkové populácie zo zredukovaných dát sme vizualizovali v stromovej štruktúre a porovnali s výsledkami existujúcich riešení, s ktorými sme dosiahli konzistentné výsledky a správnosť výsledkov nášho návrhu potvrdili aj doménový experti. Navrhli sme proces extrakcie črt z cytometrických a klinických dát, vykonali viacero predikcií klinického stavu a porovnali výsledky s existujúcimi riešeniami, kde sme dosiahli najlepšie výsledky. Na záver sme identifikovali možné vylepšenia nášho návrhu.





# Annotation

Slovak University of Technology Bratislava

FACULTY OF INFORMATICS AND INFORMATION TECHNOLOGIES

Degree Course: Information Systems

Author: Bc. Martin Nemček

Master thesis: Density based downsampling of cytometry data and clinical outcome prediction using clinical data

Supervisor: doc. RNDr. Mária Lucká, PhD.

May 2018

Identification of the cellular populations is one of the first and important steps in analysis of cytometry data. The cellular populations are commonly identified by clustering algorithms which are ineffective on big cytometry data sets. Significant differences in the representations of the cellular populations can lead to inability of the clustering algorithms to identify the rare populations. Our proposed algorithm for density-based downsampling and uniform representations of the cellular populations is based on novel weighted density and addresses multiple drawbacks of existing solutions. We proposed an effective algorithm for density calculation utilizing a space partitioning and an index tree structure followed by a novel iterative approach to density-based downsampling. The proposed solution achieved significant improvements of time complexity over the existing solutions. The cellular populations identified from the downsampled data were visualized in a tree structure and compared with the results of existing solutions which yielded consistent results. The results were also confirmed by the domain specialists. We proposed a process of extracting features from the cytometry and clinical data used for predictions of clinical outcome. Multiple predictions were realized and compared with results of the existing solutions where the best results were achieved. At the conclusion we identified possible improvements to our proposed solution.



# Obsah

<b>1</b>	<b>Úvod</b>	<b>1</b>
1.1	Motivácia . . . . .	2
1.2	Prehľad práce . . . . .	2
<b>2</b>	<b>Metódy analýzy cytometrických dát</b>	<b>4</b>
2.1	Cytometria . . . . .	4
2.1.1	Cytometrické dáta . . . . .	5
2.1.2	Predspracovanie cytometrických dát . . . . .	5
2.2	Gating . . . . .	6
2.2.1	DeepCyTOF . . . . .	7
2.3	Vizualizácia cytometrických dát . . . . .	9
2.3.1	viSNE . . . . .	9
2.3.2	PhenoGraph . . . . .	10
2.3.3	Minimálna kostra stromu . . . . .	11
2.4	Zhlukovacie metódy . . . . .	11
2.4.1	SPADE . . . . .	12
2.4.2	FlowSOM . . . . .	13
2.5	Detekcia vývojovej trajektórie buniek . . . . .	15
2.5.1	Wanderlust . . . . .	15
2.6	Predikcia klinického stavu . . . . .	16
2.6.1	Súťaž FlowCAP-II . . . . .	17
2.6.2	Citrus . . . . .	19
2.6.3	COMPASS . . . . .	20
2.6.4	Náhodný les . . . . .	21
2.6.5	Sieť Elastic . . . . .	22
2.7	Redukcia dát . . . . .	23
2.7.1	Redukcia dát na základe hustoty . . . . .	23
2.7.2	Problémy redukcie dát na základe hustoty . . . . .	25
2.8	Dátové štruktúry . . . . .	26
2.8.1	kd-strom . . . . .	26

2.8.2	r-strom	27
2.9	Zhrnutie	27
<b>3</b>	<b>Návrh riešenia</b>	<b>29</b>
3.1	Špecifikácia	29
3.1.1	Dáta	31
3.2	Výpočet hustoty buniek	31
3.2.1	Strom hustôt	33
3.2.2	Stavba stromu hustôt	33
3.2.3	Pamäťová náročnosť	34
3.2.4	Časová náročnosť	35
3.2.5	Paralelizácia riešenia	36
3.3	Váhovaná hustota	37
3.4	Redukcia dát na základe hustoty	38
3.4.1	Postup redukcie	39
3.4.2	Iteratívny prístup	40
3.4.3	Úprava váh	41
3.4.4	Odstránenie šumu	42
3.5	Upsampling	43
3.5.1	Navrhnutý algoritmus	43
3.5.2	Paralelizácia riešenia	44
3.6	Predikcia klinického stavu pacienta	45
3.6.1	Extrakcia črt z cytometrických dát	45
3.6.2	Extrakcia črt z klinických dát	46
3.6.3	Predikcia použitím cytometrických črt	47
3.6.4	Predikcia použitím klinických a cytometrických črt	47
3.6.5	Interpretácia výsledkov	49
3.7	Vizualizácia	50
3.7.1	Stromová vizualizácia bunkových populácií	50
3.7.2	Vizualizácia výsledkov predikcie teplotnými mapami	51
3.8	Zhrnutie	52
<b>4</b>	<b>Implementácia</b>	<b>54</b>
4.1	Výpočtový projekt	54
4.2	Vizualizačný projekt	55
4.3	Projekt analýzy	55
4.4	Zhrnutie	55



<b>5</b>	<b>Výsledky</b>	<b>57</b>
5.1	Výpočet hustoty . . . . .	58
5.2	Váhovaná hustota . . . . .	58
5.3	Redukcia dát na základe hustoty . . . . .	59
5.3.1	Iteratívny prístup redukcie dát na základe hustoty . . . . .	60
5.3.2	Výsledky redukcie dát na základe hustoty . . . . .	61
5.3.3	Odstránenie šumu v procese redukcie dát na základe hustoty . . . . .	62
5.3.4	Úprava váh v procese redukcie dát na základe hustoty . . . . .	63
5.4	Upsampling . . . . .	64
5.5	Porovnanie rýchlostí . . . . .	65
5.5.1	Porovnanie rýchlosti výpočtu hustoty buniek . . . . .	66
5.5.2	Porovnanie rýchlosti redukcie dát na základe hustoty . . . . .	67
5.6	Overenie biologických výsledkov . . . . .	68
5.6.1	Porovnanie na kostnej dreni myši . . . . .	70
5.7	Predikcia klinického stavu pacienta . . . . .	71
5.7.1	Predikcia z cytometrických dát . . . . .	71
5.7.2	Predikcia použitím klinických dát . . . . .	75
5.7.3	Porovnanie s FlowCAP-II . . . . .	76
5.8	Interpretácia výsledkov predikcie . . . . .	78
5.9	Zhrnutie . . . . .	80
<b>6</b>	<b>Zhodnotenie</b>	<b>82</b>
6.1	Možnosti rozšírenie práce . . . . .	84
	<b>Literatúra</b>	<b>86</b>
<b>A</b>	<b>Dokumentácia</b>	<b>A-1</b>
A.1	Inštalácia . . . . .	A-1
A.2	Spustenie implementácie . . . . .	A-1
A.3	Používateľská príručka . . . . .	A-2
<b>B</b>	<b>Electronické médium</b>	<b>B-1</b>
<b>C</b>	<b>Plán práce</b>	<b>C-1</b>
<b>D</b>	<b>Vizualizácie</b>	<b>D-1</b>
D.1	Porovnanie stromových vizualizácií . . . . .	D-1
D.2	Porovnanie vizualizácií na dátach kostnej drene myši . . . . .	D-4
<b>E</b>	<b>IIT.SRC článok</b>	<b>E-1</b>



# Kapitola 1

## Úvod

Cytometria je veda zaoberajúca sa skúmaním a meraním charakteristík buniek, pričom existujú dve rozšírené metódy na meranie cytometrických dát a to prietoková a hmotnostná cytometria. Obe spomenuté metódy majú veľa využití v biológii, medicíne alebo v imunológii, kde sa používajú na identifikáciu bunkových populácií a objavovanie nových biologických ukazovateľov, na základe ktorých sa dá následne sledovať a predikovať stav pacienta [29].

Dáta vygenerované pri cytometrických experimentoch sú komplexné, viacrozmerné dáta obsahujúce údaje z niekoľko stotisíc buniek a 18 až 40 ukazovateľmi pre každú bunku podľa použitej metódy, pričom v teórii je možné s hmotnostnou cytometriou vyprodukovať až 100 ukazovateľov pre jednu bunku [32, 33, 29].

Dátová analýza cytometrických dát sa tradične vykonávala manuálnym gatingom. Ide o manuálne zaradenie jednotlivých buniek do diskretných skupín na základe bunkového typu určeného podľa ukazovateľov buniek. Tento proces je vykonávaný na dvojrozmernom bodovom grafe, kde sa nedá v jednom momente zachytiť viac ako dva bunkové ukazovatele a preto pri viacrozmerných cytometrických dátach sa stáva tento proces takmer nepoužiteľný a neškálovateľný. Manuálny gating má aj ďalšie výrazné nedostatky a tými sú vnášanie subjektivity expertov do výsledkov a z toho vyplývajúca ťažká reprodukovateľnosť a porovnateľnosť. Pri gatingu sa expert zameriava na konkrétne populácie buniek a preto sa podstatne znižuje šanca objavenia nových bunkových populácií [37, 15, 5].

Je nevyhnutné použitie výpočtových metód na uľahčenie a automatizáciu procesu manuálneho gatingu a celkovo analýzy komplexných cytometrických dát. Na riešenie spomenutých problémov manuálneho gatingu bolo doposiaľ vyvinutých viacero metód a nástrojov, ktoré využívajú hlavne zhukovacie algoritmy na objavenie bunkových populácií a následná vizualizácia týchto dát je často vykonávaná použitím techník redukcie dimenzionality. Zaujímavé je aj nedávne použitie hlbokých neurónových sietí na automatizáciu procesu gatingu [22].

Veľká početnosť cytometrických dát je problémom pre zaužívané prístupy analýzy dát. Navyše bunkové populácie identifikovateľné z cytometrických dát sú v dátach reprezentované nerovnomerne a to redundantné a vzácne bunkové populácie. Pre správnosť analýz cytometrických dát je nevyhnutná korektná identifikácia všetkých aj vzácných, malo zastúpených

bunkových populácií. Použitím zhlukovacieho algoritmu na identifikáciu bunkových populácií nebudú vzácne bunkové populácie správne, prípadne vôbec, identifikované práve kvôli veľkému nepomeru zastúpení v dátach. Na vyriešenie tohto problému je potrebné vykonať redukciu dát, ktorá zabezpečí rovnomernú reprezentáciu bunkových populácií v dátach [28].

Okrem cytometrických dát sú v imunológii často používané aj klinické dáta pacientov ako doplnkové informácie ku cytometrickým dátam. Klinické dáta je možné použiť spolu s cytometrickými dátami na predikciu klinického stavu pacientov. Výsledky predikcie poskytujú možnosť na nájdenie a extrahovanie nových vzťahov medzi cytometrickými a klinickými dátami, objavenie nových biologických ukazovateľov a vyvodenie doposiaľ neznámych záverov, ktoré môžu byť použité v nasledujúcich analýzach. Na predikciu klinického stavu pacientov s využitím ako aj cytometrických, tak aj klinických dát, bolo vytvorených niekoľko softvérových nástrojov ako Citrus [13] alebo COMPASS [24].

Vyvodenie relevantných biologických, medicínskych alebo imunologických záverov z analýzy cytometrických dát vyžaduje veľkú znalosť problémovej oblasti. Často je to náročný, zdĺhavý a z časti subjektívny proces. Vhodná vizualizácia výsledkov, ako aj medzi-výsledkov analýzy cytometrických dát dokáže značne zrýchliť a uľahčiť doménovým expertom proces vyvodzovania záverov z výsledkov analýzy.

## 1.1 Motivácia

Analýza cytometrických dát je časovo náročný proces, na ktorého automatizáciu a zrýchlenie bolo vyvinutých viacero prístupov, ktoré sú z hľadiska biologických výsledkov na veľmi dobrej úrovni, ale zaostávajú v efektívnosti výpočtovej sily. Našu prácu sme riešili v spolupráci s doktorkami zo Slovenskej akadémie vied, ktoré používajú viacero softvérových nástroj na analýzu cytometrických dát, ako aj jeden z celkovo najrozšírenejších softvérových nástrojov SPADE [28, 27], ktorý umožňuje identifikáciu bunkových populácií a ich následnú vizualizáciu. Doktorky pri práci s nástrojom SPADE narazili na viacero problémov, z ktorých hlavným bol veľmi neefektívny výpočtový čas na ich veľkých dátových množinách ako aj nereprodukovateľnosť výsledkov kvôli využitiu stochastických prístupov pri spracovaní dát.

Efektívne a inteligentne spracovanie cytometrických dát poskytuje možnosť ušetriť veľa času v procese analýzy. Nami navrhovaný prístup sa zameriava na efektívne spracovanie a analýzu veľkých cytometrických dátových množín za použitia deterministických prístupov pri zachovaní správnosti výsledkov aj z biologického hľadiska.

## 1.2 Prehľad práce

V práci analyzujeme dostupné softvérové nástroje, algoritmy a iné prístupy na analýzu cytometrických dát. Pri analýze sa zameriavame hlavne na automatizované prístupy spracovania

cytometrických dát, vizualizácie bunkových populácií a predikcie klinického stavu pacienta. Následne navrhujeme riešenie na efektívny výpočet hustoty buniek v priestore a redukciu dát na základe hustoty na vyrovnanie hustoty a reprezentácií jednotlivých bunkových populácií v priestore a vizualizáciu výsledkov analýzy. V ďalšej časti opisujeme použité programovacie jazyky, knižnice a nástroje na implementovanie nami navrhnutého prístupu. Na záver overujeme nami dosiahnuté výsledky na viacerých dátových množinách, ale aj s doménovými expertmi.





## Kapitola 2

# Metódy analýzy cytometrických dát

Tradične sa na analýzu cytometrických dát využíval manuálny gating, ktorý sa ale s nárastom komplexnosti cytometrických dát stal takmer nepoužiteľný. Na spracovanie komplexných cytometrických dát bolo vyvinutých viacero metód a nástrojov. Väčšina z nich využíva zhukovacie algoritmy na identifikovanie bunkových populácií na základe podobností nameraných hodnôt ukazovateľov, pričom niektoré na to používajú prístupy s učiteľom a niektoré bez učiteľa. Vizualizácia cytometrických dát je zložitá úloha, keďže ide o mnohopočetné a viacrozmerné dáta. Z toho dôvodu väčšina vizualizačných metód používa techniky redukcie dimenzionality, prípadne najskôr vykoná zhukovanie buniek a následne sa vizualizujú zhuky buniek. Špeciálnou možnosťou nástrojov na analýzu cytometrických dát sú nástroje, ktoré detegujú vývojovú trajektóriu buniek, teda prechod medzi jednotlivými stavmi bunky v čase, čo sa využíva na predikciu vývoja bunky. Okrem toho existujú metódy na koreláciu klinických dát s cytometrickými dátami, z ktorej je možné identifikovať vplyv bunkových populácií na zmenu v klinických dátach, prípadne predikovať klinické výsledky podľa cytometrických a klinických dát.

### 2.1 Cytometria

Cytometria je veda zaoberajúca sa skúmaním a meraním charakteristík buniek. Na meranie sa používajú dva spôsoby a to prietoková cytometria a hmotnostná cytometria. Pre komplexnosť cytometrických dát je nevyhnutný vývoj nových dátových analýz a výpočtových a štatistických nástroj na podporu biologického výskumu [32]. Cytometria má veľa využití v biológii aj medicíne, a napríklad v imunológii sa používa najmä na identifikáciu a kvantifikáciu bunkových populácií, čo sa dá ďalej využiť na monitorovanie imunitného stavu pacienta a detegovanie nových biologických ukazovateľov, na základe ktorých by sa dal predikovať stav pacienta [29].

Prietoková cytometria je dlho používaný prístup na meranie ukazovateľov buniek, ktorý spravidla dokáže pre jednu bunku vyprodukovať 14 – 18 parametrov, pričom nedávno sa stali komerčne dostupné nástroje prietokovej cytometrie, ktoré umožňujú namerať až 30 parametrov pre bunku [29, 1]. Podľa autorov článku [14] je možné mať v blízkej budúcnosti prietokovú

cytometriu, ktorá vyprodukuje 50 parametrov pre jednu bunku.

Hmotnostná cytometria je pomerne nový druh cytometrie, ktorá dokáže namerať 40, a v teórii až 100, parametrov pre bunku. V procese hmotnostnej cytometrie sú merané bunky zničené, čo má za dôsledok potrebu nástrojov, ktoré vedia bunky zoradiť [32, 33, 29].

### 2.1.1 Cytometrické dáta

Cytometrické dáta sú viacrozmerné dáta, ktoré obsahujú 14 až 40 ukazovateľov pre každú inštanciu. Dátové množiny spravidla obsahujú stotisíce inšancií, kde inštancia zväčša reprezentuje jednu bunku a parametre reprezentujú ukazovatele jednotlivých buniek. Dáta sa dajú preto reprezentovať maticou, kde riadky sú jednotlivé bunky a stĺpce zodpovedajú nameraným ukazovateľom buniek. Ukážka cytometrických dát reprezentovaných maticou je na obrázku 2.1.

Štandardom na ukladanie cytometrických dát je binárny súborový formát FCS. Skladá sa z troch segmentov, a to textový segment, ktorý obsahuje metadáta vo formáte kľúč → hodnota, dátový segment obsahujúci pole dát a segment analýzy. Dátové pole sa skladá z jednotlivých výskytov (angl. event) v riadkoch, z ktorých väčšina sú jednotlivé bunky a ukazovatele (angl. markers) v stĺpcoch. FCS súborový formát umožňuje uchovanie viacerých dátových množín v jednom súbore [31].

	Time	Event_length	Rh103Di	In113Di	In115Di	Xe131Di
[1,]	2327.344	21	0.0000000	0.0000000	0.0000000	0.2380648
[2,]	2487.604	26	0.0000000	0.0000000	0.0000000	0.0000000
[3,]	3066.536	24	0.0000000	0.0000000	0.0000000	0.0000000
[4,]	3217.083	22	5.9205546	0.0000000	4.658334	0.0000000
[5,]	3377.018	26	0.0000000	0.0000000	0.0000000	0.0000000
[6,]	3524.609	28	0.0000000	2.442713	0.0000000	0.0000000
[7,]	3820.651	33	0.0000000	4.955865	0.0000000	0.5844795
[8,]	3846.862	22	0.0000000	0.0000000	0.0000000	0.0000000
[9,]	4278.893	16	1.6994513	0.0000000	0.0000000	0.0000000
[10,]	4329.857	16	0.8504162	5.358715	0.0000000	0.0000000

Obr. 2.1: Ukážka cytometrických dát

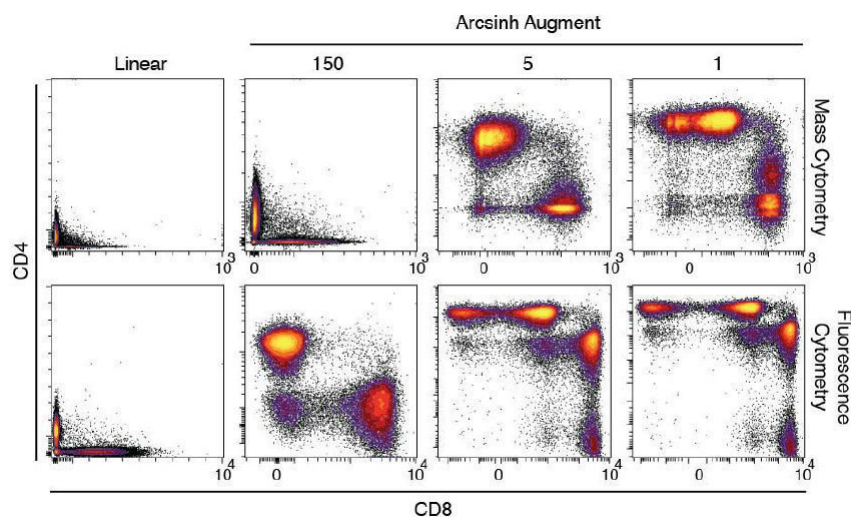
### 2.1.2 Predspracovanie cytometrických dát

Pedspracovanie cytometrických dát je dôležitý krok pred ich analýzou aby sme dosiahli biologicky presné výsledky. Bunkové populácie extrahované z cytometrických dát získaných prietokovou cytometriou majú zväčša log-normálnu distribúciu a preto sa zvyknú transformovať do logaritmickej škály [30].

Avšak cytometrické dáta môžu obsahovať záporné hodnoty, na ktoré nie je vhodná logaritmickej transformácia a preto vzniklo viacero transformácií na adresovanie tohto problému,

napríklad *Logicle* transformácia, ktorú využili aj autori nástroja FloReMi [36] na predspracovanie dát, ktorí dosiahli najlepšie výsledky predikcie klinického stavu pacienta v porovnaní viacerých nástrojov autormi [2]. Okrem *Logicle* transformácie sa na predspracovanie používajú aj *Hyperlog* [6], *Box-Cox* [25] transformácie a najrozšírenejšia inverzná hyperbolická sínusová transformácia – *arcsinh*. *Arcsinh* transformáciu využíva, okrem iného, nástroj SPADE [28] a taktiež ju použili autori [9] pri analýze cytometrických dát a používa sa aj v softvérovom nástroji Cytobank <sup>1</sup>.

*Arcsinh* transformácia je závislá od viacerých parametrov a hlavne od parametru kofaktor. Autori pri analýze cytometrických dát identifikovali vhodnú hodnotu parametru kofaktor ako pre cytometrické dáta získane prietokovou aj hmotnostnou cytometriou. Pre dáta získane prietokovou cytometriou sa jedná o hodnotu kofaktoru rovnú  $1/150$  a pri dátach z hmotnostnej cytometrie to je  $1/5$ . Na obrázku 2.2 sú vizualizované cytometrické dáta získane hmotnostnou a prietokovou cytometriou bez aplikovania transformácie a po aplikovaní *arcsinh* transformácie s rôznymi hodnotami parametra kofaktor [9].



Obr. 2.2: Aplikovanie *arcsinh* transformácie na cytometrické dáta získane hmotnostnou a prietokovou cytometriou [9]

## 2.2 Gating

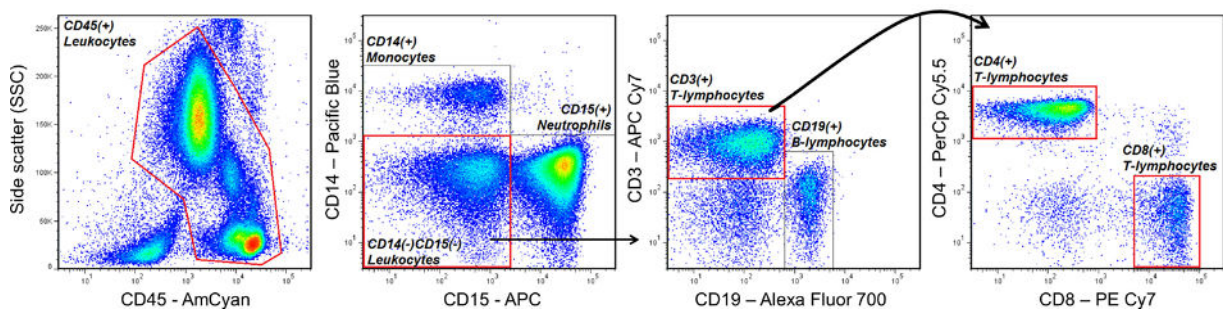
Dátová analýza cytometrických dát sa tradične vykonávala manuálnym gatingom, ktorý je jedným zo základných prístupov a procesov v cytometrickej analýze. Ide o manuálne zaradenie jednotlivých buniek do diskretných skupín na základe bunkového typu určeného podľa ukazovateľov jednotlivých buniek za použitia vizuálnej prehliadky dvojrozmerných bodových grafov. Bunkové populácie, ktoré sú zaujímavé pre výskumníka sú identifikované z rodičovských po-

<sup>1</sup> [www.cytobank.org](http://www.cytobank.org) – nástroj zoskupujúci viacero nástrojov na analýzu cytometrických dát

populácií pomocou vizuálnej prehliadky bodov na grafe reprezentujúce hodnoty jednotlivých ukazovateľov buniek. Gating je veľmi dôležitý krok, a zároveň prekážkou, v analýze cytometrických dát [22, 37, 38].

Gating sa skladá z časovo náročných manuálnych krokov, ktoré vnášajú do celého procesu viacero problémov. Výsledky podliehajú subjektivite jednotlivých expertov a spôsobujú ťažkú reprodukovateľnosť a porovnateľnosť. Gating sa spolieha na presné prahové hodnoty pri klasifikácii buniek do podmnožín. Z toho dôvodu bunky, ktorých ukazovatele sú tesne nad alebo pod prahovou hodnotou, sa nemusia klasifikovať korektne, prípadne vôbec. Keďže sa expert pri manuálnom gatingu zameriava na vopred určené bunkové populácie, zabraňuje to možnosti detekcií neznámych, biologicky relevantných, bunkových populácií. Pre veľkú dimenzionalitu cytometrických dát je manuálny gating neškálovateľný. V neposlednom rade, proces gatingu je veľmi pomalý, pričom rýchlosť je závislá od počtu skúmaných ukazovateľov a buniek v danej dátovej množine. Použitie vhodných automatických metód namiesto manuálneho gatingu dokáže znížiť časovú náročnosť z hodín na minúty [37, 15, 5].

Pri viacrozmerných údajoch sú spomenuté problémy ešte výraznejšie, keďže existuje veľké množstvo dvojrozmerných projekcií. Na podporu manuálneho gatingu boli vyvinuté viaceré čiastočne alebo úplne automatické analytické metódy [38]. Najpoužívannejšie metódy pre automatizáciu procesu gatingu sú zhukovacie algoritmy a to hlavne k-means a algoritmy zhukovania založené na modeli. Zhukovacie algoritmy založené na modeli nepotrebujú vopred pomocné údaje na vstupe a sú robustné voči tvaru bunkových populácií, avšak za cenu vyššej časovej náročnosti [37]. Nedávno sa podarilo na automatizáciu gatingu použiť aj hlbokú neurónovú sieť [22].



Obr. 2.3: Úkážka procesu manuálneho gatingu [37]

## 2.2.1 DeepCyTOF

Metódy hlbokého učenia sú momentálne veľmi obľúbené metódy s dosiahnutými výbornými výsledkami vo viacerých problematikách ako spracovanie obrazu, spracovanie prirodzeného jazyka alebo rozpoznávanie vzorov. Navyše, nedávne experimenty použitia metód hlbokého učenia v genomike a biomedicínskych aplikáciach potvrdili ich flexibilitu na spracovanie komplexných dát. Prietoková a hmotnostná cytometria vyprodukuje priemerne dáta pre  $10^5$  až  $10^6$



buniek, ktoré majú omnoho viac inštancií (buniek) ako atribútov. Keďže metódy hlbokého učenia vyžadujú veľký počet tréningových inštancií, tak cytometrické dáta sú vhodné práve na analýzu pomocou metód hlbokého učenia [22].

Pri cytometrických experimentoch prichádza k variáciám spôsobených napríklad odlišnou kalibráciou nástrojov, ktorá sa nazýva „batch” efekt. Tieto variácie komplikujú manuálny gating. Aby sa predišlo nutnosti vykonať gating nad každou dátovou množinou zvlášť, je potrebné vykonať procedúru adaptácie domén. Procedúra adaptácie domén je zoskupenie techník, ktoré umožňujú použitie modelu učenia, natrénovaného na dátach zo zdrojovej domény s danou distribúciou, na cieľovú doménu s podobnou, ale nie rovnakou distribúciou. Cieľom doménovej adaptácie je minimalizácia chyby inštancií v cieľovej doméne [22].

DeepCyTOF je rámec s integrovaným hlbokým učeníom a adaptáciou domén, ktorý vyžaduje jednu referenčnú vzorku z dát spracovanú manuálnym gatingom na následné spracovanie celej vzorky dát automatickým gatingom. DeepCyTOF obsahuje dva kroky predspracovania a to odstránenie šumu a doménovú adaptáciu. Následne po vykonaní predspracovania je vykonaná klasifikácia buniek. Všetky tri kroky sú implementované pomocou neurónových sietí. Na odstránenie šumu je použitá *denoising autoencoder* (DAE) neurónová sieť. Doménová adaptácia na kalibráciu medzi zdrojovou vzorkou a cieľovými vzorkami je vykonaná pomocou MMD-ResNet a klasifikácia jednotlivých buniek je vykonaná *depth-4 feed-forward* neurónovou sieťou. [22]

Dáta z hmotnostnej cytometrie môžu obsahovať nulové dáta na pozíciách rôznych ukazovateľov, ktoré nereflektujú skutočné biologické javy, ale sú spôsobené meracími prístrojmi. DeepCyTOF odstraňuje takýto šum využitím malej neurónovej siete DAE, ktorá je natrénovaná na bunkách, ktoré neobsahujú nulové dáta alebo iba veľmi málo. DAE sieť je použitá na rekonštrukciu čistých dát bez šumu z dát, ktoré šum obsahovali.

Na doménovú adaptáciu je použitý prístup *MMD-ResNet*. MMD-ResNet je prístup hlbokého učenia, ktorý sa učí mapovanie, ktoré dokáže kalibrovať distribúciu zdrojovej vzorky, aby zodpovedala distribúcii cieľovej vzorky.

Pred vykonaním klasifikácie buniek sa vyberie z dát referenčná zdrojová vzorka. Na výber referenčnej zdrojovej vzorky je pre každú vzorku vypočítaná kovariančná matica  $d \times d$ , kde  $d$  je počet ukazovateľov v týchto vzorkách. Následne sa pre všetky dvojice vzoriek vypočíta Frobeniová norma rozdielu medzi ich kovariančnými maticami. Vzorka, ktorá má najmenšiu priemernú vzdialenosť od všetkých ostatných vzoriek, je vybraná ako referenčná. Táto vzorka je následne spracovaná manuálnym gatingom a výstupné dáta sú ako jediné opísané dáta použité na klasifikáciu. Klasifikácia je vykonaná použitím *depth-4 feed-forward* neurónovej siete, skladajúca sa z troch softplus skrytých vrstiev a softmax výstupnej vrstvy.

Autori otestovali DeepCyTOF na piatich datasetoch zo štvrtej úlohy z FlowCAP-I súťaže a výsledky porovnali s výsledkami víťazov na jednotlivých datasetoch. DeepCyTOF dosiahol lepšie výsledky v štyroch z piatich datasetoch, pričom v jednom dosiaholi podobné výsledky ako aktuálny víťaz.

## 2.3 Vizualizácia cytometrických dát

Na vizualizácia vysoko-rozmerných cytometrických dát nie sú dostatočné tradičné bodové diagramy a preto bolo vyvinutých viacero alternatív, ktoré dokážu presnejšie zachytiť viac-rozmernosť dát v dvojrozmernom priestore. Prístupy k vizualizácii cytometrických dát sa dajú rozdeliť na dva hlavné prístupy, a to s využitím metód redukcie dimenzionality a s využitím zhlučovacích algoritmov s následnou vizualizáciou výsledných zhlučkov [29].

Cieľom prístupov využívajúcich metódy redukcie dimenzionality je čo najlepšie zachytiť nelineárnosť vzťahov v dátach a lepší pohľad na dáta, čo napomáha objavovaniu nových, skrytých bunkových populácií. Avšak tieto prístupy zahrňujú simplifikáciu dát, keďže nie všetky detaily z viac-rozmerného priestoru je možné zachovať v nízko-rozmernom priestore [29, 1].

Prístupy využívajúce zhlučovacie algoritmy najskôr zoskupenia dáta do zhlučkov, ktoré následne vizualizujú v dvojrozmernom priestore [29].

Väčšina vizualizačných metód je stochastická. Aby boli dosiahnuté akceptovateľné výpočtové časy, metódy využívajú prvky náhodnosti, čo má za dôsledok rozdielne výsledky vizualizácie nad rovnakými dátami. Pri prístupoch využívajúcich redukciu dimenzionality, ktoré dokážu vizualizovať dáta na úrovni jednotlivých buniek, na dosiahnutie akceptovateľných výpočtových časov je zväčša potrebné vykonať vizualizáciu iba na podmnožine celkových dát. Pričom, ak potrebujeme pre jednotlivé bunky mať aj informáciu o jej type, zhlučovací algoritmus je nevyhnutnosťou [29]. Možným riešením problému s rôznymi výsledkami pri vizualizácii rovnakých dát práve kvôli náhodnosti môže byť viacnásobná vizualizácia rovnakých dát a následná hlavná vizualizácia ako priemer predošlých vizualizácií.

### 2.3.1 viSNE

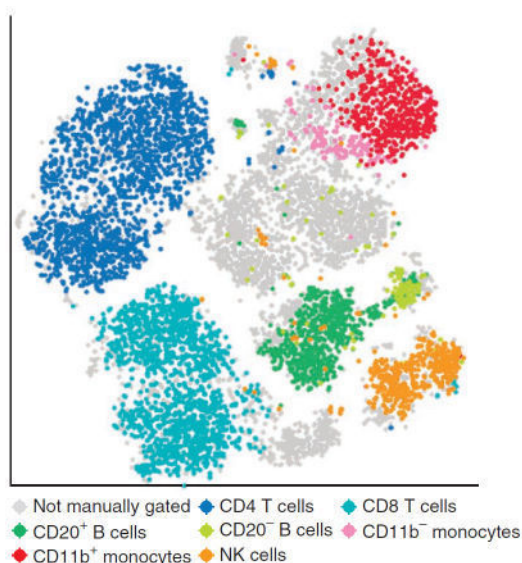
Vysoko-rozmerné cytometrické dáta sa ťažko zrozumiteľne vizualizujú. Jednobunkové dáta sa často vizualizujú v dvojrozmernom priestore, avšak s pribúdajúcim počtom ukazovateľov je množina možných párov na dvojrozmernú vizualizáciu priveľká. Navyše pri zobrazovaní viac-rozmerných dát v dvojrozmernom priestore môžu chýbať informácie, ktoré sa nedajú v dvojrozmernom priestore zobraziť [5].

Na vyriešenie tohto problému existuje niekoľko metód, ako napríklad SPADE alebo Principal component analysis (PCA). Avšak, problém pri metóde SPADE je získanie popisov jednotlivých zhlučkov po zhlučovaní, čo je zväčša priemer jednotlivých zhlučkov, pri ktorom sa stratí jednobunková informácia. PCA zobrazuje dáta v dvojrozmernom priestore so zachovaním jednobunkových informácií, avšak PCA je lineárna transformácia, a preto nedokáže zachytiť nelineárne vzťahy [5].

viSNE je algoritmus bez učiteľa, ktorý sa zaraďuje medzi algoritmy nelineárnej redukcie dimenzionality. Umožňuje vizualizáciu vysoko-rozmerných jednobunkových dát a je založený na algoritme t-distribúovaného stochastického vkladania susedov (t-Distributed Stochastic Ne-

ighbor Embedding - t-SNE). viSNE hľadá dvojrozmernú reprezentáciu jednobunkových dát, ktorá najlepšie zachová lokálne a globálne zostavenie. Vizualizácia je podobná bixiálnemu grafu, avšak pozícia buniek reflektuje ich blízkosť vo viacrozmernom priestore a nie v dvojrozmernom. Farba je použitá ako tretia dimenzia na vizualizáciu vlastností buniek [5].

Autori použili viSNE na vizualizáciu zdravých a chorých (leukémia kostnej drene a akútna myeloidna leukémia) vzorkách. Pri zdravých vzorkách viSNE zobrazil populácie buniek. Pri chorých vizualizácia zobrazila veľký jednotný útvar, ktorý sa líšil od zdravých vzoriek. Na vyhodnotenie rozdielov medzi zdravými a chorými vzorkami použili Jenson-Shannon odchýlku, ktorá sa pohybovala medzi hodnotami 0,42 – 0,45 pri porovnaní chorých a zdravých vzoriek a pri porovnaní zdravých vzoriek medzi sebou dosahovala hodnotu 0,04.



Obr. 2.4: Výsledky algoritmu viSNE na dátach z hmotnostnej cytometrie [5]

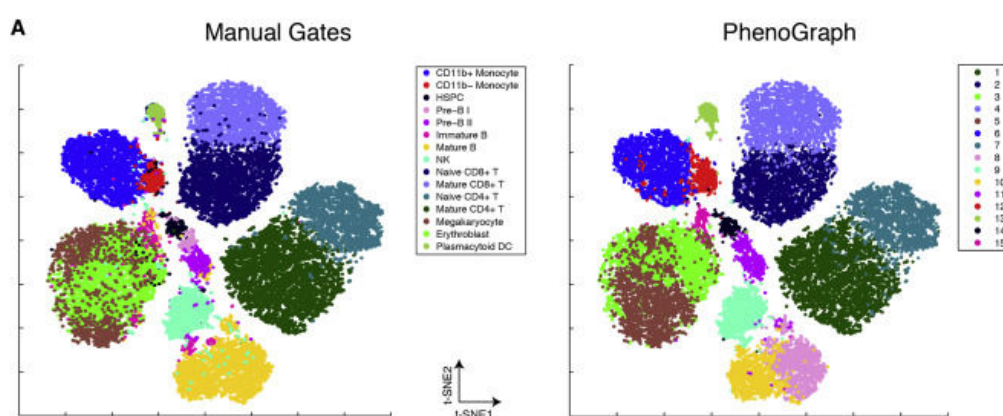
### 2.3.2 PhenoGraph

Techniky redukcie dimenzionality, ako napríklad t-SNE, napomáhajú vizualizovať dáta, ale neidentifikujú a nerozdeľujú dáta na jednotlivé populácie. PhenoGraph je algoritmus, ktorý vo viacrozmernom priestore nájde a roztriedi populácie buniek [21].

Problémom je výpočtová náročnosť a štatistická nepresnosť pri detekcii hustoty vo vysoko-rozmernom priestore. PhenoGraph modeluje vysoko-rozmerný priestor grafom najbližších susedov. V tomto grafe je každá bunka reprezentovaná uzlom spojeným s hranami s ich najpodobnejšími bunkami. Takýto graf kompaktne uchováva viacrozmerné dátové štruktúry, pričom zachováva fenotypovú príbuznosť. Po zostrojení grafu najbližších susedov je problém detekcie hustoty pretransformovaný na problém nájdenia vrcholov s veľa medziprepojeniami. Na vyriešenie tohto problému bol použitý algoritmus z oboru sociálnych sietí na rozdelenie veľkých sociálnych sietí na komunity [21].

V prípade cytometrických dát komunity predstavujú zoskupenie fenotypovo podobných buniek. Rozdelenie grafu na komunity teda rozdelí dáta na fenotypovo súvisiace populácie buniek. Algoritmus rozdelenia na komunity nepredpokladá veľkosť, počet ani tvar populácií, avšak určuje, že komunity nemôžu byť konvexné, symetrické alebo elipsoidné, čo su otázne predpoklady pre bunkové populácie [21].

V nesprávne zostrojenom grafe, prípadne v grafe obsahujúci veľké množstvo šumu môže dôjsť ku zakrytiu zriedkavých bunkových populácií veľkými populáciami. Na vyriešenie tohto problému sa graf zostrojuje dvakrát, pričom sa používa Jaccardov koeficient podobnosti pri druhom zostrojení tak, že podobnosť buniek je po druhej iterácii vyhodnotená ako počet spoločných susedných buniek vrámci oboch zostrojení grafu. Jaccardova metrika odhalí lokálnu hustotu pre každý bod, pričom odstráni falošné hrany. Spoločný výskyt zriedkavých populácií v blízkosti rovnakého fenotypu produkuje silne poprepájané moduly, ktoré sú ľahko rozoznateľné od šumu a celkovo sú ľahšie viditeľné vo výslednom grafe [21].



Obr. 2.5: Porovnanie výsledkov manuálneho gatingu a PhenoGraph algoritmu [21]

### 2.3.3 Minimálna kostra stromu

Graf minimálnej kostry stromu spája uzly tak, že suma vetiev je minimálna, tým pádom je uzol spojený s uzlami, s ktorými si je najpodobnejší, pričom sa berie do úvahy viacrozmernosť. Výsledkom je acyklický graf [35].

## 2.4 Zhlučovacie metódy

Cieľom zhlučovacích algoritmov je zoskupiť bunky do odlišných zhlukov tak, že podobné bunky, reprezentujúce rovnaký fenotyp, sa nachádzajú v rovnakom zhluku a odlišné bunky sa nachádzajú v rôznych zhlukoch [15].

Na zhlučovanie cytometrických dát je možné využiť viacero zhlučovacích algoritmov, pričom najpoužívanjšie sú zhlučovanie na základe hustoty, zhlučovanie na základe modelu,

samo-organizujúce sa mapy, hierarchické zhľukovanie, k-means a k-najbližších-susedov algoritmy [38].

Hlavnou nevýhodou použitia zhľukovacieho algoritmu pri analýze cytometrických dát je strata možnosti vizualizácie na úrovni jednotlivých buniek [15].

### 2.4.1 SPADE

SPADE je softvérový nástroj zastrešujúci celý proces analýzy cytometrických dát, od redukcie dát, cez identifikáciu bunkových populácií zhľukovaním po vizualizáciu výsledkov v stromovej štruktúre. SPADE vníma bunky ako body v priestore, teda vo viacrozmernom oblaku bodov.

Algoritmus je rozdelený do štyroch hlavných krokov. V prvom kroku sa vykonáva redukcia dát na základe hustoty. Bunkové populácie môžu byť redundantne, hojne vyskytujúce sa, ktoré sú v dátach zastúpené veľkým počtom buniek a vzácne, ktoré sú zastúpené podstatne menším počtom buniek v dátach. Cieľom redukcie dát na základe hustoty je teda vykonanie vyrovňania hustoty v priestore tak, aby vo výsledku boli hojné aj vzácne bunkové populácie zastúpené približne rovnakým počtom buniek. SPADE využíva viacero stochastických prístupov pri výpočte hustoty a následnej redukcie dát, ako aproximácia hustoty a veľkosti bunkového okolia a je závislý od viacerých parametrov výslednej hustoty a veľkosti okolia [28].

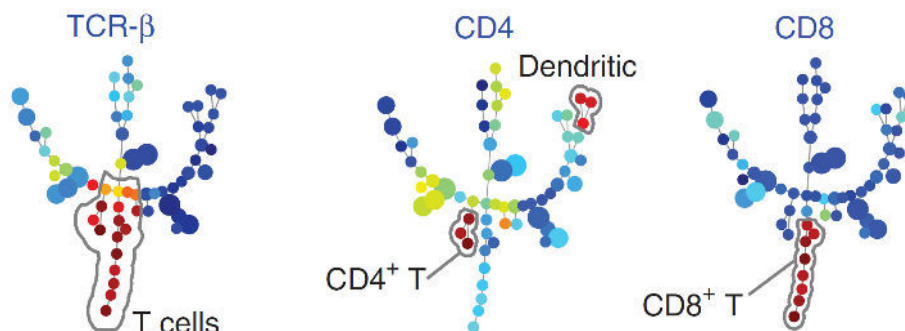
V druhom kroku algoritmus SPADE vykonáva identifikáciu bunkových populácií učením bez učiteľa a to konkrétne použitím algomeratívneho zhľukovania. Bunkové populácie sa zhľukovacím algoritmom identifikujú na základe podobností fenotypov, teda hodnôt nameraných cytometrických znakov [28].

V nasledujúcom kroku je zo zhľukov – bunkových populácií – zostavený graf a na základe vzdialeností vrcholov je z neho extrahovaná minimálna kostra grafu [28].

Posledný krok vykonáva proces priradenia všetkých buniek, ktoré boli v prvom kroku, na základe redukcie dát, odstránené do prislúchajúcich zhľukov identifikovaných v druhom kroku na základe vzdialeností medzi odstránenými bunkami a bunkami nachádzajúcimi sa v zhľukoch. Podľa hodnôt jednotlivých cytometrických znakov buniek v zhľukoch sa použitím modifikovaného Fruchtermam-Reingold algoritmu rozmiestnia vrcholy a hrany grafu minimálnej kostry grafu získaného v treťom kroku. Následne sa do grafu doplnia vizualizačné prvky, ako farba a veľkosť vrcholov, ktoré sú závislé od hodnôt cytometrických znakov a počtu buniek v zhľukoch [28].

Softvérový nástroj spade autori overovali na viacerých dátových množinách. Ako prvú dátovú množinu použili umelo vytvorené dáta, aby overili správnosť ich návrhu. V ďalšom kroku použili cytometrické dáta kostnej drene myši, v ktorých sú známe nachádzajúce sa bunkové populácie, ako aj ich hierarchia. Ako poslednú použili dátovú množinu cytometrických dát extrahovaných z kostnej drene odobratej od tridsiatich pacientov. V tejto dátovej množine sa im podarilo identifikovať doposiaľ známe bunkové populácie, ako aj nové, ktoré neboli identifikované použitím klasického manuálneho gatingu [28].

Na obrázku 2.6 je časť z výsledkov nástroja SPADE na dátovej množine cytometrických dát z kostnej drene myši. Z jednotlivých stromových vizualizácií výsledkov, ofarbených práve podľa hodnôt cytometrických znakov buniek v zhlukoch, kde červená farba reprezentuje vysoké hodnoty a modrá nízke, je vidno, že sa im podarilo identifikovať T bunky, ako aj ich podmnožiny  $CD4^+T$  a  $CD8^+T$  bunky a dendritické bunky, ktoré pri manuálnom gatingu neboli identifikované [28].



Obr. 2.6: Úkážka výsledkov algoritmu SPADE na dátovej množine cytometrických dát z kostnej drene myši [28]

## 2.4.2 FlowSOM

FlowSOM je zhlukovacia a vizualizačná metóda založená na použití samo-organizovaných máp (self-organizing maps - SOM).

SOM je špecifický typ umelej neurónovej siete na zhlukovanie bez učiteľa a redukciu dimenzionality. Skladá sa z mriežky uzlov, v ktorej každý uzol predstavuje bod vo viacrozmernom priestore. Pri zhlukovaní je nový bod klasifikovaný s uzlom, ktorý mu je najbližším susedom. Mriežka uzlov je natrénovaná tak, že uzly spojené krátkou cestou sú si podobnejšie ako uzly prepojené dlhou cestou. Takáto mriežka obsahuje topologické informácie a jeden trénovací bod môže ovplyvniť viacero [35].

FlowSOM používa omnoho väčší počet zhlukov, ako je očakávaný počet typov buniek na podporu vizualizácie a bunky, ktoré sú medzi typmi buniek majú svoje miesto v sieti, pričom to umožňuje spozorovať aj menšie zmeny v bunkových typoch. Pri zhlukovaní sa usiluje o čo najväčšiu čistotu každého zhluku. Maximálna čistota zhluku je dosiahnutá, ak zhluk obsahuje iba bunky s rovnakým typom. Algoritmus pozostáva zo štyroch hlavných krokov, a to načítanie dát, zostavenie samo-organizujúcej sa mapy, zostavenie minimálnej kostry stromu a meta-zhlukovanie [35].

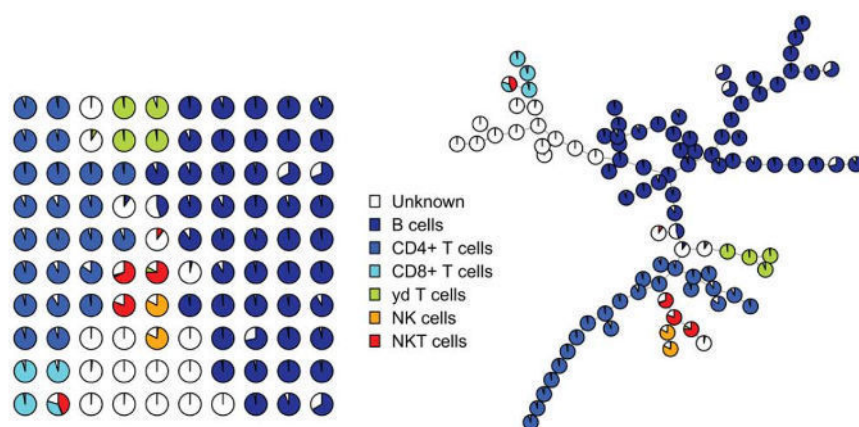
Pri načítavaní dát sú vzorky z viacerých súborov agregované do jednej dátovej množiny, nad ktorou sa následne trénuje model, čo umožňuje jednoduché porovnania vrámci celého experimentu. Kompenzácia a logická transformácia môžu byť vykonané pri načítavaní dát. Každý ukazovateľ dostane iniciálne hodnoty reprezentujúce rovnakú dôležitosť. Tieto hodnoty je možné

upraviť. Pri zostavovaní SOM je použitá Chebyshevová vzdialenosť ako funkcia susednosti v sieti / mriežke uzlov. Výsledok zhlukovania SOM je zobrazený minimálnou kostrou stromu. Meta-zhlukovanie slúži na získanie finálneho zhlukovania. Zhlukovacia metóda v meta-zhlukovaní je použitá hierarchické zhlukovanie typu *consensus*, ktorá je založená na opakovanom výbere podmnožiny a vykonanie hierarchického zhlukovania na každej podmnožine a výpočtu, ktoré body sú ako často v jednotlivých podmnožinách krokov zhlukované spolu. Počet zhlukov môže byť zadaný na základe vedomosti o očakávanom počte zhlukov alebo môže byť použité „elbow“ kritérium [35].

Pri elbow kritériu je vyskúšaných viacero hodnôt  $k$ , kde  $k$  je počet zhlukov a pre každé zhlukovanie je vypočítaná odchýlka v zhlukoch. Ak je počet zhlukov veľmi malý, odchýlka bude vysoká. Odchýlka bude rýchlo klesať, ak sa počet zhlukov zvýši. Ak je počet zhlukov správny, tak odchýlka bude relatívne nízka. Ak sa aj naďalej bude zvyšovať počet zhlukov, tak odchýlka bude aj naďalej klesať, ale podstatne pomalšie, pričom ďalšie pridané zhluky spôsobia len veľmi malý rozdiel. Cieľom tohto kritéria je detegovať bod, kde odchýlka prestane prudko klesať a bude klesať už iba mierne. Tento bod je možné nájsť položením dvoch lineárne regresívnych čiar na namerané odchýlky. Bod, v ktorom sa pretnú bude bod s minimálnou zbytkovou chybovosťou [35].

Na vizualizáciu sa okrem minimálnej kostry grafu použili aj kruhové diagramy, ktoré zobrazujú hodnoty ukazovateľov pre jednotlivé zhluky, ktoré je jednoduché medzi sebou porovnať.

Autori porovnali FlowSOM algoritmus s algoritmom SPADE, pričom dosiahli lepšie výsledky v čistote, zatiaľ čo výpočtový čas bol 10 až 50 krát rýchlejší [35]. V nedávnom porovnaní viacerých zhlukovacích algoritmov na cytometrických dátach, FlowSOM dosiahol najlepšie výsledky vo výpočtovom čase ako aj F1-skóre, precision a recall na väčšine použitých dátových množinách. [38]



Obr. 2.7: Výsledky algoritmu FlowSOM vizualizované kruhovými diagramami a minimálnou kostrou grafu [35]



## 2.5 Detekcia vývojovej trajektórie buniek

Konvenčné analýzy cytometrických dát sa zväčša orientujú na nájdenie dobre vymedzených bunkových populácií, ale jednobunkové dáta umožňujú modelovať aj prechody alebo zmeny medzi bunkovými stavmi. Bunky prechádzajú za svojej existencie viacerými stavmi. Zarovnaním buniek daného rodokmeňa na jednotnú trajektóriu umožňuje predikovať vývoj bunky. Takáto predikcia nie je možná s použitím klasických manuálnych analýz a preto bolo vyvinutých viacero metód na modelovanie bunkového procesu. [8, 29].

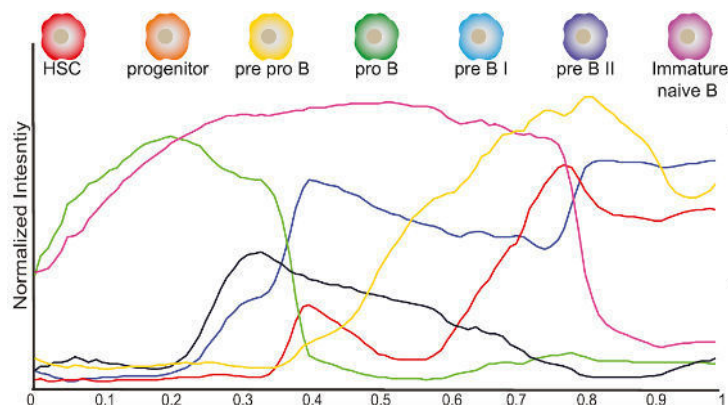
Hlavná myšlienka je modelovať procesy vývoja bunky sledovaním prechodov v dátach a tým odvodzovať trajektóriu bunkového vývoja použitím algoritmov bez učiteľa [29].

### 2.5.1 Wanderlust

Wanderlust je algoritmus hľadajúci trajektóriu vývoja buniek v cytometrických dátach. Využitie je pri predikcii vývoja buniek z počiatočného stavu do koncového [8].

Algoritmus sa skladá z dvoch krokov: inicializačný a krok iteratívnej detekcie trajektórie. V inicializačnom kroku sú náhodne uniformným rozdelením vybrané medzibody. Tieto body umožňujú presnejšiu kalkuláciu vzdialenosti medzi bodmi, keďže aj vďaka šumu v dátach sa stávajú vzdialenosti dvoch bodov nepresnejšie so zväčšujúcou sa vzdialenosťou. Preto je vzdialenosť bodu vypočítaná ako vzdialenosť ku všetkým týmto medzibodom, nie iba k jednému začiatočnému bodu. Váha vzdialenosti bodu od medzibodu je vypočítaná na základe vzdialenosti daného medzibodu k bodu. V druhom kroku sú dáta konvertované do grafu k-najbližších susedov, kde každá bunka je reprezentovaná ako vrchol spojený s hranami s k najpodobnejšími bunkami a ohodnotenie hrany zodpovedá vzdialenosti medzi dvoma bodmi. k-NNG je použitý ako predloha na vytvorenie l-k-NNG (l z k najbližších susedov) náhodným uniformným výberom l susedov z k-najbližších-susedov. Tento krok je použitý na mitigáciu krátkych cyklov v grafe, ktoré môžu narušiť správne zostrojenie trajektórie. Trajektória je následne vypočítavaná jednotlivo pre každý l-k-NNG graf a výsledná trajektória je priemerom výsledných trajektórií jednotlivých grafov. Dátový model je založený na podobnosti buniek a nie na vzdialenosti medzi parametrami, čo umožňuje lepšie spracovanie nelineárnych dát [8].





Obr. 2.8: Výsledná trajektória použitím algoritmu Wanderlust [8]

## 2.6 Predikcia klinického stavu

Existuje viacero metód na identifikáciu bunkových populácií z cytometrických dát, ako napríklad zhlukovacie algoritmy opísané v sekcii 2.4, pričom výstupom týchto algoritmov sú zväčša vizualizácie identifikovaných bunkových populácií. Tieto vizualizácie musia byť následne manuálne spracované expertom na získanie dodatočných informácií.

Metódy na koreláciu cytometrických dát s dostupnými klinickými dátami a klinickými výsledkami dokážu automatizovane predikovať možný klinický výsledok pre pacienta. Odbúrava sa tým potreba manuálneho kroku, pričom je možné identifikovať nové vzťahy vrámci klinických dát, cytometrických dát alebo medzi klinickými dátami a cytometrickými dátami, ktoré vplývajú na predispozíciu na chorobu, prípadne odlišnú odozvu alebo reakciu na liečbu.

Klinický stav pacienta môže byť príznak, či je daný pacient chorý alebo zdravý, prípadne o aký typ choroby ide alebo iný klinický príznak, ktorým sa rozdeľujú pacienti do dvoch alebo viacerých tried umožňujúcich predikciu zaradenia pacienta do daných tried. Z pravidla sa jedná o predikciu klinického stavu pacienta na základe pacientových cytometrických dát. Z nich sa extrahujú cytometrické črty, v ktorých sa následne, použitím vhodného modelu, hľadajú prediktívne črty umožňujúce vykonať čo najpresnejšiu predikciu.

Existuje veľa softvérových nástrojov a modelov na analýzu cytometrických dát. Výskumníci majú často problém vybrať z veľkého množstva dostupných prístupov ten správny pre ich výskum. Hlavným problémom veľkého množstva dostupných softvérových nástrojov a modelov je, že boli vyhodnocované a validované na odlišných dátových množinách a teda neexistuje porovnanie medzi jednotlivými prístupmi a preto je náročné, bez dodatočnej analýzy, vyhodnotiť, ktorý je najvhodnejší pre riešenie konkrétneho problému.

Na adresovanie tohto problému vzniklo zoskupenie FlowCAP <sup>2</sup>, ktoré sa zameriava na porovnanie jednotlivých softvérových nástrojov, modelov a prístupov. Porovnania sú vykonávané

<sup>2</sup>[www.flowcap.flowsite.org](http://www.flowcap.flowsite.org)

vo forme súťaží, ktorých sa môžu jednotlivé softvérové nástroje, modely a prístupy zúčastniť a následne sú vyhodnocované všetky na rovnakej dátovej množine. Jedna zo súťaží, konkrétne FlowCAP-II, bola zameraná práve na vyhodnotenie predikčnej sily softvérových nástrojov a modelov na predikciu klinického stavu pacienta [3].

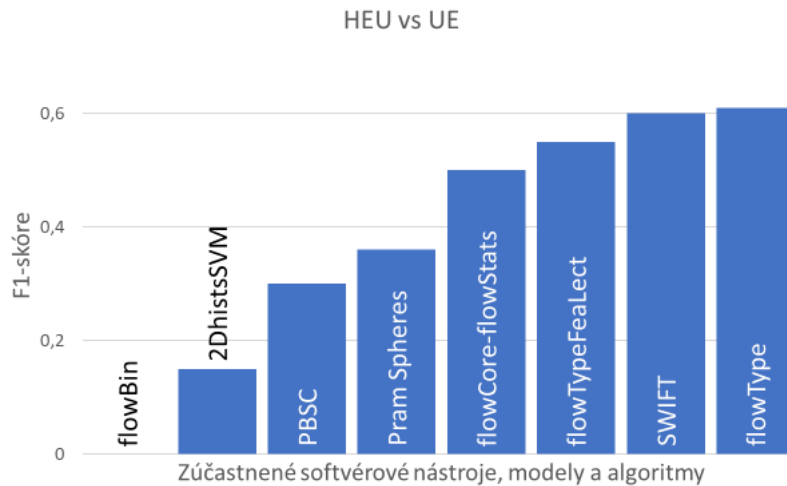
### **2.6.1 Súťaž FlowCAP-II**

Súťaž FlowCAP-II, pod záštitou zoskupenia FlowCAP, sa zamerala na porovnanie viacerých používaných softvérových nástrojov, modelov a algoritmov na predikciu klinického stavu pacienta na základe črt extrahovaných z cytometrických dát. Súťaže sa zúčastnilo 14 účastníkov, ktorí boli vyhodnocovaní použitím troch dátových množín, aj keď u väčšiny účastníkov bola vyhodnotená predikcia iba na základe dvoch z troch dátových množín [3].

V súťaži boli použité tri dátové množiny. Prvá dátová množina pozostávala z cytometrických dát afrických detí takých, ktoré boli vystavené vírusu HIV, ale neboli infikované (HEU) a takých, ktoré neboli vystavené vírusu HIV (UE). Cieľom bolo predikovať zaradenie do tried HEU a UE. Cieľom druhej dátovej množiny bolo identifikovať, či má pacient akútnu myeloidnú leukémiu alebo nie na základe cytometrických dát chorých a zdravých pacientov (AML). Posledná dátová množina sa zamerala na rozlíšenie dvoch antigén stimulujúcich skupín T buniek po vakcinácii HIV (HVTN). Polovica každej dátovej množiny bola poskytnutá jednotlivým účastníkom na tréning a druhá polovica sa použila na nezávislé testovanie [3].

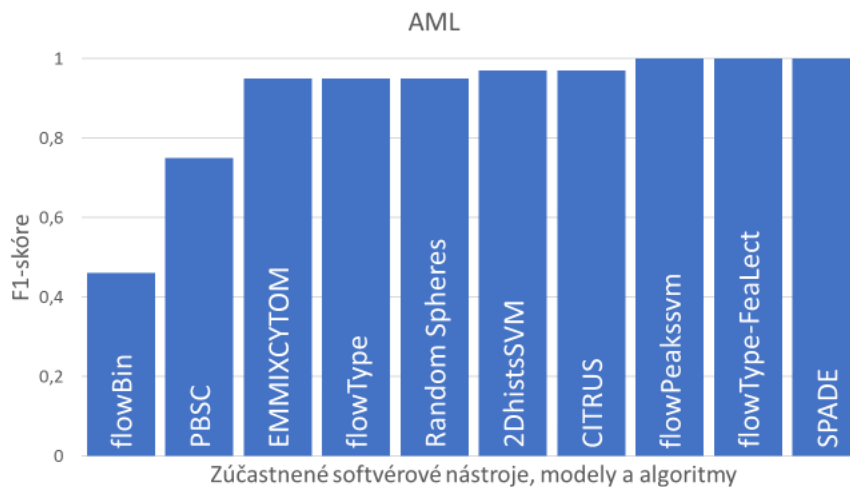
Následne boli zúčastnené softvérové nástroje, algoritmy a modely vyhodnotené použitím testovacích dát. Úspešnosť sa vyhodnocovala na základe viacerých metrík, ako presnosť, pokrytie a F1-skóre. Najväčší dôraz sa kládol práve na F1-skóre.

Na obrázku 2.9 sú vyobrazené hodnoty F1-skóre výsledkov účastníkov na testovacích dátach dátovej množiny HEU vs UE. Z dôvodu zlých výsledkov a biológie súvisiacej s touto dátovou množinou sa autori domnievajú, že rozpoznanie daných tried nebude možné iba na základe cytometrických dát [3]. Na dosiahnuté výsledky bude určite vplývať aj fakt, že veľkosť trénovacej a testovacej množiny bolo iba 48 vzoriek.

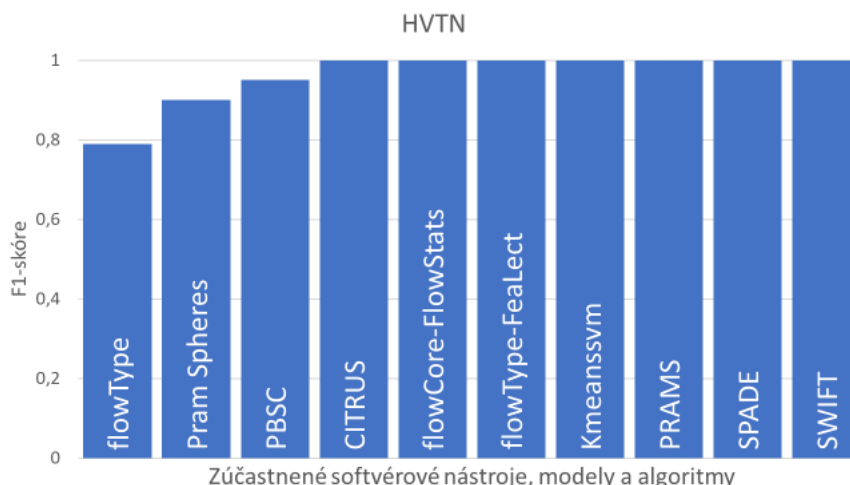


Obr. 2.9: *F1-skóre výsledkov zúčastnených softvérových nástrojov, algoritmov a modelov súťaže FlowCAP-II na dátovej množine HEU vs UE [3]*

Na testovacích dátach dátových množín AML an HVTN dosahovali účastníci podstatne lepšie výsledky, pričom niekoľko z nich dosiahlo bezchybné predikcie. Na grafoch 2.10 a 2.11 sú vyobrazené hodnoty F1-skóre výsledkov na zvyšných dvoch dátových množinách.



Obr. 2.10: *F1-skóre výsledkov zúčastnených softvérových nástrojov, algoritmov a modelov súťaže FlowCAP-II na dátovej množine AML [3]*



Obr. 2.11: *F1-skóre výsledkov zúčastnených softvérových nástrojov, algoritmov a modelov súťaže FlowCAP-II na dátovej množine HVTN [3]*

Z dosiahnutých výsledkov účastníkov súťaže FlowCAP-II vyplýva, že určenie klinického stavu pacienta na základe črt extrahovaných z cytometrických dát, je veľmi dobre predikovateľný problém.

## 2.6.2 Citrus

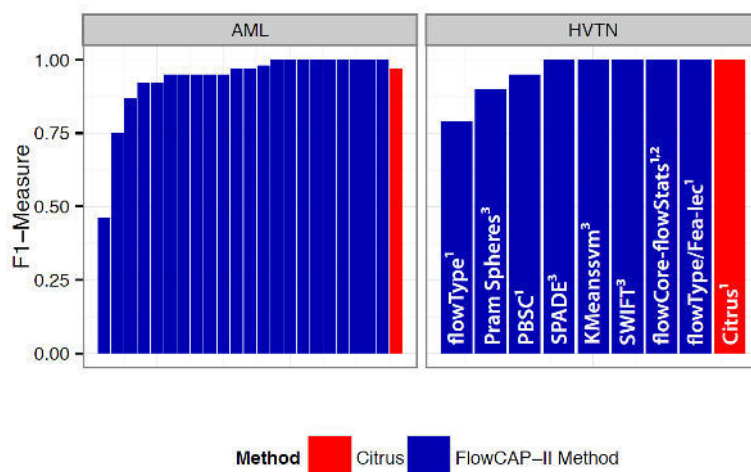
Citrus je automatická, dátovo-orientovaná metóda na identifikáciu rozvrstvujúcich bunkových populácií. Citrus spracováva cytometrické dáta z viacero vzoriek, ku ktorým sú priradené doplňujúce informácie ako dobrý alebo zlý pacientov stav, doba prežitia a pod. Bunky sú zhľukované na základe fenotypovej podobnosti a jednotlivé zhľuky sú charakterizované biologicky interpretovateľnými metrikami. Z týchto zhľukov je pomocou metódy učenia s učiteľom identifikovaná podmnožina zhľukov, ktorých charakteristika je predikovateľná ku doplňujúcim informáciám. Výstupom Citrus algoritmu sú grafy znázorňujúce fenotyp jednotlivých zhľukov a predikčný model, ktorý môže byť použitý na analýzu alebo validáciu nových dát [13].

Algoritmus vyberie z každej vstupnej vzorky dát určitý počet buniek a spojí ich do jednej množiny dát. Počet buniek, ktoré sa vyberajú z každej vzorky zadáva používateľ. Zhľuky buniek z agregovaných dát sú identifikované použitím hierarchického zhľukovania na základe podobnosti ukazovateľov. Citrus vychádza z predpokladu, že fyziologicky alebo klinicky relevantné bunkové populácie, ktoré sú reprezentatívne pre určitý fenotyp budú robustne sa opakujúce javy v dátach. Do ďalších krokov spracovania sú vybrané iba zhľuky, ktoré obsahujú minimálne 5% všetkých nameraných javov - buniek. Následne sú zhľuky rozdelené späť na prvky podľa jednotlivých vzoriek, z ktorých pochádzajú a sú vypočítané popisujúce vlastnosti pre každý zhľuk na báze jednotlivých vzoriek. Na identifikáciu zhľukov rozvrstvujúcich buniek je použitý regularizovaný učiaci algoritmus s učiteľom. Predikcia skupiny pacienta, napríklad zdravý a chorý, je vykonaná natrénovaním a použitím regularizovaného klasifikačného modelu. Podľa

definície, regularizovaný model automaticky identifikuje podmnožinu zhukovavých vlastností, ktoré najlepšie predikujú skupiny pacientov, čím odhaľuje zhluky buniek s rozvrstvujúcim správaním v dátovej množine. Citrus konštruuje klasifikačný model použitím lasso-regularizovanou logistickou regresiou a metódou *nearest shrunken centroid* [13].

S použitím teoretických informácií by mohol byť pridaný krok algoritmu, ktorý by inteligentne preriedil populácie tak, že by odstránil tie, ktoré obsahujú redundantné informácie pred predikciou klinického výsledku, čo by mohlo predikciu spresniť. Existujú regresné modely, ktoré explicitne počítajú koreláciou parametrov a použitím takých modelov by sa eliminoval krok zlučovania podobných populácií po regresii. Keďže Citrus identifikuje rozvrstvujúci sa signál na základe viacerých vzoriek, nie je vhodný na jeho identifikáciu v dátach s malým počtom vzoriek - pacientmi - a preto nedokáže vyhodnotiť ani rozdiely medzi dvoma pacientmi. Citrus algoritmus vyberá zhluky, ktoré obsahujú aspoň minimálny percentuálny podiel buniek k celkovému počtu buniek a nespracováva zhluky s menším počtom. Za následok to má neschopnosť identifikovať vzácne, zriedkavé populácie buniek, ktoré môžu mať taktiež vplyv na klinický výsledok [13].

Autori overovali prediktívnu schopnosť algoritmu Citrus na dátových množinách AML a HVTN zo súťaže FlowCAP-II (viď. 2.6.1). Grafy na obrázku 2.12 znázorňujú porovnanie prediktívnej schopnosti algoritmu Citrus s ostatnými zúčastnenými algoritmami súťaže. Porovnanie je vykonané použitím metriky F1-skóre. Algoritmu Citrus sa podarilo na dátovej množine HVTN dosiahnuť bezchybnú predikciu a na dátovej množine AML sa algoritmus pri predikcii pomýlil iba raz.



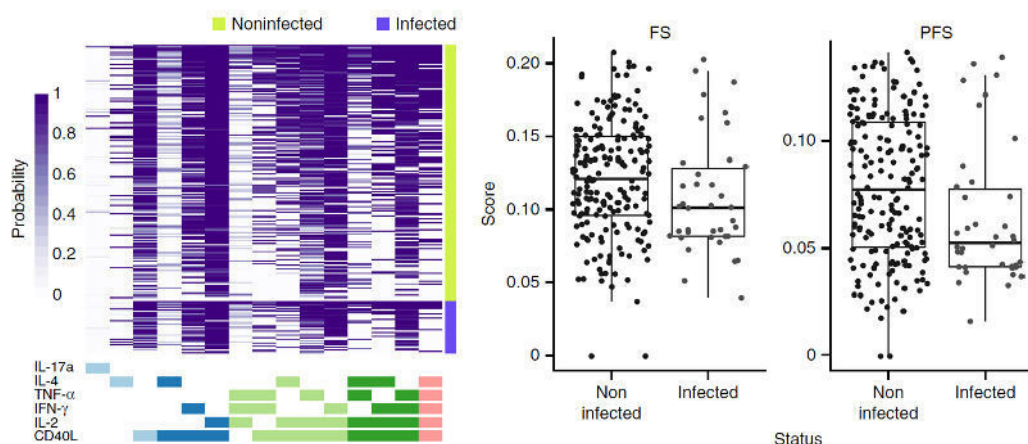
Obr. 2.12: Výsledky predikcie algoritmu Citrus na dvoch dátových množinách zo súťaže FlowCAP-II

### 2.6.3 COMPASS

COMPASS je rámec slúžiaci na výpočet korelácie medzi klinickými dátami a dátami o odozvách na úrovni pacienta a na úrovni bunkových populácií [24].

COMPASS používa Bayesov zhlukovací zmiešaný model na identifikovanie zmien naprieč všetkými bunkami simultánne, čo umožňuje definovanie opisov, ktoré môžu byť sumarizované a korelované s klinickými výsledkami. Táto metóda používa výpočtovo efektívny Markovov Chain Monte Carlo algoritmus na vyhľadanie všetkých podmnožín a vypočítanie ich pravdepodobností [24].

Vo výsledku COMPASS poskytuje skóre pre každú vzorku, ktoré môžu byť následne korelované s klinickými výsledkami [24].



Obr. 2.13: Výsledky algoritmu COMPASS [24]

## 2.6.4 Náhodný les

Metódy učenie súborom modelov využívajú viacero klasifikátorov a agregujú ich výsledky. Známe a rozšírené metódy sú napríklad *boosting* a *bagging* klasifikačných stromov. Pri metóde *boosting* sa upravujú váhy bodom, ktoré boli chybné predikované a predikcia sa vyhodnocuje na základe váhovaného hlasovania klasifikačných stromov. Metóda *bagging* zostavuje stromy na podmnožine celkových dát a predikcia je vyhodnotená na základe väčšinového hlasu [23].

Náhodný les [11] je metóda učenia súborom modelov na klasifikáciu a regresiu využívajúca metódu *bagging* a náhodný výber prediktorov. Využitím metódy *bagging* je každý strom v náhodnom lese zostavený použitím podmnožiny celkovej vstupnej množiny dát. Každý strom je samostatný klasifikátor, ktorý priradzuje triedu dátovému bodu bez triedy. Finálna klasifikácia je vyhodnotená väčšinovým hlasom samostatných klasifikátorov. Každý dátový bod má štatisticky 35% pravdepodobnosť, že sa bude nachádzať medzi „*out-of-bag*” (OOB) dátovými bodmi, teda medzi bodmi, ktoré sa nenachádzajú vo zvolenej podmnožine dát na zostavenie stromu. Tieto OOB dátové body sa používajú na výpočet OOB chybovosti [16, 23].

V klasických stromoch sa v každom vrchole stromu určuje rozdelenie, na základe ktorého sa vykonáva rozhodnutie, ako najlepšie rozdelenie spomedzi všetkých prediktorov. V náhodnom lese sa pre každý vrchol určuje najlepšie rozdelenie iba na náhodne zvolenej podmnožine

všetkých prediktorov. Tento, na prvý pohľad neintuitívny, prístup dosahuje veľmi dobré výsledky v porovnaní s inými klasifikátormi [23]. Ďalšie z výhod náhodného lesa je robustnosť voči okrajovým dátam, pretrénovaniu a interpretovateľnosť výsledkov vďaka výpočtu dôležitosti prediktorov [11, 16].

Náhodný les závisí od viacerých parametrov, ale hlavne od dvoch – počet stromov ( $n_{tree}$ ) a počet náhodne vybraných prediktorov pre vrcholy stromov ( $m_{try}$ ). Postup algoritmu by sa dal opísať troma krokmi a to:

- extrahuj  $n_{tree}$  podmnožín dát zo vstupnej množiny
- pre každú podmnožinu zostav klasifikačný alebo regresný strom s použitím  $m_{try}$  náhodných prediktorov v každom vrchole stromu na určenie rozdelenia
- vykonaj predikciu alebo regresiu na nových dátach agregáciou výsledkov  $n_{tree}$  stromov (väčšinový hlas pre klasifikáciu a priemer pre regresiu) [23].

Výpočet dôležitosti prediktorov umožňuje identifikovanie dôležitých a prediktívnych prediktorov, čo uľahčuje interpretovateľnosť výsledkov, ako aj umožňuje zostavenie jednoduchšieho modelu s rovnakou alebo veľmi podobnou prediktívnou silou. Dôležitosť prediktorov sa zväčša počíta ako zmena v presnosti predikcie permutovaním konkrétneho prediktora v OOB dátach a zachovania poradia dát pre ostatné prediktory [23]. Použitie vysoko korelovaných prediktorov sa neodrazí na výslednej presnosti predikcie, avšak môže spôsobiť nesprávne identifikovanie dôležitých prediktorov, keď sa strom nevie rozhodnúť medzi dvoma alebo viacerými prediktormi, na základe ktorých sa rozhodovať a priradiť im viac-menej rovnakú a celkovo nižšiu dôležitosť [17]. Preto sa odporúča odstrániť vysoko korelované prediktory pred interpretáciou výsledkov náhodného lesa a extrahovaní záverov z nich.

### 2.6.5 Sieť Elastic

Elastic sieť je metóda regularizácie a výberu črt, ktorá zoskupuje metódy *Ridge* a *Lasso* regresie a dosahuje lepšie výsledky ako pri použití týchto metód samostatne, keďže dokáže využiť klady a minimalizovať zápory oboch prístupov [40].

Ridge a Lasso regresie sú regularizačné metódy využívajúce sa na tvorbu menej komplexných modelov, ak platí  $p \gg n$ , kde  $p$  je počet črt a  $n$  je počet záznamov v dátovej množine. Menej komplexný model sa dosiahne zredukovaním počtu črt, na základe ktorých sa musí model rozhodovať. Obe metódy priradujú penalizáciu koeficientom jednotlivých črt. Veľký koeficient ľubovoľnej črty hovorí o tom, že daná črta je výrazne prediktívna pre výsledok. Ak sa nepenalizujú koeficienty, tak dosiahnutím príliš veľkej hodnoty koeficienta sa bude model snažiť naučiť aj odchýlky v dátach pre danú črtu, čo vo výsledku spôsobí pretrénovanie modelu na tréningových dátach [40, 20].

Ridge regresia vykonáva L2 regularizáciu, zatiaľ čo Lasso regresia vykonáva L1 regularizáciu. Ridge aj Lasso regresia sú závislé od parametru  $\alpha \in [0, \infty]$ , ktorý udáva veľkosť penalizácie

priradenej všetkým črtám. Zvyšujúcou hodnotou  $\alpha$  sa stáva model menej komplexný a znižuje sa pretrénovanie modelu, avšak príliš veľká hodnota parametru  $\alpha$  môže spôsobiť podtrénovanie modelu [40, 20].

Cieľ oboch prístupov je rovnaký, ale ich výsledok sa mení. Ridge regresia koeficienty črt, ktoré nie sú prediktívne pre výsledok znižuje a tým redukuje komplexnosť modelu, naopak Lasso regresia koeficienty týchto črt nastaví na 0, čím okrem redukcie komplexnosti vykoná aj výber prediktívnych črt. Lasso regresia je vhodnejšia pre modely s dátovými množinami, ktoré obsahujú veľké množstvo črt (v miliónoch). V prípade korelovaných črt Ridge regresia zahrnie všetky z nich do výsledného modelu a hodnoty koeficientov jednotlivých korelovaných črt budú na základe daných korelácií. Naopak Lasso regresia vyberie iba jednu črtu z korelovaných črt a ostatným nastaví koeficient na hodnotu nula [40, 20].

## 2.7 Redukcia dát

V cytometrických dátach sa nachádzajú hojne vyskytujúce sa bunkové populácie, ktoré sú reprezentované veľkým počtom buniek v dátach a vzácné bunkové populácie, ktoré sú reprezentované oveľa menším počtom buniek [21]. Ak budeme takéto dáta spracovávať zhlukovacím algoritmom, vzácné bunkové populácie nebudú vo výsledku zohľadnené práve preto, že ich počet voči ostatným bunkovým populáciám je veľmi malý a stratí sa tým informácia o výskyte týchto populácií v cytometrických dátach pacienta.

Autori nástroja SPADE (viď. 2.4.1) prišli s riešením urobiť redukciu dát na základe hustoty buniek v dátach, aby vyrovnali početnosť reprezentatívnych buniek pre bunkové populácie.

### 2.7.1 Redukcia dát na základe hustoty

Na dáta sa dá pozeráť ako na vysoko-rozmerný oblak bodov, kde každý bod v oblaku je jedna bunka a dimenzionalita oblaku je počet ukazovateľov pre bunku. V takomto oblaku husté oblasti zodpovedajú hojným populáciám a oblasti s nízkou hustotou predstavujú vzácné bunkové populácie, prípadne bunky, v prechode medzi hojnými bunkovými populáciami [28].

Lokálna hustota ( $LD_i$ ) bunky  $i$  je definovaná, ako počet buniek v jej  $\varepsilon$ -okolí. To sú bunky, ktoré majú vzdialenosť od bunky  $i$  menšiu alebo rovnú hodnote  $\varepsilon$ . Veľkosť  $\varepsilon$ -okolía je vybraná tak, aby každá bunka mala aspoň jednu bunku vo svojom okolí. Následne na základe cieľovej hustoty ( $TD$ ), teda výslednej hustoty dát a okrajovej hustoty ( $OD$ ), teda hustoty, ktorú nadobúdajú okrajové dáta, sa pre každú bunku  $i$  vypočíta pravdepodobnosť jej ponechania ako:

$$P(i) = \begin{cases} 0, & LD_i \leq OD \\ 1, & OD < LD_i \leq TD \\ \frac{TD}{LD_i}, & LD_i > TD \end{cases}$$



Tým sa odstránia bunky, ktorých lokálna hustota je menšia ako okrajová hustota, a teda sú považované za okrajové hodnoty. Bunky, ktorých lokálna hustota je v rozmedzí medzi  $(OD, TD]$  nie sú redukované. Naopak, bunky v hustých oblastiach sú na základe pravdepodobnosti vysoko redukované tak, aby ich lokálna hustota klesla na cieľovú hustotu [28].

Takéto stochastické riešenie spôsobuje nereprodukovateľnosť výsledkov, keďže kvôli náhodnému výberu buniek na základe pravdepodobnosti sa budú výsledky analýz nad rovnakými dátami líšiť. Toto sa dá vyriešiť použitím deterministického prístupu redukovania na základe hustoty, ktoré použili autori v novej verzii nástroja SPADE [27]. Stochasticita predošlého riešenia spočíva hlavne v náhodnom výbere reprezentatívnych buniek, teda buniek, ktoré sa ponechajú. Tento krok sa nahradil deterministickým krokom, kde sa ako najreprezentatívnejšie bunky vyberajú tie s najväčšou lokálnou hustotou.

Pre každú bunku  $i$  sa vypočíta lokálna hustota ( $LD_i$ ) rovnako ako pri stochastickom riešení. Podľa hodnôt lokálnych hustôt a používateľom definovaných hodnôt okrajovej hustoty ( $OD$ ) a výslednému počtu buniek ( $T_N$ ) vieme pre každú bunku  $i$  vypočítať jej váhu nasledovne:

$$w_i = \begin{cases} 0, & LD_i \leq OD \\ 1, & OD < LD_i \leq TD \\ \frac{TD}{LD_i}, & LD_i > TD \end{cases}$$

Kde  $TD$  je zvolené tak, aby platilo  $\sum_{i=1}^N w_i = T_N$ , a teda ak sa majú dáta zredukovať na  $T_N$  počet buniek, váhy popisujú, ako veľmi má daná bunka z pôvodných dát prispievať do výsledných redukovaných dát. Podobne ako pri stochastickom prístupe, bunky, pre ktoré platí  $LD_i \leq OD$  sú považované za okrajové hodnoty a preto nie sú vybraté do výsledných redukovaných dát. Tak isto všetky bunky, ktorých lokálna hustota je v intervale  $(OD, TD]$  sú vybrané do výsledných dát. Bunky s vysokou lokálnou hustotou sú počítané ako zlomok inverzne úmerný ku lokálnej hustote. Cieľová hustota je zvolená tak, že suma váh sa rovná požadovanému výslednému počtu buniek a je vypočítaná lineárnym prehľadávaním a teda časová náročnosť výpočtu  $TD$  je  $O(n)$  kde  $n$  je počet buniek pôvodných dátach [27].

Po vypočítaní váh sa iteratívnym procesom vykoná redukcia dát. V prvej iterácii sú všetky bunky inicializované ako dostupné. Bunka s najväčšou lokálnou hustotou je zvolená ako vybratá bunka. Všetky ostatné bunky sú usporiadané podľa ich vzdialenosti k zvolenej bunke a vypočíta sa bežiaci kumulatívny súčet váh. Od vrchných buniek až po kumulatívny súčet rovný 1 sú bunky označené ako nedostupné. V ďalších iteráciách je zvolená dostupná bunka s najväčšou lokálnou hustotou ako vybratá. Bunky sú opäť usporiadané podľa vzdialeností ku zvolenej bunke na výpočet bežiaceho kumulatívneho súčtu. Vrchné bunky až po bežiaci kumulatívny súčet rovný 1 alebo kým sa nenarazí na nedostupnú bunku sú označované za nedostupné. Proces končí, keď sú všetky bunky označené ako nedostupné [27].

Tento algoritmus je inšpirovaný stochastickým prístupom nástroja SPADE [28] a takzvaným verným redukovaním dát. To je algoritmus, ktorý iteratívne zo všetkých dát vyberie náhodne

dostupný bod ako reprezentanta a všetkých jeho susedov označí za nedostupných. Algoritmus končí, keď sú všetky body označené ako nedostupné [39].

Oba tieto prístupy po výbere reprezentanta vytvoria v oblaku bodov diery o veľkosti jeho okolia, z ktorého už následne nemôže byť zvolený ďalší reprezentant, pričom tieto diery sa nemôžu pretínať.

### 2.7.2 Problémy redukcie dát na základe hustoty

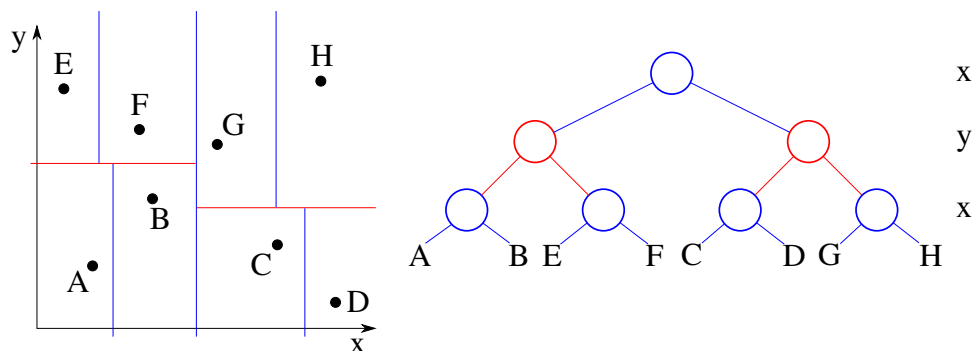
Vyššie opísané riešenie na redukciiu dát na základe hustoty má niekoľko problémov. Na vykonanie redukcie potrebujeme mať vypočítanú hustotu pre každú bunku. Podľa voľne dostupnej implementácie nástroja SPADE je pre vypočítanie hustoty jednej bunky zistená jej vzdialenosť ku každej ďalšej bunke. Aby sme teda vypočítali hustotu každej bunky, dostávame časovú zložitosť  $O(n^2)$  kde  $n$  je počet buniek. Takéto riešenie je nepoužiteľné pri veľkej hodnote  $n$ .

Samotné redukovanie dát pri stochastickom prístupe je rýchle, kde po vypočítaní pravdepodobnosti ponechanie bunky pre každú bunku v čase  $O(n)$  vieme opäť v čase  $O(n)$  na základe pravdepodobností pre bunky vykonať redukciiu. Kvôli použitiu pravdepodobností sú výsledky nereprodukovateľné a rozdielne pre rovnaké dáta a preto nepoužiteľné. Tento problém rieši deterministický prístup redukcie na základe hustoty, kde sa bunky redukujú na základe im priradených váh vypočítaných na základe ich lokálnych hustôt. Je to iteratívny prístup, kde sa v každej iterácii vyberie jedna reprezentatívna bunka a odstránia sa ostatné bunky v jej okolí na základe ich vzdialeností a bežiaceho kumulatívneho súčtu. Na tento krok musia byť bunky zoradené podľa vzdialeností, kde operácia zoradenia je časovo náročná hlavne pri veľkom počte dát. Keďže sa táto operácia zoradenia vykonáva v každej iterácii, jednotlivé iterácie budú časovo náročné a algoritmus bude konvergovať veľmi pomaly, čo ho robí nepoužiteľným na veľké dáta.

Oba prístupy potrebujú používateľom zadané hodnoty parametrov okrajová hustota ( $OD$ ) a cieľová hustota ( $TD$ ) resp. výsledný počet buniek ( $T_N$ ). Na získanie správnych hodnôt týchto parametrov je potrebná apriori analýza dát, ktorá do výsledkov vnáša nepresnosti, subjektivitu a tým pádom aj ťažšiu reprodukovateľnosť výsledkov a zvyšuje čas potrebný na vykonanie celej analýzy.

V deterministickom prístupe pri počítaní váh buniek vidno, že  $TD$  závisí od  $T_N$ , od ktorého závisí  $w_i$  a zároveň  $w_i$  závisí od  $TD$ , z čoho vyplýva, že hodnoty parametrov  $TD$  a  $w_i$  sa nedajú vypočítať naraz jedným lineárnym vyhľadávaním v dátach, ale treba dve prehľadania.

Na určenie veľkosti  $\varepsilon$ -okolia takého, aby každá bunka mala aspoň jedného suseda sa používa heuristický prístup, kedy sa pre  $m$  náhodne vybraných buniek, kde  $m < n$ , vypočíta ich vzdialenosť ku všetkým ostatným bunkám. Pre každú túto bunku sa následne zoberie hodnota vzdialenosti ku jej najbližšiemu susedovi a z tejto množiny sa zvolí medián a päťnásobok tejto hodnoty sa použije ako hodnota parametra  $\varepsilon$ . Keďže sa bunky vyberajú náhodne, tak hodnota  $\varepsilon$  sa môže medzi behmi líšiť, čo zhoršuje reprodukovateľnosť výsledkov kvôli zavedeniu ďalšej náhodnosti do celého procesu.



Obr. 2.14: Ukážka kd-stromu.

## 2.8 Dátové štruktúry

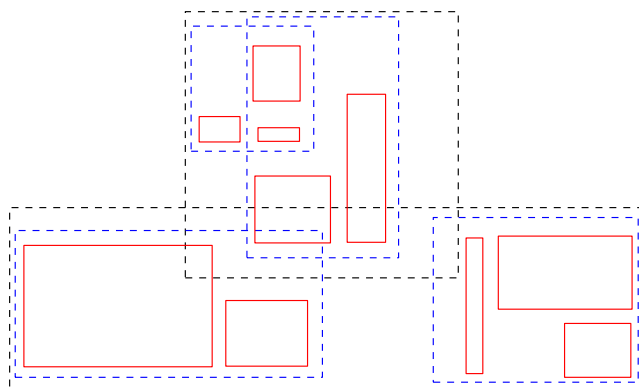
Softvérové nástroje a algoritmy využívajú mnohé dátové štruktúry na lepšiu reprezentáciu a prácu s dátami. Často používané sú práve stromové dátové štruktúry, poskytujúce rýchle stavenie a vyhľadávanie v strome. Stromové štruktúry sa okrem indexovania dát používajú aj na delenie priestoru, kde zväčša úrovně stromu reprezentujú časť priestoru, prípadne jednu dimenziu a listy obsahujú jednotlivé dátové body alebo zoskupujú podmnožinu bodov. Takéto delenie priestorov umožňuje rýchly prístup k dátam a inteligentné rozdelenie a zoskupenie dát.

### 2.8.1 kd-strom

Kd-strom [10] je dátová štruktúra využívajúca sa na rozdeľovanie a rýchle vyhľadávanie v  $k$ -rozmernom priestore. Jedná sa o binárny strom, ktorý uchováva  $k$ -rozmerné dáta a na každej úrovni stromu, ktorá reprezentuje dimenzia priestoru, rozdeľuje dáta na dve časti. Avšak nárôzdiel od binárneho stromu používa kd-strom na každej úrovni iný kľúč, podľa ktorého sa rozdeľujú dáta. Ak je výška stromu väčšia ako  $k$ , tak sa jednotlivé kľúče pre úrovně stromu opakujú [12].

V kd-strome môže byť použitá ľubovoľná hodnota ako kľúč, podľa ktorého sa dáta v dimenzii rozdeľujú. Ak však chceme dosiahnuť vyvážený kd-strom, je potrebné rozdeľovať na každej úrovni stromu podľa mediánovej hodnoty dimenzie, ktorú daná úroveň stromu reprezentuje [12, 26]. Vypočítanie mediánu môže byť časovo náročná operácia, ktorá spomalí stavenie vyváženého kd-stromu nad veľkými dátovými množinami. Quicksort algoritmus dokáže v najlepšom prípade nájsť medián hodnotu v čase  $O(n)$ , ale v najhoršom prípade v čase  $O(n^2)$  a algoritmy merge sort a heap sort v čase  $O(n \log n)$  [12].

Na obrázku 2.14 sú zobrazené body v dvojrozmernom priestore a kd-strom, ktorý vznikne zostavením z týchto bodov. Farebne sú odlíšené jednotlivé rozdelenia priestoru, ku ktorým farebne korešponduje aj úroveň stromu. Na pravo, pri každej úrovni stromu je naznačené, ktorú dimenziu daná úroveň v strome reprezentuje.



Obr. 2.15: Ukážka *r-stromu*.

### 2.8.2 *r-strom*

R-strom [18] je stromová dátová štruktúra na indexovanie viacrozmerných dát, nad ktorými je možné následne vyhľadávať.

Hlavná myšlienka *r-stromu* je zoskupovanie objektov a ich reprezentácia na základe minimálnych ohraničujúcich skriniek. Každou úrovňou stromu sa zoskupenia a teda aj minimálne ohraničujúce skrinky zmenšujú a listy stromu reprezentujú jednotlivé dátové objekty. Využitím minimálnych ohraničených skriniek sa vyhľadávanie v strome urýchľuje, lebo ak minimálna ohraničená skrinka nespĺňa podmienky vyhľadávania, tak vieme, že ani objekty, ktoré sa v nej nachádzajú nebudú spĺňať podmienky vyhľadávania [18].

Efektívne zostavenie *r-stromu*, ktorý je vyvážený a zároveň sa minimálne ohraničujúce skrinky veľmi nepretínajú je problémové. Existuje viacero variácií *r-stromu*, napríklad *r\*-strom* [7], ktoré sa zameriava na minimalizovanie prekrytia minimálne ohraničujúcich skriniek, čo má za dôsledok zlepšenie vyhľadávania, avšak za cenu rýchlosti stavby stromu [7].

Na obrázku 2.15 je zobrazený príklad rozdelenia priestoru dátovou štruktúrou *r-strom*. Červené útvary reprezentujú objekty v priestore a čierne a modré skrinky znázorňujú úrovne minimálne ohraničujúcich skriniek.

## 2.9 Zhrnutie

Cytometria je veda zaoberajúca sa meraním a skúmaním charakteristík buniek. Bunky sú charakteristické podľa hodnôt cytometrických znakov. Tieto charakteristiky buniek predstavujú viacrozmerné a mnohopočetné cytometrické dáta, ktoré z pravidla obsahujú 18 až 40 znakov a stotisíce záznamov.

Tradičné spracovanie manuálnym gatingom je veľmi neefektívny, subjektívny a neškálovateľný prístup analýzy cytometrických dát. Na adresovanie týchto problémov bolo vytvorených viacero metód, ktoré zrýchľujú a automatizujú niektoré, prípadne všetky kroky analýzy cytometrických dát.

Jedným z prvých a veľmi dôležitým krokom v analýze cytometrických dát je identifikácia bunkových populácií. Na identifikovanie bunkových populácií sa používajú rôzne zhukovacie algoritmy, ako napríklad hierarchické zhukovanie, zhukovanie na základe hustoty alebo modelu, samo-organizujúce mapy a ďalšie. V cytometrických dátach sa nachádzajú redundantné, hojné bunkové populácie, ktoré sú v dátach zastúpené veľkým počtom buniek a vzácne bunkové populácie, ktoré sú v dátach zastúpené malým počtom buniek. Na cytometrické dáta sa dá pozeráť ako na viacrozmerný oblak bodov, v ktorom hojné bunkové populácie budú predstavovať husté oblasti a naopak vzácne bunkové populácie budú reprezentované riedkymi oblasťami. Použitím zhukovacieho algoritmu na takéto dáta nebude možné identifikovať vzácne bunkové populácie, práve z dôvodu veľkých rozdielov v hustote. Preto sa používa prístup redukcie dát na základe hustoty, ktorého cieľom je vyrovnať hustotu v priestore a vo výsledku reprezentovať vzácne a hojné bunkové populácie približne rovnakým počtom bodov v priestore, čo spresní identifikáciu vzácných bunkových populácií zhukovacím algoritmom.

Ďalším dôležitým krokom v analýze cytometrických dát je vizualizácia bunkových populácií a výsledkov analýzy. Na vizualizáciu cytometrických dát vzniklo viacero metód, ktoré by sa dali zaradiť do dvoch skupín (i) využívajúce redukciu dimenzionality; (ii) využívajúce zhukovacie algoritmy a následne vizualizovanie výsledkov zhukovania. Prístupy založené na redukcii dimenzionality sa pokúšajú zachytiť lineárne a nelineárne vzťahy v dátach a zobraziť ich v dvojrozmernom priestore. Metódy z kategórie využívajúcich zhukovacie algoritmy vykonajú na dátach najskôr proces zhukovania a až následne vizualizujú výsledky zhukovania. Pri oboch prístupoch sa strácajú určité informácie.

S dostupnými klinickými údajmi o pacientovi, od ktorého pochádzajú cytometrické dáta, je možné vykonávať predikciu klinického stavu pacienta. Klinický stav môže reprezentovať, či je pacient chorý alebo zdravý, prípadne typ choroby, podstupenú liečbu alebo odozvu na liečbu. Predikcia klinického stavu sa vykonáva na základe číť extrahovaných z cytometrických dát. Na predikciu a korelácie medzi cytometrickými dátami a klinickým stavom pacientom bolo vyvinutých viacero metód, ktoré používajú rôzne predikčné modely. Úspešnosť metód bola vyhodnotená súťažou FlowCAP-II použitím troch rôznych dátových množín. Na dvoch z troch použitých dátových množín viacero zúčastnených softvérových nástrojov, modelov a algoritmov dosiahlo bezchybné predikcie, z čoho vyplýva, že predikcia klinického stavu pacienta, na základe cytometrických číť je veľmi dobre predikovateľný problém.



# Kapitola 3

## Návrh riešenia

### 3.1 Špecifikácia

Cieľom nášho výskumu je navrhnuť a implementovať plne deterministický algoritmus redukcie dát na základe hustoty a navrhnuť model na predikciu klinického stavu pacienta.

Mnohé dostupné softvérové nástroje a algoritmy na analýzu cytometrických dát sú závislé od veľkého množstva vstupných parametrov, ktoré vyžadujú apriori analýzu dát na získanie vhodných hodnôt parametrov, čo spomaľuje celý proces analýzy, ako aj zanáša subjektivitu do výsledkov a preto je vhodné sa zamerať na minimalizovanie závislostí od vstupných parametrov.

Bunkové populácie v cytometrických dátach môžu byť redundantné, teda zastúpené veľkým množstvom buniek alebo vzácne, ktoré sú oproti redundantným bunkovým populáciám zastúpené v dátach omnoho menším počtom buniek. Na identifikovanie týchto bunkových populácií sa používajú zhlučovací algoritmy, avšak bez predspracovania dát môže dôjsť k nesprávnemu identifikovaniu, prípadne neidentifikovaniu vzácných bunkových populácií, práve z dôvodu veľkých rozdielov v reprezentáciách vzácných a redundantných bunkových populácií. Na vyriešenie tohto problému je potrebné vykonať redukciu cytometrických dát na základe hustoty, ktorá vyrovná hustotu v rámci priestoru tak, aby boli vzácne aj redundantné bunkové populácie vo výsledku reprezentované rovnomerne, kde každá bunka je reprezentovaná bodom v priestore s umiestnením vyplývajúcim z jej hodnôt. Tento krok umožní správne identifikovanie, ako redundantných, tak aj vzácných bunkových populácií v rámci vzorky a navyše urýchli celý proces zhlučovania použitím menšej dátovej množiny.

Na vykonanie redukcie dát na základe hustoty je potrebné poznať hustotu v rámci priestoru a teda hustotu jednotlivých buniek, ktoré sa v priestore nachádzajú. Na výpočet hustoty konkrétnej bunky je potrebné poznať počet buniek v jej okolí a teda vzdialenosti ku ostatným bunkám. Naivný výpočet vzdialeností jednotlivých buniek má časovú náročnosť  $O(n^2)$ , respektívne  $O(\frac{n^2-n}{2})$ , ak by sa vzdialenosť medzi dvoma odlišnými bunkami počítala práve raz. Takýto prístup je neefektívny a nepoužiteľný na veľké dátové množiny a preto je nevyhnutné navrhnuť efektívnejší spôsob. Viacero existujúcich prístupov, ako napríklad softvérový nástroj

SPADE [28], využívajú stochastické prístupy aby adresovali použiteľnosť na veľkých dátových množinách, čo znemožňuje reprodukovateľnosť výsledkov z dôvodu odlišných výsledkov pri viacerých výpočtoch použitím rovnakej dátovej množiny. Z tohto dôvodu sa zameriame na navrhnutie a implementáciu plne deterministického riešenie výpočtu hustoty vrámci priestoru, ktoré bude zároveň použiteľné aj na veľké dátové množiny.

Predikcia klinického stavu sa vykonáva na základe extrahovaných črt z cytometrických dát a pridaním črt extrahovaných z klinických dát je možné vyvodiť nové poznatky. Črty z cytometrických dát sa vytvárajú na základe identifikovaných bunkových populácií v dátach. Aby boli identifikované aj vzácne bunkové populácie, tak je potrebné na zhlukovanie použiť zredukované dáta na základe hustoty. Na extrahovanie črt je potrebné mať k dispozícii dáta všetkých buniek z bunkových populácií a preto je nevyhnutné pred extrakciou črt vykonať proces, ktorý bunky odstránené v procese redukcie zaradi do príslušných bunkových populácií na základe zhlukovaných dát.

Z podstaty tvorby cytometrických črt vzniká pre každú vzorku pomerne veľký počet črt [13, 36], kde pri  $n$  vzorkách a  $m$  črtách pre jednu vzorku platí  $m \gg n$ . Preto je potrebné použitie vhodného modelu, ktorý dokáže identifikovať malú podmnožinu prediktívnych črt, ako aj poskytovať, či už automatickú alebo manuálnu, redukcia črt na základe prediktívnosti.

Dôležitým krokom v celom procese analýzy cytometrických dát je vyvodenie nových záverov a poznatkov z výsledkov analýzy, na ktoré je však potrebné mať vedomosti z doménovej oblasti a preto ich vykonáva výskumník. Existujú prístupy, ktorými je možné výskumníkom vyvodzovanie nových záverov a poznatkov uľahčiť, zrýchliť, spresniť a minimalizovať závažnosť subjektivity. Vhodným prístupom je využitie vizualizácie na adekvátne vizualizovanie výsledkov analýzy takým spôsobom, ktorý bude prospešný pre výskumníkov. Možnosti typov vizualizácie, ako aj konkrétnych výsledkov, je mnoho. Navrhujeme preto zamerať sa na vizualizáciu bunkových populácií a výsledkov predikcie, ktoré by mohli mať pre výskumníkov, pri vyvodzovaní nových záverov a poznatkov, najväčšiu pridanú hodnotu.

Výsledky navrhnutého riešenia je nevyhnutné overiť a porovnať s existujúcimi prístupmi. Pri výpočte hustoty je potrebné overiť správnosť výpočtu a následne porovnať výpočtový čas na veľkých dátových množinách s existujúcimi riešeniami. Proces redukcie dát na základe hustoty treba overiť na vygenerovaných dvojrozmerných dátach, na ktorých bude zreteľne vidieť správnosť navrhnutého riešenia a následne porovnať výpočtové časy redukcie dát na základe hustoty navrhnutého riešenia s existujúcimi riešeniami. Vizualizácie identifikovaných bunkových populácií sa musia porovnať s vizualizáciami existujúcich prístupov na identifikáciu bunkových populácií a vyhodnotiť, či výsledky navrhovaného riešenia dosahujú konzistentné výsledky v porovnaní s existujúcimi riešeniami. Na vyhodnotenie správnosti výsledkov z biologického hľadiska je potrebné overenie výsledkov doménovými expertmi. Správnosť extrakcie črt a predikcie klinického stavu je potrebné vyhodnotiť porovnaním výsledkov navrhnutého riešenia s výsledkami riešení zúčastnených súťaže FlowCAP-II použitím dátovej množiny z tejto súťaže.



### 3.1.1 Dáta

K dispozícii sú cytometrické a klinické dáta. Cytometrické dáta sú viacrozmerne a mnohopočetné dátové množiny, ktoré sú uložené v binárnych súboroch typu FCS. Bližšia špecifikácia cytometrických dát a FCS súborov je popísaná v časti 2.1.1.

Cytometrické dáta sú rozdelené do 855 vzoriek odobratých od 313 pacientov. Každá vzorka je v jednom FCS súbore a 855 vzoriek spolu obsahuje 139246956 meraní – buniek, pričom najmenšia vzorka obsahuje iba 1667 a najväčšia až 1065696 buniek a všetky majú 48 cytometrických znakov. Pacienti sú rozdelení do troch skupín a to zdraví (10 pacientov), s chorobou waldenstrom myelóm (72 pacientov) a s chorobou mnohopočetný myelóm (231 pacientov).

Jednotlivé vzorky cytometrických dát sú rozdelené do troch panelov a to panel P2, panel P3 a panel P4. Jednotlivé panely zoskupujú vzorky, ktoré vznikli v rovnakom procese extrahovania cytometrických dát zo vzoriek kostnej drene odobratej od pacientov. Každý panel sa zameriaval na inú analýzu a preto aj použité cytometrické znaky sú medzi panelmi rozdielne. Na analýzu panelov P2 a P3 bolo definovaných trinásť zhlukovacích cytometrických znakov a to CD10, CD19, CD20, CD22, CD27, CD34, CD38, CD45, CD138, IGM, IGD, IGA a IGG.

Klinické dáta sú dostupné iba pre pacientov s chorobou typu waldenstrom myelóm a mnohopočetný myelóm. V klinických dátach sú pre jednotlivých pacientov rôzne údaje zozbierané pri odbere kostnej drene pacienta, ako napríklad úroveň hemoglobínu, kreatinínu v krvi a počet krvných doštičiek, ale aj údaje o podstupených liečbach a odozvách na tieto liečby a doplňujúce informácie k jednotlivým liečbam, ale aj pacientom samotným.

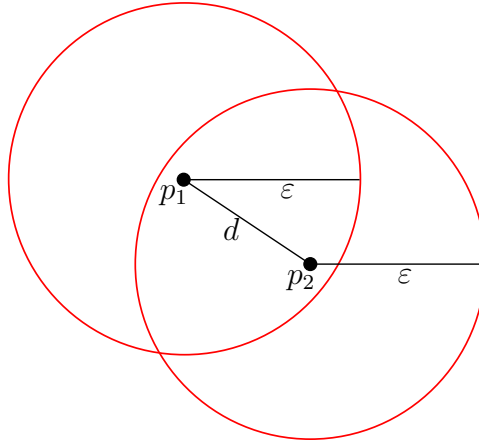
Hlavným problémom klinických dát je veľké množstvo chýbajúcich údajov. Ak zoberieme, že počet údajov v klinických dátach je  $n \times m$ , kde  $n$  je počet riadkov a  $m$  je počet stĺpcov, tak výsledný počet je 90945 údajov. Z tohto počtu údajov chýba až 57221, čo je približne 63%.

## 3.2 Výpočet hustoty buniek

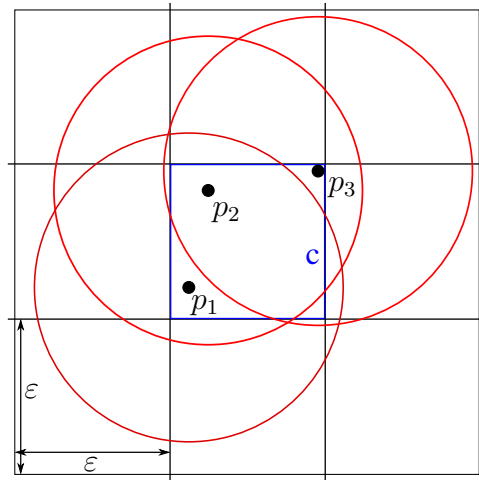
Hustota bunky je počet buniek v jej okolí. Bunky sú susedné ak sa nachádzajú navzájom v okoliach, teda ak ich vzdialenosť je menšia alebo rovná veľkosti okolia. Veľkosť okolia je definovaná parametrom  $\varepsilon$ , ktorý vyjadruje polomer kruhu, ktorý reprezentuje okolie. Tieto vlastnosti sú vyobrazené na obrázku 3.1, kde je vidno dve bunky  $p_1$  a  $p_2$ , ktoré sú si navzájom vo svojich  $\varepsilon$ -okoliach, lebo ich vzájomná vzdialenosť  $d$  je menšia ako  $\varepsilon$ , a teda hustota oboch buniek je jedna. Vzdialenosť medzi dvoma bunkami sa počíta manhattanskou vzdialenosťou.

Ak by sme si pre bunku nijak neobmedzili množinu jej potencionálnych susedov a teda by množina potencionálnych susedov zahŕňala všetky ostatné bunky, museli by sme pre každú bunku počítat vzdialenosť s každou ďalšou bunkou, čo je časová náročnosť  $O(n^2)$  pri  $n$  bunkách. Potencionálny sused bunky je taká bunka, ktorá sa nachádza v jej  $\varepsilon$ -okolí ( $\varepsilon_N$ ) alebo v blízkosti jej okolia.

Rozdelením priestoru v každej dimenzii na intervaly o veľkosti  $\varepsilon$  vieme zredukovať pre každú



Obr. 3.1: Ukážka  $\varepsilon$ -okolía



Obr. 3.2: Ukážka priestoru

bunku jej množinu potencionálnych susedov. Ak si dvojrozmerný priestor obsahujúci bunky umiestnené v priestore na základe hodnôt ich parametrov rozdelíme na  $\varepsilon$ -intervaly ( $\varepsilon_I$ ), vznikne mriežka s okienkami o veľkosti  $\varepsilon$ .  $\varepsilon_I$  je interval na danej dimenzii o veľkosti  $\varepsilon$ . Ak si zoberieme bunku  $p$  v tomto priestore, ktorá sa nachádza v okienku  $c$ , tak vieme s určitou povedať, že susedné bunky  $p$  bunky sa nachádzajú iba v susedných okienkách okienka  $c$ . Susedné okienka okienka  $c$  sú také okienka, ktoré sú na danej dimenzii v  $\varepsilon$ -intervale okienka  $c$  posunutého o hodnotu z množiny  $\{-1, 0, 1\}$ . Určitosť tohto výroku vyplýva z toho, že ak by sa bunka  $p$  nachádzala na ľubovolnom mieste v okienku  $c$  s veľkosťou  $\varepsilon$ , tak  $\varepsilon$ -okolie bunky  $p$  určite nebude siahť ďalej ako do susedných okienok okienka  $c$ . Na obrázku 3.2 vidno, že kdekoľvek sa nachádzajúce bunky v okienku  $c$  nebudú  $\varepsilon$ -okolím presahovať mimo susedných listov listu  $c$ , aj v prípade bunky  $p_3$ , ktorá sa nachádza na okraji okienka  $c$ .

Preto vieme, že potencionálne susedné bunky pre bunku  $p$  z okienka  $c$  sú všetky bunky zo susedných okienok okienka  $c$ . Tým sa výrazne zredukuje množina buniek, s ktorými musíme počítať spoločnú vzdialenosť aby sme získali hustotu bunky  $p$ . Množinu buniek z konkrétneho okienka v mriežke vieme získať v konštantnom čase  $O(1)$  ak je mriežka implementovaná

dvojrozmerným polom. Na prístup a získanie buniek z okienka teda potrebujeme indexy  $\varepsilon$ -intervalov do mriežky. Tie vieme vypočítať pre každú dimenziu, ak číslujeme intervaly od nuly, nasledovne

$$\varepsilon_I(p, d) = \text{int}\left(\frac{p_d}{\varepsilon}\right) \quad (3.1)$$

Kde  $d$  je dimenzia, pre ktorú rátame index  $\varepsilon$ -intervalu bunky  $p$ ,  $p_d$  je hodnota bunky  $p$  v dimenzii  $d$  a  $\text{int}$  je funkcia, ktorá z desatinného čísla zoberie iba celé číslo pred desatinnou čiarkou.

Ak predpokladáme, že dáta sú normalizované do intervalu  $[0, 1]$  a  $\varepsilon$  rozdelí každú dimenziu na  $m$  intervalov, tak pri  $k$  dimenziách vznikne  $m^k$  okienok, pričom  $n \ll m^k$ , kde  $n$  je počet buniek. Z toho dôvodu, aj keby bola každá bunka vo vlastnom okienku, tak veľká väčšina okienok by bola prázdna a zároveň už pri pomerne nízkom  $k$  je pamäťová náročnosť obrovská. Na vyriešenie tohto problému sme navrhli strom hustôt.

### 3.2.1 Strom hustôt

Strom hustôt využíva princíp  $\varepsilon$ -okolí za nízkej pamätevej náročnosti. Strom sa skladá z vrcholov, hrán a každá úroveň stromu reprezentuje jednu dimenziu v priestore okrem koreňa, ktorý nerepresentuje žiadnu dimenziu. Hĺbka stromu preto bude  $k + 1$ , kde  $k$  je počet dimenzií v priestore. Každý vrchol obsahuje hešovací mapu, v ktorej kľúčom je index  $\varepsilon$ -intervalu, ktorý reprezentuje potomka daného vrchola a hodnotou v mape je hrana ku potomkovi. Strom sa stavia postupne podľa buniek z dát a preto bude obsahovať iba tie  $\varepsilon$ -intervaly a ich prieniky, ktoré budú obsahovať aspoň jednu bunku.

### 3.2.2 Stavba stromu hustôt

Koreň stromu nerepresentuje žiadnu dimenziu, iba obsahuje mapu do  $\varepsilon$ -intervalov prvej dimenzie. Strom sa zostavuje postupne vkladaním buniek zo vstupnej dátovej množiny. Pre prvú dimenziu bunky sa podľa vzorca 3.1 na výpočet indexu  $\varepsilon$ -okolí vypočíta index. Zistí sa, či koreň stromu obsahuje v mape kľúč s hodnotou indexu. Ak nie, vytvorí sa nový stĺpec a pridá sa do mapy vrchola. Následne sa podľa indexu  $\varepsilon$ -intervalu prejde do nasledujúceho vrchola. V tomto momente sa algoritmus nachádza na prvej úrovni v strome, ktorá reprezentuje prvú dimenziu, ak berieme, že koreň je nultá úroveň stromu. Z tejto úrovne sa vieme dostať do ďalšej úrovne, ktorá reprezentuje nasledujúcu dimenziu. Algoritmus postupuje rovnako pri každej dimenzii vstupných dát, až kým bunka nie je zaradená do jedného z existujúcich listov stromu, prípadne do novo vytvoreného listu. Takto sa postupuje pokiaľ nie sú všetky bunky zo vstupných dát zaradené do listov stromu. Algoritmus stavby stromu je naznačený pseudokódом 1.

---

**Algorithm 1** Algoritmus vytvorenia stromu hustôt

---

```
1: procedure BUILDENSITYTREE(dataset,  $\varepsilon$ )
2:   for each cell in dataset do
3:     Insert(root, cell,  $\varepsilon$ , dimensions(dataset))
4:   end for
5: end procedure
6: function INSERT(node, cell,  $\varepsilon$ , height)
7:   if height = 0 then ▷ Sme v liste
8:     AddCell(node, cell)
9:   else
10:    index  $\leftarrow$  GetIntervalIndex(cell,  $\varepsilon$ )
11:    if NotContainsIndex(node, index) then
12:      CreateNodeAtIndex(index)
13:    end if
14:    node  $\leftarrow$  NodeAtIndex(node, index)
15:    Insert(node,  $\varepsilon$ , height - 1)
16:  end if
17: end function
```

---

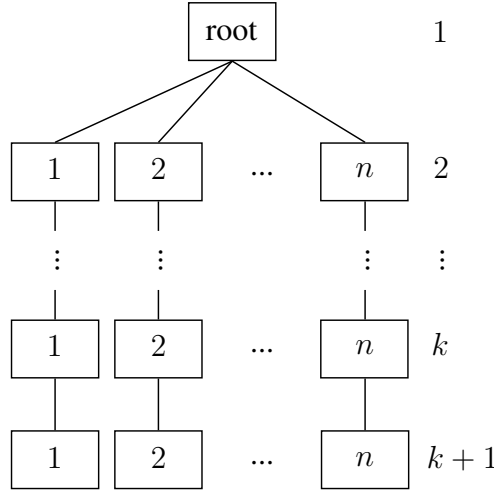
### 3.2.3 Pamäťová náročnosť

Pre prístup do konkrétneho listu stromu na základe bunky, ktorá sa v ňom nachádza postupujeme rovnako ako pri vkladaní novej bunky do stromu. Pomocou indexov  $\varepsilon$ -intervalov sa posúvame po úrovniach stromu až po konkrétny list. Preto v takto zostavenom strome bude jeden konkrétny prechod stromom predstavovať cestu do konkrétneho listu, čo predstavuje  $k$ -rozmerné okienko v prípade  $k$ -rozmernej mriežky. Z toho vyplýva, že prístup do listu na konkrétnych súradniciach v  $k$ -rozmernej mriežke je v čase  $O(k)$ , lebo prístup do hešovacej mapy je  $O(1)$ , kde  $k$  je počet dimenzií. Keďže platí  $k \ll n$ , tak  $O(k)$  je zanedbateľné.

Ako bolo ukázané, výška stromu bude  $k+1$ , keďže každá úroveň predstavuje jednu dimenziu. Keďže strom predstavuje  $k$  dimenzionálny priestor, ktorý je na každej dimenzii rozdelený na  $m$  intervalov, tak intuitívne z toho vychádza, že šírka stromu bude  $m^k$  aby dokázal strom zohľadniť všetky pozície v priestore. Avšak vďaka vlastnosti postupného stavania stromu vieme zabezpečiť, že strom bude reprezentovať iba tie pozície v priestore, ktoré budú obsahovať aspoň jednu bunku. Pomocou tejto vlastnosti vieme ukázať, že šírka stromu bude  $O(n)$ .

Ak si zoberieme najhorší prípad, kedy by hodnota  $\varepsilon$  bola zvolená taká, že každá bunka zo vstupných dát by skončila vo vlastnom liste, tak na poslednej úrovni ( $k+1$ ) stromu by sme mali  $n$  listov a teda šírka stromu bude  $n$ . Na prístup k týmto listom by bolo potrebné ďalších  $n$  vrcholov na úrovni  $k$ . Z toho vyplýva, že počet vrcholov v strome je ohraničený  $O(k \times n + 1)$  ak  $m^k > n$  a koreňa má  $n$  potomkov. Medzi dvoma vrcholmi môže existovať iba jedna hrana a teda vieme s určitosťou povedať, že počet hrán je ohraničený  $O(k \times n)$ . Na obrázku 3.3 je vyobrazená štruktúra stromu pri najhoršom prípade, kedy by každá bunka z dát, po zaradení do listov, skončila vo vlastnom liste.

Takto zostrojeným stromom sa nám podarilo dostať pamäťovú náročnosť do reálnych hodnôt



Obr. 3.3: Štruktúra stromu hustôt v najhoršom prípade

za cenu malého zhoršenia prístupu na ľubovoľnú pozíciu v priestore z  $O(1)$  na  $O(k)$ .

### 3.2.4 Časová náročnosť

Pri výpočte hustôt buniek v strome hustôt sa využíva princíp potencionálne susedných buniek zo susedných listov konkrétneho listu. Pri stavbe stromu nevieme efektívne určiť pre všetky listy jeho susedné listy, lebo v momente vytvárania listu ešte nemusia byť vytvorené všetky jeho susedné listy. Priestorové indexy susedných listov si vieme vypočítať karteziánskym súčinom. Množiny súčinu vytvoríme tak, že pre každú dimenziu zoberieme index  $\varepsilon$ -intervalu ( $\varepsilon_I$ ) znížený o  $\pm 1$  a z výsledku odstránime samotný list. Pre list  $l$  z dvojrozmerného priestoru by teda indexy  $\varepsilon$ -intervalov boli výsledkom  $\{\varepsilon_I(l, 0) - 1, \varepsilon_I(l, 0), \varepsilon_I(l, 0) + 1\} \times \{\varepsilon_I(l, 1) - 1, \varepsilon_I(l, 1), \varepsilon_I(l, 1) + 1\} \setminus \{\varepsilon_I(l, 0), \varepsilon_I(l, 1)\}$ . Pri  $k$  dimenziách bude mať list  $3^k - 1$  susedov, čo už pri pomerne malom  $k$  bude veľké číslo a je teda neefektívne počítat si všetky indexy susedných listov, keď  $n \ll 3^k - 1$  a teda väčšina z týchto susedných listov ani nebude existovať. Práve preto sa susedné listy vypočítavajú dynamicky počas prechádzania stromom.

Výpočet hustôt buniek sa vykonáva po listoch stromu. Pre každý list  $l_i$  sa zistia jeho susedia dynamicky, prechodom cez strom od koreňa až po susedné listy tak, že na každej úrovni sa okrem indexu  $\varepsilon$ -intervalu listu  $l_i$  pozerá na index  $\pm 1$ , ktoré zodpovedajú susedným listom. Ak sa na danom vrchole v danej úrovni taký index nenachádza, potom taký list nebude existovať a navštívia sa len existujúce susedné listy listu  $l_i$ .

Keď máme všetky susedné listy tak vieme medzi bunkami jednotlivých susedných listov  $nl_i$  a listom  $l_i$  vypočítať vzdialenosť a na základe nej vyhodnotiť, či sa nachádzajú v  $\varepsilon$ -okolí. Pri hustote dvoch bodov  $p_i, p_j$  platí, že ak  $d(p_i, p_j) \leq \varepsilon$  tak potom aj  $d(p_j, p_i) \leq \varepsilon$  a teda ak  $p_i \in \varepsilon_N(p_j)$  tak potom aj  $p_j \in \varepsilon_N(p_i)$ , kde  $d$  je funkcia vzdialenosti dvoch bodov v priestore a  $\varepsilon_N$  predstavuje  $\varepsilon$ -okolie daného bodu. Na základe týchto dvoch vlastností vyplýva, že ak sme vyrátali vzdialenosť  $d(p_i, p_j)$ , tak už nemusíme počítat vzdialenosť  $d(p_j, p_i)$  a teda ak počítame

vzdialenosti medzi dvoma rovnakými množinami buniek, nemusíme to robiť v čase  $O(n^2)$ , ale  $O(\frac{n^2}{2})$  a navyše nepočítame vzdialenosť bunky samú so sebou, z čoho dostávame výslednú časovú náročnosť  $O(\frac{n^2-n}{2})$ .

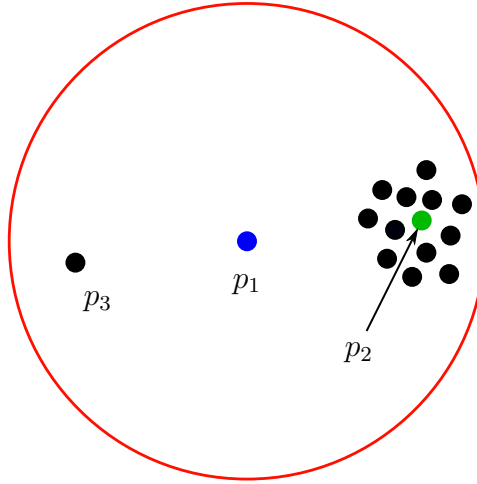
Pri počítaní vzdialenosti medzi bunkami listov rozoznávame dva prípady. Prvý keď sa počítajú vzdialenosti medzi bunkami rovnakého listu. Ako bolo ukázané, tak tento prípad má časovú náročnosť  $O(\frac{m^2-m}{2})$ , kde  $m$  je počet buniek v liste. Druhý prípad nastáva keď sa počítajú vzdialenosti medzi bunkami listu  $l_i$  a jeho susedným listom. V tomto prípade nie sú množiny buniek rovnaké a preto nemôžeme využiť vylepšenie času z prvého prípadu, ale musíme vypočítať vzdialenosť každej bunky s každou a teda v čase  $O(m \times s)$  kde  $m$  je počet buniek v liste a  $s$  počet v susednom liste. S využitím vyššie spomenutých vlastností vieme, že ak sme vypočítali vzdialenosti medzi bunkami listov  $l_i$  a  $l_j$ , tak už nemusíme počítať medzi listami  $l_j$  a  $l_i$ , čo redukuje celkový počet potrebných výpočtov.

Ak by bola hodnota parametra  $\varepsilon$  taká, že by rozdelením priestoru na základe hodnoty  $\varepsilon$  boli všetky bunky vo výslednom strome v jednom liste, napríklad pri hodnote  $\varepsilon = 1$ , tak by bolo potrebné vypočítať vzdialenosti medzi všetkými bunkami. Z toho vyplýva, že časová náročnosť je v najhoršom prípade ohraničená  $O(\frac{n^2-n}{2})$ .

### 3.2.5 Paralelizácia riešenia

Výpočet hustoty buniek listu stromu je proces nezávislý od výpočtu hustoty buniek iného listu stromu a preto je tento proces ľahko oddeliteľný a teda paralelizovateľný. V jednom momente vieme počítať hustoty buniek  $m$  listov stromu, kde  $m$  je zhora ohraničené počtom virtuálnych jadier procesora stroja, na ktorom sa algoritmus vykonáva. Aby sme zachovali vylepšenie počítania vzdialenosti medzi bunkami dvoch listov iba raz, musíme si pre každý list uchovávať informáciu, s ktorými listami sme v danom liste už vzdialenosti počítali. Pri paralelizácii riešenia nám v tomto prípade môže vzniknúť súťaženie o prístup. Tento prípad sme vyriešili tak, že v jednom momente môže iba jedno vlákno pristupovať k informáciám o susedných listoch, s ktorými vzdialenosť už bola počítaná a modifikovať ju na základe ktorej sa rozhoduje, či sa pokračuje s výpočtom alebo sa proces ukončí. Tým zabránime, aby sa vzdialenosti medzi dvoma listami počítali dvakrát a teda hustota pre dané bunky bola zarátaná dvakrát.

Podľa časovej náročnosti výpočtu hustoty buniek opísanej v časti 3.2.4 vidno, že rýchlosť celého výpočtu závisí od počtov buniek v jednotlivých listoch a keby väčšina buniek bola situovaná v jednom liste, tak celá časová náročnosť bude ovplyvnená hlavne rýchlosťou výpočtu vzdialeností buniek listu s bunkami toho istého listu, a teda  $O(\frac{m^2-m}{2})$ . Keďže proces výpočtu hustoty jednej bunky je nezávislý od procesu výpočtu hustoty iných buniek, tak vieme tento proces tiež paralelizovať. Pri malom počte  $m$ , resp.  $m \times s$  podľa časti 3.2.4, by režia paralelizmu bola časovo náročnejšia ako získane zrýchlenie a preto sme definovali prahovú hodnotu  $t$ . Podľa hodnoty  $t$  sa rozhoduje, či proces výpočtu vzdialenosti buniek medzi dvoma listami (listom so sebou samým prípadne listom a susedným listom) bude paralelizovaný alebo nie. Pri situácii



Obr. 3.4: Nesprávne vyhodnotenie najreprezentatívnejšej bunky

počítania vzdialeností medzi bunkami rovnakého listu musí platiť  $\frac{m^2-m}{2} > t$  a pri počítaní vzdialeností medzi bunkami listu  $l$  a susedného listu  $l_i$  musí platiť  $m \times s > t$ , kde  $m$  je počet buniek v liste  $l$  a  $s$  počet buniek v susednom liste  $l_i$  aby bol proces paralelizovaný.

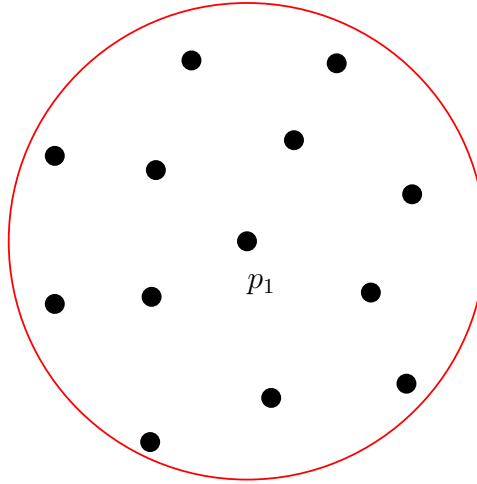
### 3.3 Váhovaná hustota

Pri klasickej hustote každá bunka v  $\varepsilon$ -okolí prispieva rovnako do výslednej hodnoty hustoty bez ohľadu na to, ako ďaleko sa vrámci okolia nachádza od centrálnej bunky. Hodnotu klasickej hustoty bunky pre bunku  $p$  možno vyrátať ako  $\sum_1^m 1$ , kde  $m$  je počet susedných buniek bunky  $p$ . Na tomto vzťahu vidno, že každá bunka prispieva práve hodnotou jedna do výslednej hodnoty hustoty. Takáto reprezentácia hustoty bunky má viacero problémov, ktoré môže viesť ku nesprávnym vyhodnoteniam. Ak by sme mali situáciu ako na obrázku 3.4, tak bunka  $p_1$  bude mať najväčšiu hustotu, a teda bude najreprezentatívnejšia, lebo okrem zhluku buniek okolo bunky  $p_2$  prispieva do jej hustoty ešte aj okrajová bunka  $p_3$ . Avšak intuitívne vidno, že bunka  $p_2$  je reprezentatívnejšia, keďže má vo svojej tesnej blízkosti viacero buniek, ktoré reprezentuje. Pri situácií vyobrazenej na obrázku 3.5 pri pomernej veľkej hodnote  $\varepsilon$  bude mať bunka  $p_1$  najväčšiu hustotu, pričom bunky v jej okolí pripomínajú skôr šumové dáta.

Preto na vyriešenie spomenutých problémov navrhujeme použitie váhovanej hustoty, ktorej hodnota nezávisí iba od počtu, ale aj od vzdialeností susedných buniek. Hodnota, ktorou prispieva susedná bunka do výslednej hodnoty váhovanej hustoty centrálnej bunky sa vyráta ako

$$c = \left(1 - \frac{d}{\varepsilon}\right)^{\varepsilon} \quad (3.2)$$

kde  $c$  je príspevok susednej bunky do výslednej hodnoty váhovanej hustoty centrálnej bunky,  $d$  je vzdialenosť medzi susednou a centrálnou bunkou a  $\varepsilon$  je veľkosť bunkového okolia. Váhovaná



Obr. 3.5: Určenie reprezentatívnosti pri šumových dátach

hustotu  $w$  bunky  $p$  je teda

$$w(p) = \sum_{i=1}^M \left(1 - \frac{d(p, m_i)}{\varepsilon}\right)^e \quad (3.3)$$

kde  $M$  je počet susedných buniek bunky  $p$  a  $d(p, m_i)$  je vzdialenosť medzi bunkou  $p$  a jej  $i$ -tou susednou bunkou. Použitím takéhoto vzťahu sme dosiahli, že susedné bunky, ktoré sú v tesnej blízkosti bunky  $p$  prispievajú do výslednej hustoty bunky  $p$  podstatne väčším podielom ako susedné bunky, ktoré sú pri okraji  $\varepsilon$ -okolía bunky  $p$ . Túto vlastnosť dobre vidieť aj na obrázku 3.6, ktorý zobrazuje nelineárny vzťah medzi vzdialenosťou a príspevkom do výslednej hustoty pri hodnotách  $\varepsilon = 1$  a  $d \in [0, \varepsilon]$ .

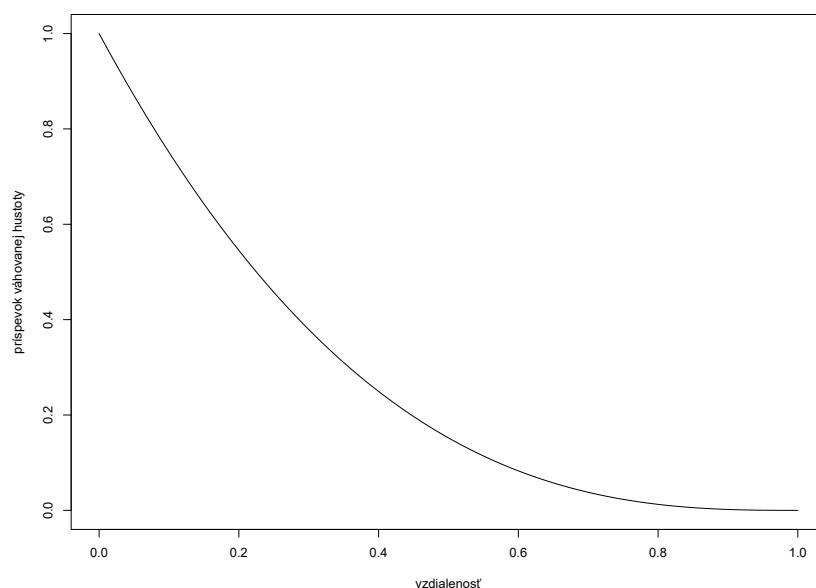
Váňovanú hustotu je možné vypočítat rovnako ako klasickú hustotu a aj obe v jednom kroku pomocou stromu hustôt.

### 3.4 Redukcia dát na základe hustoty

Cieľom redukcie dát na základe hustoty je pôvodne hojne zastúpené a vzácne bunkové populácie vo výsledku reprezentovať rovnomerne. Pri redukcii dát na základe hustoty je nevyhnuté správne identifikovať najreprezentatívnejšie bunky v rámci priestoru. V nami navrhnutom algoritme je reprezentatívnosť bunky rovná jej váňovanej hustote (viď. 3.3), z čoho vyplýva, že čím viac blízkych susedných buniek bunka má, tým je reprezentatívnejšia.

Proces redukcie dát na základe hustoty sa skladá z výberu najreprezentatívnejšej bunky spomedzi buniek, ktoré ešte neboli vybraté alebo odstránené, odstránenia susedných buniek vybratej bunky a iterovania pokým nie sú všetky bunky buď vybraté alebo odstránené. Časová náročnosť redukcie dát na základe hustoty je z veľkej časti ovplyvnená časovou náročnosťou výpočtu hustoty buniek, lebo proces redukcia je podstatne rýchlejší, keďže pri ňom nie je potrebné vykonať také množstvo operácií ako pri výpočte hustôt. Voľba nesprávnej hodnoty  $\varepsilon$ , podľa ktorej ma byť vykonaný výpočet hustôt a teda aj redukcia dát na základe hustoty,





Obr. 3.6: *Vzťah medzi vzdialenosťou a príspevkom do váhovanej hustoty*

môže viesť ku pomalému procesu redukcia dát na základe hustoty. Navyše na zvolenie vhodnej hodnoty  $\varepsilon$  je potrebná apriori analýza dát, opäť zvyšujúca celkovú časovú náročnosť celého procesu. Preto navrhujeme iteratívny prístup k redukcii dát na základe hustoty, ktorý značne urýchľuje celý proces redukcie dát na základe hustoty a vyžaduje iba jeden vstupný parameter percentuálnej veľkosti výslednej množiny zredukovaných dát.

Na zlepšenie výsledku je nevyhnutné vykonať odstránenie šumu, aby sme odstránili okrajové, zväčša chybné merania – bunky, ktoré by mohli v ďalších krokoch analýzy negatívne ovplyvniť výsledky. Okrem odstránenia šumu vykonávame aj úpravu váh v iteratívnom procese na minimalizovanie „dier“ vo výsledkom zredukovanom priestore.

### 3.4.1 Postup redukcie

Princíp redukcie dát na základe hustoty je výber reprezentatívnych a odstránenie redundantných buniek. Reprezentatívnosť bunky je určená jej váhovanou hustotou. Navrhujeme efektívny prístup, ako tento proces vykonať.

Algoritmus začína usporiadaním buniek zostupne na základe ich váhovaných hustôt. Z usporiadaných buniek sa následne zostaví dátová štruktúra fronta. Z fronty sa vyberie prvý prvok – bunka. Skontroluje sa, či daná bunka nebola už odstránená a ak áno, pokračuje sa výberom nasledujúcej vrchnej bunky z fronty až kým nie je vybratá doposiaľ neodstránená bunka. Táto bunka je označená ako vybratá. Keďže sme na začiatku usporiadali bunky podľa ich hustôt tak vieme, že na vrchu fronty sa nachádzajú bunky s najväčšou hustotou a teda najreprezentatívnejšie bunky a preto keď vyberieme bunku z fronty, ktorá nebola ešte odstránená, tak s určitosťou vieme, že sa jedná o aktuálne najreprezentatívnejšiu bunku. Následne sa všetky susedné bunky

vybratej bunky označia za odstránené. Pokračuje sa výberom aktuálne najreprezentatívnejšej bunky a iteruje sa pokým nie sú všetky bunky označené za vybraté alebo odstránené. Navrhnutý algoritmus je naznačený pseudokódom 2.

Ak by sa na začiatku bunky neusporiadali a nenaplnila by sa dátová štruktúra fronta, tak by bolo v každej iterácii redukcie, až pokým by neboli všetky bunky označené za vybraté alebo odstránené, nevyhnutné nájsť aktuálne najreprezentatívnejšiu bunku prehľadáním celej vstupnej množiny dát v čase  $O(n)$ , kde  $n$  je počet buniek vo vstupnej množine. Pri  $m$  potrebných iteráciách na redukciu je to časová náročnosť  $O(m \times n)$  a v najhoršom prípade, kedy by všetky bunky nemali žiadne susedné bunky a teda by sa v každej iterácii vybrala iba jedna bunka a neodstránila žiadna, tak by potrebný čas na vyhľadávania aktuálne najreprezentatívnejších buniek bol  $O(n^2)$ .

---

**Algorithm 2** Algoritmus redukcie dát na základe hustoty

---

```

procedure DOWNSAMPLE(dataset, densities)
  sortedPoints  $\leftarrow$  sort(dataset, densities)
  cells  $\leftarrow$  queue(sortedPoints)
  while not all cells chosen or discarded do
    topCell  $\leftarrow$  GetMostRepresentativeCell(cells)
    SetChosen(topCell)
    for each neighbourCell in GetNeighbourCells(topCell) do
      SetDiscarded(neighbourCell)
    end for
  end while
  return all chosen cells
end procedure

```

---

### 3.4.2 Iteratívny prístup

Zvolením relatívne nízkej hodnoty  $\varepsilon$  sa priestor rozdelí na veľa malých pod-priestorov. Tieto pod-priestory budú reprezentované listami stromu hustôt (viď. 3.2.1) a keďže pod-priestory budú kvôli hodnote  $\varepsilon$  pomerne malé, tak väčšina buniek skončí vo vlastnom liste stromu s výnimkou hustých oblastí priestoru, kde jeden list stromu bude obsahovať viac ako jednu bunku. Časová náročnosť výpočtu hustoty pre takto rozdelený priestor je veľmi nízka (viď. 3.2) z dôvodu potreby vykonania malého množstva výpočtov pre každú bunku. Vykonaním redukcie dát na základe hustoty, podľa hustoty v takto rozdelenom priestore, by sa odstránili iba redundantné bunky z hustých oblastí priestoru, lebo väčšina buniek by nemala žiadne alebo iba veľmi málo susedných buniek.

Na základe využitia vyššie spomenutej vlastnosti delenia priestoru navrhujeme iteratívny prístup pre redukciu dát na základe hustoty. Na začiatku sa zvolí relatívne nízka hodnota  $\varepsilon$ , vypočíta sa hustota v priestore a vykoná sa redukcia dát na základe hustoty. Potom, ak veľkosť výstupnej množiny dát redukcie ( $p_c$ ) je rovná  $\pm 1\%$  zvolenej výslednej percentuálnej časti

pôvodnej množiny ( $p_r$ ), algoritmus skončí. V opačnom prípade sa upraví hodnota  $\varepsilon$  nasledovne:

$$\varepsilon = \begin{cases} \varepsilon \times 2, & p_c > p_r \\ \frac{|\varepsilon - \varepsilon_p|}{2}, & \text{inak} \end{cases}$$

kde  $\varepsilon_p$  je hodnota  $\varepsilon$  z predchádzajúcej iterácie. Zvýšením hodnoty  $\varepsilon$  dosiahneme v nasledujúcej iterácii odstránenie viacerých a ponechanie menšieho počtu buniek z dôvodu väčších listov stromu, ktoré budú obsahovať viac buniek a teda bunky budú mať viacero susedných buniek. Tým sa budeme klesajúcim prístupom blížiť ku požadovanej veľkosti výslednej množiny. Naopak, ak hodnotu  $\varepsilon$  v nasledujúcej iterácii znížime, dosiahneme tým odstránenie menšieho počtu a ponechanie väčšieho počtu buniek, čím sa budeme stúpajúcim prístupom blížiť ku veľkosti výslednej množiny.

Ak platí  $p_c > p_r$ , tak iba bunky, ktoré neboli v aktuálnej iterácii odstránené sú použité ako vstupné dáta do nasledujúcej iterácie. Naopak, ak platí  $p_c \leq p_r$ , tak neodstránené bunky z poslednej iterácie, kde platilo  $p_c > p_r$  sú použité ako vstupné dáta do nasledujúcej iterácie, lebo ak platí  $p_c \leq p_r$ , tak vieme, že v aktuálnej iterácii boli odstránené aj tie bunky, ktoré by odstránené byť nemali a preto je potrebné sa vrátiť do stavu, v ktorom ešte neboli odstránené.

Následne algoritmus pokračuje výpočtom hustoty podľa novej hodnoty  $\varepsilon$  a redukcie dát na základe hustoty na nových dátach z predchádzajúcej iterácie. Ďalšia z vlastností, ktoré sa pri iteratívnom prístupe redukcia dát na základe hustoty, využívajú je postupné redukovanie vstupnej množiny dát. Ako je vyššie opísané, iterácia v procese redukcie pracuje iba na podmnožine dát z predchádzajúcej iterácie, čo ma za následok podstatné zníženie časovej náročnosti celého procesu, lebo keď algoritmus dosiahne pomerne vysoké hodnoty  $\varepsilon$ , kde by výpočet hustoty mohol zabráť podstatne viac času, tak vstupná množina dát je už v tom momente zredukovaná tak, že výpočet hustoty je rýchly, lebo je potrebné vykonať menej výpočtov, čo zrýchľuje celý proces redukcia dát na základe hustoty.

Výslednou množinou dát sú vybraté najreprezentatívnejšie bunky z poslednej iterácie, kde platilo  $p_c = p_r \pm 1\%$ . Algoritmus iteratívnej redukcie dát na základe hustoty je naznačený pseudokódom 3.

### 3.4.3 Úprava váh

Pri výbere najreprezentatívnejšej bunky v ľubovolnej iterácii nemáme informácie o jej reprezentatívnosti z predchádzajúcich iterácií, čo môže viesť k nesprávnemu vyhodnoteniu a následnému odstráneniu reprezentatívnej bunky, čím vzniknú „diery“ vo výsledkom redukovanom priestore.

Takýto prípad môže nastať, keď v iterácii  $i$  s hodnotou  $\varepsilon_1$  je bunka  $p_1$  vyhodnotená ako najreprezentatívnejšia a teda je vybratá a jej susedné bunky sú odstránené. Následne v iterácii  $i + 1$  s hodnotou  $\varepsilon_2$  takou, že platí  $\varepsilon_2 > \varepsilon_1$ , má bunka  $p_1$  menej susedných buniek ako niektorá jej nová susedná bunka  $p_2$ , ktorá je reprezentatívnejšia z dôvodu, že väčšina susedných buniek

---

**Algorithm 3** Algoritmus iteratívnej redukcie dát na základe hustoty

---

```
procedure ITERATIVEDOWNSAMPLING(dataset, resultRatio)  
   $\varepsilon \leftarrow$  choose initial value of  $\varepsilon$   
  currentRatio  $\leftarrow$  100%  
  while currentRatio  $\neq$  resultRatio  $\pm$  1% do  
    weightedDensities  $\leftarrow$  CalculateDensities(dataset,  $\varepsilon$ )  
    weightedDensities  $\leftarrow$  WeightsAdjusting(weightedDensities)  
    dataset, currentRatio  $\leftarrow$  Downsample(dataset, weightedDensities)  
     $\varepsilon \leftarrow$  Adjust( $\varepsilon$ )  
  end while  
  dataset  $\leftarrow$  RemoveNoise(dataset)  
  return dataset  
end procedure
```

---

bunky  $p_1$  bola odstránená v iterácii  $i$  a teda je bunka  $p_1$  odstránená.

Na odstránenie tohto problému navrhujeme upraviť v každej iterácii váhy buniek nasledovne:

$$w_i(p) = w_i(p) + \frac{w_j(p)}{2}$$

kde  $w_i(p)$  je váhovaná hustota bunky  $p$  v iterácii  $i$  a  $w_j(p)$  je váhovaná hustota z poslednej iterácie, kde platilo  $p_c > p_r$  (viď. 3.4.2). Takouto úpravou váhovanej hustoty dosiahneme minimalizovanie „dier“ vo výsledkom zredukovanom priestore.

### 3.4.4 Odstránenie šumu

Keďže využívame iteratívny prístup k redukcii dát na základe hustoty, tak nemôžeme jednoducho označiť za šumové dáta tie bunky, ktoré nemajú ani jednu susednú bunku, lebo podľa definície iteratívneho prístupu (viď. 3.4.2) je v prvých iteráciách použitá relatívne malá hodnota  $\varepsilon$ , čo má za dôsledok, že väčšina buniek skončí vo vlastnom liste stromu a má málo až žiadne susedné bunky, čo by spôsobilo chybné vyhodnotenie šumových dát.

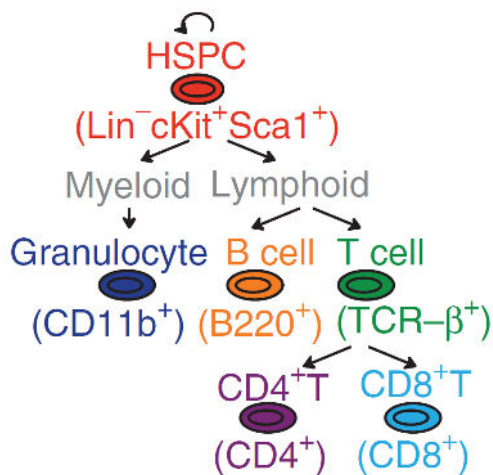
Na vyriešenie tohto problému navrhujeme nový prístup na identifikovanie šumových dát pri použití iteratívneho prístupu. V poslednej iterácii procesu redukcie, v ktorej platí  $p_c = p_r \pm 1\%$  a algoritmus v nej končí, sa identifikujú šumové dáta. Tie bunky z výslednej zredukovanej množiny, ktoré nemajú v poslednej iterácii žiadne susedné bunky sú označené za šumové dáta a odstránené z výslednej zredukovanej množiny.

Iteratívny prístup na redukcii dát na základe hustoty terminuje pri relatívne vysokých hodnotách  $\varepsilon$  a teda tento prístup vychádza z pozorovania, že ak bunka nemá žiadne susedné bunky ani pri relatívne vysokých hodnotách  $\varepsilon$ , tak sa jedná o šumové dáta.

## 3.5 Upsampling

*Upsampling* je priradzovanie bodov, ktoré boli v procese redukcie dát na základe hustoty odstránené, do zhlukov a teda tried bodov, ktoré boli v procese redukcie vybraté.

Po vykonaní redukcie na základe hustoty je potrebné vykonať zhľukovanie buniek na ich priradenie do diskretných skupín – zhlukov, ktoré reprezentujú triedy, resp. bunkové populácie. Bunkové populácie tvoria hierarchickú štruktúru, kde napríklad T-bunky sa na nižšej úrovni hierarchie delia na  $CD4^+$ T-bunky a  $CD8^+$ T-bunky. Na obrázku 3.7 je vyobrazená celá hierarchia bunkových populácií u myši. Z tohto hľadiska navrhujeme použitie aglomeratívneho hierarchického zhľukovacieho algoritmu, ktorý dokáže zachytiť takúto hierarchiu v rámci zhľukovaných dát.



Obr. 3.7: Hierarchia bunkových populácií u myši [28]

Po identifikovaní bunkových populácií zo zredukovaných dát na základe hustoty je potrebné priradiť do príslušných bunkových populácií aj bunky, ktoré boli v procese redukcie odstránené, na získanie výsledných počtov buniek a hodnôt pre jednotlivé bunkové populácie.

### 3.5.1 Navrhnutý algoritmus

Máme množinu buniek  $K$ , ktoré boli v procese redukcie dát na základe hustoty vybraté a množinu buniek  $D$ , ktoré boli odstránené a množinu všetkých buniek  $A$ , pričom platí  $D + K = A$ . Ak máme hodnotu  $p \in [0, 1]$ , ktorá hovorí o pomere zredukovaných dát, tak potom platí  $|D| = (1 - p) \cdot |A|$  a  $|K| = p \cdot |A|$ . Cieľom je každej bunke  $d \in D$  priradiť bunkovú populáciu podľa bunky  $k \in K$ , ktorá je pre bunku  $d$  najreprezentatívnejšia. Jednoduchý spôsob je porovnať každú bunku  $d$  s každou bunkou  $k$ . Takýto prístup má časovú náročnosť  $O(|D| \cdot |K|)$  alebo  $O(n^2 \cdot (1 - p) \cdot p)$ , kde  $n = |A|$ . Pri väčších hodnotách  $n$  môže byť takýto prístup neefektívny a preto navrhujeme využiť strom hustôt (viď. 3.2.1) na vykonanie upsampling-u.

Ak by sa počítali vzdialenosť medzi každou bunkou  $d$  a  $k$ , tak by sa vykonávalo veľa nepotrebných výpočtov, lebo by sa počítali vzdialenosti aj medzi bunkami, kde by platilo, že

bunka  $k$  nie je reprezentatívna pre bunku  $d$ . Z tohto hľadiska je potrebné zredukovať množinu potencionálne reprezentatívnych buniek  $k$  pre bunky  $d$ . To sa dosiahne využitím stromu hustôt. Strom hustôt sa postaví použitím buniek  $k \in K$  a hodnota  $\varepsilon$  sa použije dvojnásobok hodnoty  $\varepsilon$  poslednej iterácie iteratívneho procesu redukcie dát na základe hustoty (viď. 3.4.2).

Takto vieme podstatne zredukovať množinu potencionálnych reprezentatívnych buniek  $k$  pre bunky  $d$ , podľa ktorých sa určuje bunková populácia, keďže sa využíva rovnaký princíp ako pri výpočte hustoty, kde sa počítajú vzdialenosti len s okolitými bunkami (viď. 3.2). Navrhujeme dva prístupy, ako na základe bunkových populácií množiny potencionálne reprezentatívnych buniek  $k_r$  pre bunky  $d$  určiť ich bunkové populácie a to:

1. najbližšia susedná bunka
2. váhované hlasovanie

Pri metóde najbližšej susednej bunky sa spomedzi potencionálne reprezentatívnych buniek  $k_r$  pre každú bunku  $d$  nájde, na základe vzdialenosti, najbližšia bunka a bunke  $d$  sa priradí bunková populácia najbližšej bunky  $k_r$ .

Pri váhovanom hlasovaní má každá bunka  $d$  množinu tried bunkových populácií svojich potencionálne reprezentatívnych buniek  $k_r$ . Každá bunka  $k_r$  prispieva ku triede bunkovej populácie, ktorú reprezentuje váhou rovnou príspevku váhovanej hustoty, vyrátanej podľa vzorca 3.2, ktorou by prispievala do výslednej hustoty bunky  $d$ . Po vypočítaní vzdialenosti bunky  $d$  so všetkými bunkami  $k_r$  sa nájde trieda bunkovej populácie z množiny tried bunky  $d$ , ktorá má najväčšiu váhu a tá sa bunke  $d$  priradí.

Výsledkom procesu upsampling-u sú priradené bunkové populácie pre všetky bunky z množiny  $A$ .

### 3.5.2 Paralelizácia riešenia

Využitím stromu hustôt je možné upsampling buniek podstatne zrýchliť oproti algoritmu, ktorý by prehľadával všetky možnosti. Avšak priradenie bunkovej populácie odstránenej bunke na základe vybratých buniek je proces nezávislý od priraďovania bunkovej populácie na základe vybratých buniek ktorejkoľvek inej bunky a preto sa môžu tieto výpočty vykonávať simultánne.

Navrhujeme paralelizovať celý proces priraďovania bunkových populácií odstráneným bunkám podľa vybratých buniek z procesu redukcia dát na základe hustoty tak, že sa množina odstránených buniek rozdelí na  $x$  rovnomerných podmnožín, kde  $x$  je počet virtuálnych jadier stroja a výpočty každej podmnožiny sa budú vykonávať nezávisle od ostatných podmnožín a na konci sa výsledok spojí do výslednej množiny. Týmto prístupom sa ešte väčšmi urýchli celý proces.

## 3.6 Predikcia klinického stavu pacienta

Cieľom predikcie je na základe cytometrických a klinických dát predikovať stav pacienta, čo môže byť pacientova odpoveď na liečbu, príznak či je chorý alebo zdravý, resp. typ choroby. K dispozícii sú dva typy dát a to cytometrické a klinické. V cytometrických dátach je možné identifikovať bunkové populácie a následne z nich extrahovať črty pre pacienta. Klinické dáta hovoria o klinických stavoch všetkých pacientov a navyše obsahujú viacero klinických a biologických informácií, z ktorých sa dá extrahovať práve klinický stav pacientov a ďalšie črty. Po konzultácii s doménovými expertmi navrhujeme vykonať dva typy predikcie klinického stavu pacienta a to:

1. na základe cytometrických dát predikovať, či je pacient zdravý alebo chorý a typ choroby
2. a na základe cytometrických a klinických dát predikovať pacientovu odozvu na podstúpenú liečbu.

Voľba dvoch rôznych predikcií, pričom jedna nevyužíva klinické dáta, vyplýva hlavne z dôvodu nedostupnosti klinických dát pre zdravých pacientov a preto by nebolo možné takú predikciu vykonať.

Pri predikcií je dôležité dosiahnuť čo najlepšie výsledky, avšak pri predikcii klinického stavu treba brať do úvahy aj fakt, že predikcia tried nemusí mať rovnakú váhu. Presnosť, pokrytie a F1-skóre sú dôležité metriky na vyhodnotenie správnosti riešenia, avšak aby boli výsledky relevantné aj z hľadiska medicíny a biológie, je nutné ich správne interpretovať, keďže pri extrakcii črt vzniká veľké množstvo črt, ktoré sú bez vhodnej interpretácie pre používateľa nepoužiteľné.

Na vykonanie predikcií navrhujeme použiť dva modely a náhodný les a sieť Elastic a ich výsledky navzájom porovnať a vyhodnotiť. Lasso regularizáciu zahrnutú v sieti Elastic z dôvodu jej použitia autormi nástroja Citrus na predikciu klinického stavu [13] a náhodný les pre jeho robustnosť voči okrajovým hodnotám, pretrénovaniu a vysoko-korelovaným črtám a jeho dobré výsledky v porovnaní s inými modelmi [11, 23, 16]. Oba prístupy ponúkajú vhodnú interpretovateľnosť výsledkov.

### 3.6.1 Extrakcia črt z cytometrických dát

Po vykonaní redukcie cytometrických dát na základe hustoty, zhľukovania na identifikovania bunkových populácií a následnom upsampling-u máme všetky bunky zo vstupných dát priradené do bunkových populácií, resp. zhľukov. Z týchto zhľukov je možné extrahovať črty, ktoré sa delia na dva typy a to črty početnosti a črty hodnôt cytometrických znakov buniek v zhľukoch.

Črty početnosti reprezentujú podiel buniek v zhľuku ku všetkým bunkám a teda koľko percent zo všetkých buniek sa nachádza v danom zhľuku. Táto hodnota je normalizovaná na interval  $[0, 1]$ , aby zodpovedala ostatným hodnotám, ktoré sú taktiež normalizované (viď. 3.2)

a teda hodnota 1 znamená, že sa v danom zhluke nachádza 100% všetkých buniek z pôvodnej množiny a 0 reprezentuje 0% buniek z pôvodnej množiny.

Črty hodnôt cytometrických znakov predstavujú jednotlivé hodnoty buniek týchto znakov. Pre každý zhluk sa identifikujú konkrétne bunky, ktoré sa v danom zhluke nachádzajú. Pre každý cytometrický znak identifikovaných buniek sa vypočíta ich mediánová hodnota. Z dôsledku normalizácie dát na interval  $[0, 1]$  pred redukciou dát na základe hustoty a výpočte hustôt sú vypočítané mediánové hodnoty a teda hodnoty črt cytometrických znakov taktiež normalizované.

Ak sa zhlukovaním cytometrických dát identifikuje  $l$  bunkových populácií, resp. zhlukov a používa sa  $m$  cytometrických znakov, tak pre jeden zhluk sa extrahuje  $m$  črt hodnôt cytometrických znakov a jedna črta počtosti, čo je  $m + 1$  črt pre jeden zhluk a  $l \times (m + 1)$  črt pre  $l$  zhlukov. Z toho vyplýva, že pre jedného pacienta je možné extrahovať vektor črt o dĺžke  $l \times (m + 1)$ . Avšak črty sa extrahujú na báze pacientov a preto pre každého pacienta je možné extrahovať rovnako dlhý vektor črt, čo pri  $n$  pacientoch tvorí  $n \times l \times (m + 1)$  črt, ktoré sú reprezentované maticou o  $n$  riadkoch a  $l \times (m + 1)$  stĺpcoch, kde každý riadok reprezentuje jedného pacienta a stĺpce reprezentujú črty a teda konkrétne hodnoty črt pacientov. Reprezentácia matice črt pacientov je vyobrazená na schéme 3.4, kde  $f_{xy}$  predstavuje hodnotu  $y$ -tej črty pre  $x$ -tého pacienta.

$$\begin{bmatrix} f_{11} & f_{12} & \dots & f_{1l \times (m+1)} \\ f_{21} & f_{22} & \dots & f_{2l \times (m+1)} \\ \vdots & \vdots & \ddots & \vdots \\ f_{n1} & f_{n2} & \dots & f_{nl \times (m+1)} \end{bmatrix}_{n \times l \times (m+1)} \quad (3.4)$$

Po takto vykonanej extrakcii črt sú pre každého pacienta dostupné črty, ktoré je možné použiť pri predikcii.

### 3.6.2 Extrakcia črt z klinických dát

Klinické dáta sú biologické a medicínske dáta o pacientoch, ktoré boli zozbierané počas ich liečby a obsahujú dáta ako napríklad diagnóza, podstúpené liečby, hodnoty odberu krvi, odpoveď na jednotlivé liečby a mnoho ďalšieho. Pri extrakcii črt z klinických dát je potrebné vyriešenie viacerých problémov.

Aj napriek tomu, že klinické dáta obsahujú veľa potencionálnych klinických črt pre pacienta, tak sa v nich nachádza veľké množstvo chýbajúcich dát. Tieto chýbajúce dáta nie je možné dodatočne vytvoriť, lebo nemáme informácie o tom, v akom vzťahu sú prípadne črty k iným črtám a v akých hodnotách by sa chýbajúce dáta mali pohybovať, čo by ovplyvnilo proces predikcie a znehodnotilo výsledky.

Kategorické dáta v klinických dátach, ako napríklad podstúpená liečba alebo reakcia na liečbu, je potrebné transformovať na numerické reprezentácie. Taktiež dátumy, podľa ktorých sa párujú cytometrické dáta na klinické dáta, je nutné správne naformátovať.



Po extrakcii dát z klinických dát sa z nich vytvorí črty pre jednotlivých pacientov. Tieto črty je možné podľa unikátneho identifikátora spárovať s konkrétnymi pacientmi. Tieto črty je následne možné použiť v kombinácii s cytometrickými črtami na predikciu klinického stavu pacienta.

### 3.6.3 Predikcia použitím cytometrických črt

Z dôvodu nedostupnosti klinických dát pre zdravých pacientov sme sa po konzultácii s doménovými expertmi rozhodli vykonať predikcia klinického stavu pacienta iba na základe črt extrahovaných z cytometrických dát.

Pre každého pacienta vieme, aký je jeho stav. Stavy môžu byť:

- zdravý,
- choroba typu WM (walderstrom myelóm),
- choroba typu MM (mnohopočetný myelóm).

Tým vznikajú tri triedy, ktoré budeme predikovať. Pre každého pacienta extrahujeme črty z cytometrických dát a identifikujeme, do ktorej z troch tried patrí. Množinu dát rozdelíme na tréningovú a testovaciu množinu. Na tréningovej množine natrénujeme modely a na testovacej vyhodnotíme ich presnosť. Pre čo najpresnejšie vyhodnotenie výsledkov použijeme krížovú validáciu, kde sa celá množina dát rozdelí na  $k$  rovnomerných podmnožín a v každej z  $k$  iterácií sa použije iná z  $k$  podmnožín na vyhodnotenie. Predikčné modely použijeme náhodný les a lasso regularizáciu a výsledky oboch modelov porovnáme a vyhodnotíme.

### 3.6.4 Predikcia použitím klinických a cytometrických črt

Keďže nie sú dostupné klinické dáta pre zdravých pacientov, tak predikcia s použitím klinických dát bude zahŕňať iba chorých pacientov. Predikciu s využitím klinických dát sme konzultovali s doménovými expertmi, od ktorých máme dostupné klinická dáta, aby sme identifikovali, ktoré dáta, črty, vzťahy a vlastnosti v dátach sú pre nich najdôležitejšie.

Po konzultáciách sme navrhli, že sa nebude predikovať typ choroby, ale odozva jednotlivých pacientov na podstúpenú liečbu. Táto predikcia sa rozdelí na dve predikcie a to na predikciu pacientov s chorobou WM (walderstrom myelóm) a MM (mnohopočetný myelóm). Rozdelenie na dve predikcie je z dôvodu, ako aj odlišných klinických dát pre jednotlivé skupiny, tak hlavne pre to, že pre doménových expertov sú pre jednotlivé skupiny zaujímavé a dôležité odlišné črty a výstupy.

#### Predikcia odozvy na liečbu pacientov s chorobou WM

Pri pacientoch s chorobou WM je pre doménových expertov najdôležitejšia podstúpená liečba *ibrutinib*. Cieľom tejto predikcie je nájsť prediktívne črty a vzťahy v dátach, na základe ktorých

by sa dalo určiť, či má pacient predpoklad na pozitívnu alebo negatívnu odozvu na túto liečbu. Z tohto dôvodu sa bude predikovať práve odozva na liečbu.

Okrem liečby ibrutinib, ktorá je v dátach najpočetnejšia, podstúpili pacienti s chorobou WM aj iné liečby, ktoré však nie sú pre doménových expertov zaujímavé, a preto je potrebné tieto kategorické dáta reprezentovať numericky. Navrhujeme nasledovnú reprezentáciu, kde číslo 1 reprezentuje liečbu ibrutinib a číslo 0 všetky ostatné liečby.

Odozva na podstúpenú liečbu je kategorizovaná do štyroch hlavných kategórií a to VGPR (very good partial response - veľmi dobrá čiastočná odozva), PR (partial response - čiastočná odozva), MR (minor response - nižšia odozva) a SD (stable disease - bez odozvy, choroba pretrvávajúca). Kategórie odoziev, po konzultácii s doménovými expertmi, navrhujeme rozdeliť do dvoch skupín a to pozitívna odozva, skladajúca sa z odoziev VGPR a PR a negatívna odozva zahŕňajúca všetky ostatné kategórie. Jednotlivé skupiny je potrebné numericky reprezentovať a preto navrhujeme reprezentáciu, kde 1 reprezentuje skupinu pozitívnych odoziev na liečbu a 0 skupinu negatívnych odoziev.

Okrem typu podstúpenej liečby a odozvy na podstúpenú liečbu sú pre túto predikciu dôležité aj ďalšie dáta z klinických dát a tými sú hodnota *IgM*, ktorá reprezentuje bunky, ktoré sú najdôležitejšie na identifikovanie myelómového ochorenia a *ipss skóre*, ktoré bolo na základe ďalších klinických dát vypočítané podľa publikácie [4] a nám dodané doménovými expertmi. Toto skóre kategorizuje pacientov do troch skupín a preto navrhujeme jeho reprezentáciu numerickými hodnotami.

Jeden pacient mohol podstúpiť viacero liečení, ale cytometrické dáta boli odobraté iba raz. Preto je nevyhnutné pre cytometrické dáta pacientov nájsť najaktuálnejšiu liečbu, s ktorou odobraté cytometrické dáta súvisia. Pre cytometrické dáta a aj jednotlivé liečby sú dostupné dátumy ich odberu, resp. vykonania a na základe týchto dátumov treba prepojiť cytometrické dáta s liečbou. Následne je z hľadiska správnosti výsledkov potrebné použiť informácie o liečbe, ako typ liečby, odozva, *IgM* a pod., podľa najaktuálnejšej liečby.

Z dôvodu veľkého počtu chýbajúcich dát sme sa po konzultáciách s doménovými expertmi rozhodli použiť iba tri črty a jednu predikovanú hodnotu na základe klinických dát.

### **Predikcia odozvy na liečbu pacientov s chorobou MM**

Pri pacientoch s chorobou MM doménových expertov zaujíma hlavne podstúpená liečba *RVD* a teda, či sa dá nájsť prediktívna črta, prípadne vzťah v dátach, na základe ktorých by bolo možné určiť, či má pacient predispozíciu na pozitívnu alebo negatívnu odozvu na túto liečbu.

Pacienti s chorobou MM podstupujú okrem najpočetnejšej liečby *RVD* aj iné typy liečení, na ktoré sa však doménový experti nezameriavajú. Ide o kategorické dáta, ktoré treba reprezentovať numericky a preto navrhujeme reprezentáciu, kde číslo 1 reprezentuje chorobu *RVD* a 0 reprezentuje všetky ostatné liečby.

Odozva na podstúpenú liečbu pacientmi s chorobou MM je kategorizovaná do piatich hlavných

kategórií a to VGPR (very good partial response - veľmi dobrá čiastočná odozva), PR (partial response - čiastočná odozva), CR, MR (minor response - nižšia odozva) a SD (stable disease - bez odozvy, choroba pretrváva). Kategórie odoziev je vhodné rozdeliť do dvoch skupín podľa toho, či sa jedná o pozitívnu alebo negatívnu odozvu. Navrhujeme preto po konzultácii s doménovými expertmi odozvy VGPR, PR a CR zaradiť do skupiny pozitívnych odoziev a ostatné odozvy zaradiť do skupiny negatívnych odoziev. Následne navrhujeme numerickú reprezentáciu skupín odoziev tak, že číslo 1 bude reprezentovať skupinu pozitívnych odoziev a číslo 0 bude reprezentovať skupinu negatívnych odoziev.

Pre doménových expertov sú pre túto predikciu zaujímavé a dôležité aj iné údaje, ako len typ liečby a odozva na liečbu. Tieto údaje sú napríklad *IgL*, ktoré hovorí, či pre daného pacienta bola meraná hodnota *kappa* alebo *lambda*, a teda ktorú z týchto hodnôt použiť. Ďalej je pre liečbu pacientov s chorobou mnohopočetného myelómu špecifické, že prvá podstupená liečba je rozdelená na konsolidačnú a indukčnú fázu. Avšak pre niektorých pacientov nie sú dostupné dáta pre obe fázy prvej liečby a preto po konzultácii s doménovými expertmi navrhujeme prioritne použiť dáta z konsolidačnej fázy a ak nie sú pre túto fázu pre pacienta dáta dostupné, tak použiť dáta z indukčnej fázy.

Rovnako ako pri predikcii odozvy na liečbu pacientov s chorobou WM tak aj pri chorobe MM podstupujú pacienti viacero liečení, počas ktorých sú zbierané klinické dáta, ale cytometrické dáta sú zozbierané iba raz. Preto je nevyhnutné spárovať odobranie cytometrických dáta s najaktuálnejšou liečbou pre daný odber a extrahovať črty z klinických dát spárovanej liečby. Treba dbať na to, že ak sa jedná o prvú podstupenú liečbu, tak treba prioritne extrahovať črty z konsolidačnej fázy prvej liečby.

Z klinických dát je z hľadiska tejto predikcie potrebné extrahovať črty podstupenej liečby, odozvy na podstupenú liečbu a hodnotu *kappa*, resp. *lambda*. Extrakcia viacerých črt nie je možná z dôvodu veľkého množstva chýbajúcich dát.

### 3.6.5 Interpretácia výsledkov

Zmysluplná interpretácia výsledkov je nevyhnutná pre vyvodenie relevantných záverov predikcie a v našom prípade extrakciu biologických závislostí. Sieť Elastic aj náhodný les poskytujú dobrú interpretáciu výsledkov pomocou identifikovania prediktívnych črt [34, 23], avšak náhodný les môže mať problém správne identifikovať dôležité črty, ak sú tieto črty vzájomne silno korelované [17].

Na odstránenie problému pri identifikovaní dôležitých črt náhodným lesom navrhujeme vypočítať koreláciu medzi jednotlivými črtami. Navrhujeme použiť spearmanov korelačný koeficient na výpočet korelácií, lebo je vhodnejší aj na lineárne neseparovateľné dáta, ako napríklad pearsonov korelačný koeficient. Následne po vypočítaní korelácií medzi črtami odstrániť tie črty, ktoré sú silne korelované s aspoň jednou inou črtou.

Keďže nemáme dodatočné informácie o bunkových populáciách, napríklad do akých bunko-

vých populácií vo vyšších úrovniach hierarchie patria, a teda ktoré konkrétne bunkové populácie dané črty reprezentujú, tak nevieme jednoducho vyvodiť a extrahovať biologické závislosti z výsledkov. Tento krok zostáva na doménových expertov. Avšak pomocou výberu iba menšej podmnožiny najdôležitejších črt a vhodnou vizualizáciou týchto črt vieme dopomôcť k jednoduchšej práci s výsledkami.

## 3.7 Vizualizácia

Vizualizácia výsledkov je veľmi dôležitý krok, ktorý napomáha analýze dát. V nami navrhovanom riešení sú dva výsledky, ktoré je možné vizualizovať a tým dopomôcť ku analýze. Jednou z nich je vizualizácia bunkových populácií extrahovaných zhľukovaním buniek po procese redukcie dát na základe hustoty. Vizualizácia bunkových populácií napomáha porovnaniu jednotlivých pacientov medzi sebou a vyvodzovanie biologických záverov. Druhou vizualizáciou je vizualizovanie výsledkov predikcie a to konkrétne vizualizovanie hlavných prediktívnych črt na lepšiu interpretáciu výsledkov predikcie a nájdenie nových vzťahov v dátach.

Na vizualizovanie bunkových populácií sme zvolili stromovú vizualizáciu, lebo doménový experti, s ktorými sme spolupracovali, sú z nástroja SPADE [28] zvyknutý práve na stromovú vizualizáciu. Na vizualizáciu interpretácie výsledkov predikcie navrhujeme prístup využívajúci teplotné mapy.

### 3.7.1 Stromová vizualizácia bunkových populácií

Základom stromovej vizualizácie bunkových populácií je zobrazenie bunkových populácií ako vrcholov stromu, ktoré sú ofarbené na základe hodnôt cytometrických znakov obsiahnutých v konkrétnych bunkových populáciách.

Vrcholy stromu sú vytvorené na základe zhľukov získaných zhľukovaním zredukovaných cytometrických dát na základe hustoty. Zo všetkých buniek obsiahnutých v jednotlivých zhľukoch, sa vypočíta priemerná hodnota pre každý cytometrický znak. Ak sa používa  $m$  cytometrických znakov, tak každý zhľuk je definovaný vektorom obsahujúcim  $m$  hodnôt. Pri  $n$  extrahovaných zhľukoch vzniká matica zhľukov o veľkosti  $n \times m$ , ktorá obsahuje pre  $n$  zhľukov  $m$  extrahovaných priemerných hodnôt cytometrických znakov. Štruktúra matice zhľukov je naznačená na schéme 3.5.

$$\begin{bmatrix} c_{11} & c_{12} & \dots & c_{1m} \\ c_{21} & c_{22} & \dots & c_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n1} & c_{n2} & \dots & c_{nm} \end{bmatrix}_{n \times m} \quad (3.5)$$

Podľa matice zhľukov sa vypočíta matica príľahlosti na základe vzdialenosti medzi jednotlivými zhľukmi definovanými riadkami v matici zhľukov použitím manhattanskej vzdialenosti. Z matice

príľahlosti sa vytvorí ohodnotený plne prepojený neorientovaný graf s hranami ohodnotenými práve podľa vzdialeností z matice príľahlosti. Následne sa z plne prepojeného neorientovaného váhovaného grafu extrahuje minimálna kostra grafu použitím Primovho algoritmu pre váhované grafy. Takýmto postupom sa získa stromová štruktúra pospájaných zhlukov bunkových populácií na základe hodnôt buniek nachádzajúcich sa v jednotlivých zhlukoch.

Vrcholy stromovej štruktúry pospájaných zhlukov na základe hodnôt obsiahnutých buniek je možné rozmiestniť v priestore rôznymi algoritmami. My navrhujeme použiť Fruchterman-Reingold algoritmus, ktorý je založený na simulácii fyzikálnych síl, kde sa samotné vrcholy od seba odpudzujú a hranami príľahujú.

Následne keď máme rozmiestnenú štruktúru stromu poskladanú z redukovaných dát v priestore, tak na základe dát po upsampling-u sa vytvoria vizualizačné prvky pre strom. Veľkosť jednotlivých vrcholov v strome je úmerná počtu buniek v zhluku daného vrchola. Ako ďalší vizuálny prvok sa využíva farba. Jednotlivé vrcholy stromu sú ofarbené podľa priemerných hodnôt cytometrických znakov buniek obsiahnutých v daných vrcholoch.

Aby bola vizualizácia pridanou hodnotou je potrebné aby bola interaktívna a výskumník s ňou vedel jednoducho pracovať. Vizualizácia musí umožňovať vykonať základné operácie ako zoom a „drag & drop”, aby si výskumník vedel výsledný strom upraviť podľa predstáv. Okrem základných interakcií musí vizualizácia umožňovať výber viacerých vrcholov stromu, výber cytometrického znaku, podľa ktorého sa má strom ofarbiť, ako aj výber pacienta, podľa ktorého cytometrických dát má byť strom zostavený. Vizualizácia by taktiež mala poskytovať spätnú kontrolu cez dvojrozmerný bodový graf zobrazujúci dáta zvolených vrcholov stromu na dvoch vybratých cytometrických znakoch.

### **3.7.2 Vizualizácia výsledkov predikcie teplotnými mapami**

Ako bolo spomenuté vyššie (viď. 3.6.5), nedisponujeme doplňujúcimi informáciami ani znalosťami o bunkových populáciách, vďaka ktorým by sme vedeli jednoducho interpretovať výsledky predikcie a preto je potrebná vhodná vizualizácia výsledkov, ktorá pomôže výskumníkovi pri interpretácii výsledkov. Cieľom vizualizácie výsledkov predikcie je jednoduché vyvodenie záverov, ktoré bunkové populácie, prípadne cytometrické znaky v bunkových populáciách vplývajú na klinický stav pacienta. Dôležitosť jednotlivých črt by mohla byť dostupná ako ďalšia možnosť ofarbenia stromu (viď. 3.7.1), lenže tento postup by vyžadoval veľa vizualizácií rôznych stromov, respektíve zmeny voľby zobrazovaného pacienta a cytometrického znaku na vyvodenie záverov, nehovoriac o tom, že by si výskumník musel pamätať veľa medzikrokov. Na adresovanie tohto problému je potrebné použiť vizualizáciu, ktorá by zároveň ponúkala celkový pohľad na dáta, ako aj možnosť porovnania výsledkov predikcie medzi jednotlivými skupinami tried, resp. pacientov. Z tohto dôvodu navrhujeme využitie teplotných máp.

Teplotné mapy sú vhodné na vyriešenie opísaného problému, lebo poskytujú globálny pohľad na cele dáta a zároveň umožňujú aj porovnanie medzi jednotlivými triedami predik-

cie, respektíve pacientmi. Vytvorením teplotnej mapy podľa zhlukov a priemerných hodnôt cytometrických znakov daných zhlukov, poskytneme alternatívnu vizualizáciu ku stromovej vizualizácii, ktorá taktiež umožňuje identifikáciu bunkových populácií. Výberom vhodne veľkej podmnožiny najdôležitejších črt z hľadiska predikcie a ich následná vizualizácia teplotnou mapou na základe priemerných hodnôt pre jednotlivých pacientov, umožní výskumníkovi vidieť, na akých hodnotách sa predikčný model rozhodoval pri zaraďovaní do tried. Ak takúto teplotnú mapu rozdelíme podľa predikovaných tried pacientov, umožní to výskumníkovi jednoducho vidieť rozdiely v priemerných hodnotách črt, na základe ktorých sa model rozhodoval pri zaraďovaní do tried. S využitím takéhoto porovnania a teplotnej mapy zobrazujúcej globálny pohľad na dáta, bude výskumník schopný ľahšie extrahovať relevantné závery z výsledkov predikcie.

Ako dodatočná vizualizačná pomoc pri extrahovaní záverov z výsledkov by mohla pomôcť teplotná mapa, ako rozdiel medzi teplotnými mapami dvoch predikovaných tried. Ak sa pre každú triedu, pre každú najdôležitejšiu črtu spriemerujú hodnoty pre danú črtu všetkých pacientov v danej triede, vzniknú teplotné mapy s jednou hodnotou pre každú črtu naprieč všetkými pacientmi. Takéto teplotné mapy je následne možné od seba odpočítať, čo umožní vizualizovať iba rozdiely medzi teplotnými mapami, ktoré zodpovedajú rozdielom v črtách, na základe ktorých sa predikčný model rozhodoval.

### 3.8 Zhrnutie

Navrhli sme riešenie na efektívny výpočet hustoty buniek a následnú redukciu dát na základe hustoty aplikovaním iteratívneho procesu. Na vizualizovanie bunkových populácií získaných zhlučovacím algoritmom sme navrhli vizualizáciu stromovou štruktúrou. Ako ďalšie sme navrhli proces extrakcie črt z cytometrických dát a následnú predikciu s klasifikátormi náhodný les a sieť Elastic. Na uľahčenie interpretácie sme navrhli vizualizáciu teplotnými mapami.

Pri výpočte hustoty sa priestor rozdeľuje na základe hodnoty parametra  $\varepsilon$ , čím sa redukujú množiny potencionálne susedných buniek pre všetky bunky. Navrhnutá stromová indexová štruktúra umožňuje efektívnu reprezentáciu priestoru a vyhľadávanie množín buniek ako z časovej, tak aj z pamäťovej náročnosti. Navrhli sme paralelizáciu procesu výpočtu hustoty, ktorá podstatne urýchlí celý proces.

Na vylepšenie procesu redukcie dát na základe hustoty sme navrhli iteratívny proces, ktorý začína pri nízkej hodnote parametra  $\varepsilon$  a postupne ju upravuje aby sa proces približoval ku požadovanému výsledku a jednotlivé iterácie pracujú s dátami, ktoré boli zredukované v predchádzajúcej iterácii.

Na vizualizáciu bunkových populácií sme navrhli použiť stromovú štruktúru s rozmiestnenými vrcholmi v priestore podľa Fruchterman-Reingold algoritmu. Navrhli sme proces extrakcie črt z cytometrických dát a ich následne využitie pri predikcii klinického stavu klasifikátormi náhodný les a sieť Elastic. Proces interpretácie výsledkov predikcie a analýzy je veľmi dôležitý.

Navrhli sme použiť teplotné mapy na vizualizáciu najdôležitejších črt predikcie podľa určenia dôležitosti jednotlivými klasifikátormi.

Nami navrhnuté riešenie vyžaduje iba jeden vstupný parameter veľkosti výslednej množiny po redukcii dát na základe hustoty, čím sa minimalizuje možnosť zavedenie subjektivity do výsledkov.





# Kapitola 4

## Implementácia

Implementáciu návrhu sme rozdelili do troch projektov (i) výpočtový projekt; (ii) vizualizačný projekt; (iii) projekt analýzy. Výpočtový projekt zastrešuje hlavné, časovo náročné výpočtové operácie a teda výpočet hustoty, redukciu dát na základe hustoty a upsampling. Vizualizačný projekt implementuje väčšinu navrhovaných vizualizácií a to vizualizáciu bunkových populácií a výsledkov predikcie. Projekt analýzy je hlavný projekt, ktorý zabezpečuje celý proces analýzy cytometrických dát, spracovanie a prípravu dát pre ostatné balíky, ktoré zjednocuje a prepája.

### 4.1 Výpočtový projekt

Hlavná funkcionálna výpočtového projektu – výpočet hustoty, redukcia dát na základe hustoty a upsampling – je implementovaná v programovacom jazyku C++14 (C++ verzie 14). V projekte sa využíva viacero funkcionalít jazyka, ktoré boli pridané vo verzii 11 (C++11), respektíve 14, a hlavne *smart pointer*, ktorý umožňuje využiť obmedzený *garbage collector*, ktorý bol v týchto verziách pridaný. Zdrojové súbory sú súčasťou CMake<sup>1</sup> projektu, pomocou ktorého sa aj kompilujú.

V projekte sa využíva voľne dostupná knižnica *OpenMP*<sup>2</sup> na tvorbu paralelných programov, prácu s vláknami a synchronizáciu.

Na komunikáciu a prenos dát medzi jazykom R a jazykom C++ sme použili voľne dostupnú knižnicu *Rcpp*<sup>3</sup>, ktorá umožňuje vytvoriť R-balík<sup>4</sup>, ktorý slúži ako rozhranie medzi jazykom R a jazykom C++ zastrešujúci konverziu dátových štruktúr a volania funkcií.

Výpočtový projekt je vo výsledku Rcpp balík s CMake C++ balíkom vnútri, ktorého funkcionálnu je možné využiť pri analýze vykonanej v jazyku R.

---

<sup>1</sup>[www.cmake.org](http://www.cmake.org) - skupina nástrojov na kompiláciu, testovanie a tvorbu softvérových balíkov

<sup>2</sup>[www.openmp.org](http://www.openmp.org)

<sup>3</sup>[www.rcpp.org](http://www.rcpp.org)

<sup>4</sup>balík, pomocou ktorého sa zverejňujú knižnice v jazyku R

## 4.2 Vizualizačný projekt

Vizualizačný projekt zastrešuje väčšinu vizualizácií v návrhu a hlavne vizualizáciu bunkových populácií stromovou štruktúrou.

Pri vizualizácii bunkových populácií sa využíva Fruchterman-Reingold algoritmus na rozmiestnenie vrcholov a hrán v priestore, ktorý je implementovaný v jazyku C++ a prepojený s jazykom R použitím knižnice *Rcpp* a *Rcpp* balíka. Samotná vizualizácia stromovej štruktúry podľa rozmiestnenia vrcholov a hrán v priestore, ktorá zároveň poskytuje interakciu, je implementovaná v jazyku JavaScript s využitím voľne dostupnej knižnice *D3.js* <sup>5</sup>. Knižnica *D3.js* umožňuje implementovať vizualizáciu grafov ako SVG vektorovej grafiky a interakciu medzi jednotlivými prvkami grafu.

Na vizualizáciu výsledkov predikcie teplotnými mapami sa používa teplotná mapa *heatmap.2* z knižnice *gplots*.

Dáta na vizualizáciu sa predspracovávajú v jazyku R a vizualizačný projekt je zostavený ako *htmlwidget* <sup>6</sup> balík, ktorý sprostredkováva prepojenie medzi jazykom R a jazykom JavaScript a umožňuje zobrazenie vizualizácie na webe priamo z jazyka R a taktiež ponúka integráciu do R servera.

## 4.3 Projekt analýzy

Projekt analýzy zastrešuje celý proces analýzy cytometrických dát – predspracovanie dát, redukcia na základe hustoty, zhľukovanie, upsampling, predikcia a vizualizácia. Predspracováva dáta potrebné pre výpočtový a vizualizačný projekt a zastrešuje komunikáciu s nimi.

Na prácu s FCS súbormi a spracovanie cytometrických dát sa využíva knižnica *flowCore* [19]. Zhľukovací algoritmus sa používa aglomeračné zhľukovanie z knižnice *Rclusterpp*, ktorá implementuje efektívne, paralelné zhľukovacie algoritmy. Pri predikcií sa používa sieť Elastic implementovaná v knižnici *glmnet* a náhodný les z knižnice *randomForest*. V procese vizualizácie sa využíva knižnica *igraph* na zostavenie minimálnej kostry grafu ako aj knižnica *shiny* na vytvorenie R servera do ktorého je integrovaný vizualizačný projekt.

## 4.4 Zhrnutie

Implementácia nášho riešenia bola rozdelená do troch balíkov. Výpočtový balík zastrešuje všetky výpočtovo náročné operácie a je implementovaný v jazyku C++ s využitím knižnice *OpenMP* na prácu s vláknami. Na prepojenie výpočtového balíka s jazykom R bola použitá knižnica *Rcpp*.

---

<sup>5</sup>[www.d3js.org](http://www.d3js.org)

<sup>6</sup>[www.htmlwidget.org](http://www.htmlwidget.org)

Vizualizácia nášho riešenia je implementovaná vo vizualizačnom balíku v jazyku JavaScript využívajúci knižnicu *D3.js* na tvorbu grafov. V tomto balíku je taktiež implementovaná upravená verzia Fruchterman-Reingold algoritmu v jazyku C++. Na prepojenie balíka v jazyku JavaScript s jazykom R bola použitá knižnica *htmlwidgets*.

Hlavný balík, ktorý zoskupuje výpočtový aj vizualizačný balík je implementovaný v jazyku R. Funkcionalita hlavného balíka je spracovanie cytometrických dát, predikcia použitím klasifikátorov z knižníc *randomForest* a *glmnet* a zabezpečenie komunikácie a toku dát medzi ostatnými balíkmi.



# Kapitola 5

## Výsledky

Pri vyhodnocovaní výsledkov nami navrhnutého riešenia sme overovali všetky kroky riešenia a jednotlivé kroky sme vyhodnocovali použitím viacerých dátových množín.

Ako prvé sme overovali výsledky výpočtu hustoty porovnaním s výpočtom naivným prístupom. Následne sme overovali správnosť návrhu váhovanej hustoty na dvojrozmerných vygenerovaných dátach. Ako ďalšie sme vyhodnocovali správnosť redukcie dát na základe hustoty použitím vygenerovanej dvojrozmernej dátovej množiny. Po tom, čo sme zredukované dáta priradili do tried, sme overili správnosť priradenia odstránených dát v procese redukcie do tried zredukovaných dát na základe vzdialenosti medzi zredukovanými a odstránenými dátami. Po tom, ako sme overili správnosť výpočtu hustoty a redukcie dát na základe hustoty sme vyhodnocovali rýchlosť nami navrhnutého riešenia v porovnaní so softvérovým nástrojom SPADE [27] – najpoužívanejším softvérovým nástrojom na analýzu cytometrických dát využívajúci redukciu dát na základe hustoty.

Pokračovali sme overením výsledkov z biologického hľadiska. Vizualizovali sme výsledky analýzy cytometrických dát a porovnali sme ich s vizualizáciami výsledkov softvérového nástroja SPADE s použitím rovnakej dátovej množiny. Vizualizované výsledky sme overili na správnosť z biologického hľadiska s doménovými expertmi.

V poslednom rade sme vyhodnocovali presnosť predikcie. Predikciu sme overovali použitím dostupných cytometrických dát a na dátovej množine AML z FlowCAP-II súťaže. Ako prvé sme overovali predikciu klinického stavu pacienta (zdravý, choroba waldenstrom myelóm, choroba mnohopočetný myelóm) na základe črt extrahovaných z cytometrických dát pacientov. Následne sme tieto cytometrické črty spojili s klinickými dátami a predikovali odozvu na liečbu. Ako posledné sme použili dátovú množinu AML a porovnali výsledky predikcie s existujúcimi riešeniami.

Porovnania rýchlosti jednotlivých algoritmov boli kvôli objektívnosti vykonané na rovnakom stroji s procesorom Intel(R) Core(TM) i7-6700HQ CPU @2,60GHz (8 CPUs) s operačnou pamäťou 16GB a 64-bitovým operačným systémom.

## 5.1 Výpočet hustoty

Výpočet hustoty sme overovali na viacerých dátových množinách, ako aj na vygenerovaných, tak aj na cytometrických dátach. Ako prvé sme dali vypočítať pre každý bod / bunku klasickú hustotu, kde hustota bunky je rovná počtu jej susedných buniek, naivným spôsobom. Naivný spôsob vypočíta hustoty v čase  $O(\frac{n^2-n}{2})$ , keďže počíta vzdialenosti medzi všetkými bunkami, pričom medzi dvoma rovnakými bunkami počíta vzdialenosť iba raz a nepočíta vzdialenosť bunky samej so sebou. Potom sme vypočítali klasickú hustotu nami navrhovaným spôsobom a porovnali sme výsledky. Pri použití všetkých dátových množín boli výsledky výpočtu hustoty naivným a nami navrhovaným spôsobom rovnaké. Tým sa nám podarilo overiť správnosť výpočtu hustoty buniek.

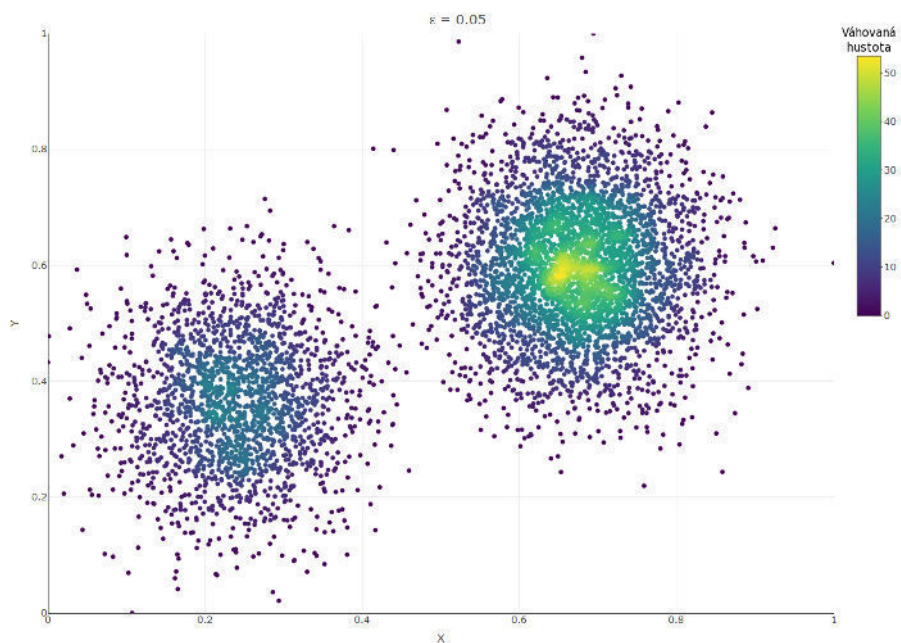
Rovnaký proces overovania, výpočtu hustoty naivným a nami navrhovaným spôsobom sme použili aj pri váhovanej hustote. Výsledky boli taktiež pre všetky použité dátové množiny zhodné. Na základe dosiahnutých výsledkov sme overili správnosť výpočtu aj váhovanej hustoty.

## 5.2 Váhovaná hustota

Overovali sme nami navrhnutú váhovanú hustotu (viď. časť 3.3) a schopnosť identifikácie hustých oblastí v priestore s jej aplikovaním. Vygenerovali sme si dvojrozmerné dáta reprezentujúce cytometrické dáta pozostávajúce zo 6000 bodov, ktoré sú rozdelené do dvoch oblastí s rôznou hustotou, kde redšia oblasť pozostáva z 2000 a hustejšia zo 4000 bodov. Vygenerované dáta sú zobrazené na obrázku 5.1. Na týchto dátach sme vypočítali hustoty jednotlivých bodov využitím váhovanej hustoty, pri hodnote parametra  $\varepsilon = 0,05$ , nami navrhovaným spôsobom. Následne sme vizualizovali vygenerované dáta s použitím farby, ktorá reprezentuje hodnotu váhovanej hustoty bodov. Z výsledku na obrázku 5.2 vidno, že body nachádzajúce sa v hustej oblasti majú vyššie hodnoty váhovanej hustoty, čím sa nám podarilo overiť správnosť navrhutej váhovanej hustoty, ako aj schopnosť identifikácie hustých oblastí s jej použitím.



Obr. 5.1: Vygenerované dvojrozmerné dáta obsahujúce dve oblasti s rozdielnou hustotou



Obr. 5.2: Identifikovanie hustých oblastí v dátach použitím váňovanej hustoty

### 5.3 Redukcia dát na základe hustoty

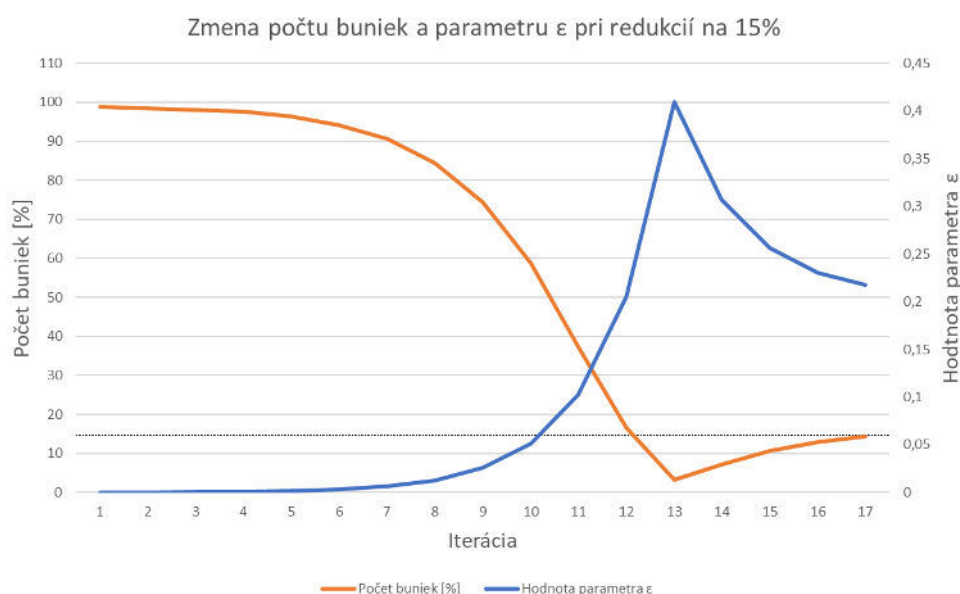
Pri vyhodnocovaní redukcie dát na základe hustoty sme overovali správnosť viacerých častí tohto procesu a to (i) iteratívny prístup redukcie dát na základe hustoty; (ii) výsledky redukcie

dát; (iii) odstránenie šumu a (iv) úpravu váh pri redukcii dát na základe hustoty.

### 5.3.1 Iteratívny prístup redukcie dát na základe hustoty

Pri overovaní iteratívneho prístupu redukcia dát na základe hustoty sme sledovali iteratívnu zmenu parametra  $\varepsilon$  a jej vplyv na veľkosť redukovaných dát. Na overenie sme použili viacero cytometrických dátových množín, ktoré sme redukovali iteratívnym prístupom. Na vizualizovanie sme zvolili dátovú množinu, ktorá obsahovala 287497 buniek a bola redukovaná na 15% veľkosti pôvodnej množiny.

Na grafe 5.3 je vyobrazený priebeh redukcie dát na základe hustoty na veľkosť 15% veľkosti pôvodnej množiny. Os  $x$  reprezentuje jednotlivé iterácie, v tomto prípade ich bolo vykonaných práve sedemnásť. Ľavá os  $y$  znázorňuje percentuálnu veľkosť redukovanej množiny ku veľkosti pôvodnej množiny po vykonaní konkrétnej iterácie a pravá os  $y$  hovorí o hodnote parametra  $\varepsilon$  použitom v danej iterácii. Prerušovaná čiara indikuje výsledný percentuálny počet buniek, ktorý sa má v procese redukcie dosiahnuť. Z grafu vidno vplyv zvyšovania hodnoty parametra  $\varepsilon$  na znižovanie veľkosti množiny buniek, ako aj vplyv znižovania hodnoty parametra  $\varepsilon$  na zvyšovanie veľkosti množiny po tom, čo bola v iterácii 13 množina buniek zredukovaná na základe hodnoty parametra  $\varepsilon$  príliš.



Obr. 5.3: Zmena počtu buniek a hodnoty parametra  $\varepsilon$  pri iteratívnom prístupe redukcie dát na základe hustoty

Vizualizovaním a vyhodnotením výsledkov priebehu iteratívneho prístupu procesu redukcie dát na základe hustoty sme overili vhodnosť použitia iteratívneho prístupu, ktorý postupne upravuje hodnoty parametra  $\varepsilon$  na dosiahnutie výsledku.

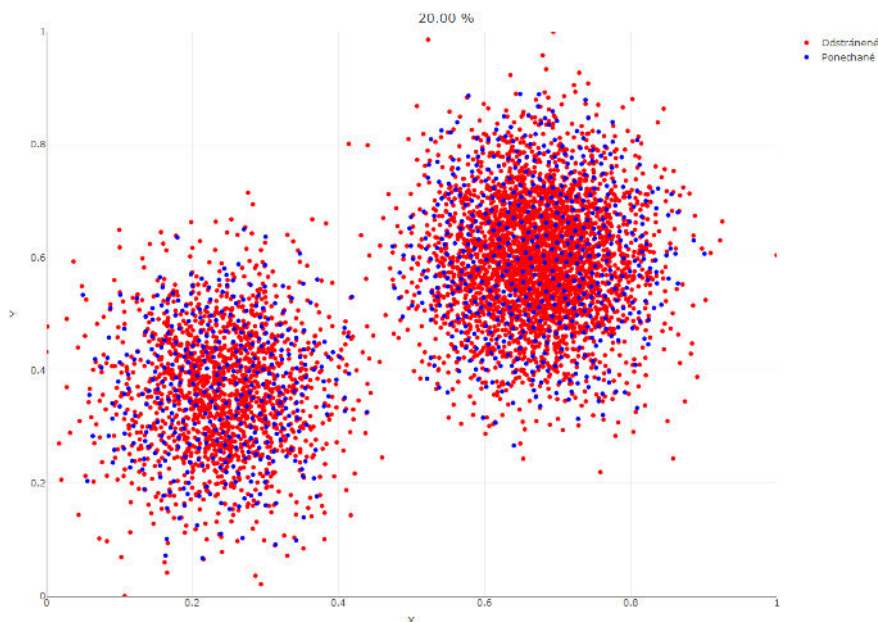


### 5.3.2 Výsledky redukcie dát na základe hustoty

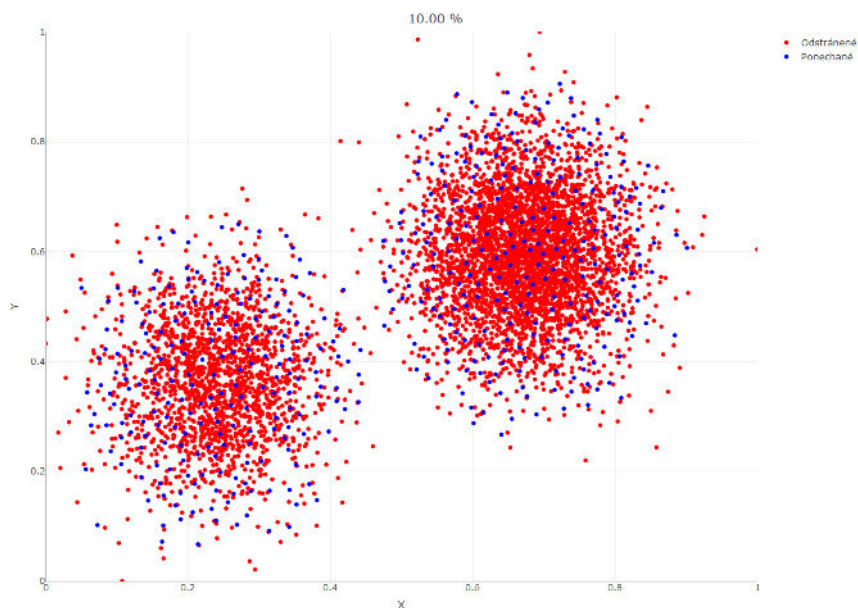
Na overenie a jednoduchšie vizuálne vyhodnotenie výsledkov redukcie dát na základe hustoty sme použili vygenerované dáta použité na overenie váhovanej hustoty vyobrazené na obrázku 5.1. Dáta sme nami navrhnutým prístupom zredukovali na základe hustoty na 20%, 10% a 5% veľkosti pôvodnej množiny. Výsledky redukcií sú vyobrazené na obrázkoch 5.4 až 5.6.

Vo vizualizáciách červené body reprezentujú bunky, ktoré boli pri redukcií dát na základe hustoty odstránené a naopak modré body reprezentujú bunky vybraté ako reprezentatívne a teda ponechané v procese redukcie. Z vizualizácií vidno, ako sú z hustých oblastí na základe hustoty vyberané reprezentatívne bunky, ktorých susedné bunky sú odstraňované, čím sa postupne vyrovnáva hustota v priestore. Taktiež vidno, ako sa postupne klesajúcou hodnotou výslednej veľkosti množiny (20% – 5%) zachováva proces redukcie a vyrovnávanie hustoty v priestore. Pri zredukovaných dátach na 5% veľkosti pôvodnej množiny bola vo výslednej množine pôvodne hustá oblasť reprezentovaná 1,98% bodmi z pôvodných dát a pôvodne riedka oblasť bola vo výslednej množine zastúpená 1,78% bodov z pôvodnej množiny. Vo výslednej množine nebolo presne 5% bodov z pôvodnej množiny, lebo bol v procese redukcie aplikovaný krok odstránenia šumu.

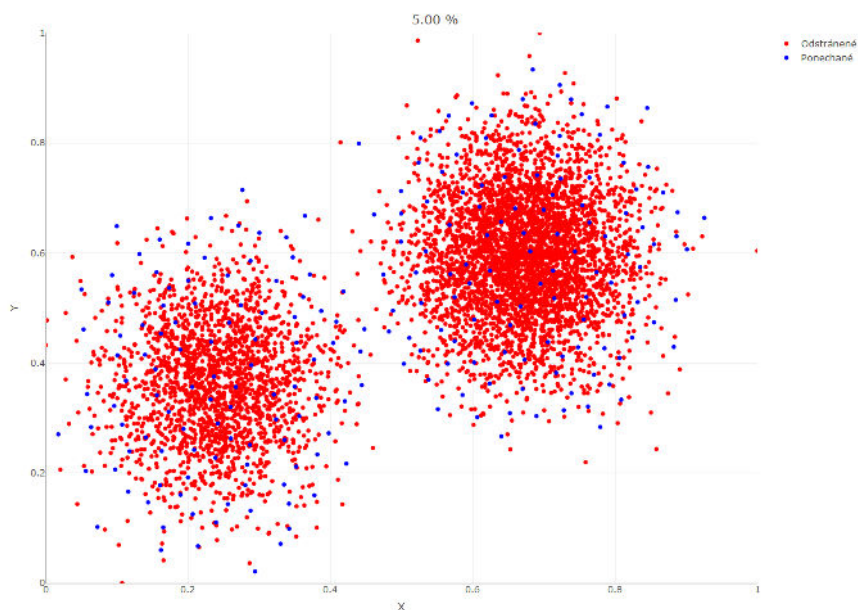
Na základe vizualizácií výsledkov redukcie dát na základe hustoty a číselnému vyhodnoteniu veľkosti množín v zredukovaných dátach sme overili správnosť nami navrhnutého prístupu a jeho schopnosť zredukovať dáta v priestore tak, aby vo výsledku bola hustota v priestore vyrovnaná.



Obr. 5.4: Vygenerované dvojrozmerné dáta zredukované na základe hustoty na 20% veľkosti pôvodnej množiny



Obr. 5.5: Vygenerované dvojrozmerné dáta zredukované na základe hustoty na 10% veľkosti pôvodnej množiny



Obr. 5.6: Vygenerované dvojrozmerné dáta zredukované na základe hustoty na 5% veľkosti pôvodnej množiny

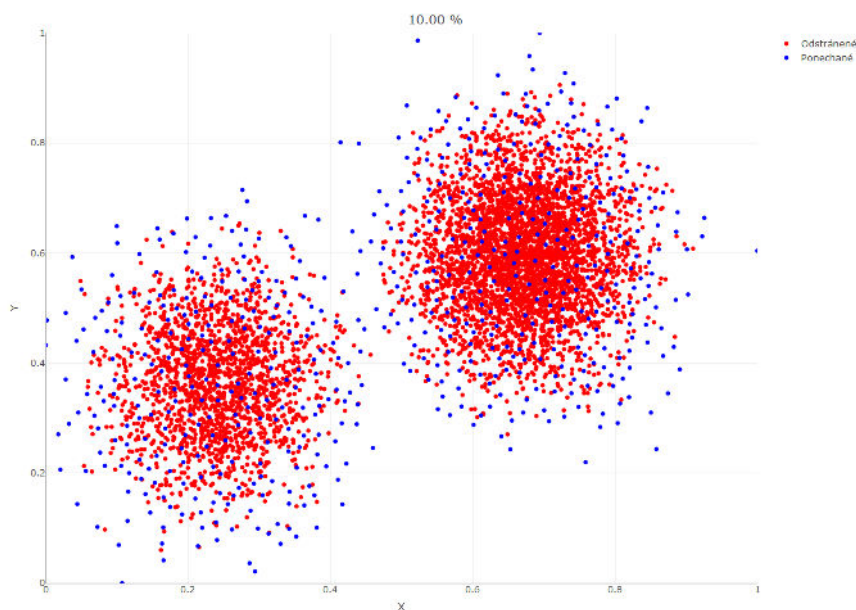
### 5.3.3 Odstránenie šumu v procese redukcie dát na základe hustoty

Pri vyhodnocovaní výsledkov redukcie dát na základe hustoty bol v procese redukcie aplikovaný krok odstránenia šumu. Pre porovnanie, na obrázku 5.7 je vizualizácia rovnakých dát pri redukcii na 10% veľkosti pôvodnej množiny, rovnako ako na obrázku 5.5, ale bez aplikovania kroku

odstránenia šumu.

Z porovnaní spomínaných dvoch vizualizácií vidno, že bez aplikovania kroku odstránenia šumu sa vo výsledných dátach ponechaných, reprezentatívnych buniek nachádzajú aj také bunky, ktoré sú viditeľne šumové dáta.

Porovnaním výsledkov s aplikovaným krokom odstránenia šumu a bez sa nám podarilo overiť správnosť fungovania nami navrhnutého spôsobu identifikovania okrajových hodnôt, ktoré reprezentujú šumové dáta a ich následne odstránenie v procese redukcie dát na základe hustoty.



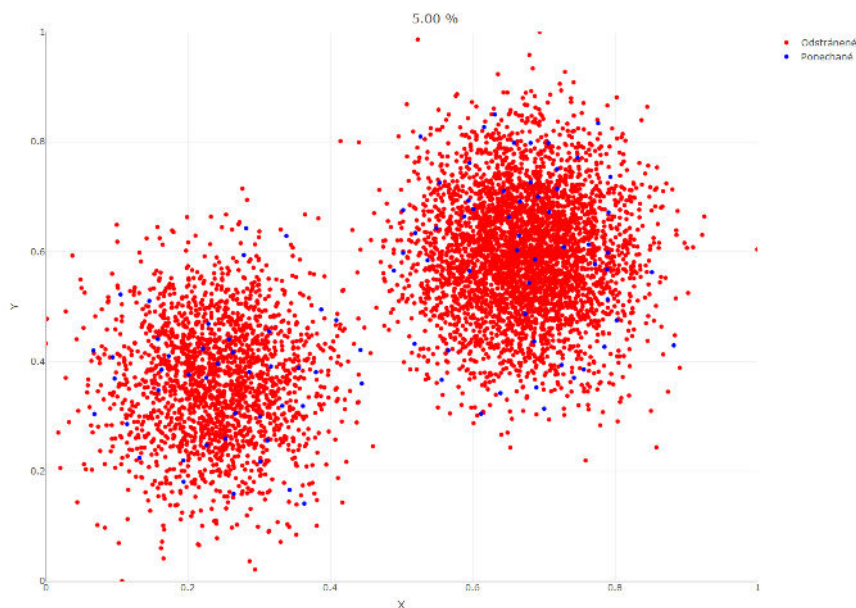
Obr. 5.7: Vygenerované dvojrozmerné dáta zredukované na základe hustoty na 10% veľkosti pôvodnej množiny bez aplikovania odstránenie šumu

### 5.3.4 Úprava váh v procese redukcie dát na základe hustoty

Pri vyhodnocovaní výsledkov redukcie dát na základe hustoty bol v procese aplikovaný krok úpravy váh. Na obrázku 5.8 je vyobrazený výsledok redukcie dát na výslednú veľkosť množiny 5% veľkosti pôvodnej množiny bez aplikovania kroku úpravy váh. Pre porovnanie, na obrázku 5.6 sú vyobrazené výsledky s rovnakou výstupnou veľkosťou množiny, avšak s aplikovaním kroku úpravy váh.

Zo samotných výsledkov redukcie dát na základe hustoty bez aplikovania kroku úpravy váh vyobrazených na obrázku 5.8 vidno, že krok úpravy váh je nevyhnutný na dosiahnutie správnych výsledkov. Bez aplikovania tohto kroku sú v procese redukcia odstránené aj reprezentatívne bunky, čo má za dôsledok vznik „dier“ vo výsledkom zredukovanom priestore.

Z vizualizácií sa nám podarilo overiť správne aplikovanie nami navrhnutého kroku úpravy váh, ako aj jeho nevyhnutnosť pre dosiahnutie správnych výsledkov pri redukcii dát na základe hustoty.



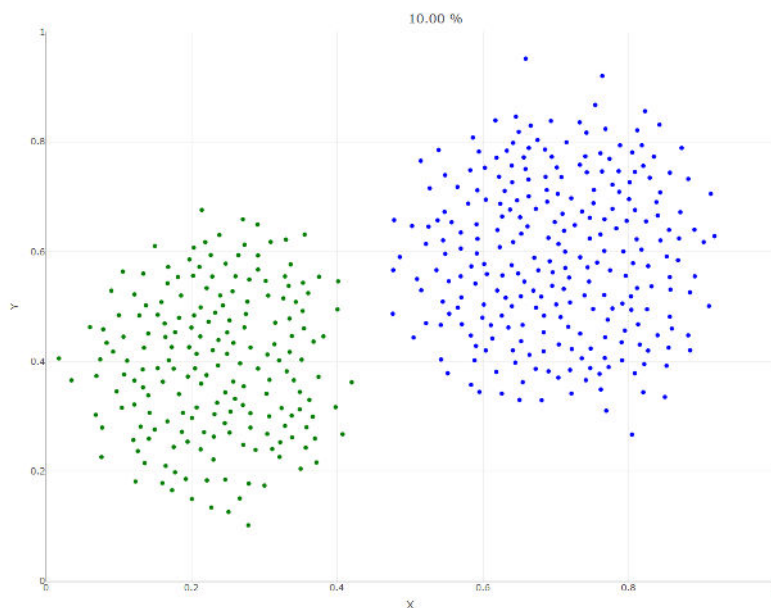
Obr. 5.8: Vygenerované dvojrozmerné dáta zredukované na základe hustoty na 5% veľkosti pôvodnej množiny bez aplikovania kroku úpravy váh

## 5.4 Upsampling

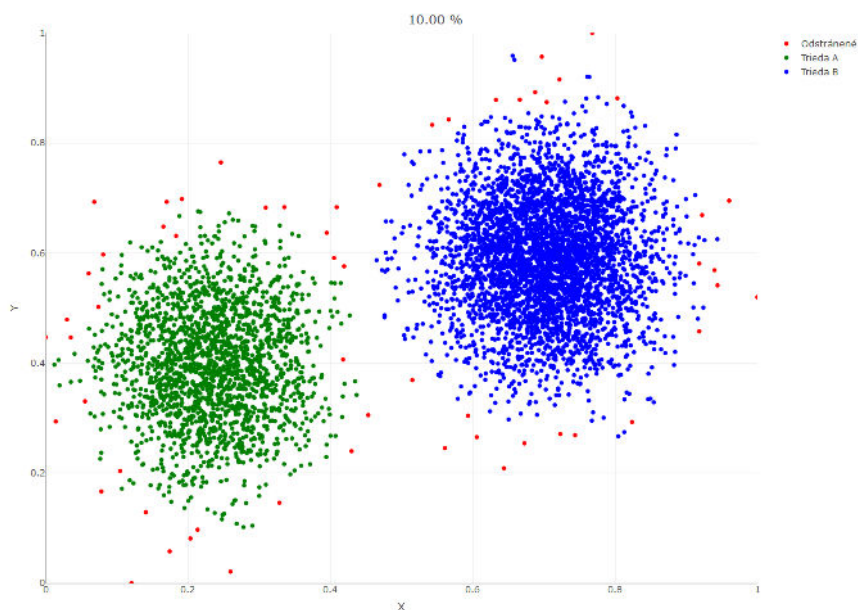
Upsampling je proces priradenie buniek, ktoré boli v procese redukcie dát na základe hustoty odstránené do tried priradením bunkám, ktoré boli v procese redukcie vybraté ako reprezentatívne, na základe podobnosti medzi odstránenými a reprezentatívnymi bunkami. Podobnosť dvoch buniek závisí od ich vzdialenosti v priestore.

Ak po redukcií vygenerovaných dvojrozmerných dát po redukcií na základe hustoty zaradíme reprezentatívne bunky do dvoch tried, vznikne situácia vyobrazená na obrázku 5.9. Na takýchto dátach sme následne vykonali upsampling, čím sme zaradili všetky bunky z pôvodnej množiny do daných dvoch tried a výsledok sme vizualizovali na obrázku 5.10. Ako vidno na vizualizácii, v procese upsampling-u je taktiež aplikovaný krok odstránenia šumu, ktorý identifikuje a odstráni okrajové, šumové dáta.

Pomocou vizualizácií sa nám podarilo overiť správne fungovanie nami navrhnutého procesu upsampling.



Obr. 5.9: Reprezentatívne bunky získane redukciou dát na základe hustoty vygenerovaných dvojrozmerných dát zaradené do dvoch tried



Obr. 5.10: Zaradenie odstránených buniek do tried na základe podobností s reprezentatívnymi bunkami

## 5.5 Porovnanie rýchlostí

Nami navrhnuté riešenie sme porovnávali v rýchlosti vykonávania výpočtu hustoty buniek a redukcie dát na základe hustoty s deterministickou verziou softvérového nástroja SPADE [27] implementovaného v jazyku matlab. Na porovnanie sme použili viacero dátových množín,



	~ 81000	~ 248000	~ 382000	~ 491000	~ 624000
SPADE	0,68 min	7,65 min	16,88 min	26,36 min	38 min
densamp	0,67 min	3,35 min	3,17 min	21,36 min	18 min

Tabuľka 5.1: Porovnanie potrebného výpočtového času na výpočet hustoty buniek

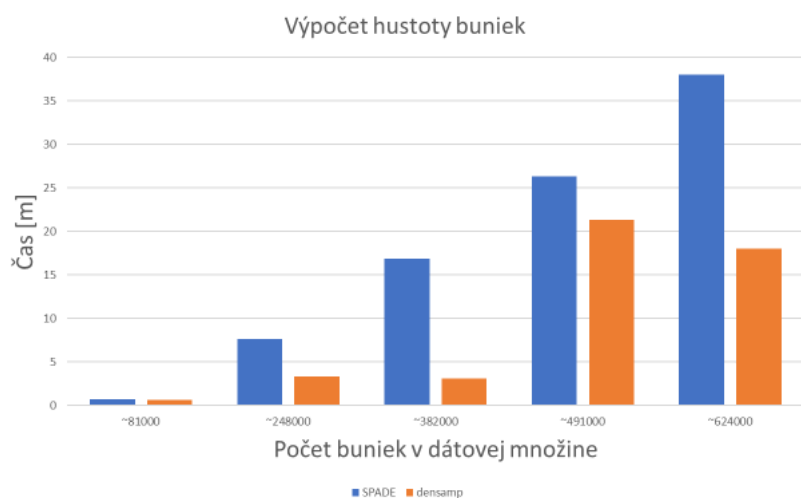
pričom najmenšia z nich obsahovala okolo 81000 buniek, najväčšia okolo 710000 a všetky obsahovali 13 parametrov – cytometrických znakov. Keďže nami navrhnuté riešenie pracuje s dátami normalizovanými na interval  $[0, 1]$ , tak sme pre zachovanie objektívnosti výsledkov všetky dáta normalizovali, aby nástroj SPADE vykonával procesy nad rovnakými dátami.

Pri overovaní výsledkov z biologického hľadiska sme objavili, že pre dosiahnutie biologicky správnych výsledkov je nevyhnutné cytometrické dáta pred analýzou transformovať. My sme zvolili inverznú hyperbolickú sínusovú transformáciu ( $\arcsinh$ , viď. časť 3.2.4). Táto transformácia ovplyvní rozmiestnenie buniek v priestore, čo môže ovplyvniť výpočtové časy nami navrhnutého riešenia. Preto pri porovnaní rýchlostí redukcie dát uvádzame tri hodnoty a to čas potrebný na vykonanie daného procesu softvérovým nástrojom SPADE, nami navrhnutým prístupom po aplikovaní transformácie  $\arcsinh$  a bez aplikovania  $\arcsinh$  transformácie. SPADE nie je ovplyvnený rozmiestnením buniek v priestore a preto u neho uvádzame iba jednu hodnotu.

### 5.5.1 Porovnanie rýchlosti výpočtu hustoty buniek

Pri procese výpočtu hustoty je pre nami navrhnutý prístup a aj pre softvérový nástroj SPADE potrebný parameter  $\varepsilon$ . Aby boli výsledky dosiahnuté za rovnakých podmienok, tak sme nástrojom SPADE vypočítali hodnotu  $\varepsilon$ , s ktorou následne boli vykonané výpočty hustôt.

Výsledné časy potrebné na výpočet hustoty buniek jednotlivých dátových množín sú zobrazené grafom 5.11 a presné hodnoty sú v tabuľke 5.1.



Obr. 5.11: Porovnanie času potrebného na vykonanie výpočtu hustoty softvérovým nástrojom SPADE a nami navrhnutým prístupom

	~ 81000	~ 248000	~ 346000	~ 421000	~ 675000	~ 710000
SPADE	0,94 min	9,73 min	15,78 min	20,20 min	60,53 min	63,28 min
densamp (+ arcsinh)	0,61 min	2,14 min	4,04 min	3,41 min	12,37 min	13,42 min
densamp	0,16 min	0,69 min	0,91 min	3,695 min	1,39 min	2,072 min

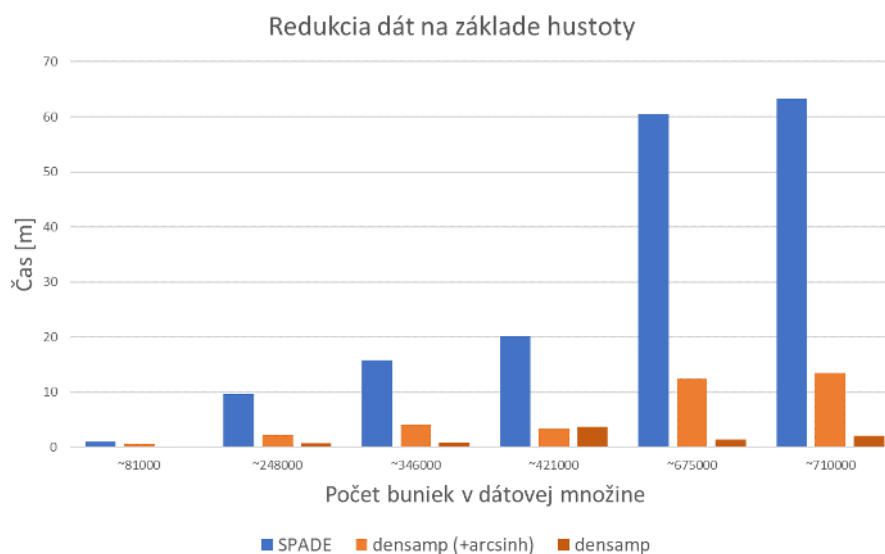
Tabuľka 5.2: Porovnanie potrebného výpočtového času na redukciu dát na základe hustoty

Z výsledkov vidno, že rozdiely na menších dátových množinách sú nízke a zanedbateľné. Pri väčších dátových množinách je vylepšenie nami navrhnutým prístupom zreteľnejšie, pričom dosahuje niekoľko násobné zrýchlenie. Porovnaním výsledkov sa nám podarilo overiť zlepšenie používaného prístupu softvérového nástroja SPADE, ako aj použiteľnosť na veľkých dátových množinách.

Podľa grafu 5.11 vidno, že potrebný čas na výpočet hustoty nami navrhnutým riešením nesleduje žiadnu krivku, lebo ako bolo opísané v časti 3.2.4, nami navrhnuté riešenie je z časti závislé od rozloženia dát v priestore.

### 5.5.2 Porovnanie rýchlosti redukcie dát na základe hustoty

Pri procesoch redukcie dát na základe hustoty bola výsledná hodnota veľkosti výslednej množiny v SPADE aj v nami navrhnutom prístupe zvolená na hodnotu 10% veľkosti pôvodnej množiny. Následne sa oboma prístupmi vykonala redukcia dát na základe hustoty na jednotlivých dátových množinách a výsledne výpočtové časy sa porovnali. Graf 5.12 zobrazuje porovnanie výpočtových časov potrebných na vykonanie redukcií a tabuľka 5.2 obsahuje presné hodnoty výpočtových časov.



Obr. 5.12: Porovnanie času potrebného na vykonanie redukcie dát na základe hustoty softvérovým nástrojom SPADE a nami navrhnutým prístupom

Softvérový nástroj SPADE pri redukcii dát na základe hustoty najskôr vypočíta hodnotu parametra  $\varepsilon$ , podľa ktorého následne vykoná výpočet hustoty a redukcii dát na základe hustoty. Z tohto dôvodu sa výpočtový čas potrebný na vykonanie redukcie nástrojom SPADE skladá z výpočtového času potrebného pre vykonanie výpočtu hustoty s konkrétnou hodnotou  $\varepsilon$  a výpočtového času potrebného na vykonanie redukcie.

Nami navrhovaný prístup na redukcii dát využíva nami navrhnutý iteratívny prístup, vďaka ktorému nie je potrebné počítať hustotu pre jednu, relatívne vysokú, hodnotu parametra  $\varepsilon$  nad veľkou dátovou množinou (ako boli porovnané výpočtové časy v časti 5.5.1), ale hodnota parametra  $\varepsilon$  sa postupne zvyšuje, zatiaľ čo sa veľkosť dátovej množiny znižuje a keď sa dosiahnú relatívne vysoké hodnoty parametra  $\varepsilon$ , tak je už dátová množina podstatne zredukovaná, čo urýchľuje celý proces. Tento jav je znázornený na grafe 5.3.

Využitím iteratívneho prístupu ku redukcii dát na základe hustoty s efektívnym výpočtom hustoty buniek sa nám podarilo výrazne, niekoľkonásobne zrýchliť tento proces, čo má za dôsledok celkové zrýchlenie analýzy cytometrických dát.

## 5.6 Overenie biologických výsledkov

Niekoľkonásobné zrýchlenie redukcie dát na základe hustoty by bolo nepodstatné, ak by výsledky neboli správne aj z biologického hľadiska. Vizualizovali sme v stromovej štruktúre bunkové populácie extrahované redukciiu dát a zhľukovaním a tieto vizualizácie sme porovnali s vizualizáciami bunkových populácií extrahovaných z rovnakých dát softvérovým nástrojom SPADE. Okrem porovnania vizualizácií sme vizualizácie bunkových populácií overili aj s doménovými expertmi.

Bunkové populácie sa vo vizualizáciách vyhľadávajú na základe farieb jednotlivých vrcholov stromu, ktorá reprezentuje hodnoty zvoleného cytometrického znaku v danom vrchole. Červené ofarbenie hovorí o vysokých hodnotách – expresiách a modré ofarbenie o nízkych. Bunkové populácie sa vyhľadávajú práve podľa vysokých expresií.

Od doménových expertov máme viacero informácií, ako vyhodnocovať správnosť biologických výsledkov. Napríklad cytometrický znak CD45 je veľmi heterogénny a teda by sa mal vyskytovať takmer vo všetkých bunkových populáciách a teda aj v takmer všetkých vrcholoch stromu. Cytometrický znak CD38 ma podobné vlastnosti ako CD45, ale je o niečo menej heterogénny. Naopak znak CD34 je veľmi homogénny, čiže sa nachádza iba v konkrétnych bunkových populáciách a v strome vrcholy reprezentujúce tieto bunkové populácie sú veľmi dobre oddeliteľné od ostatných. Vysoké hodnoty znaku IgM sú typické pre pacientov s chorobou waldenstom myelóm a teda v stromoch zostavených z pacientov s touto chorobou sa vyskytuje väčší počet vrcholov s vysokou expresiou tohto znaku ako v stromoch zostavených z dát zdravých pacientov. Cytometrické znaky IgD a IgA sú typické pre rozdielne bunkové populácie a preto sa aj v strome vysoké expresie týchto znakov nachádzajú na odlišných vrcholoch, pričom vrcholov

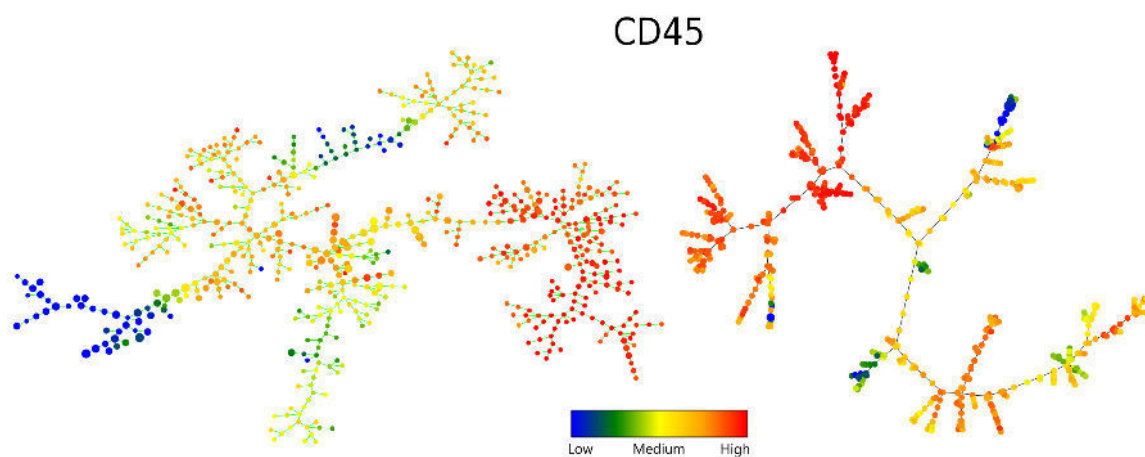


s vysokou expresiou znaku IgD má byť viac. Vysoké hodnoty expresí znakov CD19 a CD20 sú špecifické pre rovnaké bunkové populácie a preto sa aj v strome nachádzajú na rovnakých vrcholoch. Cytometrické znaky CD22 a CD27 nie sú reprezentatívne iba pre konkrétne bunkové populácie, ale nachádzajú sa vo viacerých a preto nie sú dobre oddeliteľné ani v strome.

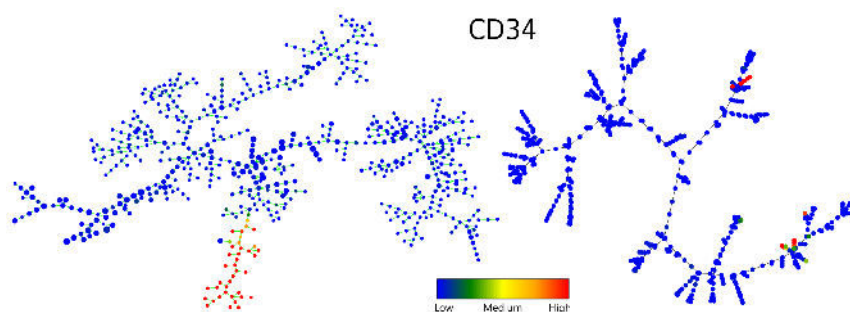
Na základe týchto a ďalších informácií sme vyhodnocovali správnosť výsledkov z biologického hľadiska. Okrem toho sme dali overiť výsledky aj doménovým expertom, ktorí potvrdili ich správnosť.

Na porovnanie s nástrojom SPADE sme vizualizovali strom zostavený z cytometrických dát pacienta s chorobou waldenstrom myelóm a aplikovali na ne farbu podľa cytometrických znakov CD45, CD34. Výsledné vizualizácie sú na obrázkoch 5.13 a 5.14, kde na ľavej strane je strom zostavený algoritmom nástroja SPADE a na pravej strane je strom zostavený nami navrhnutým algoritmom. Na porovnaníach vidno konzistentné výsledky medzi nástrojom SPADE a nami navrhnutým riešením. Okrem toho vizualizácie stromom zobrazujú vysokú heterogenitu znaku CD45 a dobrú oddeliteľnosť znaku CD34. Porovnania zvyšných jedenástich cytometrických znakov použitých pri analýze sú v prílohe D.1.

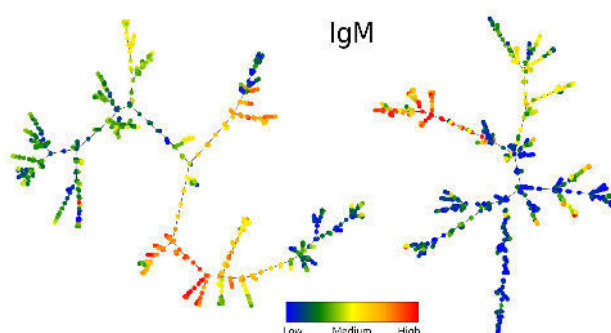
Na porovnanie znaku IgM, ktorý má byť zastúpený väčším počtom vysokých expresí práve u pacientov s ochorením waldenstrom myelóm sme zostavili strom z cytometrických dát pacienta s týmto ochorením a z cytometrických dát zdravého pacienta a aplikovali sme ofarbenie podľa znaku IgM. Na vizualizácii 5.15 je na ľavej strane strom chorého pacienta a na pravej strane strom zdravého. Z porovnania vidno konzistentné výsledky s biologickými predpokladmi.



Obr. 5.13: Porovnanie vizualizácie bunkových populácií podľa cytometrického znaku CD45



Obr. 5.14: Porovnanie vizualizácie bunkových populácií podľa cytometrického znaku CD34

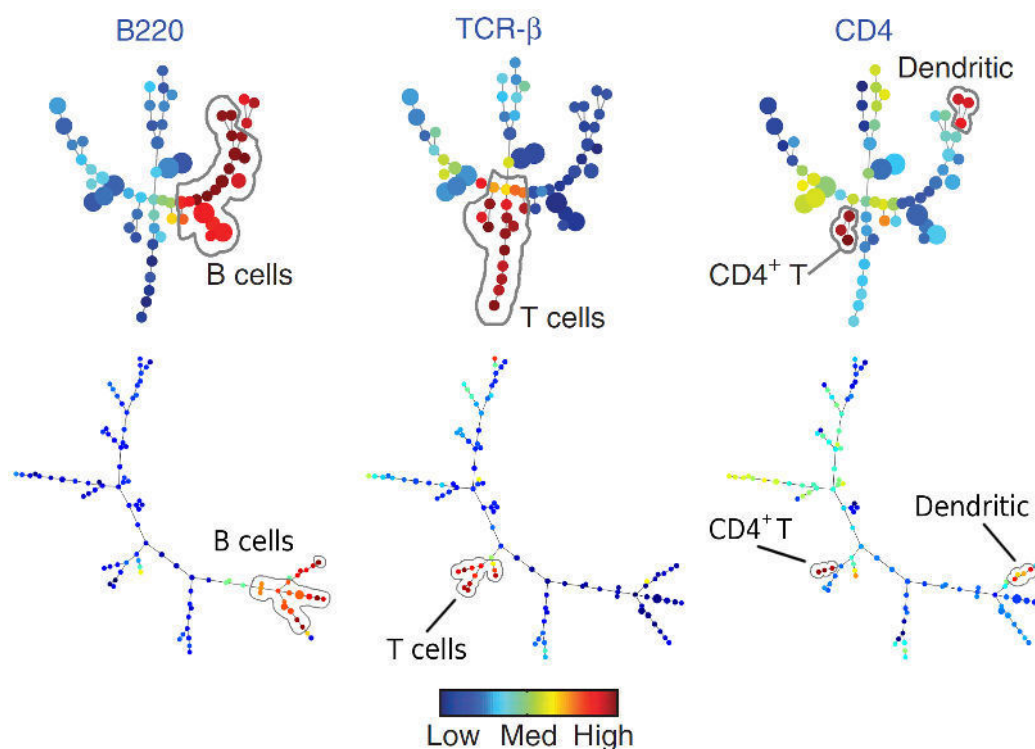


Obr. 5.15: Porovnanie vizualizácie bunkových populácií pacienta s chorobou waldenstrom myelóm a zdravého pacienta podľa cytometrického znaku IgM

### 5.6.1 Porovnanie na kostnej dreni myši

V kostnej dreni myši sú známe bunkové populácie, ktoré sa tam nachádzajú, ako aj ich hierarchia, ktorá je zobrazená na obrázku 3.7 v návrhu. Ako sme písali v analýze softvérového nástroja SPADE (viď. časť 2.4.1), pri analýze cytometrických dát kostnej drene myši sa im podarilo identifikovať bunkovú populáciu dendritických buniek, ktoré neboli manuálnym gatingom identifikované. Dendritické bunky sú definované ako  $\text{TCR-}\beta^- \text{ B220}^+ \text{ CD4}^+$ .

Cytometrické dáta kostnej drene myši sme spracovali použitím ôsmich cytometrických znakov c-kit, Sca-1, CD150, CD11b, B220, TCR-B, CD4 a CD8. Na dáta sme aplikovali arcsinh transformáciu s hodnotou parametra kofaktor 150 a zredukovali sme ich na základe hustoty na 10% pôvodnej množiny. Zredukováné dáta sme zhukovacím algoritmom rozdelili do 100 zhukov, vykonali upsampling a vizualizovali výsledok obrázkom 5.16. Horné tri stromové vizualizácie boli vytvorené nástrojom SPADE a spodné tri našim riešením. Ako vidno na vizualizácii, tak isto ako SPADE, tak aj nami navrhnuté riešenie dokázalo identifikovať bunkovú populáciu dendritických buniek, ktoré manuálnym gatingom neboli identifikované. Porovnanie vizualizácií podľa hodnôt ostatných cytometrických znakov je v prílohe D.2.



Obr. 5.16: Identifikovanie bunkovej populácie dendritických buniek

## 5.7 Predikcia klinického stavu pacienta

Presnosť predikcie sme vyhodnocovali na viacerých dátových množinách použitím náhodného lesa a siete Elastic (viď. časti 2.6.4 a 2.6.5). Ako prvé sme vyhodnocovali predikciu klinického stavu pacienta na základe črt extrahovaných z cytometrických dát. Následne sme k cytometrickým črtám pridali črty extrahované z klinických dát a predikovali sme odozvu pacienta na podstúpenú liečbu. Nakoniec sme vyhodnotili predikciu klinického stavu pacienta na dátovej množine zo súťaže FlowCAP-II a porovnali sme naše výsledky s výsledkami účastníkov.

Množiny dát sme rozdeľovali na trénovacie a testovacie množiny v pomere 80 : 20, pričom kvôli veľkému nepomeru tried v dátach sme rozdelenie vykonali takým spôsobom, aby v testovacej množine bola každá trieda reprezentovaná práve dvadsiatimi percentami z celkovej reprezentácie danej triedy.

### 5.7.1 Predikcia z cytometrických dát

Dáta sú rozdelené podľa panelov na dve skupiny P2 a P3. Na dátach z každého panelu jednotlivo sme vykonali predikciu. Začali sme panelom P2, z cytometrických dát sme extrahovali črty a vykonali klasifikáciu do troch tried – zdravý, s chorobou waldenstrom myelóm (WM) a s chorobou mnohopočetný myelóm (MM). S použitím týchto informácií sme vykonali predikciu. Viac informácií k predikcii a extrakcii črt je v častiach 3.6.3 a 3.6.1.

Predikcia	Referencia		
	Zdravý	WM	MM
	Zdravý	0	0
	WM	0	11
	MM	3	1
			42

Tabuľka 5.3: Matica zámen predikcie náhodným lesom na dátach panela P2

	Presnosť	Pokrytie	F1-skóre
Zdravý	—	0	—
WM	1,00	0,92	0,96
MM	0,91	1,00	0,95

Tabuľka 5.4: Vyhodnotenie predikcie náhodným lesom na dátach panel P2

Ako prvé sme na predikciu použili náhodný les. Najlepšie výsledky sa nám podarilo dosiahnuť pri hodnotách parametrov  $n_{tree} = 500$ ,  $m_{try} = 200$ , kde  $p$  je počet črt a s randomizáciou poradia pacientov. V matici zámen 5.3 sú výsledky predikcie, z ktorých vidno problém modelu na triede *Zdravý*, ktorá bola v dátach zastúpená malým počtom pacientov.

V tabuľke 5.4 je vyhodnotenie natrénovaného modelu metrikami presnosť, pokrytie a F1-skóre pre jednotlivé triedy, po otestovaní na testovacej množine. Presnosť je pomer počtu správnych predikcií triedy ku celkovému počtu predikcií danej triedy modelom. Pokrytie je vyčíslené ako pomer počtu správnych predikcií triedy ku celkovej početnosti triedy. F1-skóre sa používa na vyhodnocovanie predikčných modelov, zahŕňa metriky presnosť a pokrytie a je vypočítané ako  $F1 = 2 \times \frac{\text{presnosť} \times \text{pokrytie}}{\text{presnosť} + \text{pokrytie}}$ .

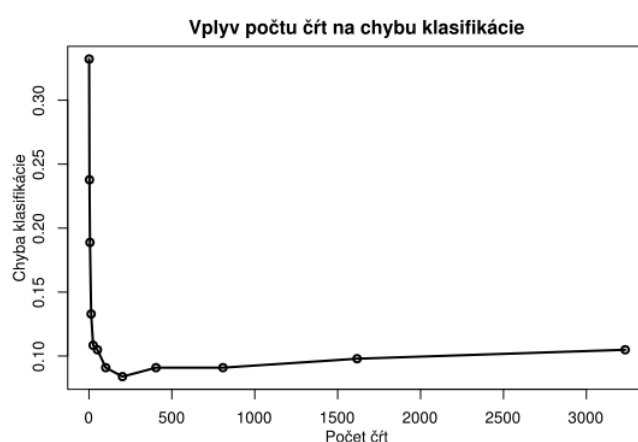
Pri tejto predikcii bolo extrahovaných 3234 črt, z čoho intuitívne vyplýva, že iba podmnožina všetkých črt bude prediktívna a preto sme vykonali viacero iterácií predikcií. V každej iterácii sme znížili počet črt o polovicu na základe prediktívnej sily črt tak, že sme odstránili polovicu najmenej prediktívnych črt. Chybu klasifikácie v jednotlivých iteráciách spolu s počtom črt sme vizualizovali na grafe 5.17, z ktorého vidno, že menej komplexný model, ktorý obsahuje okolo 200 črt, dosahuje najmenšiu chybu predikcie. Z tohto dôvodu sme sa rozhodli použiť aj sieť Elastic, ktorá umožňuje automatický výber prediktívnych črt.

		Referencia		
		Zdravý	WM	MM
Predikcia	Zdravý	3	0	1
	WM	0	12	0
	MM	0	0	41

Tabuľka 5.5: Matica zámen predikcie sieťou Elastic na dátach panela P2

	Presnosť	Pokrytie	F1-skóre
Zdravý	0,75	1,00	0,86
WM	1,00	1,00	1,00
MM	1,00	0,98	0,99

Tabuľka 5.6: Vyhodnotenie predikcie sieťou Elastic na dátach panela P2



Obr. 5.17: Vplyv počtu črt na chybu klasifikácie modelu

Pri predikcií so sieťou Elastic sme najlepšie výsledky, kedy sa nepodarilo správne klasifikovať iba jedného pacienta, dosiahli pri krížovej validácii typu *leave-one-out*, randomizácii poradia vo vstupnej dátovej množine a parametrom  $\alpha = 0,5$ . Z hodnôt matice zámen 5.5 vidno, že sieť Elastic sa podarilo správne klasifikovať aj málo zastúpenú triedu *Zdravý*.

Na základe hodnôt metrík v tabuľke 5.6 vieme vyhodnotiť, že sieť Elastic dosiahla na dátach z panela P2 lepšie výsledky predikcie ako model náhodného lesa.

V nasledujúcom kroku sme vykonali klasifikáciu do troch tried na dátach z panela P3. Použili sme náhodný les a sieť Elastic. Použitím modelu náhodný les sme dosiahli najlepšie výsledky pri randomizácii poradia vstupných dát a hodnotách parametrov  $n_{tree} = 500$  a  $m_{try} = 200$ . Výsledky predikcie náhodným lesom sú v matici zámen 5.7 a vyhodnotenie v tabuľke 5.8.

So sieťou Elastic sme najlepšie výsledky klasifikácie dosiahli pri randomizácii poradia vstupných dát a hodnotách parametru  $\alpha = 0,5$  *leave-one-out* krížovej validácii. Matica zámen 5.9 zobrazuje výsledky klasifikácie a v tabuľke 5.10 je vyhodnotenie klasifikácie metrikami.

Z výsledkov klasifikácie na dátach panela P3 jasne vidno, že náhodnému lesu a ani sieti Elastic sa nepodarilo správne klasifikovať triedu *Zdravý* s malým zastúpením a oba modely

		Referencia		
		Zdravý	WM	MM
Predikcia	Zdravý	0	0	0
	WM	0	10	0
	MM	3	0	40

Tabuľka 5.7: Matica zámen predikcie náhodným lesom na dátach panela P3

	Presnosť	Pokrytie	F1-skóre
Zdravý	—	0,00	—
WM	1,00	1,00	1,00
MM	0,93	1,00	0,96

Tabuľka 5.8: Vyhodnotenie predikcie náhodným lesom na dátach panela P3

		Referencia		
		Zdravý	WM	MM
Predikcia	Zdravý	0	0	0
	WM	1	10	0
	MM	2	0	40

Tabuľka 5.9: Matica zámen predikcie sieťou Elastic na dátach panela P3

	Presnosť	Pokrytie	F1-skóre
Zdravý	—	0,00	—
WM	0,91	1,00	0,95
MM	0,95	1,00	0,98

Tabuľka 5.10: Vyhodnotenie predikcie sieťou Elastic na dátach panela P3

Predikcia	Referencia	
	Pozitívna odozva	Negatívna odozva
	Pozitívna odozva	Negatívna odozva
Pozitívna odozva	7	2
Negatívna odozva	0	1

Tabuľka 5.11: Matica zámen predikcie náhodným lesom použitím klinických dát pacientov s chorobou WM

Predikcia	Referencia	
	Pozitívna odozva	Negatívna odozva
	Pozitívna odozva	Negatívna odozva
Pozitívna odozva	7	2
Negatívna odozva	0	1

Tabuľka 5.12: Matica zámen predikcie sieťou Elastic použitím klinických dát pacientov s chorobou WM

dosiahli veľmi podobné výsledky. Naopak na dátach z panela P2 sa sieť Elastic podarilo správne klasifikovať aj triedu *Zdravý* čím dosiahla lepšie výsledky klasifikácie ako náhodný les.

## 5.7.2 Predikcia použitím klinických dát

Pri predikcii použitím klinických dát sa predikovala odozva pacienta na podstúpenú liečbu. Z dôvodu nedostupnosti klinických dát pre zdravých pacientov sa predikcia, po konzultácii s doménovými expertmi, vykonávala iba na pacientoch s chorobou walderstrom myelóm (WM) a mnohopočetný myelóm (MM). Pacienti boli zaradení do týchto dvoch tried a okrem extrahovaných cytometrických črt sa pre nich extrahovali aj črty z klinických dát. Na základe týchto črt sa následne vykonala klasifikácia do dvoch tried – pozitívna odozva na liečbu a negatívna odozva na liečbu. Podrobnejší opis predikcie s použitím klinických dát, ako aj extrakcie klinických črt, je v časti 3.6.4 a 3.6.2.

Na klasifikáciu sme použili náhodný les a sieť Elastic a pri oboch sme dosiahli rovnaké výsledky. Poradie vstupných dát bolo randomizované a v náhodnom lese boli parametre nastavené na  $n_{tree} = 500$  a predvolenou hodnotou  $m_{try} = \lfloor \sqrt{p} \rfloor$ , kde  $p$  je počet črt a pri sieti Elastic  $\alpha = 0,5$  s vykonaním krížovej validácie typu *leave-one-out*. V jednotlivých maticiach zámen pre náhodný les 5.11 a sieť Elastic 5.12 sú výsledky predikcie a porovnanie výsledkov je v tabuľke 5.13.

Vplyv na predikciu odozvy pacientov na liečbu mala aj veľkosť dátovej množiny. V tejto množine sa nachádzalo iba 53 záznamov, čo je veľmi málo na klasifikáciu a vyvodenie záverov tejto predikcie.

	Presnosť	Pokrytie	F1-skóre
náhodný les	0,78	1,00	0,88
sieť Elastic	0,78	1,00	0,88

Tabuľka 5.13: Vyhodnotenie predikcií náhodným lesom a sieťou Elastic použitím klinických dát pacientov s chorobou WM

Predikcia	Referencia	
	Pozitívna odozva	Negatívna odozva
	Pozitívna odozva	Negatívna odozva
Pozitívna odozva	18	6
Negatívna odozva	0	2

Tabuľka 5.14: Matica zámen predikcie náhodným lesom použitím klinických dát pacientov s chorobou MM

Predikcia	Referencia	
	Pozitívna odozva	Negatívna odozva
	Pozitívna odozva	Negatívna odozva
Pozitívna odozva	17	4
Negatívna odozva	1	4

Tabuľka 5.15: Matica zámen predikcie sieťou Elastic použitím klinických dát pacientov s chorobou MM

Následne sme vykonali rovnakú klasifikáciu na dátovej množine pacientov s chorobou MM. Táto dátová množina je väčšia, ako množina dát pacientov s chorobou WM, keďže obsahuje 133 záznamov. Na klasifikáciu do dvoch tried sme opäť použili náhodný les a sieť Elastic.

Najlepšie výsledky náhodným lesom sme dosiahli pri randomizácii poradia vstupných dát a hodnotách vstupných parametrov  $n_{tree} = 500$  a  $m_{try} = 25$ . Matica zámen 5.14 obsahuje výsledky predikcie, podľa ktorých vidno, že náhodný les mal problém s klasifikáciu triedy *Negatívna odozva*. So sieťou Elastic sa nám podarilo dosiahnuť najlepšie výsledky so vstupným parametrom  $\alpha = 0,5$ , *leave-one-out* krížovou validáciou a randomizáciou poradia vstupných dát. V matici zámen 5.15 sú výsledky klasifikácie použitím siete Elastic. Podľa matice vidno, že sieť elastic o niečo lepšie klasifikovala triedu *Negatívna odozva* oproti náhodnému lesu.

Vyhodnotenie klasifikácie náhodným lesom a sieťou Elastic je v tabuľke 5.16. Oba klasifikátory dosiahli veľmi podobné výsledky podľa metriky F1-skóre, pričom sieť Elastic o trochu lepšie hlavne vďaka lepšej klasifikácii triedy *Negatívna odozva*, pri ktorej mal náhodný les väčšie problémy.

### 5.7.3 Porovnanie s FlowCAP-II

Na objektívne vyhodnotenie je potrebné predikciu porovnať s existujúcimi prístupmi. Združenie FlowCAP organizuje súťaže, v ktorých objektívne vyhodnocuje softvérové nástroje, algoritmy a modely na spracovanie a analýzu cytometrických dát. Súťaž FlowCAP-II sa zmerala aj na predikciu klinického stavu pacienta. K jednotlivým súťažiam sú voľne dostupné dátové množiny, ktoré boli použité na evaluáciu účastníkov súťaže. Viac informácií o súťaži FlowCAP-II je v

	Presnosť	Pokrytie	F1-skóre
náhodný les	0,75	1,00	0,86
sieť Elastic	0,81	0,94	0,87

Tabuľka 5.16: Vyhodnotenie predikcií náhodným lesom a sieťou Elastic použitím klinických dát pacientov s chorobou MM



Predikcia	Referencia	
	AML	Zdravý
AML	8	1
Zdravý	0	62

Tabuľka 5.17: Matica zámen predikcie náhodným lesom na dátovej množine AML zú súťaže FlowCAP-II

Predikcia	Referencia	
	AML	Zdravý
AML	8	0
Zdravý	0	63

Tabuľka 5.18: Matica zámen predikcie sieťou Elastic na dátovej množine AML zú súťaže FlowCAP-II

časti 2.6.1.

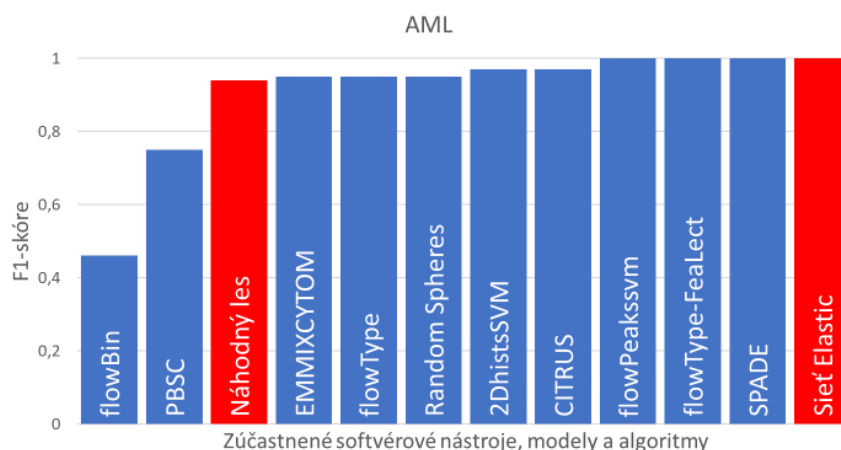
Na vyhodnotenie predikcie a porovnanie s existujúcimi riešeniami sme použili dátovú množinu AML zo súťaže FlowCAP-II. Táto dátová množina obsahuje cytometrické dáta 359 pacientov, ktorí sú zaradení do dvoch skupín – s chorobou akútna myeloidná leukémia (AML) a zdraví. Cieľom predikcie bolo správne určiť, či sa jedná o pacienta s chorobou AML alebo zdravého pacienta.

Na cytometrických dátach sme vykonali redukciu dát na základe hustoty, zhlukovanie na identifikovanie bunkových populácií, upsampling a následne extrakciu cytometrických črt. Dátovú množinu sme rozdelili na trénovaciu a testovaciu množinu. Na trénovacej množine sme natrénovali klasifikátory náhodný les a sieť Elastic. Najlepšie výsledky pre klasifikátor náhodný les sme dosahovali pri hodnotách vstupných parametrov  $n_{tree} = 500$ ,  $m_{try} = 40$  a pre klasifikátor sieť Elastic  $\alpha = 0,5$ , *leave-one-out* krížová validácia a randomizácia poradia vstupných dát pri oboch prístupoch. Výsledky klasifikácie náhodným lesom sú v matici zámen 5.17 a výsledky siete Elastic v matici zámen 5.18. Vyhodnotenie metrikami a porovnanie oboch klasifikátorov je v tabuľke 5.19.

Klasifikátoru náhodný les sa nepodarilo správne klasifikovať iba jedného pacienta a klasifikátor sieť Elastic dosiahol bezchybný výsledok. Výsledky oboch klasifikátorov sme porovnali s výsledkami zo súťaže FlowCAP-II na dátovej množine AML na základe metriky F1-skóre. Porovnania výsledkov sú vizualizované grafom 5.18. Oba klasifikátory nami navrhnutého riešenia dosiahli veľmi dobré výsledky v porovnaní s ostatnými účastníkmi, pričom sieť Elastic dosiahla najlepšie výsledky ešte s prístupmi *flowPeakssvm*, *flowType-FeaLect* a *SPADE*.

	Presnosť	Pokrytie	F1-skóre
náhodný les	0,89	1,00	0,94
sieť Elastic	1,00	1,00	1,00

Tabuľka 5.19: Vyhodnotenie predikcií náhodným lesom a sieťou Elastic na dátovej množine AML zo súťaže FlowCAP-II



Obr. 5.18: Porovnanie výsledkov zo súťaže FlowCAP-II s výsledkami nášho riešenia

Porovnaním výsledkov sa nám podarilo overiť správnosť nami navrhnutého riešenia, ako aj ukázať, že nami navrhnuté riešenie dosahuje rovnaké, až lepšie výsledky ako zaužívané prístupy pri analýze cytometrických dát a predikcie klinického stavu pacienta.

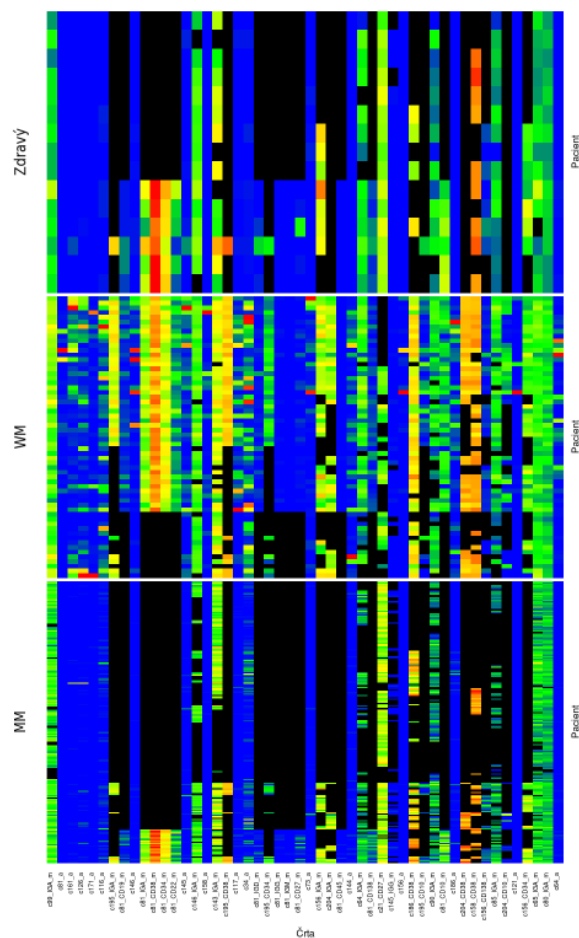
## 5.8 Interpretácia výsledkov predikcie

Extrahovanie nových znalostí z výsledkov predikcie je dôležitý krok v analýze, ktorý je časovo náročný, podlieha subjektivite a vyžaduje značnú znalosť doménovej oblasti. Vhodnou vizualizáciou výsledkov je možné proces extrahovania nových znalostí urýchliť a znížiť zanášanie subjektivity do výsledkov.

Nami navrhnutú vizualizáciu výsledkov predikcie použitím teplotných máp opísanú v časti 3.6.5 sme vyhodnocovali na predikcii klinického stavu pacienta z cytometrických dát panel P2 podľa vyhodnotenia dôležitosti črt klasifikátorom náhodný les. Na vizualizáciu sme vybrali 50 črt, ktorým náhodný les priradil najvyššiu dôležitosť. Následne sme podľa tried klasifikácie zatriedili pacientov do troch skupín. Pre každú skupinu sme vizualizovali hodnoty črt pre všetkých pacientov zaradených v danej skupine pomocou teplotnej mapy. Výsledkom boli tri teplotné mapy, jedná pre každú triedu predikcie. Tieto teplotné mapy sme následne vizualizovali pod sebou v jednej vizualizácii, s tým že poradie črt je rovnaké v každej teplotnej mape, čo umožňuje vizuálne porovnanie jednotlivých teplotných máp. Výsledná vizualizácia je na obrázku 5.19.

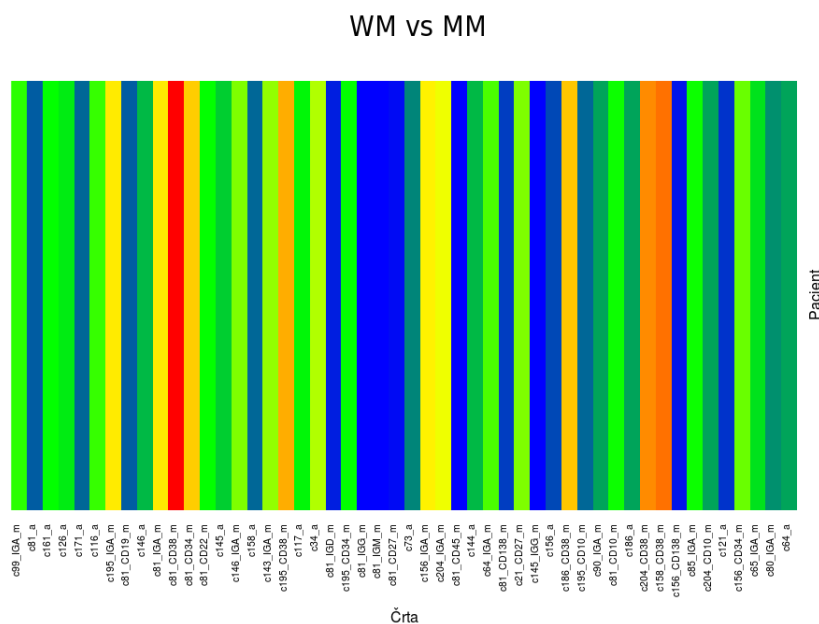
Na vizualizácii sú zobrazené tri teplotné mapy, každá pre jednu triedu predikcie a teda *Zdravý*, *WM*, a *MM*. Stĺpce teplotných máp reprezentujú črty, ktorých názvy su na konci spodnej teplotnej mapy a riadky jednotlivých pacientov v konkrétnej triede, ktorú teplotná mapa vizualizuje. Ak pacient nemal pre črtu hodnotu, lebo daná črta bola vytvorená na základe zhľuku, v ktorom daný pacient nemal žiadne bunky, tak je v teplotnej mape zobrazená pre

danú črtu a daného pacienta čierna oblasť. Vo vizualizácií je vidno rozdiely medzi jednotlivými triedami na úrovni črt.



Obr. 5.19: Vizualizácia výsledkov predikcie teplotnými mapami

Vizuálne porovnanie teplotných máp môže viesť k nesprávnym výsledkom. Tento problém je možné minimalizovať vizualizáciou rozdielu teplotných máp. Keďže trieda *Zdravý* obsahovala malý počet pacientov, pre overenie nášho návrhu sme vybrali triedy *WM* a *MM*. Všetky črty sme si reprezentovali priemernými hodnotami danej črty v konkrétnej triede. Odčítaním priemerných hodnôt črt v absolútnej hodnote sme získali reprezentáciu rozdielu hodnôt jednotlivých črt, ktorá je vizualizovaná na obrázku 5.20. Z vizualizácie jednoducho a jasne vidno rozdiely v hodnotách jednotlivých črt medzi triedami *WM* a *MM*, na ktorých sa klasifikátor rozhodoval pri klasifikácii.



Obr. 5.20: Rozdiel hodnôt prediktívnych črt tried WM a MM

Jednotlivé prediktívne črty použité pri predikcii sú pomenované podľa zhlukov, respektívne vrcholov stromovej vizualizačnej štruktúry a zoradené podľa prediktívnosti. Na podrobnú interpretáciu výsledkov je potrebné zhluky, z ktorých vznikli črty, zaradiť do konkrétnych bunkových populácií podľa hodnôt cytometrických znakov, napríklad využitím spomínanej stromovej vizualizačnej štruktúry. Na takéto zaradenie je potrebná znalosť doménovej oblasti a preto sme plánovali tieto výsledky overiť s doménovými expertmi, čo sme žiaľ nestihli. Z toho dôvodu navrhujeme nami navrhnutý spôsob vizualizácie na podporu interpretáciu výsledkov iba ako jeden z možný spôsobov ako doménovým expertom uľahčiť interpretáciu výsledkov predikcie.

## 5.9 Zhrnutie

Overili a vyhodnotili sme všetky kroky nášho riešenia. Ako prvé sme overovali správnosť výsledkov nami navrhnutého prístupu výpočtu hustoty buniek porovnaním s naivným riešením a podarilo sa nám overiť správnosť nášho návrhu. Ako ďalšie sme overovali správnosť nášho návrhu váhovanej hustoty. Na vygenerovaných dvojrozmerných dátach obsahujúcich dve oblasti s rozdielnou hustotou bodov sme vypočítali váhovanú hustotu pre každý bod. Empirickým vyhodnotením sme overili správnosť nášho riešenia, keďže husté oblasti vygenerovaných dát dosahovali výrazne vyššie hodnoty váhovanej hustoty ako riedke oblasti.

Následne sme overovali správnosť nášho návrhu na redukciu dát na základe hustoty. Ako prvé sme overili fungovanie iteratívneho prístupu ako vhodného prístupu na redukciu dát na základe hustoty. Pokračovali sme overovaním procesu redukcie dát na základe hustoty na dvojrozmerných vygenerovaných dátach, ktoré uľahčujú vizualizáciu. Overili sme návrh odstránenia

šumu v procese redukcie ako aj nami navrhovaný prístup na úpravu váh. Overenia sme podložili viacerými vizualizáciami.

Ako ďalšie sme overili návrh procesu upsampling a pokračovali sme porovnaním výpočtovej náročnosti nami navrhovaných riešení výpočtu hustoty a redukcie dát na základe hustoty s výpočtovou náročnosťou týchto procesov implementovaných v deterministickej verzii softvérového nástroja SPADE. Z výsledkov porovnaní jasne vidno, že sa nám podarilo dosiahnuť niekoľkonásobné zrýchlenie vo výpočte hustoty a ešte niekoľkonásobne väčšie zrýchlenie v procese redukcie dát na základe hustoty.

V neposlednom rade sme overovali správnosť výsledkov z biologického hľadiska. Porovnali sme výsledky našim riešením identifikovaných známych bunkových populácií v cytometrických dátach kostnej drene myši s identifikovanými bunkovými populáciami nástrojom SPADE. Následne sme porovnali vizualizácie identifikovaných bunkových populácií v našim dátach. Na základe vizualizácií sme ukázali, že sme dosiahli konzistentné výsledky s nástrojom SPADE z biologického hľadiska. Navyše sme výsledky overili s doménovými expertmi, ktorí potvrdili ich správnosť.

Následne sme vyhodnocovali predikciu klinického stavu pacienta nami navrhnutým riešením. Vykonali sme viacero predikcií klinického stavu na našim dátach s použitím klasifikátorov náhodný les a sieť Elastic a vyhodnotili výsledky predikcie metrikami presnosť, pokrytie a F1-skóre. Nakoniec sme vykonali predikciu klinického stavu na dátovej množine zo súťaže FlowCAP-II a porovnali sa s existujúcimi riešeniami zúčastnených tejto súťaže. V porovnaní sme dosiahli najlepšie výsledky ešte s ďalšími tromi zúčastnenými riešeniami. Pri všetkých vykonaných predikciách dosahoval klasifikátor sieť Elastic lepšie výsledky v porovnaní s náhodným lesom.

Ako posledné sme vykonali vizualizáciu interpretácie výsledkov predikcie použitím teplotných máp. Tieto výsledky sme nestihli overiť s doménovými expertmi.



# Kapitola 6

## Zhodnotenie

Tradičný prístup k analýze cytometrických dát je manuálny gating. Ukázali sme, že s narastajúcou dimenzionalitou a početnosťou cytometrických dát je manuálny gating neškálovateľný, neefektívny a subjektívny prístup. Manuálnym gatingom zanesená subjektivita do výsledkov spôsobuje nereprodukovateľnosť a komplikuje porovnateľnosť výsledkov, čo má za následok vytváranie prekážok výskumníkom pri analýze. Na adresovanie problémov týchto metód je nevyhnutné použitie nových automatizovaných výpočtových metód. V práci sme analyzovali najnovšie a najpoužívannejšie prístupy na analýzu cytometrických dát. Rozdelili sme ich do štyroch hlavných skupín podľa zamerania a to metódy vizualizácie, zhlukovania, detekcie vývojových trajektórií buniek a predikcie klinického stavu. Pre každú skupinu sme priblížili hlavných predstaviteľov a ukázali aký problém riešia a aké postupy pri tom využívajú.

Jedným z prvých a dôležitých krokov v analýze cytometrických dát je identifikácia bunkových populácií. Pri automatizovaných prístupoch sa na ich identifikáciu využívajú zhlukovacie algoritmy. Bunkové populácie môžu byť, podľa počtu zastúpení v dátach, hojné alebo vzácne. Vzácne bunkové populácie sú v dátach zastúpené podstatne menším počtom ako hojné. V analýze sme ukázali, že nerovnomerné zastúpenie a rozdiely v hustote vrámci priestoru môžu spôsobiť nesprávnu identifikáciu bunkových populácií zhlukovacími algoritmami. Na vyriešenie problému identifikácie vzácných bunkových populácií zhlukovacím algoritmom bol navrhnutý proces redukcie dát na základe hustoty, ktorý vyrovná reprezentáciu a hustotu bunkových populácií v priestore.

Analyzovali sme existujúce prístupy na redukciu dát na základe hustoty a poukázali sme na hlavné problémy týchto riešení, ako nedeterminizmus a výpočtová neefektívnosť na veľkých dátových množinách. Prístupy často využívajú stochastické prvky, aby ich riešenie bolo použiteľné aj na veľké dátové množiny, čo má za dôsledok nereprodukovateľnosť výsledkov. Naopak prístupy, ktoré využívajú iba deterministické prvky sú navrhnuté neefektívne a výpočtová náročnosť na veľkých dátových množinách je vysoká, čo spôsobuje predĺženie celkového času potrebného na analýzu.

Navrhli sme efektívny prístup na výpočet hustoty a redukcie dát na základe hustoty, ktoré adresujú problémy existujúcich riešení, sú plne deterministické a použiteľné na veľké dátové

množiny. Nami navrhnutý výpočet hustoty využíva delenie priestoru, čím výrazne redukuje množiny buniek, s ktorými treba vypočítať vzdialenosť pre zistenie hustoty konkrétnej bunky. Využitím vhodnej stromovej indexovej štruktúry sa nám podarilo efektívne reprezentovať rozdelený priestor, ako aj vyhľadávať bunky a množiny buniek v priestore a vykonať efektívnu paralelizáciu riešenia. Algoritmus sme overili s naivným prístupom na správnosť výsledkov a s nástrojom SPADE, ktorý je najpoužívanejší softvérový nástroj na analýzu cytometrických dát. Podľa výsledkov je nami navrhnutý algoritmus veľmi vhodný prístup na výpočet hustoty s dosiahnutým podstatným, až niekoľkonásobným zrýchlením v porovnaní s nástrojom SPADE. Na efektívne vykonanie redukcie dát na základe hustoty sme navrhli iteratívny prístup, ktorý využíva dve hlavné vlastnosti. Pri relatívne nízkej hodnote  $\varepsilon$  je priestor rozdelený tak, že sa väčšina buniek bude nachádzať vo vlastnom liste stromu, s výnimkou hustých oblastí, kde listy stromu budú obsahovať viacero buniek. Po redukcii dát na základe hustoty v takomto priestore sa zredukujú hlavne husté oblasti priestoru. Ďalšou využívanou vlastnosťou je rýchly výpočet hustoty pri relatívne nízkej hodnote  $\varepsilon$  lebo väčšina buniek nebude mať žiadne, prípadne iba veľmi málo susedných buniek, s ktorými bude potrebné počítať vzdialenosť. Iteratívny prístup teda začína s relatívne nízkou hodnotou  $\varepsilon$ , ktorá sa postupne upravuje a ako vstupné dáta do iterácie sa používajú zredukované dáta z prechádzajúcej iterácie, čím sa postupne redukuje množina na výslednú veľkosť. Nami navrhnuté riešenie sme porovnali s nástrojom SPADE a z výsledkov niekoľkonásobného zrýchlenia vyplýva, že použitie iteratívneho prístupu je veľmi vhodné na vykonanie redukcie dát na základe hustoty.

Na identifikáciu bunkových populácií zo zredukovaných dát sme použili aglomeratívne zhľukovanie. Navrhli sme použiť Fruchterman-Reingold algoritmus na vizualizáciu bunkových populácií v stromovej štruktúre. Takto vizualizované bunkové populácie sme porovnali s vizualizáciami bunkových populácií nástroja SPADE. Z výsledkov porovnania vidno, že nami navrhnuté riešenie dosahuje konzistentné výsledky s najpoužívanejším nástrojom na identifikovanie bunkových populácií za potreby niekoľkonásobne menej výpočtového času. Nami identifikované bunkové populácie, na základe hodnôt cytometrických znakov, sme overili aj s doménovými expertmi, ktorí potvrdili ich správnosť.

Z výsledkov súťaže FlowCAP-II zameranej na vyhodnotenie predikcií klinického stavu na základe cytometrických dát sme ukázali, že sa jedná o veľmi dobre predikovateľný problém. S použitím cytometrických a klinických dát, ktoré sme mali dostupné od doktoriek na SAV sme vykonali viacero predikcií klinického stavu pacienta. Zamerali sme sa na predikciu príznaku choroby, respektívne typu choroby na základe cytometrických dát, ale aj na predikciu odozvy na podstúpenú liečbu využitím cytometrických a klinických dát. Na predikcie sme použili klasifikátory náhodný les a sieť Elastic, ktorá dosahovala lepšie výsledky. Predikciu sme vykonali aj na dátovej množine cytometrických dát zo súťaže FlowCAP-II a výsledky sme porovnali s účastníkmi súťaže. Podarilo sa nám dosiahnuť najlepšie výsledky totožné ešte s ďalšími tromi zúčastnenými riešeniami.

Vhodnou interpretáciou výsledkov predikcie je možné uľahčiť extrahovanie relevantných

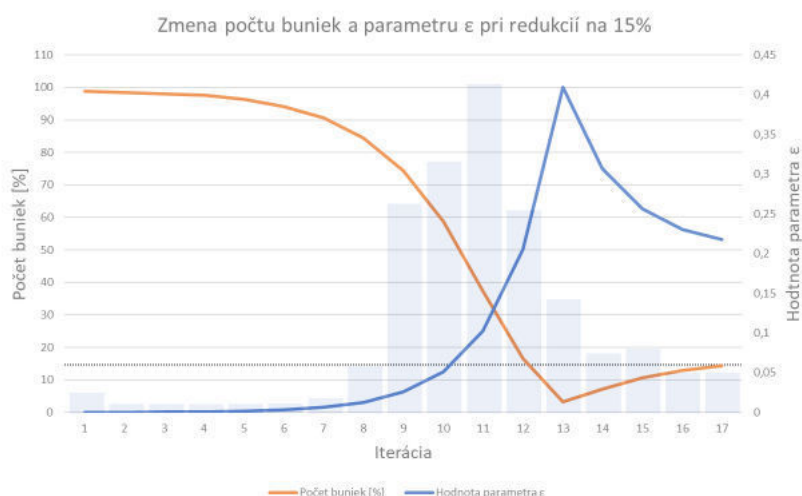


záverov analýzy. Na interpretáciu výsledkov sme navrhli vizualizáciu teplotnými mapami a vykonali ju na výsledkoch predikcie na dátach panela P2. Vizualizáciu interpretácie výsledkov sme nestihli overiť s doménovými expertmi a preto ju navrhujeme iba ako jednu z možných vizualizácií na uľahčenie extrahovania záverov z analýzy.

Nami navrhnuté riešenia na výpočet hustoty a redukcie dát na základe hustoty sa podľa výsledkov ukázali ako veľmi vhodné. Ich niekoľkonásobne nižšia časová náročnosť umožňuje zrýchlenie celého procesu analýzy cytometrických dát. Zo zredukovaných dát sme zhlukovacím algoritmom identifikovali bunkové populácie, ktoré sme vizualizovali stromovou štruktúrou, porovnali ich s existujúcimi riešeniami a overili s doménovými expertmi. Dosiahli sme konzistentné výsledky z biologického hľadiska v porovnaní s existujúcimi riešeniami pri nižšej časovej náročnosti riešenia. Vykonali sme predikcie klinického stavu pacienta a porovnali naše výsledky s existujúcimi riešeniami, kde sme dosiahli najlepšie výsledky ešte s ďalšími tromi riešeniami. Nami navrhnuté prístupy majú veľký potenciál na zlepšenie procesov analýzy cytometrických dát.

## 6.1 Možnosti rozšírenie práce

Sekvencia zmeny hodnoty  $\varepsilon$  pri iteratívnom procese redukcie dát na základe hustoty je podľa geometrického radu, čo je vidno aj z obrázka 5.3. Ak si pre každú iteráciu zobrazíme čas potrebný na jej vykonanie, tak ako vidno na obrázku 6.1, čas iterácií je najvyšší práve pri rýchlom náraste hodnoty  $\varepsilon$ . Celkový výpočtový čas iteratívneho procesu je ovplyvnený práve týmito iteráciami. Navrhnutím a využitím sekvencie, podľa ktorej by sa upravovala hodnota  $\varepsilon$ , ktorá nebude geometrický rad s veľkými skokmi v hodnote  $\varepsilon$ , ale zároveň nebude stúpať veľmi pomaly, je možné približne vyrovnať výpočtové časy jednotlivých iterácií, čím by sa zredukoval celkový výpočtový čas na redukciu dát na základe hustoty.



Obr. 6.1: Výpočtové časy jednotlivých iterácií v iteratívnom procese redukcie dát na základe hustoty

Použitý Fruchterman-Reingold algoritmus na zostavenie stromovej štruktúry vizualizácie je závislý od vstupného parametru počtu iterácií. Tento algoritmus je potrebné upraviť, aby jeho ukončenie nebolo podmienené počtom vykonaných iterácií, čo môže spôsobiť, že v zložitejšej stromovej štruktúre nemusia byť za daný počet iterácií vrcholy rozmiestnené tak, aby sa hrany stromu nepretínali. Naopak, algoritmus by mal sám skonvergovať a preto je potrebné do algoritmu doplniť dve podmienky a to minimálny posun a nepretínajúce sa hrany. Podmienka minimálneho posunu je splnená, ak platí  $\forall c : m(c) \leq \delta$ , kde  $c$  je vrchol stromu a  $m(c)$  je posun vrchola  $c$  v danej iterácii a  $\delta$  je definovaná prahová hodnota minimálneho posunu. Nepretínajúce hrany dosiahne strom vtedy, ak v danej iterácii neexistuje dvojica hrán, ktoré by sa vzájomne pretínali. Ak sú obe podmienky splnené, algoritmus rozmiestnenia sa ukončí.

Ďalší problém použitého Fruchterman-Reingold algoritmu je nedeterministickosť. Aj napriek zachovaniu topologickej informácie sa vizualizácie nad rovnakými dátami môžu vizuálne líšiť, čo spôsobí komplikácie pri porovnateľnosti a reprodukovateľnosti výsledkov. Navrhnutím a využitím deterministického algoritmu na rozloženie vrcholov stromovej štruktúry by sa tento problém odstránil.

Použitím klasifikátorov náhodného lesa a siete Elastic sme dosiahli dobré výsledky. Autori v softvérovom nástroji na analýzu cytometrických dát CITRUS [13] použili aj metódu *nearest shrunken centroids* na predikciu a tiež dosiahli dobré výsledky v porovnaní so súťažou FlowCAP-II. Použitím a vyhodnotením napríklad tejto metódy je možnosť dosiahnutia ešte o niečo lepších výsledkov predikcií.

# Literatúra

- [1] A. Adan, G. Alizada, Y. Kiraz, Y. Baran, and A. Nalbant. Flow cytometry: basic principles and applications. *Critical Reviews in Biotechnology*, 37(2):163–176, 2017. PMID: 26767547.
- [2] N. Aghaeepour, P. Chattopadhyay, M. Chikina, T. Dhaene, S. Van Gassen, M. Kurs, B. N. Lambrecht, M. Malek, G. J. McLachlan, Y. Qian, P. Qiu, Y. Saeys, R. Stanton, D. Tong, C. Vens, S. Walkowiak, K. Wang, G. Finak, R. Gottardo, T. Mosmann, G. P. Nolan, R. H. Scheuermann, and R. R. Brinkman. A benchmark for evaluation of algorithms for identification of cellular correlates of clinical outcomes. *Cytometry Part A*, 89(1):16–21, 2016.
- [3] N. Aghaeepour, G. Finak, H. Hoos, T. R. Mosmann, R. Brinkman, R. Gottardo, and R. H. Scheuermann. Critical assessment of automated flow cytometry data analysis techniques. *Nat Meth*, 10(3):228–238, Mar. 2013.
- [4] C. Alhan, T. M. Westers, E. M. P. Cremers, C. Cali, B. I. Witte, G. J. Ossenkoppele, and A. A. Loosdrecht. High flow cytometric scores identify adverse prognostic subgroups within the revised international prognostic scoring system for myelodysplastic syndromes. *British Journal of Haematology*, 167(1):100–109, 2014.
- [5] E.-a. D. Amir, K. L. Davis, M. D. Tadmor, E. F. Simonds, J. H. Levine, S. C. Bendall, D. K. Shenfeld, S. Krishnaswamy, G. P. Nolan, and D. Pe’er. visne enables visualization of high dimensional single-cell data and reveals phenotypic heterogeneity of leukemia. *Nat Biotech*, 31(6):545–552, June 2013.
- [6] C. B. Bagwell. Hyperlog—a flexible loglike transform for negative, zero, and positive valued data. *Cytometry Part A*, 64A(1):34–42, 02 2005.
- [7] N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger. The  $r^*$ -tree: An efficient and robust access method for points and rectangles. *SIGMOD Rec.*, 19(2):322–331, May 1990.
- [8] S. Bendall, K. Davis, E. adDavid Amir, M. Tadmor, E. Simonds, T. Chen, D. Shenfeld, G. Nolan, and D. Pe’er. Single-cell trajectory detection uncovers progression and regulatory coordination in human b cell development. *Cell*, 157(3):714 – 725, 2014.

- [9] S. C. Bendall, E. F. Simonds, P. Qiu, E.-a. D. Amir, P. O. Krutzik, R. Finck, R. V. Bruggner, R. Melamed, A. Trejo, O. I. Ornatsky, R. S. Balderas, S. K. Plevritis, K. Sachs, D. Pe'er, S. D. Tanner, and G. P. Nolan. Single-cell mass cytometry of differential immune and drug responses across a human hematopoietic continuum. *Science*, 332(6030):687–696, 2011.
- [10] J. L. Bentley. Multidimensional binary search trees in database applications. *IEEE Transactions on Software Engineering*, SE-5(4):333–340, July 1979.
- [11] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, Oct 2001.
- [12] R. A. Brown. Building a balanced  $k$ -d tree in  $o(kn \log n)$  time. *Journal of Computer Graphics Techniques (JCGT)*, 4(1):50–68, March 2015.
- [13] R. V. Bruggner, B. Bodenmiller, D. L. Dill, R. J. Tibshirani, and G. P. Nolan. Automated identification of stratifying signatures in cellular subpopulations. *Proceedings of the National Academy of Sciences*, 111(26):E2770–E2777, 2014.
- [14] P. Chattopadhyay, S. Perfetto, B. Gaylord, A. Stall, L. Duckett, J. Hill, R. Nguyen, D. Ambrozak, R. Balderas, and M. Roederer. Toward 40+ parameter flow cytometry. *Congress of the International Society of Advancement of Cytometry*, 215, 2014.
- [15] C. Chester and H. T. Maecker. Algorithmic tools for mining high-dimensional cytometry data. *The Journal of Immunology*, 195(3):773–779, 2015.
- [16] K. Fawagreh, M. M. Gaber, and E. Elyan. Random forests: from early developments to recent advancements. *Systems Science & Control Engineering: An Open Access Journal*, 2(1):602–609, 2014.
- [17] B. Gregorutti, B. Michel, and P. Saint-Pierre. Correlation and variable importance in random forests. *Statistics and Computing*, 27(3):659–678, May 2017.
- [18] A. Guttman. R-trees: A dynamic index structure for spatial searching. *SIGMOD Rec.*, 14(2):47–57, June 1984.
- [19] F. Hahne, N. LeMeur, R. R. Brinkman, B. Ellis, P. Haaland, D. Sarkar, J. Spidlen, E. Strain, and R. Gentleman. flowcore: a bioconductor package for high throughput flow cytometry. *BMC Bioinformatics*, 10:106–106, Apr. 2009.
- [20] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: data mining, inference and prediction*. Springer, 2 edition, Feb. 2009.
- [21] J. Levine, E. Simonds, S. Bendall, K. Davis, E.-a. Amir, M. Tadmor, O. Litvin, H. Fienberg, A. Jager, E. Zunder, R. Finck, A. Gedman, I. Radtke, J. Downing, D. Pe'er, and G. Nolan.

- Data-driven phenotypic dissection of aml reveals progenitor-like cells that correlate with prognosis. *Cell*, 162(1):184–197, 2015.
- [22] H. Li, U. Shaham, K. P. Stanton, Y. Yao, R. R. Montgomery, and Y. Kluger. Gating mass cytometry data by deep learning. *Bioinformatics*, 33(21):3423–3430, 2017.
- [23] A. Liaw, M. Wiener, et al. Classification and regression by randomforest. *R news*, 2(3):18–22, 2002.
- [24] L. Lin, G. Finak, K. Ushey, C. Seshadri, T. R. Hawn, N. Frahm, T. J. Scriba, H. Mahomed, W. Hanekom, P.-A. Bart, G. Pantaleo, G. D. Tomaras, S. Rerks-Ngarm, J. Kaewkungwal, S. Nitayaphan, P. Pitisuttithum, N. L. Michael, J. H. Kim, M. L. Robb, R. J. O’Connell, N. Karasavvas, P. Gilbert, S. C De Rosa, M. J. McElrath, and R. Gottardo. Compass identifies t-cell subsets correlated with clinical outcomes. *Nat Biotech*, 33(6):610–616, June 2015.
- [25] K. Lo, R. R. Brinkman, and R. Gottardo. Automated gating of flow cytometry data via robust modelbased clustering. *Cytometry Part A*, 73A(4):321–332, 02 2008.
- [26] R. H. C. Lopes, I. D. Reid, and P. R. Hobson. A well-separated pairs decomposition algorithm for k-d trees implemented on multi-core architectures. *Journal of Physics: Conference Series*, 513(5):052011, 2014.
- [27] P. Qiu. Toward deterministic and semiautomated spade analysis. *Cytometry. Part A : the journal of the International Society for Analytical Cytology*, 91:281–289, Mar 2017.
- [28] P. Qiu, E. F. Simonds, S. C. Bendall, K. D. Gibbs Jr, R. V. Bruggner, M. D. Linderman, K. Sachs, G. P. Nolan, and S. K. Plevritis. Extracting a cellular hierarchy from high-dimensional cytometry data with spade. *Nat Biotech*, 29(10):886–891, Oct. 2011.
- [29] Y. Saeys, S. V. Gassen, and B. N. Lambrecht. Computational flow cytometry: helping to make sense of high-dimensional immunology data. *Nat Rev Immunol*, 16(7):449–462, July 2016.
- [30] H. M. Shapiro. *Data Analysis*, pages 225–256. Wiley-Blackwell, 2005.
- [31] J. Spidlen, W. Moore, D. Parks, M. Goldberg, C. Bray, P. Bierre, P. Gorombey, B. Hyun, M. Hubbard, S. Lange, R. Lefebvre, R. Leif, D. Novo, L. Ostruszka, A. Treister, J. Wood, R. F. Murphy, M. Roederer, D. Sudar, R. Zigon, and R. R. Brinkman. Data file standard for flow cytometry, version fcs 3.1. *Cytometry. Part A : the journal of the International Society for Analytical Cytology*, 77(1):97–100, Jan. 2010.
- [32] M. Spitzer and G. Nolan. Mass cytometry: Single cells, many features. *Cell*, 165(4):780–791, May 2016.

- [33] S. D. Tanner, V. I. Baranov, O. I. Ornatsky, D. R. Bandura, and T. C. George. An introduction to mass cytometry: fundamentals and applications. *Cancer Immunology, Immunotherapy*, 62(5):955–965, 2013.
- [34] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.
- [35] S. Van Gassen, B. Callebaut, M. J. Van Helden, B. N. Lambrecht, P. Demeester, T. Dhaene, and Y. Saeys. Flowsom: Using self-organizing maps for visualization and interpretation of cytometry data. *Cytometry Part A*, 87(7):636–645, 2015.
- [36] S. Van Gassen, C. Vens, T. Dhaene, B. Lambrecht, and Y. Saeys. Floremi: Flow density survival regression using minimal feature redundancy. *Cytometry Part A*, 89(1):22–29, 08 2015.
- [37] C. P. Verschoor, A. Lelic, J. L. Bramson, and D. M. E. Bowdish. An introduction to automated flow cytometry gating tools and their implementation. *Frontiers in Immunology*, 6:380, July 2015.
- [38] L. M. Weber and M. D. Robinson. Comparison of clustering methods for high-dimensional single-cell flow and mass cytometry data. *Cytometry Part A*, 89(12):1084–1096, 2016.
- [39] H. Zare, P. Shooshtari, A. Gupta, and R. R. Brinkman. Data reduction for spectral clustering to analyze high throughput flow cytometry data, 2010.
- [40] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.

# Príloha A

## Dokumentácia

Implementácia navrhnutého riešenia sa nachádza v adresári `Application` elektronického média, ako je uvedené v prílohe elektronického média B. V tomto priečinku sa nachádzajú všetky tri balíky implementácie, ako je opísané v časti 4.

Na spustenie implementácie je potrebné mať nainštalovaný interpreter jazyka R pre verziu  $R \geq 3.4.2$  a korektne nainštalovanú knižnicu OpenMP.

### A.1 Inštalácia

V priečinku `Application` elektronického média sa nachádza skript `build.bat`, ktorý slúži na nainštalovanie výpočtového a vizualizačného balík implementácie (viď. kapitolu 4). Tento skript obsahuje príkazy jazyka R, ktoré nainštalujú príslušné balíky.

Na stroji s operačným systémom Windows postačuje skript `build.bat` spustiť, prípadne v konzole spustiť príkazom `.\build.bat`. Pod operačným systémom Linux je možné spustiť skript príkazom `bash build.bat`

### A.2 Spustenie implementácie

Kvôli veľkosti cytometrických dát sú k implementácií dodané iba ukážkové dáta v priečinku `Application/magic/data/examples` elektronického média. Nachádzajú sa tu dve vzorky cytometrických dát a to `healthy.fcs` od zdravého pacienta a `WM.fcs` od pacienta s chorobou walderstrom myelóm.

Spustenie implementácie vykoná proces analýzy cytometrických dát na ukážkových dátach a zobrazí na výstupe vizualizácie výsledkov. Implementáciu je možné spustiť dvoma spôsobmi (i) z príkazového riadka (ii) cez IDE. Z príkazového riadka sa implementácia spusti nasledovne – vojsť do priečinka hlavného výpočtového balíka `Application` príkazom `cd Application/magic` a spustenie implementácie príkazom `Rscript R/run.R`. Im-

plementácia vykoná načítanie ukážkových dát z priečinka `magic/data/examples` ako aj dodaných predspracovaných dát z priečinka `Application/magic/data/output`, vykoná zhľukovanie na 200 zhľukov, upsampling, vytvorí vizualizácie a spustí lokálny R server. V konzole sa zobrazí výstup v tvare `Listening on http://127.0.0.1:<port>` indikujúci na akom porte je spustený lokálny R server. Po zadaní tejto adresy do webového prehliadača sa zobrazia výsledne interaktívne vizualizácie.

Pre automatické zobrazenie výsledkov vo webovom prehliadači a úpravu vstupných parametrov analýzy odporúčame otvoriť súbor `Application/magic/magic.Rproj` v IDE, ktoré podporuje projekty jazyka R. Odporúčame voľne dostupné *Rstudio*<sup>1</sup>. Následne vykonaním všetkých riadkov v súbore `R/run.R` alebo spustením funkcie `source()` nad týmto súborom sa spustí proces analýzy ukážkových dát a výsledky sa automaticky zobrazia vo webovom prehliadači. Pri takto spustenej implementácii je možné v súbore `R/run.R` upravovať vstupné dáta funkcie `Magic.Pipeline()`. Vstupné parametre sú nasledovné:

`files` - zoznam FCS súborov na analýzu

`k` - počet zhľukov

`eps` - hodnota z intervalu  $[0, 100]$  určujúce na koľko percent sa majú analyzované vzorky zredukovať

`weight.adjusting` - príznak, či má byť v procese redukcie dát na základe hustoty vykonaný krok úpravy váh

`noise.removal` - príznak, či má byť v procese redukcie dát na základe hustoty vykonaný krok odstránenia šumu

`force.recalculation` - príznak, či má byť vynútené vykonanie procesu redukcie dát, aj keď existuje súbor so spracovanými dátami pre dané vzorky

## A.3 Používateľská príručka

Vizualizácia výsledkov sa zobrazí vo webovom prehliadači v rozhraní zobrazenom na obrázku A.1. Rozhranie je rozdelené na päť častí označených červenými číslami takto:

1. voľba vzorky / pacienta, pre ktorého ma byť zobrazená stromová vizualizácia výsledkov,
2. voľba cytometrického znaku, podľa ktorého hodnôt má byť na strom aplikovaná farba,
3. výsledný strom podľa zvolenej vzorky v 1 a zvoleného cytometrického znaku v 2,
4. použitá farebná škála na vizualizáciu,
5. dvojrozmerný bodový graf hodnôt zvolených cytometrických znakov

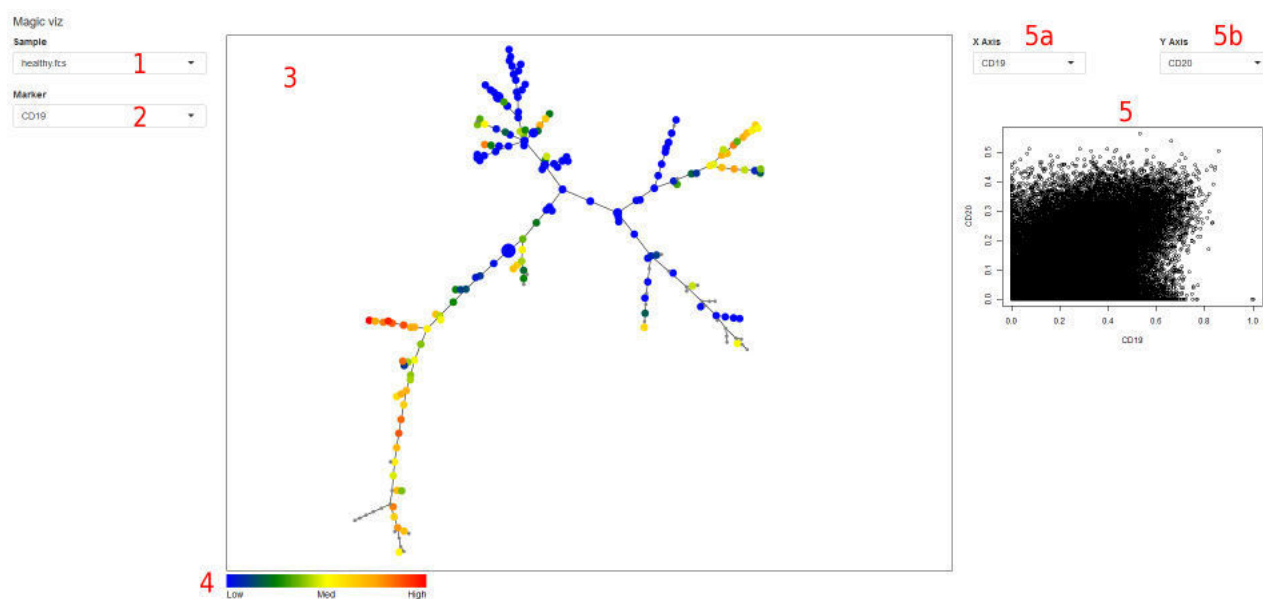
---

<sup>1</sup>[www.rstudio.com/](http://www.rstudio.com/)



- (a) voľba cytometrického znaku zobrazovaného na osi X,
- (b) voľba cytometrického znaku zobrazovaného na osi Y.

Rozhranie ponúka základe prvky interakcie na prácu s vizualizáciami. Kliknutím a potiahnutím je možné presúvanie vrcholov stromu vo vizualizácii. Podržaním klávesy `ctrl`, kliknutím na plochu vizualizácie a potiahnutím myši je možný výber a následný posun viacerých vrcholov naraz. Zrušenie hromadného výberu vrcholov je klávesou `ESC`. Po výbere viacerých vrcholov naraz sa hodnoty buniek vybratých vrcholoch zvýraznia na dvojrozmernom bodovom grafe (bod 5).



Obr. A.1: Používateľské rozhranie vizualizácií



# Príloha B

## Electronické médium

Ku diplomovej práci je priložené elektronické médium s nasledovnou štruktúrou súborov:

/Application

- implementácia opísaného riešenia

/Application/magic/data/examples

- ukážkové dáta

/Documentation

- bakalárska práca spolu s anotáciami v slovenskom a anglickom jazyku

/Documentation/Latex

- latex zdrojové súbory dokumentácie

/Documentation/BibTeX

- BibTeX súbor s použitými referenciami

/Documentation/Resources

- dostupné použité zdroje

read.me - popis obsahu média v slovenskom a anglickom jazyku



# Príloha C

## Plán práce

Do prvého kontrolného bodu DP1 sme si naplánovali zanalyzovať doménovú oblasť a problematiku a spísať do práce analýzu. Do nasledujúceho kontrolného bodu DP2 sme mali naplánované navrhnúť riešenie na nami vybraný zanalyzovaný doménový problém z DP1 a začať s implementáciou. Do posledného kontrolného bodu sme mali naplánované dokončenie implementácie, overenie výsledkov nami navrhnutého riešenia a dokončenie práce.

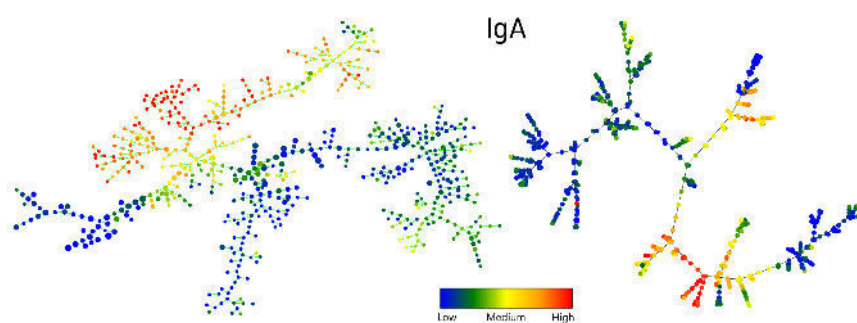
Plán práce sme splnili v plnom rozsahu, ako bol naplánovaný.



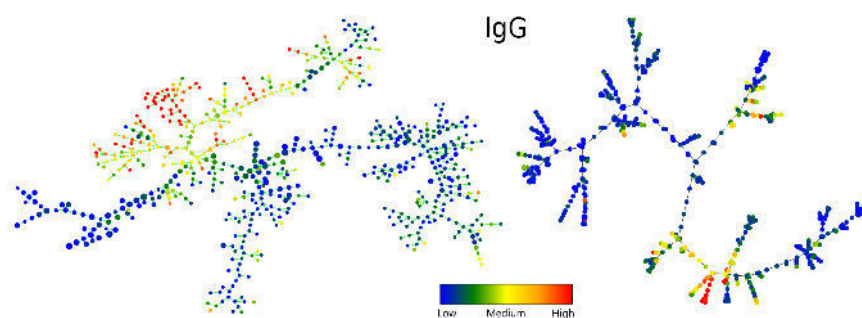
# Príloha D

## Vizualizácie

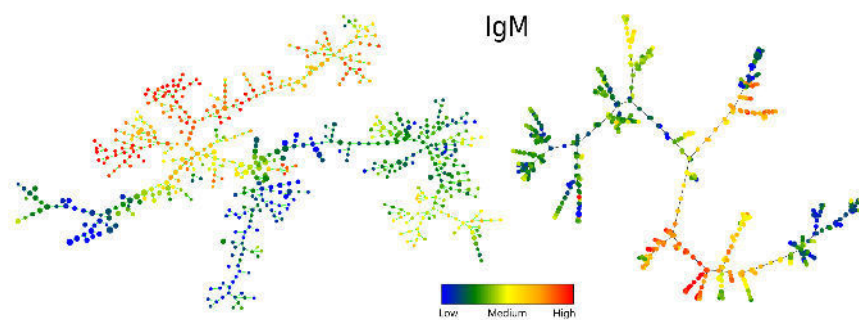
### D.1 Porovnanie stromových vizualizácií



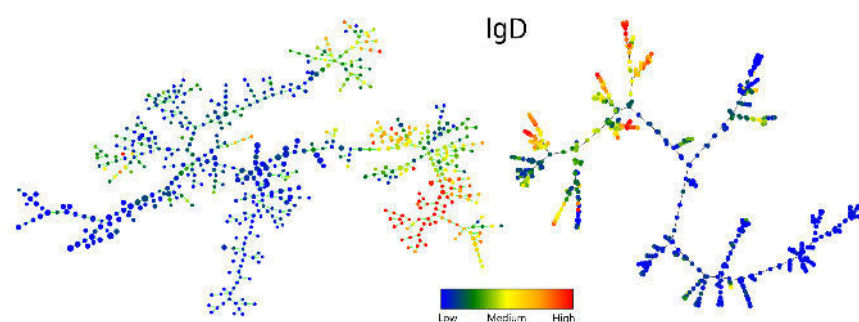
Obr. D.1: Porovnanie vizualizácie bunkových populácií podľa cytometrického znaku IgA



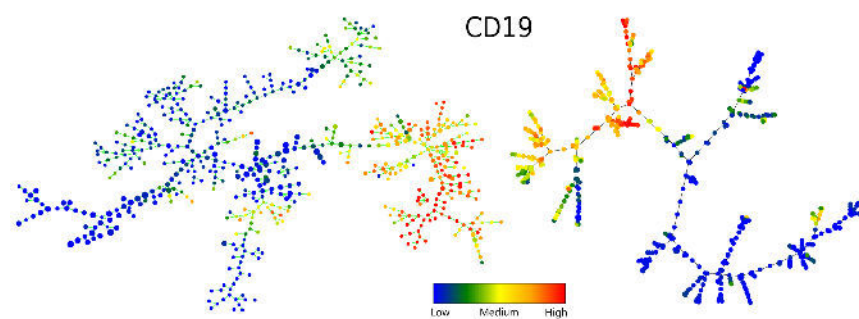
Obr. D.2: Porovnanie vizualizácie bunkových populácií podľa cytometrického znaku IgG



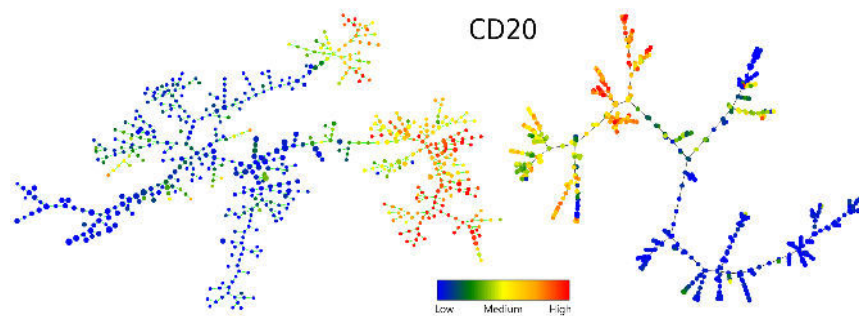
Obr. D.3: Porovnanie vizualizácie bunkových populácií podľa cytometrického znaku IgM



Obr. D.4: Porovnanie vizualizácie bunkových populácií podľa cytometrického znaku IgD

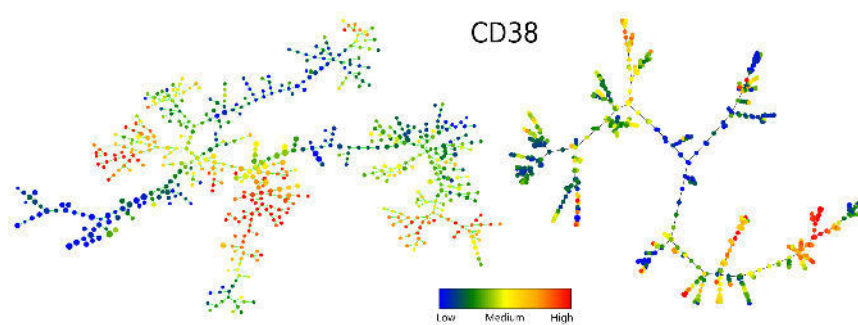


Obr. D.5: Porovnanie vizualizácie bunkových populácií podľa cytometrického znaku CD19

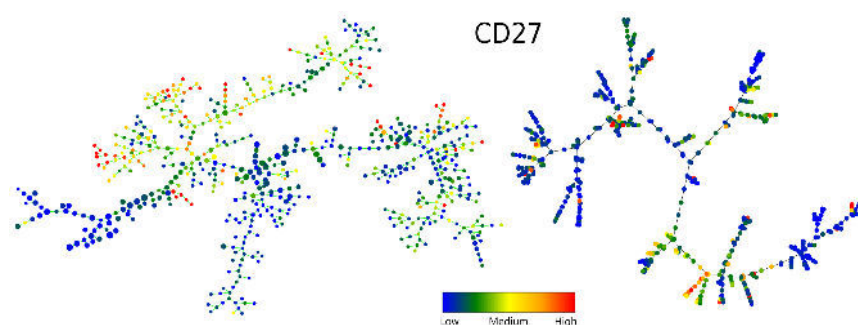


Obr. D.6: Porovnanie vizualizácie bunkových populácií podľa cytometrického znaku CD20

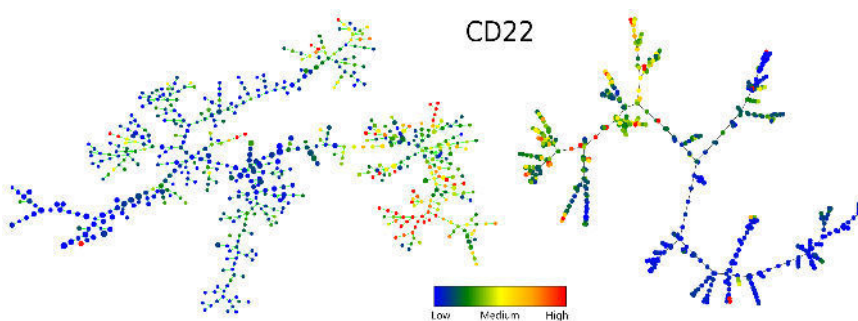




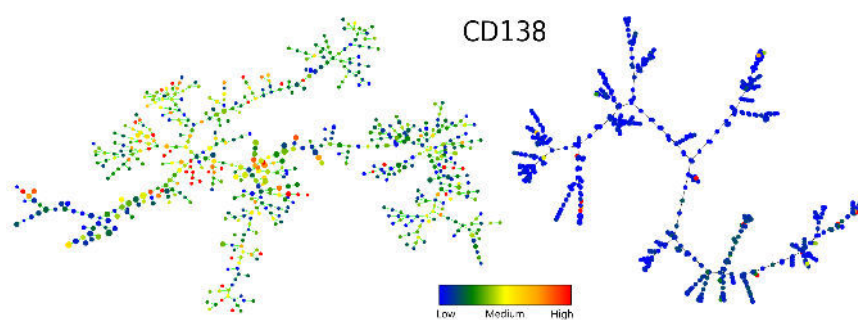
Obr. D.7: Porovnanie vizualizácie bunkových populácií podľa cytometrického znaku CD38



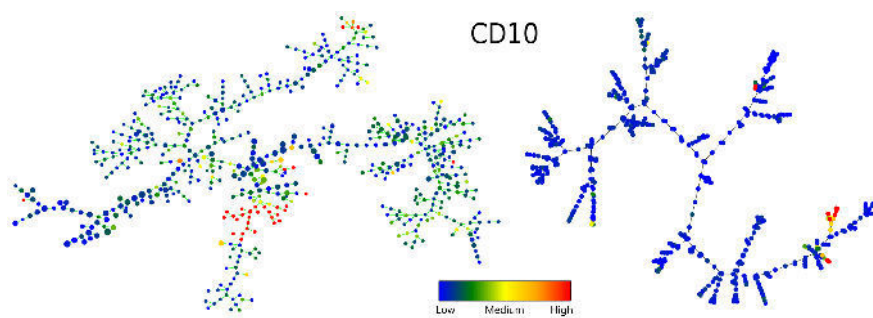
Obr. D.8: Porovnanie vizualizácie bunkových populácií podľa cytometrického znaku CD27



Obr. D.9: Porovnanie vizualizácie bunkových populácií podľa cytometrického znaku CD22

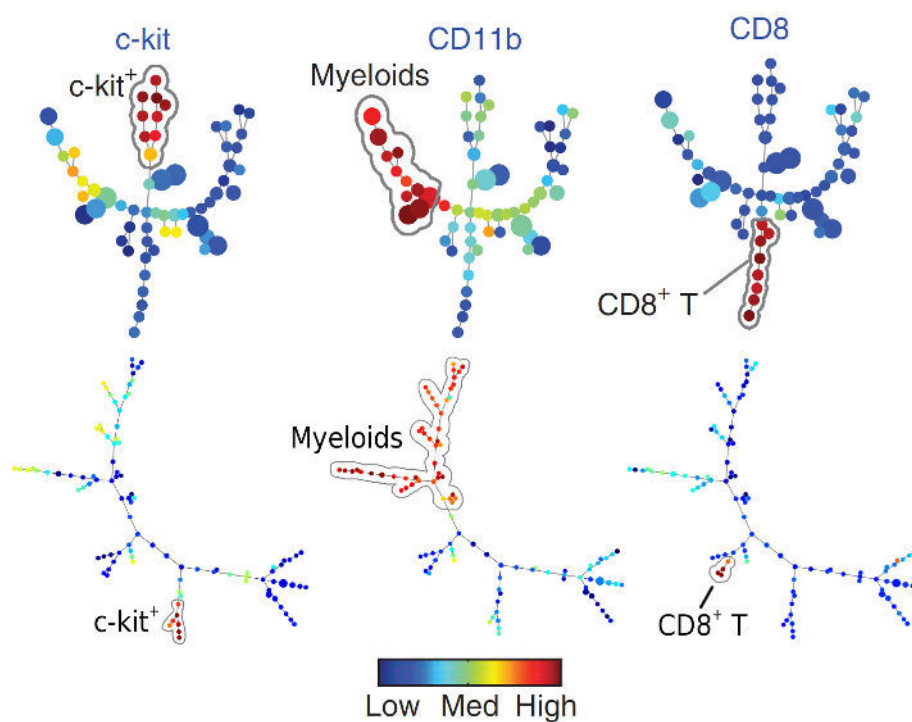


Obr. D.10: Porovnanie vizualizácie bunkových populácií podľa cytometrického znaku CD138



Obr. D.11: Porovnanie vizualizácie bunkových populácií podľa cytometrického znaku CD10

## D.2 Porovnanie vizualizácií na dátach kostnej drene myši



Obr. D.12: Porovnanie vizualizácií cytometrických znakov na dátach kostnej drene myši

# **Príloha E**

## **IIT.SRC článok**

# Fast density-based downsampling of cytometry data

Martin NEMČEK\*

*Slovak University of Technology in Bratislava  
Faculty of Informatics and Information Technologies  
Ilkovičova 2, 842 16 Bratislava, Slovakia  
mrt.n.nemcek@gmail.com*

**Abstract.** Identification of the cellular populations is one of the first and important steps in an analysis of cytometry data. To correctly identify both the abundant and the rare cellular populations a density-based preprocessing of data to equalize representations of the populations is needed. A density-based downsampling keeps representative points in the space while discarding the irrelevant ones based on their densities. Using stochastic approaches to make the algorithms usable on big datasets in real-time render the produced results irreproducible which is a key problem. Therefore we propose a fast and fully deterministic algorithm for density calculation based on the space partitioning and a tree representation and iterative approach to the downsampling utilizing fast calculation of the density.

## 1 Introduction

Cytometry focuses on measuring and analyzing of multiple parameters of cells. The ability to analyze cells has many various applications in medicine, biology and immunology to identify and quantify immune cell populations which allows monitoring of the patient's immune system and detecting novel biomarkers [4].

For measuring of cell parameters a mass cytometry is used which allows up to the 40 parameters to be measured for a single cell resulting in multidimensional big data sets [5, 6].

Traditional approach to analysis of cytometry data is a manual gating - identification of the cellular populations from two dimensional dot plots. This method has several issues such as subjectivity and irreproducibility of the results and the time consumption. Therefore approaches automating gating process using clustering algorithms were developed [1, 7, 8]. A cellular population can be viewed as a dense region of cells in space. However, abundant and rare cell populations

can have significantly different densities which might lead to incorrect identification of the rare cell populations using clustering algorithms. To tackle this issue a preprocessing of data consisting of the density-based downsampling is necessary. This step selects subset of original data with equalized density throughout the space [3].

## 2 Density-based downsampling of cytometry data

Measurement of cell parameters is viewed as a point in a point cloud. Density-based downsampling is performed based on the densities of points. Density of a point is equal to the number of points in its  $\epsilon$  neighborhood. Those points are also called the neighbor points. In such point cloud areas with high density corresponds to abundant cellular populations while areas with low densities to rare ones [3]. After calculation of the density of every point the density-based downsampling can be performed in such manner that representative points are kept while discarding others. The idea is to pick the most representative point and make a "hole" of size equal to the value of  $\epsilon$  neighborhood in point cloud with the center at the picked point and repeating the process until the desired result is reached [9].

Time complexity of such approach is heavily influenced by the time complexity of density calculation. Some methods like [3] utilize stochasticity to tackle the problem of the time complexity, which on the other hand render results irreproducible. Deterministic version exists but suffers from time complexity of such approach [2]. There is possibility of improvements in both density calculation and downsampling.

## 3 Proposed algorithm

We propose a fully deterministic fast algorithm for calculation of density using tree representation of the par-

\* Master study programme in field: Information Systems

Supervisor: doc. RNDr. Mária Lucká, PhD., Institute of Informatics, Information Systems and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

tioned space and an iterative approach to the density-based downsampling utilizing the fast calculation of density.

### 3.1 Density calculation

A density of a point is the number of other points in its neighborhood called the neighbor points. The size of the neighborhood is denoted by parameter  $\varepsilon$ . If we would want to calculate density of the point we would need to know distance to each other point to determine whether those two points are neighbors. This approach is inefficient on big data sets and thus the main idea of our proposed algorithm is a minimization of set of points with which we have to calculate distance to find out the density of a point.

By partitioning space in each dimension to intervals of size  $\varepsilon$  one can reduce the size of a set of the possible neighbor points — points with which a distance is needed to be calculated. If we consider a two dimensional space as on figure 1 normalized to  $[0, 1]$  and partitioned to intervals we get a grid of cells of size  $\varepsilon \times \varepsilon$ . Then if we get any point  $p$  in a cell  $c$  we can with certainty say that neighbor points of point  $p$  are only in neighbor cells of cell  $c$ . The truth of the assertion is based on fact that  $\varepsilon$ -neighborhood of any point in cell  $c$  would not extend further than to neighbor cells of cell  $c$ . That being said if we would need to calculate a density of point  $p$  we would need to calculate distances only to all points of neighbor cells of  $p$  point's cell.

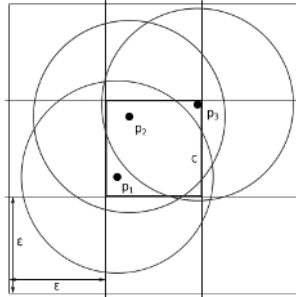


Figure 1. 2D space partitioned to intervals of size  $\varepsilon$

The points are placed into the cells based on their values at each dimension. At each dimension an index of  $\varepsilon$ -interval is calculated by formula

$$\varepsilon(p, d) = \text{int}\left(\frac{p_d}{\varepsilon}\right)$$

where  $p$  is the point,  $d$  is an index of the dimension,  $p_d$  is the value of the point  $p$  at the dimension  $d$  and  $\text{int}$  is a function returning integer part of a number. By calculating an index of  $\varepsilon$ -interval at each dimension we get a position of cell of point  $p$  in space.

If  $\varepsilon$  splits each of  $k$  dimensions to  $m$  intervals it creates  $m^k$  cells and even with rather small  $k$  we get  $n \ll m^k$  where  $n$  is number of points. Therefore the most of the cells would be empty while having enormous memory requirements. We propose a tree representation called a density tree to represent such space and utilize space partitioning by  $\varepsilon$  to calculate the densities of the points. Each level defines one dimension and the leaf nodes represent cells of partitioned space. Every node contains map of  $\varepsilon$ -interval indices of the next dimension. The density tree is being built sequentially as new points are pushed into it. When new point is pushed into the tree we iterate over dimensions of the point and for each dimension we calculate the  $\varepsilon$ -interval index and move point from the node representing  $\varepsilon$ -interval in current dimension to the next node. If such node does not exist then it is created. This sequential building of the density tree ensures that only non-empty cells are represented.

With  $k$  dimensions and root node not representing any, the height of the density tree is  $k + 1$ . By utilizing sequential building of the tree the worst case scenario is if all points would end up in their own nodes and therefore the width of the tree is  $O(n)$ . With the worst case scenario the tree would have  $n$  leaf nodes. To access those  $n$  leaf nodes in worst case scenario the tree would need another  $n$  nodes at level  $k - 1$ . This implies that number of nodes is  $O(k \times n + 1)$  and number of edges is  $O(k \times n)$ .

Assuming the value of  $\varepsilon$  large enough to partition the space in such manner that all points would end up in one leaf node then the time complexity of density calculation is  $O(\frac{n^2 - n}{2})$  because we would need to calculate the distances to all other points for every point while calculating the distance between two points only once and not calculating the distance for point with itself. However if we assume non-edge values of  $\varepsilon$  then the time complexity is further reduced due to creation of more smaller leaf nodes which results in reduction of sets of possible neighbor points for each point and resulting in fewer calculations needed.

By parallelization we can even further improve time complexity because calculation of densities of points of a node is independent of calculation of densities of points of the other nodes and therefore the calculations for each node can be performed simultaneously.

### 3.2 Weighted density

Density of a point is calculated by a formula  $\sum_1^m 1$  where  $m$  is number of neighbor points. Using such calculation of density can lead to some misinterpretations as seen on figure 2 where point  $p_1$  has the highest density while intuitively point  $p_2$  is more representative due to having more neighbor points close to itself.

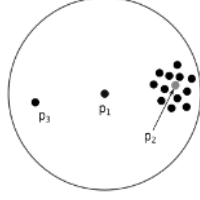


Figure 2. Misinterpretation of the most representative point

Therefor we propose weighted density where contribution of neighbor point to the density of a point is based on their distance and is calculated as  $c = (1 - \frac{d}{\varepsilon})^e$  where  $c$  is the contribution,  $d$  is the distance between the neighbor point and the point and  $\varepsilon$  is the size of the neighborhood. Accordingly a weighted density  $w$  of point  $p$  is calculated as

$$w(p) = \sum_{i=1}^M (1 - \frac{d(p, m_i)}{\varepsilon})^e$$

where  $M$  is number of the neighbor points of point  $p$  and  $d(p, m_i)$  is the distance between point  $p$  and its  $i$ -th neighbor point. Using this definition we ensure that closer neighbor points will contribute significantly more to the final weighted density of the point than neighbor points which lie near the edge of the neighborhood.

### 3.3 Iterative density-based downsampling

In our proposed algorithm the representativeness of a point is equal to the weighted density of that point (see 3.2) meaning that the more close neighbor points the point has the more representative it is. The process of downsampling is finding the most representative point among the points which were not chosen to be kept or discarded yet and keeping that point and discarding all neighbor points of the kept point and iterating until all the points are either kept or discarded. The time complexity of the density-based downsampling is largely affected by the time complexity of the density calculation because the downsampling is fast in comparison to the density calculation. Choosing the incorrect value of  $\varepsilon$  can lead to the slow density-based downsampling. Not to mention that to choose a correct value of  $\varepsilon$  an priori analysis of data is needed, further increasing the time complexity of the whole process. That being said we propose a iterative approach to density-based downsampling which vastly improves the time complexity of the whole density-based downsampling process and needs only the resulting percentage of kept points from data as input parameter.

The iterative approach utilizes one key observation and that is if a relatively small value of  $\varepsilon$  is chosen then most of the points would end up in their own leaf nodes with exception to the dense regions of space where one leaf node would contain more than one point. The time complexity of such partitioned space is very low (see 3.1) because very few calculations are needed to be performed for each point. During the downsampling only the irrelevant points from the dense regions would be discarded because most of the points would have few to non neighbor points.

The aforementioned observation is utilized in such a way that a relatively small value of  $\varepsilon$  is chosen and the densities are calculated and the points are density-based downsampled. Then if the current percentage of the kept points ( $p_c$ ) is equal to the  $\pm 1\%$  of the selected resulting percentage ( $p_r$ ) an algorithm terminates. Otherwise the value of  $\varepsilon$  is adjusted as:

$$\varepsilon = \begin{cases} \varepsilon \times 2, & p_c > p_r \\ \frac{\|\varepsilon - \varepsilon_p\|}{2}, & \text{otherwise} \end{cases}$$

where  $\varepsilon_p$  is value of  $\varepsilon$  from the previous iteration. By increasing value of  $\varepsilon$  in the next iteration more points are discarded and fewer are kept resulting in descending approaching to he resulting percentage otherwise by decreasing the value of  $\varepsilon$  fewer points are discarded and more are kept resulting in ascending approaching to resulting percentage. Also if  $p_c > p_r$  then only the kept points from the current iteration are used as input data to the next iteration otherwise if  $p_c \leq p_r$  then the points from last iteration where  $p_c > p_r$  was true are used as input data to the next iteration. Then the algorithm continues with calculating the densities and downsampling on the new data from previous iteration.

Another key observation which the iterative approach utilizes is that performing the density calculation and the density-based downsampling on the kept points from previous iterations further improves time complexity because when algorithm proceeds to relatively big value of  $\varepsilon$  where calculation of densities could take up more time, the input data is already reduced resulting in fewer calculations needed and speeding up the whole process.

Furthermore to improve results we perform weight adjusting such that to value of weighted density of each point is added a half of the value of weighted density of that point from previous iteration. This step helps to minimize the number of "holes" in resulting space containing no points while there should be at least one point.

The resulting set of the iteratively density-based downsampled points are the kept points from the last iteration where the condition  $p_c = p_r \pm 1$  was met.

## 4 Results

We compared our proposed algorithm with SPADE [2] which is the most used software tool for the density-based downsampling of cytometry data. We compared time needed for the density calculation and the density-based downsampling. For testing we used multiple datasets with varying number of points including smaller and also bigger datasets all with 13 dimensions.

First we compared the performance of a density calculation of both SPADE and our proposed algorithm *densamp*. For consistent results and comparisons we let SPADE calculate the value of  $\varepsilon$  which we then used to run the density calculation by SPADE and *densamp*. In table 1 are times in minutes needed for calculation of densities by SPADE algorithm and our algorithm using the same value of  $\varepsilon$  and also the number of points for each dataset. Using smaller datasets the differences are not meaningful. However with bigger datasets the differences become significant. If we would plot results of our algorithm we could see that they do not follow a curve due to dependency on the value of  $\varepsilon$  and space partitioning by that value (see 3.1).

Table 1. Time in minutes needed for calculation of densities

	~81k	~248k	~382k	~491k	~624k
spade	0.683	7.65	16.883	26.36	38
densamp	0.666	3.35	3.116	21.26	18

Next we compared times needed to the perform density-based downsampling. We chose 10% as the resulting percentage of points after downsampling for both SPADE and our algorithm. Table 2 shows time in minutes needed to perform density-based downsampling. Here we can see the significant differences between SPADE and our proposed algorithm resulting from using iterative approach which utilizes calculating densities with relatively small value of  $\varepsilon$  at first and iteratively approaching the result while reducing the data in each iteration, further reducing time to calculate densities in next iteration.

Table 2. Time in minutes needed for density-based downsampling

	~81k	~248k	~382k	~491k	~624k
spade	0.939	9.732	21.783	38.95	58.083
densamp	0.156	0.689	0.84	3.419	2.024

## 5 Conclusions and future work

The existing approaches to density calculation and density-based downsampling of cytometry data use

either the stochastic methods to improve their time complexity or perform badly on big data sets. We proposed a fast and deterministic way to calculate the densities based on space partitioning with tree representation and an iterative approach to density-based downsampling utilizing the fast density calculation. We compared our algorithm with the most used software tool for density-based downsampling and our proposed enhancements resulted in significantly improved time complexity on big data sets.

As the next step we plan on extracting the features from the density-downsampled data and combine them with clinical data to perform clinical status prediction. Another possible improvement of our algorithm is to incorporate noise removal to the iterative density-based downsampling step.

*Acknowledgement:* This work was partially supported by the APVV project, No. APVV-16-0484.

## References

- [1] Li, H., Shaham, U., Stanton, K.P., Yao, Y., Montgomery, R.R., Kluger, Y.: Gating mass cytometry data by deep learning. *Bioinformatics*, 2017, vol. 33, no. 21, pp. 3423–3430.
- [2] Qiu, P.: Toward deterministic and semiautomated SPADE analysis. *Cytometry. Part A : the journal of the International Society for Analytical Cytology*, 2017, vol. 91, pp. 281–289.
- [3] Qiu, P., Simonds, E.F., Bendall, S.C., Gibbs Jr, K.D., Bruggner, R.V., Linderman, M.D., Sachs, K., Nolan, G.P., Plevritis, S.K.: Extracting a cellular hierarchy from high-dimensional cytometry data with SPADE. *Nat Biotech*, 2011, vol. 29, no. 10, pp. 886–891.
- [4] Saeys, Y., Gassen, S.V., Lambrecht, B.N.: Computational flow cytometry: helping to make sense of high-dimensional immunology data. *Nat Rev Immunol*, 2016, vol. 16, no. 7, pp. 449–462.
- [5] Spitzer, M., Nolan, G.: Mass Cytometry: Single Cells, Many Features. *Cell*, 2016, vol. 165, no. 4, pp. 780–791.
- [6] Tanner, S.D., Baranov, V.I., Ornatsky, O.I., Bandura, D.R., George, T.C.: An introduction to mass cytometry: fundamentals and applications. *Cancer Immunology, Immunotherapy*, 2013, vol. 62, no. 5, pp. 955–965.
- [7] Verschoor, C.P., Lelic, A., Bramson, J.L., Bowdish, D.M.E.: An Introduction to Automated Flow Cytometry Gating Tools and Their Implementation. *Frontiers in Immunology*, 2015, vol. 6, p. 380.
- [8] Weber, L.M., Robinson, M.D.: Comparison of clustering methods for high-dimensional single-cell flow and mass cytometry data. *Cytometry Part A*, 2016, vol. 89, no. 12, pp. 1084–1096.
- [9] Zare, H., Shooshtari, P., Gupta, A., Brinkman, R.R.: Data reduction for spectral clustering to analyze high throughput flow cytometry data, 2010.

