VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

DEPARTMENT OF BIOMEDICAL ENGINEERING

NAVRŽENÍ GENOVÉ REGULAČNÍ SÍTĚ NA ZÁKLADĚ VZÁJEMNÉ INFORMACE U NEMODELOVÝCH ORGANISMŮ

GENE REGULATORY NETWORK INFERENCE BASED ON MUTUAL INFORMATION IN NON-MODEL ORGANISMS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Petr Pirkl

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Jana Musilová

BRNO 2022

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Diplomová práce

magisterský navazující studijní program **Biomedicínské inženýrství a bioinformatika**

Ústav biomedicínského inženýrství

Student: Bc. Petr Pirkl ***ID:*** 203682

Ročník: 2 ***Akademický rok:*** 2021/22

NÁZEV TÉMATU:

Navržení genové regulační sítě na základě vzájemné informace u nemodelových organismů

POKYNY PRO VYPRACOVÁNÍ:

1) Vypracujte literární rešerši metod pro navržení genové regulační sítě. Zaměřte se i na metody založené na výpočtu vzájemné informace. 2) Seznamte se s laboratorními metodami pro stanovení genové exprese. 3) Předzpracujte dataset obsahující hodnoty genové exprese. 4) Navrhněte algoritmus pro navržení genové regulační sítě. 5) Implementujte navržený algoritmus v libovolném programovacím jazyce. 6) Algoritmus otestujte na předzpracovaném datasetu. 7) Proveďte vyhodnocení a diskutujte výsledky.

DOPORUČENÁ LITERATURA:

[1] MERCATELLI, Daniele, Laura SCALAMBRA, Luca TRIBOLI, Forest RAY a Federico M. GIORGI, 2020. Gene regulatory network inference resources: A practical overview. Biochimica et Biophysica Acta (BBA) - Gene Regulatory Mechanisms. 1863(6). ISSN 18749399.

[2] BARBOSA, Sara, Bastian NIEBEL, Sebastian WOLF, Klaus MAUCH a Ralf TAKORS, 2018. A guide to gene regulatory network inference for obtaining predictive solutions: Underlying assumptions and fundamental biological and data constraints. Biosystems. 174, 37-48. ISSN 03032647.

Termín zadání: 7.2.2022 ***Termín odevzdání:*** 20.5.2022

Vedoucí práce: Ing. Jana Musilová

prof. Ing. Ivo Provazník, Ph.D.
předseda rady studijního programu

ABSTRAKT

Práce se zabývá shrnutím základních laboratorních metod pro stanovování genové exprese, postupy předzpracování dat a nástroji používanými k odvozování genových regulačních sítí. Dále se práce zabývá samotným předzpracováním dat, tedy vytvořením matice počtů a její normalizace s využitím dat nemodelového organismu *Clostridium beijerinckii* NRRL B-598. Hlavní částí práce je poté navržení algoritmu pro tvorbu genové regulační sítě s využitím vzájemné informace a jeho implementace v jazyku R, včetně testování na datech nemodelového organismu i gold standardu.

KLÍČOVÁ SLOVA

Genové regulační sítě, vzájemná informace, genová exprese, gen, RNA-Seq

ABSTRACT

The thesis is focused on summary of laboratory methods for determining gene expression, data preprocessing procedures and possible tools used to infere gene regulatory networks. Furthermore, the thesis handles with the pre-processing of data. It means create count table and normalize it. It was use data from the non-model organism *Clostridium beijerinckii* NRRL B-598. The main parts of the thesis are designed an algorithm for the creation of a gene regulatory network using mutual information and its implementation in the R language. This include testing the algorithm on data from the non-model organism and the gold standard.

KEYWORDS

Gene regulatory networks, mutual information, gene expression, gene, RNA-Seq

PIRKL, Petr. *Navržení genové regulační sítě na základě vzájemné informace u nemodelových organismů*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav biomedicínského inženýrství, 2022, 64 s. Diplomová práce. Vedoucí práce: Ing. Jana Musilová

Prohlášení autora o původnosti díla

Jméno a příjmení autora:	Bc. Petr Pirkl
VUT ID autora:	203682
Typ práce:	Diplomová práce
Akademický rok:	2021/22
Téma závěrečné práce:	Navržení genové regulační sítě na základě vzájemné informace u nemodelových organismů

Prohlašuji, že svou závěrečnou práci jsem vypracoval samostatně pod vedením vedoucí/ho závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

podpis autora*

*Autor podepisuje pouze v tištěné verzi.

PODĚKOVÁNÍ

Chtěl bych poděkovat vedoucí diplomové práce paní Ing. Janě Musilové za odborné vedení, doporučenou literaturu, poskytnutá data, trpělivost se všemi otázkami, návrhy a připomínky k textu práce, navrhovanému algoritmu a jeho implementaci.

Brno ....

podpis autora

Obsah

Seznam obrázků

Seznam tabulek

Úvod

Téma diplomové práce je Navržení genové regulační sítě na základě vzájemné informace u nemodelových organismů. Návrh genových regulačních sítí je velké téma pro dnešní systémovou biologii a bioinformatiku. Sítě fungují zejména pro reprezentaci jednotlivých regulačních drah. Jejich základem je tedy zjištění genové regulace. Objevení genových regulačních drah má velký význam pro pochopení fungování organismů a pro studium jejich biologických funkcí. Jejich znalost následně nachází uplatnění zejména v genovém inženýrství.

Navržený algoritmus by mohl najít využití při prvním pohledu na genovou regulační síť nemodelového organismu. Při prvním zkoumání máme k dispozici pouze dostupná data genové exprese v různých časech a událostech životního cyklu organismu.

Práce je členěná do sedmi kapitol, kde první kapitola nás uvádí do tématu genové exprese, na což navazuje druhá kapitola s teoretickým popisem laboratorních technik zjišťování genové exprese. Třetí kapitola popisuje data, která byla použita pro testování algoritmu a v případě nemodelového organismu *Clostridium beijerinckii* NRRL B-598, která byla také předzpracována (tj. zjištění matice počtů a normalizace). Předzpracováním dat se zabývá následující kapitola. Pátá kapitola je teoretickým úvodem k navrhování algoritmu. Jsou zde popsány jednotlivé přístupy k tvorbě genových regulačních sítí, včetně námi využívané vzájemné informace. Poslední dvě kapitoly popisují navržený algoritmus a hodnotí jeho funkčnost a úskalí.

Cílem diplomové práce je tedy provést předzpracování dat nemodelového organismu. Dále navrhnout algoritmus, který využívá vzájemnou informaci pro tvorbu genových regulačních sítí a také ho implementovat do programovacího jazyka. A nakonec otestovat jeho funkčnost a využití na předzpracovaných datech nemodelového organismu.

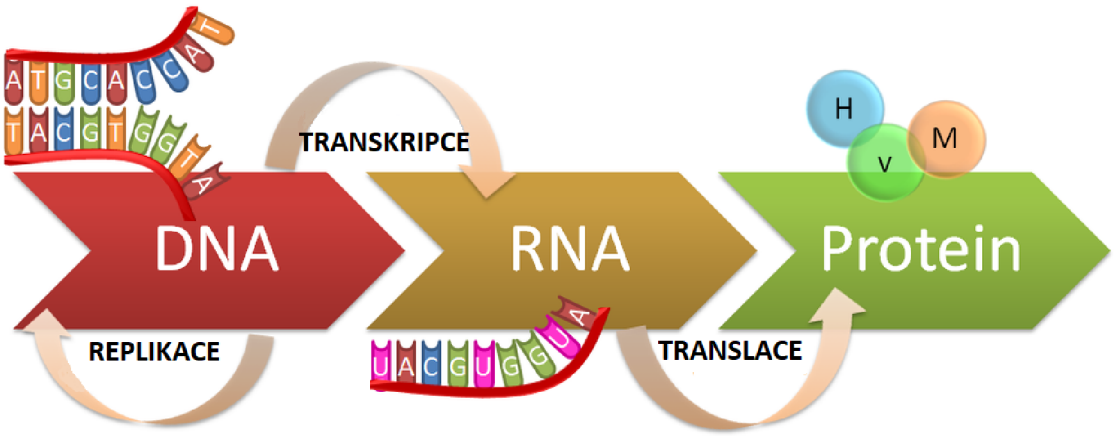
1 Genová exprese

Proces genové exprese nám informaci ze sekvencí genu převádí do funkčních produktů. DNA může být transkribována na RNA. Z RNA následně pomocí translace mohou vznikat funkční produkty, jako jsou proteiny nebo některé typy RNA. Proces genové exprese probíhá ve všech organismech a je regulován na několika úrovních. Genová exprese je tedy základem vývoje a diferenciace organismů. [1]

1.1 Ústřední dogma molekulární biologie

Základem pro pochopení fungování genové exprese je vysvětlení centrálního dogmatu molekulární biologie.

Víme, že veškerá genetická informace o organismech je kódována v DNA. Z DNA je pomocí transkripce přepsána do mRNA. Následně pomocí translace na ribozomech vznikají z RNA proteiny, které plní v organismu mnoho funkcí. [2]



Obr. 1.1: Centrální dogma molekulární biologie [3]

S větší znalostí biologických systémů víme, že složitost organismů a jejich fungování je výsledkem víceúrovňové, dynamicky řízené sítě regulace genové exprese [4]. Základním procesem a zároveň hlavním kontrolním bodem genové exprese je tedy transkripce. Pro nás je velmi důležité určit klíčové hráče, abychom pochopili, čím je hladina genové exprese regulována. [5, 6] Přičemž transkripční faktory váží genomovou informaci v komplexech, čímž regulují úroveň transkripce cílových genů. Tyto událostí nám vytváří regulační síť, jenž je následně možné zobrazit. Velké množství dat o regulačních sítích, které získáme z experimentů, je potřeba zpracovat. O některých možnostech zpracování hovoří kapitola 4 Předzpracování dat.

1.2 Reverzní inženýrství

Odvozování genových regulačních sítí (GRN) je takzvaným „problémem reverzního inženýrství.“ Reverzní inženýrství bývá také označováno jako zpětné inženýrství. Obecně můžeme reverzní inženýrství definovat jako proces, během kterého se na základě zpětné analýzy snažíme porozumět principu fungování daného objektu. Jeho cílem je navrhnout po pozorování systému model, který se snaží, co nejvěrněji kopírovat funkce systému. Obrácený problém je obecně neřešitelný pro nelineární systémy [6].

V souvislosti s GRN můžeme označit reverzní inženýrství jako rekonstrukci GRN, jejímž cílem je určit potenciální regulační vztahy mezi geny. [7]

I když bylo vyvinuto již několik metod určených k odvozování GRN, jedná se stále o velmi náročnou problematiku. Výzvou pro dnešní studie je odvozování GRN pouze s využitím informací získaných z genové exprese. Jedná se o obtížný úkol z důvodu nedostatečné přesnosti měření genových expresí, což způsobuje vznik zašuměných dat, a také z důvodu enormního objemu genů a malé velikosti vzorku. Proto je potřeba, aby vznikaly další účinnější metody pro odvozování GRN. [8]

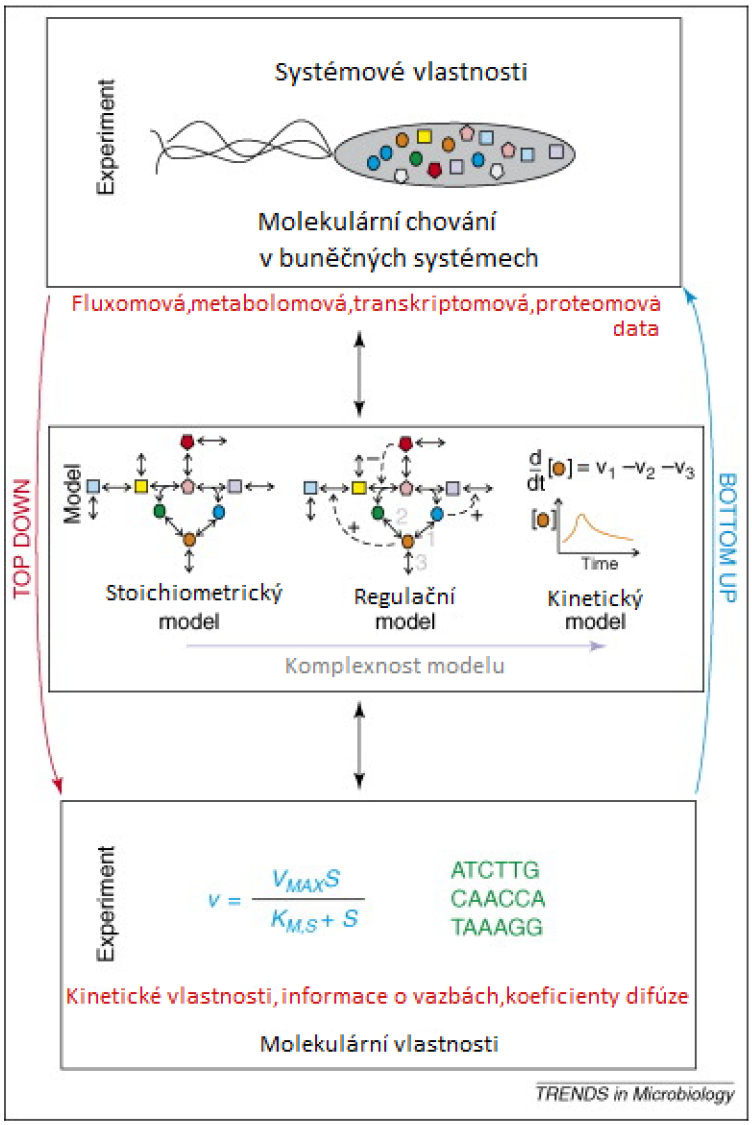
1.3 Přístup top-down a bottom-up

V dnešní době je stále velkou výzvou pochopit fungování organismů, dále objevení a popsání interakcí a jejich dynamik. A také zjišťování systémových informací a propojení mezi koncentracemi molekul a jejich fyziologií organismu. K zjišťování těchto informací můžeme využít přístupů top-down a bottom-up. Tyto přístupy můžeme zjednodušeně popsat pomocí obrázku 1.2.

1.3.1 Přístup top-down

Top-down přístup bere vzorek jako celek a začíná jeho analýzu pohledem na něj. Následně z celkového pohledu, který dále dělí, se dostává až k podstatě systému a jeho fungování. Dělení probíhá na základě minimálních nákladů a stejně jako u přístupu bottom-up je algoritmus zastaven až při splnění definovaného kritéria. [9] Jedná se o využívanou metodu, jelikož je díky ní možné přistupovat k biologickým systémům komplexně. [10]

Začínáme tedy s daty celých genomů z experimentů a z nich se snažíme zjistit vazby mezi geny, tedy interakce mezi nimi. Často při tomto přístupu nebereme v potaz již dříve získané biologické informace o vztazích mezi molekulárními složkami. Tento přístup pracuje s velkým objemem dat. Ta jsou získávána sekvenováním, které se na přelomu století v důsledku snížení ceny velmi rozšířilo [30]. Je však časté,



Obr. 1.2: Přístup top-down a bottom-up [10]

zejména v pozdějších fázích experimentu, že se výzkum zaměří na část metabolismu, genomu apod. - tzv. subsystémy, aby byl získán detailnější popis systému. Je ovšem nutné znát úskalí těchto experimentů. Dále experimentální data podrobíme navrženým analýzám, abychom stanovili korelace mezi jednotlivými velikostmi genové exprese. Následně formulujeme hypotézu o vzájemné regulaci mezi geny. Tyto hypotézy mohou předpovídat nové korelace mezi daty. Jelikož top-down přístup je iterativní, tak v dalších iteracích jsou navrženy a testovány nové experimenty, které se mohou více zaměřit na formulovanou hypotézu, nebo mohou být nastaveny jiné analýzy dat. [10]

Při zkoumání organismu je také důležité zkoumat fungování organismu při růz-

ných poruchách. Sem patří například genetické poruchy, jako jsou mutace, výrazná exprese proteinu, či environmentální vliv, kam patří například změna přísunu živin, vliv různých podávaných látek a další. Cílem studií následně bývá zjistit, co nejvíce informací o daném organismu a jak se chová za různých podmínek, aby byly prediktivní a popsaly všechny funkční mechanismy. Těchto mechanismů je ovšem velké množství a je obtížné připravit dostatek nezávislých experimentů. Jedná se zde tedy o komplexní problém, který je velmi obtížně řešitelný. [10]

Přístup top-down je tedy používán k tvorbě genových regulačních sítí a je také využíván v této práci, kdy námi navržený algoritmus pracuje s genovou expresí a pomocí informací získaných z ní navrhuje genovou regulační síť.

1.3.2 Přístup bottom-up

Bottom-up (v české literatuře se můžeme setkat s pojmem zdola nahoru) je přístup, ve kterém se s každým vzorkem pracuje jako se samostatným segmentem. Tyto segmenty jsou postupně spojovány na základě výpočtu nejnižšího nákladu na sloučení. Spojovány jsou tedy sousední segmenty s minimálními náklady. Tento proces je ukončen po dosažení stanoveného kritéria. [9]

Odvozují se zde funkční vlastnosti subsystému, které by mohly být charakterizovány do vyšších úrovní systému. Nejdříve definujeme interaktivní chování (jako jsou rychlostní rovnice) dílčích procesů částí systému. Tyto vlastnosti a jednotlivé menší modely spojujeme a snažíme se z nich předpovědět chování celého systému. [10]

K tomuto přístupu je samozřejmě nutné použít i specifických experimentů a analýz. Využívají se například studie jednotlivých složek izolovaně, kdy získáváme povědomí o jejich kinetice, či fyzikálně-chemických vlastnostech. To získáváme vytvářením a testováním modelů podle dat, které získáme testováním subsystémů při poruchách, jak bylo uvedeno v kapitole 1.3.1. Přístup top-down. Tyto přístupy jsou náchylné na nepřesnosti, jelikož je velmi obtížné popsat parametry kinetiky a jsou často zjednodušovány a je také velmi obtížné tyto parametry měřit. Možností, jak parametry měřit, mohou být experimenty *in vitro*, ty ovšem mohou být neproveditelné. Z toho důvodu by přesnou metodu pro měření kinetiky bylo měření *in vivo*. Zde je ovšem problém proveditelnosti experimentu, i když s vývojem zobrazovacích technik již tato měření nejsou nemožná. [10]

2 Laboratorní metody pro stanovení genové exprese

V této kapitole si blíže představíme základní laboratorní metody pro stanovení genové exprese. Právě data genové exprese jsou vstupními daty našeho algoritmu. Hned první podkapitola se zaměřuje na laboratorní metodu RNA-Seq, která byla použita pro získání genové exprese organismu *Clostridium beijerinckii* NRRL B-598, jenž byl použit k testování algoritmu.

2.1 RNA-Seq

RNA-sekvenování (RNA-Seq) neboli sekvenování celého transkriptomu patří mezi metody tzv. nové generace sekvenování (next generation sequencing). RNA sekvenování dokáže systematicky a celkově analyzovat transkriptom organismu pouze s minimálním zkreslením. [11]

Například oproti metodě microarray můžeme u RNA-Seq pozorovat několik výhod, jako například širší rozsahy měření, menší šum a vyšší propustnost. RNA-Seq nám dává obdobné informace jako microarray, ale zde získaná data bývají přesnější. [11]

Oproti DNA microarray RNA-Seq získává celou informaci o RNA, takže tím získáme znalosti o splicingu, isoformách apod. Oproti jiným metodám nám RNA-seq u nemodelových organismů dává možnost sestavit genom *de novo*. Nejsme ani omezeni délkou fragmentů a tato laboratorní metoda má dynamický rozsah. [12]

Prvním krokem před samotným sekvenováním musí být izolace celkové RNA z buňky. Dále je potřeba provést frakcionaci, přepis a tvorbu cDNA knihovny s připojením adaptorů na sekvence. Někdy je důležité provést amplifikaci (obzvláště při nízkých koncentracích vzorku) a kontrolu kvality knihovny. Následně již provádíme sekvenování a to buď *de novo* a nebo můžeme využít modelových organismů a známé informace při mapování na referenční sekvence. [12]

Existuje několik technik sekvenování, jak uvádí [12]:

Nejpoužívanější technologií je Illumina, která využívá 4 fluoroforů. Jdou zde přidávány nukleotidy značené fluorofory. Tyto nukleotidy jsou chráněny před navázáním více jednotek. Následuje skenování fluorescence a následné omytí vzorku. Na to navazuje znovu přidání roztoku se značenými nukleotidy. Tento proces se několikrát opakuje. Tuto metodu můžeme nazývat „sekvenování syntézou."

Ion Torrent také využívá sekvenování syntézou, ovšem tato metoda měří změnu pH. Sekvence je ukotvena v jamce na polovodičovém čipu. Čip je omýván roztokem s jedním nukleotidem, a pokud se daný nukleotidy naváže k sekvenci, tím je uvolněn

proton, který způsobí změnu pH a ta je měřena. Proces je obdobně jako u Illuminy několikrát opakován.

Další technika Roche 454 je založena na pyrosekvenování. Opět se k sekvenci umístěné na kotvící kuličce, která je uchycena v pikotitrační destičce z optických vláken, přiřazují nukleotidy. Po jejich navázání je uvolněn pyrofosfát, který s ATP-sulfurylasou vytvoří ATP. Enzym luciferasa díky ATP přemění luciferin na oxyluciferin a uvolní světelné kvantum. To je zaznamenáno na detektoru.

Poslední metodou a nejnovější technologií je Nanopore. Tato metoda je přínosnější tím, že dokáže sekvenovat čtení delší jak 1000 nt (až 150 tisíc nt), což předchozí metody nedokázaly. Přínosná je také nízkou cenou sekvenování a malou velikostí přístroje, který díky tomu umožňuje uplatnění při práci v terénu. Problémem této technologie je však současně vysoká chybovost (až 12 %).

2.2 DNA microarray

DNA microarray (DNA čipy) využívá právě čipů, což jsou tuhé povrchy z různých materiálů (skla, plastu, silikonu), na nichž jsou dopředu podle experimentu, který chceme provést, navázány známé a konkrétní fragmenty DNA. Ty umožňují hybridizaci značených nukleových kyselin. Izolovaná RNA (může být převedena na cDNA) je očištěna a následně je značená například pomocí fluorescenčně značených nukleotidů. Takto připravený roztok je následně již hybridizován na připravený čip a promytím jsou z něj odstraněny nenavázané molekuly. Nyní již probíhá skenování čipu, k čemuž se využívá konfokální mikroskop s laserem. Problémem DNA microarray může být to, že dojde k hybridizaci na čip podobných sekvencí a tím mohou být zkresleny výsledky, či ve směsných vzorcích může mít vliv pozadí, či při saturaci vzorkem nedokážeme určit přesnou hladinu. [12]

2.3 Sériová analýza genové exprese (SAGE)

Sériová analýza genové exprese dokáže současně získat expresi velkého množství transkriptů současně. SAGE funguje tak, že mRNA (cDNA) je reprezentována pomocí krátkých úseků sekvencí neboli značek (tagů) s definovanou pozicí o délce 10–17 bp. Ty jsou následně ligovány do konkatemerů, aby bylo možné analyzovat, co největší množství tagů současně a mohlo proběhnout sekvenování. Dalšími bioinformatickými úpravami jsou vytvořeny tabulky počtů tagů. [12]

Metoda SAGE se ovšem potýká s některými problémy a má několik nevýhod. Mezi ně patří potřeba velkého množství vstupní mRNA, obtížnost analýzy neznámých vzorků a sestavování SAGE knihoven, znečištění vzorků molekulami linkerů

z použité PCR pro sekvenování, velká chybovost při sekvenování a tím velké procento chybně určených bází. [13] Ovšem existuje také několik specializovaných úprav původní SAGE metody, které některé problémy řeší, jako jsou například miRAGE, SuperSAGE nebo MicroSAGE. [12]

2.4 PCR s revezní transkripcí (RT-PCR)

Polymerázovou řetězovou reakci s reverzní transkripcí lze také využít k získávání informací o genové expresi. PCR amplifikuje DNA a dokáže z malého množství vytvořit velké množství kopií DNA [12]. PCR s reverzní transkripcí dokáže použít při vstupu do reakce RNA. RNA musí být izolovaná z buňky a pomocí nukleáz z ní jsou odstraněny zbytky DNA. Připravená RNA je nejdříve převedena na cDNA pomocí reverzní transkriptázy. cDNA vstupuje jako templát do PCR a probíhá PCR normálním způsobem, přičemž je zde detekováno množství amplifikovaného produktu. V termocykleru proběhne několik teplotních cyklů, kde se postupně DNA denaturuje při vysoké teplotě (kolem 94 °C a 10-30 s) a následně dochází k annealingu primerů (přibližně teplota 50 -68 °C po 10-60 s). Posledním krokem je syntéza komplementárního řetězce DNA díky primerům, které připojují volné nukleotidy k řetězci DNA. K detekování amplifikované DNA se využívají fluorescenční reportérové molekuly, jako je například Taqman sonda. V kvantitativních termocyklerech je následně detekována fluorescence, která je přímo úměrná množství amplifikované DNA. [14]

2.5 Fluorescenční *in situ* hybridizace (FISH)

Jedná se o molekulárně genetickou metodu, která nám umožňuje stanovit určitý úsek DNA a jeho pozici. Úvodem je potřeba připravit sondu. Sonda je komplementární sekvence ke zkoumané sekvenci DNA, která je značena fluorescenční značkou. Často jsou využívány komerčně připravené sondy. Tyto sondy jsou aplikovány na vzorek. Následně je vzorek denaturován, a poté dochází k hybridizaci. Sondy (kratší sekvence) se komplementárně vážou na cílové sekvence v genu. Počet navázaných kopií na zkoumanou DNA a jejich pozice jsou detekovány ve fluorescenčním mikroskopu. [15]

3 Vstupní data

V této kapitole budou popsána data, která byla využita k testování algoritmu. Byla použita data z experimentu s *Clostridium beijerinckii* NRRL B-598 a také data z DREAM4 *in silico* challange.

3.1 *Clostridium beijerinckii* NRRL B-598

Kmeny *Clostridium* řadíme mezi přísně anaerobní sporulující bakterie. Bakterie spadající do této skupiny jsou jak toxické a patogenní, tak i průmyslově využitelné nepatogenní druhy. Solventogenní nepatogenní *Clostridia* jsou studována pro možné využití jejich metabolismů produkujících chemické sloučeniny, které jsou v současnosti vyráběny zejména z ropy. Přínosné je také to, že zde existuje snaha zlepšit využití odpadních produktů jako substrátu pro tyto bakterie. [16] Ovšem různé rody využívají různé substráty a produkují různé chemické látky a využívají rozdílné regulační dráhy, proto nelze již získané informace obecně aplikovat. [17]

Kmeny *Clostridia* zahrnují také nepatogenní bakterii *Clostridium beijerinckii*. Tato, ne příliš prozkoumaná, bakterie získala svoji pozornost při pokusech o nalezení řešení ekologické a energetické stability světa, proto je velmi významná pro průmysl, a to díky schopnosti produkovat butanol. Původní druhový název této bakterie byl *pasteurianum*. Pod tímto označením byl sestaven genom *Clostridium beijerinckii* NRRL-598 v roce 2015. [16] *Clostridium beijerinckii* NRRL-598 využívá oproti dalším kmenům širší škálu substrátu pro solventogenezi. Je také odolnější na nepříznivé prostředí a je tedy jako anaerobní bakterie tolerantní například na přítomnost kyslíku. [17, 16]

3.1.1 Experiment s *Clostridium beijerinckii* NRRL B-598

Data použitá k testování algoritmu byla získána z experimentu s *Clostridium beijerinckii* NRRL B-598 z analýzy RNA-Seq. Jedná se již o namapovaná čtení ve formátu .bam. Z raw dat je již odstraněna rRNA, jsou trimována, mapována na lidskou DNA kvůli odstranění kontaminace, a nakonec jsou data mapována na referenční genom. [16]

Jedná se o 24 souborů ze čtyř experimentů B-E. K zisku genové exprese bylo využito sekvenační technologie Illumina NextSeq500. V každém experimentu byla provedena měření v 6 časech, které byly vybrány, aby pokryly všechny důležité fáze životního cyklu buněk a metabolické změny, jako jsou acidogenní a solventogenní fáze fermentace aceton-butanol-ethanolu a přechodové stavy. Čas T1 ve 3,5 hodině

odpovídá produkci kyselin. Při T2 v 6 hodinách bylo nejnižší pH a zahájení solventogeneze (produkce aceton-butanol- etanolu). V 8,5 hodině T3 jsou znovu využity kyseliny a probíhá formování rozpouštědla. V čase 13 hodin T4 byl zvyšován počet životaschopných buněk obsahujících akumulovanou granulózu a předpokládá se začátek sporulace. V 18 hodině T5 a 23 hodině T6 již byl pozorován vývoj sporulace a akumulace rozpouštědla v médiu. [16]

3.2 DREAM4 *in silico* challange

Mimo dat genové exprese *Clostridium beijerinckii* NRRL B-598 byla k testování algoritmu využita data z DREAM4 *in silico* challange [18, 19, 20]. Jedná se o simulovaná data genové exprese v ustáleném stavu v časových řadách. Data byla vytvořena z podsítí *Escherichia coli* a *Saccharomyces cerevisiae* pro účely výzvy DREAM4 v odvozování GRN. [18, 19, 20].

Výzva se skládala ze tří podúkolů (InSilico_Size10, InSilico_Size100, InSilico_Size100_Multifactorial), které se lišily ve velikosti sítí a typem poskytnutých dat. V úkolu Size10 bylo poskytnuto pět sítí a pro Size100 deset sítí. Každá časová řada se skládala z 21 časových bodů. K jednotlivým sítím byl poskytnut tzv. zlatý standard, tedy správné složení genové regulační sítě. V datech bylo k dispozici i označení typu interakce hrany (aktivace nebo inhibice). Data odpovídala zašuměným datům z měření úrovně mRNA. Data byla také normalizována a neobsahovala self-regulace. [18, 19, 20]

4 Předzpracování dat

Signál, který je vytvořen při laboratorním experimentu je nutné nejdříve převést do digitální podoby. K tomu se využívá formátu FastQ. Ten v sobě obsahuje kromě získané sekvence a identifikátoru i informaci o kvalitě konkrétní přečtené báze. A právě kontrola kvality je prvním krokem předzpracování. Existují k tomu nástroje jako jsou například FastQC nebo MultiQC [21]. Další částí, která na kontrolu kvality navazuje, je odstranění nekvalitních dat neboli trimming. [12]

Dalším krokem je mapování sekvencí. To je závislé zejména na existenci referenčních sekvencí. Pokud existují, tak můžeme sekvence mapovat k již existujícímu referenčnímu genomu. V opačném složitějším případě, kdy neexistuje referenční genom, či jej nechceme využít, musíme přečtené fragmenty skládat *de novo* [12]. Výsledné soubory SAM můžeme transformovat do BAM souborů pomocí SAMtools. [22, 16]

Matice počtů (count table) lze z BAM souborů vytvořit pomocí funkce featureCounts z balíčku Rsubread [23]. A aby byly získané matice počtů využitelné pro další analýzy a srovnávání genové exprese mezi vzorky, je zapotřebí provést normalizaci dat. Při normalizaci se zaměřujeme na několik faktorů, mezi něž patří délka genu, hloubka čtení či složení RNA. Délku genu musíme uvažovat, jelikož na delší gen bývá mapováno více čtení oproti kratším genům. Hloubka čtení nám udává kolikrát byl daný gen sekvenován. Pokud má ovšem jeden celý vzorek hloubku čtení vysokou, tak i konkrétní gen bude mít hloubku čtení tomu úměrně vysokou. Nešlo by proto pozorovat expresi dvou stejných genů z různých vzorků bez ošetření hloubky čtení. Složení RNA může také mít vliv na normalizaci dat. Například pokud v jednom ze vzorků nejsou exprimované stejné geny jako v jiných vzorcích, nebo jsou vzorky kontaminované, či jsou některé geny v jednom ze vzorků nadměrně exprimované. Je tedy důležité toto také uvažovat. [24, 25]

Existuje několik metod, které lze využít k normalizaci. Některé komplexnější metody si příblížíme v následujících podkapitolách.

4.1 RPKM/FPKM a TPM

„Čtení/fragmentů na kilobázi exonu na milion mapovaných čtení/fragmentů“ a „Přepis na miliony“ uvažuje hloubku čtení a délku genu, které jsou vhodné pro porovnávání změn ve vzorku. Tato metoda není příliš vhodná pro zkoumání rozdílů expresí mezi vzorky. [24]

RPKM se využívalo ke srovnávání genové exprese jak ve vzorcích, tak i mezi nimi. Díky svému jednoduchému výpočtu se stále využívá. [25]

$$RPKM = 10^9 \cdot \frac{čtení\ mapovaná\ na\ transkript}{počet\ čtení \cdot délka\ transkriptu} \tag{4.1}$$

RPKM mělo být používáno k měření relativní molární koncentrace RNA (RMC) transkriptu ve vzorku. Avšak RPKM nemá neměnný průměr jako RMC k množství RNA ve vzorku. K řešení tohoto problému byla navržena TPM jako úprava rovnice RPKM [25]:

$$TPM = 10^6 \cdot \frac{čtení\ mapovaná\ na\ transkript/délka\ transkriptu}{\sum^i (čtení\ mapovaná\ na\ transkript\ genu\ i/délka\ transkriptu\ genu\ i)} \tag{4.2}$$

Ovšem při porovnávání exprese genů mohou tyto metody vykazovat problematické chování, i přes to, že se jedná již o normalizované hodnoty. Problematické pro tyto metody mohou být například rozdíly v použitých extrakčních a izolačních protokolech RNA, v přípravách knihoven či rozdíly v mitochondriálních a jaderných RNA částic ve tkáních. [25]

4.2 Normalizace poměry mediánů

Normalizace uvažuje hloubku čtení a složení RNA. Díky tomu je vhodná zejména pro porovnávání expresí mezi vzorky, avšak neuvažuje, že by geny byly velmi rozdílně exprimované, a proto tato normalizace má výsledky vhodnější pro analýzy mezi vzorky.

Nejdříve je vytvořen skrz všechny vzorky geometrický průměr jednotlivých genů. Dále vypočítáme poměry expresí genů v každém vzorku s vypočítaným geometrickým průměrem daného genu. Z těchto poměrů pro každý vzorek vybereme mediánem normalizační faktor, kterým podělíme hodnoty exprese genů pro daný vzorek. [24]

4.3 Oříznutý průměr M hodnot

Oříznutý průměr M hodnot (Trimmed mean of M value TMM) je metoda využívaná k odhadování relativních expresí genů jednoduchou, ale účinnou cestou. Předpokladem k jejímu užití je, že exprese genů, které mají vzorky stejné, se neliší. Dle studie [26] je TMM odolná vůči odchylkám od předpokladu stejné exprese o přibližně 30 % jedním směrem. Z TMM jsou vyloučeny vysoce exprimované geny a geny s velkou variancí exprese. Normalizační faktor vypočítáme pomocí vážených průměrů podskupin genů. [26, 27]

4.4 Příprava testovacích dat z experimentu

Data, která jsou použita k testování našeho algoritmu byla získána z RNA-Seq analýzy. Jejich předzpracování bylo provedeno pomocí následujícího postupu.

Prvním krokem je vytvoření matice počtů (count table) a následně její normalizace. K tomuto účelu byly využity knihovny pro programovací jazyk R. Matice počtů byla vytvořena pomocí implementované funkce featureCounts z knihovny Rsubread [23]. Dalším krokem bylo provedení normalizace dat pomocí knihovny DESeq2 [28] metodou založenou na poměrech mediánů [24], která byla popsána v kapitole 4.2 Normalizace poměry mediánů. Část normalizované tabulky je možné vidět v příloze A.2.

Výsledné normalizované exprese genů pro všechny vzorky v časech mohou být vykresleny pro vizualizaci exprese genů jako heat mapy (viz. příloha A.1).

5 Genové regulační sítě

Genové regulační sítě (gene regulatory networks, GRN) jsou v dnešní době žádané ke sledování a zobrazování informací o interakcích na úrovních gen-gen nebo genové interakce s prostředím [29], jak již bylo zmíněno v kapitole o genové expresi. Požadavek na tvorbu genových regulačních sítí a na další metody interpretací tohoto typu dat vznikl po roce 2000, kdy výrazně klesla cena za sekvenování dat [30] a tím narostlo množství vytvořených genomických a transkriptomických dat. A právě GRN se ukázaly jako velmi vhodné ve vizualizaci informací z dat [29]. Jedná se tedy aktuálně o jednu z největších výzev systémové biologie [6]. Kvůli tomu, že má genová regulace velký vliv na chování buňky, potažmo celého organismu, tak má popis těchto vztahů velký význam v medicíně k návrhu genové terapie a léčiv, v biologii pro geneticky modifikované rostliny nebo i v průmyslu [16]. Předpokladem pro vytváření GRN je, že můžeme vytvořit model, kde úroveň exprese jednoho genu je ovlivněna jedním či několika geny [31, 32]. Popisují tedy vztah mezi transkripčními faktory (nebo i jinými regulujícími elementy jako jsou například sRNA) a regulovanými geny [6]. Odvozování GRN lze provádět několika způsoby a jejich volba je závislá na biologických omezeních a na tom, za jakým účelem GRN vzniká [29].

GRN jsou obvykle zobrazovány jako grafy s určitým počtem vrcholů a hran, přičemž vrcholy reprezentují jednotlivé geny a hrany následně zobrazují vazby mezi geny. Tudíž hrany existují pouze mezi geny, které se navzájemn ovlivňují. Pokud jeden gen řídí druhý, buď ho aktivuje nebo inhibuje a daný graf se nazývá orientovaný a směr a druh řízení jsou do grafu specificky zakresleny. Můžou ovšem existovat grafy, u kterých neznáme směr či povahu řízení a tyto grafy jsou označovány jako neorientované [33], potom tedy síť nazýváme jako genovou koexpresní síť [6].

5.1 Metody tvorby genových regulačních sítí

Metody tvorby GRN lze rozdělit do šesti základních skupin: Analýza koexprese, vyhledávání sekvenčních motivů, ChIP technologie, vyhledávání ortologů, pomocí literatury a díky transkripčním komplexům. [6]

Finální výběr metody odvozování GRN však závisí na zkoumaném problému, na typu dat, velikosti sítě a jestli máme k dispozici nějaké dřívější poznatky o GRN či informace o zkoumaném druhu, transkripčních faktorech a mnoho dalšího. I přestože jsou některé metody odvození GRN vhodnější pro konkrétní data, lze vybrané metody upravit tak, abychom dosáhli zamýšlených výsledků [6, 29]. Velmi dobrými výsledky se prokázalo užití kombinací více přístupů odvození GRN a jejich spojování [34].

5.1.1 Nástroje založené na koexpresi

Tyto nástroje k rekonstrukci genových regulačních sítí založené na koexpresi začaly být využívány k získávání informace o regulacích genů po aplikacích metod kvantifikace množství transkriptů na úrovní transkriptomu. [35] Tyto nástroje využívají například DNA microarrays, SAGE či RNA-Seq k získávání informací o transkripčních expresních profilech z transkriptomických dat. Právě pomocí závislosti jejich TEP stanovujeme, jestli jsou dva geny koexprimované nebo ne. [6] Závislost mezi TEP se velice jednoduše hodnotí pomocí Pearsonovy korelace, pokud jsou v datech pouze lineární interakce. Pokud jsou interakce nelineární, je možné využít například Spearmanovy korelace, či ještě ve složitějších případech vzájemné informace. Výhodou koexpresních nástrojů je, že i z malého počtu vzorků lze odvodit celogenomovou síť, jsou jednoduché a výpočetně málo náročné. [6, 36]

Nástroje využívající vzájemné informace (mutual information, MI)

Právě mezi nástroje založené na koexpresi řadíme i nástroje využívající vzájemné informace. Teorii informace, tedy sledování chování dat při ukládání, načítání a přenosu, navrhl Claude E. Shannon [37]. Nalezla využití i při odvozování GRN, kde se hledá podobnost a odlišnost mezi páry genů. Takové sítě mohou být také nazývány sítě relevancí. Jejich přínos tkví v tom, že metody založené na teorii informace nejsou výpočetně náročné a je tedy možné je využít u rozsáhlých regulačních systémů. [29]

V GRN se uvažuje, že dva geny spolu interagují, pokud mají korelační koeficient vyšší než předem stanovený práh. Je nutné vědět, že GRN jsou vysoce propojené sítě, a proto je mezi všemi geny nenulová korelace. Proto je nutné stanovovat práh pro rozhodnutí, jestli interakce mezi geny opravdu existuje. Problémy, které vyvstávají při měření korelace jsou u MI překonány díky měření nelineárních závislostí mezi geny [38]. Vzájemná informace je hodnota, která nám říká, kolik informace je v daném genu uloženo o nějakém jiném genu. Je to tedy proměnná nesoucí informaci o interakci mezi dvěma geny. Na ní je založeno několik přístupů k odvozování GRN.[29]

Mezi tyto přístupy patří například „kontextová pravděpodobnost příbuznosti“ (CLR) [39]. Ta upravuje náhodný šum v MI pomocí pravděpodobnosti založené na z-skóre. K ohodnocení interakce mezi geny využívá transkripční profily s různými podmínkami a je schopná snižovat korelaci a nepřímé interakce mezi geny. Patří sem například algoritmus GTRNetwork. Ten nejdříve odhaduje změny aktivit transkripčních faktorů se známými geny pomocí analýzy síťových komponent (metoda založená na regresi) a následně pomocí CLR zjišťuje interakce mezi aktivitou transkripčních faktorů a genů. [29]

Dalším přístupem od autorů [40] bylo nalézání shluků genů, které mezi sebou interagují. Tím že ohodnotili vzory genové exprese a vzájemnou informaci mezi vzory exprese u každé dvojice genů, které byly filtrovány pomocí prahu, byli schopni vytvořit shluky interakcí mezi geny. Shlukování bylo využito i u [41] k zjednodušení a zvýšení přesnosti algoritmu. Na seskupených datech byla teprve odhadnuta podobnost profilů pomocí maximálního informačního koeficientu (MIC) [42].

Oproti tomu entropii uvažuje i algoritmus MIDER [43], který snížením entropie určuje, jestli jsou interakce mezi geny přímé nebo nepřímé a dokáže také určit směr interakcí. [29]

Posledním algoritmem využívajícím MI, který je zde uveden, je algoritmus ARACNE [44], který byl dále rozšířen na algoritmus pro odhad GRN na základě časových řad s názvem TimeDelay ARACNE [45]. Oba algoritmy vypočítávají odhad Gaussova jádra MI. Tyto odhady jsou dále filtrovány. Díky tomu jsou odstraněny hrany mezi geny, kde MI není dostatečně velká. Posledním krokem je vyřešení nerovnosti zpracování dat, kdy se odstraňují přímé hrany mezi dvěma geny, pokud mezi nimi existuje spojení skrz jiný gen a toto spojení má větší pravděpodobnost.

Existuje ovšem i velké množství algoritmů, které využívají kromě MI i jiné metody a dosahují tím lepších výsledků při odhadu GRN. Mezi ně patří například PIDC využívající částečnou dekompozici informace spolu s MI, nebo třeba SELDOM využívající kromě MI i logické modelování, souborové modelování, parametrické dynamické identifikace modelů a redukci modelů. [29]

5.1.2 Nástroje založené na sekvenčních motivech

K rekonstrukcím genových regulačních sítí můžeme využít i známých konzervovaných motivů sekvencí DNA, které transkripční faktory rozpoznávají v regulačních oblastech genů. DNA motiv se nachází v promotorové oblasti genu a jedná se o krátkou konzervovanou sekvenci. Motiv je vazebným místem pro regulátory transkripce. Každý transkripční faktor rozpoznává sekvence DNA, které jsou podobné jeho motivu vazebného místa. Tyto sekvence DNA podobné motivům transkripčních faktorů mohou být reprezentovány jako matice pozice-vah (PWM), nebo také pomocí pravděpodobnostních skrytých Markovových modelů. Ty popisují pravděpodobnost nukleotidu na specifické pozici v DNA sekvenci. [46] Na rozpoznávání motivů mají také vliv sekvenční kontext či tvar DNA nebo také vazba kofaktorů. [6]

Známe motivy jsou uloženy v databázích a existují výpočetní postupy pro predikci předpokládaných vazebných míst transkripčních faktorů. Díky znalosti vlastností motivů DNA můžeme vytvořit síťové moduly, které nám vytváří sady genů, které jsou regulovány stejnými motivy. Předpokládáme, že geny ve stejném síťovém modulu jsou zapojeny do stejných biologických procesů. Díky těmto znalostem, mů-

žeme vytvářet genovou regulační síť, kde nám hrany dávají informaci o vztazích mezi transkripčními faktory a cílovými geny. Tato metoda nám snižuje náročnost odhadu regulačních sítí tím, že využívá společných znaků a tím snižuje počet parametrů, které by měly být odhadnuty. Využívání sekvenčních motivů je velmi výkonnou metodou odhadu genových regulačních sít, jelikož uvažuje konkrétní transkripční faktory a cílové geny. K častějšímu využívání této metody napomáhají i nové výpočetní metody a rozšiřování databází sekvenčních motivů. Nesmíme ovšem opomínat, že přítomnost konkrétního motivu nám nezaručuje veškerou znalost exprese genu. Ta je totiž také ovlivňována i dalšími faktory, a proto je tato metoda mírně prediktivní. [6]

5.1.3 Nástroje založené na ChIP

Chromatin Immunoprecipitation (ChIP) je metoda, která dovede z živých buněk extrahovat a izolovat komplexy protein-DNA chromatin [47]. Právě díky schopnosti identifikovat spojení mezi transkripčním faktorem a genem, jenž je tímto transkripčním faktorem regulován, je zásadní a rozšiřuje nám znalost, která je založená pouze na nalezení vazebného místa pro transkripční faktor. ChIP lze využít v kombinaci s metodami sekvenování nové generace jako ChIP-Seq, či s DNA microarrays jako ChIP-chip. Toto spojení nám může pro daný transkripční faktor získat mapu vazebných míst po celém zkoumaném genomu. Informace s ChIP-Seq/ChIP-chip společně s profily transkripční exprese jsou užívané například v algoritmu GRAM, což vede k odvozování robustnějších GRN a zlepšení výsledků analýz. [6]

Dnes už samozřejmě existuje velké množství databází, které shromažďují a poskytují informace z ChIP-Seq. Jsou v nich uložena ChIP-Seq data jak pro proteiny, tak tkáně, organismy a data z různých experimentů, které obsahují informace o GRN. Patří mezi ně například ENCODE, LOLA, Chip-Array, Chip Atlas a mnohé další. [6]

Nebo existují nástroje, které anotují ChIP-Seq data s funkčními genomickými místy a využívají již zmíněných databází k odvozování genové regulace. Sem patří například ChIPpeadAnno(R), DROPA(Python), Goldmine (R) a další. Jedná se o balíčky a funkce implementované v různých programovacích jazycích. [6]

5.1.4 Ortologické zdroje

Ortologické nástroje využívají znalostí o GRN a transkripčních sítích získané již pro jiné organismy. Je ovšem důležitá správná definice ortologie. Jedná se o evoluční vztah mezi geny, které mají společného předka. Organismy mají v daném úseku sekvenční podobnost, či zachovávají vzor. Vzhledem k sekvenční podobnosti a zachování

vzorů předpokládáme, že mezi geny mající také stejnou nebo podobnou funkci, neexistuje jednoznačná metoda, která by určila funkční konzervaci genů nebo proteinů. [48] Ovšem i tak předpokládáme, že existuje i funkční podobnost.

I zde se jedná o nástroj, který dokáže zlepšit výpovědní hodnotu GRN. Přínosná je i tím, že díky ní můžeme navrhovat interakce, které budou v jednom organismu a u jeho příbuzného druhu je můžeme ověřit. [6] I u tohoto nástroje existují databáze, které nám usnadňují práci jako třeba eggNOG, inParanoid a orthoDB. [6]

Vyjma databází obsahujících informace o ortologních vztazích, existují nástroje pro odvozování GRN jako jsou MRTLE, TargetOrtho, webová služba Phylogene či OrthoClust. [6]

5.1.5 Literární zdroje

GRN lze také odvozovat z odborné literatury bez použití experimentů. Mohou být ovšem také využity k doplnění některých informací, které nebyly z experimentů či jiných nástrojů zřejmé. Patří sem například: databáze KEGG, která tvoří sítě komplexních biologických systémů. JASPAR a TRANSFAC obsahují informace o transkripčních faktorech. Harmonizome sdružuje několik genomických databází včetně těch, které již byly zmíněny atd. [6]

5.1.6 Nástroje transkripčních komplexů protein-protein interakce (PPI)

PPI sítě nám dávají informaci o interakcích na úrovni genových produktů. Podgrafy PPI sítě jsou ovšem důležité pro pochopení regulace transkripce a díky nim mohou být predikovány a také vizualizovány. [6]

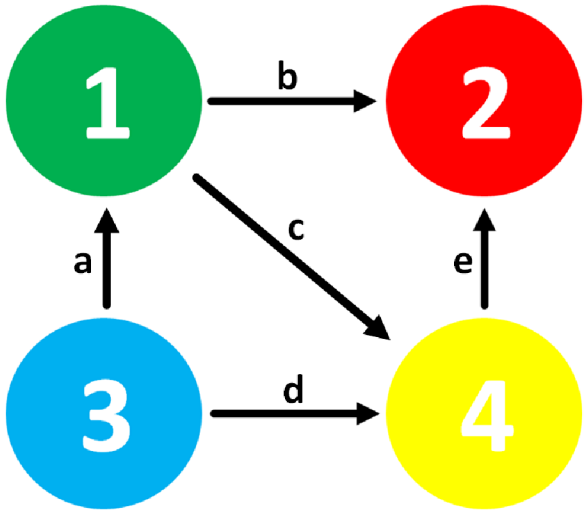
5.2 Formáty genových regulačních sítí

Pro člověka je nejpřehlednější způsob reprezentace sítí v podobě grafů. Avšak grafy ve výpočetní technice musí být nějakým způsobem zapsány. K tomu se využívají matice a listy. Pravděpodobně nejvíce intuitivní je adjacency matrix. Dále se používá například adjacency list či adjacency map, případně incidence matrix. Po výpočtu genových regulačních sítí tedy musíme tyto získané informace zapsat do některého ze zmíněných formátů a následně uložit.

V grafech nám vrcholy značí geny a hrany symbolizují vztahy mezi dvěma vrcholy. Proto v grafech chceme zobrazit tolik vrcholů, kolik pozorujeme genů a vztahy

mezi nimi odvozujeme. Tyto vztahy jsou směrové a říkají nám, který gen je regulátorem a který je regulován.[49] V našem algoritmu budeme zjišťovat při návrhu genových regulačních sítí i typ interakce, tedy jestli se jedná o aktivaci, či inhibici.

Grafy, které nám zobrazují vztahy mezi geny mohou být směrové, či nesměrové. Přičemž nesměrové grafy jsou podmnožinou grafů směrových, kde považujeme jednu hranu za dvě hrany směrového grafu z bodu A do budu B a také obráceně z B do A. V těchto grafech je tedy uloženo dvojnásobné množství hran. Tyto nesměrové grafy nejsou vhodné pro genové regulační sítě, jelikož v nich nenajdeme dostatečné informace o směrech interakcí. Nesměrové grafy mají matice, které jsou symetrické podle diagonály. Ve směrových grafech tomu tak není. Zde gen na řádku reguluje gen ve sloupci. [49, 50]



Obr. 5.1: Příklad genové regulační sítě

5.2.1 Adjacency matrix

Adjacency matrix (matice sousednosti) nám dává informace o všech hranách mezi geny. Vždy má rozměr NxN, kde N odpovídá počtu zkoumaných genů. Hrana je tedy vždy označena dvěma geny, mezi kterými se nachází, přičemž vychází z genu, který je symbolizován řádkem a vstupuje do genu příslušného sloupce. V nejjednodušším případě je adjacency matrix pouze matice nul, pokud se na daném místě hrana nenachází a jedniček, pokud se zde hrana nachází. Pak se tedy jedná o booleanovskou matici. Výhoda ve využití adjacency matrix je její jednoduchost pro pochopení a snadnost při nastavení programů. Problémem je zde velký objem dat, který není

ovlivněn počtem hran, ale počtem zkoumaných genů. A proto při velkých sítích může nastávat problém s ukládáním dat, případně se zpomalováním programu. [49]

Tab. 5.1: Příklad adjacency matrix vytořené podle obr. 5.1

	gen 1	gen 2	gen 3	gen 4
gen 1	0	1	0	1
gen 2	0	0	0	0
gen 3	1	0	0	1
gen 4	0	1	0	0

5.2.2 Adjacency list

V tomto formátu je vytvořen list, kde je pro každý vrchol vytvořen jeden prvek listu. Do těchto prvků pro určitý vrchol jsou následně ukládány cílové vrcholy. Může být vytvořen i seznam opačný, kdy jsou pro každý vrchol do listu ukládány všechny vrcholy, které skrz hranu interagují s tímto vrcholem. Tímto způsobem jsou popsány hrany pomocí vrcholů, které jsou na jejich koncích, což je vhodné pro efektivnější testy sousedství. Je však také možné každou hranu označit jejími identifikátory a mít tím seznam hran a seznam vrcholů, které jsou potom v adjacency listu propojeny. Jednou z velkých výhod tohoto přístupu k ukládání sítí do formátu adjacency list je úspora prostoru a tím i zamezení zpomalování programu. [50]

Tab. 5.2: Vzor adjacency list vytořený podle obr. 5.1

gen 1	gen2, gen4
gen 2	
gen 3	gen1, gen4
gen 4	gen 2

5.2.3 Adjacency map

Má obdobný tvar jako adjacency list, avšak list se skládá ze dvou slovníků, kde jeden je pro vstupní hrany do vrcholu, kde je zapsán identifikátor hrany společně s vrcholem, ze kterého hrana vychází a druhý slovník, kde jsou zapsány hrany vycházející z vrcholu opět s vrcholem, do kterého vstupují. [50]

Tab. 5.3: Vzor adjacency map vytořený podle obr. 5.1

	vstupující hrany	vystupující hrany
gen 1	gen 2, b; gen 4, c;	gen3, a;
gen 2		gen 1, b; gen4, e;
gen 3	gen 1, a; gen 4, d;	
gen 4	gen 2, e;	gen 1, c; gen 3, d;

5.2.4 Incidence matrix

Pro použití této matice je nutné si kromě označení vrcholů označit také hrany. Incidence matrix má následně rozměry MxN, kde M je počet hran a N je počet genů. Geny jsou zapsány do řádků a hrany jsou zapsány do sloupců. Následně se při kódování grafu do matice přičte k pozici +1, pokud daná hrana z genu vychází a -1 pokud hrana do genu vstupuje. Můžeme mít také nesměrový graf, následně tedy potom v algoritmu pouze přičítáme 1. Může nám také nastat případ self-regulace. V tu chvíli na pozici zapisujeme číslo 2. [51]

Tab. 5.4: Vzor incidence matrix vytořený podle obr. 5.1

	hrana a	hrana b	hrana c	hrana d	hrana e
gen 1	-1	1	1	0	0
gen 2	0	-1	0	0	-1
gen 3	1	0	0	1	0
gen 4	0	0	-1	-1	1

5.3 Alternativní bioinformatické přístupy ke genové expresi

Alternativou pro genovou regulační síť může být genová koexpresní síť. Stejně jako GRN využívá ke svému sestavení genové exprese. [52] Rozdíl mezi nimi je ovšem ten, že genová koexpresní síť se nesnaží mezi jednotlivými geny vytvořit směrové hrany (směr interakce), ale pouze nalézá významné vztahy mezi koexprimovanými geny. Tímto způsobem se vytvoří shluky genů, které mezi sebou interagují a spolu se účastní biologických procesů. Kvůli tomu jsou tyto sítě reprezentovány jako neorientované grafy. [53]

Aby byla v tomto případě vytvořena hrana mezi geny, je nutné určitým způsobem zjistit jejich vzájemnou korelaci a tím potvrdit jejich koexpresi. Pro bivariační normální data může být použit Pearsonův korelační koeficient. [52] Kromě korelačních

koeficientů se využívá i vzájemné informace. [53]

Máme-li mezi geny zjištěny korelační koeficienty či vzájemné informace, tak obdobně jako v námi navrženém algoritmu, provádíme prahování těchto hodnot. Určení prahu bývá často obtížné a zároveň klíčové pro získání správných výsledků. Ovšem využijeme-li k sestavení sítě pouze hodnoty nad prahem, tak sestavenou síť nazýváme síť relevance. [53]

6 Navržený algoritmus

Cílem práce bylo navrhnout a implementovat algoritmus pro tvorbu genových regulačních sítí pomocí vzájemné informace z dat genové exprese. V následující kapitole je tento navržený algoritmus popsán včetně poznámek k jeho implementaci do jazyka R.

6.1 Programovací jazyk R

K implementaci algoritmu byl použit programovací jazyk R. Jedná se o volně dostupný jazyk. Tento jazyk je podobný jazyku S, který byl využíván zejména ve statistice a k analýze dat, a lze jej považovat za odlišnou implementaci S jazyka. Proto také mnoho kódů z S jazyka funguje i v R jazyku bez změn. R tedy také obsahuje mnoho statistických nástrojů (analýzy, testy, klasifikátory...). Je také dobrý pro vytváření grafů [54]. V jazyku R je také možné si snadno vytvářet další funkce, které R neobsahuje, případně existuje také mnoho již vytvořených balíčků funkcí. Ty jsou dostupné například na internetových stránkách CRAN [55]. [54]

V současné době je tento jazyk využíván k vědeckým účelům. Velké množství dat je od počátku výzvou pro bioinformatiku. Je zapotřebí s nimi pracovat (nahrávat, přemisťovat, uchovávat je), analyzovat a graficky je zobrazovat. Jazyk R je k této práci využíván, jelikož se pomocí něj data dobře zpracovávají a modelují. Přínos v oblasti používání R v bioinformatice má Bioconductor Project. [56]

Algoritmus byl implementován ve verzi jazyka R 4.1.3, která je z března roku 2022.

Pro práci v jazyku R bylo použito open source integrované vývojové prostředí RStudio ve verzi RStudio 2022.02.1+461 "Prairie Trillium"pro Windows. Jedná se o verzi Desktop RStudio, která slouží k místnímu spouštění programu ze zdrojového editoru na osobním počítači. Proti této verzi existuje verze RStudio Server, kde je přístup do Rstudia na serveru skrz webový prohlížeč. Obě verze mohou být, jak již bylo řečeno, open source, ovšem existují i komerční verze, které přináší uživateli více možností. [57]

6.1.1 Použité knihovny

K implementaci algoritmu do jazyka R byly použity určité funkce z některých knihoven, které budou přiblíženy v této podkapitole.

Funkce **featureCounts** z knihovny **RSubread** [58] ve verzi 2.8.1 byla využita k výpočtům count tables (matice počtů) genové exprese. Vstupem funkce jsou soubory .bam mapovaných čtení. Do funkce byla poskytnuta anotovaná data ve formátu

.GFF3, která byla získána z databáze GenBank a NCBI pro zkoumaný organismus, kterým v tomto případě byla *Clostridium beijerinckii* NRRL B-598 (ID CP011966.3). Dále zde byl nastaven featuteType pro vybrání řádků z GFF3 souboru (gene/pseudogene) a strandSpecific pro reverzní řetězec. [59]

Pro normalizaci dat byly využity funkce z knihovny **DESeq2** ve verzi 1.34.0, která obsahuje mimo funkce k normalizaci dat i funkce k vizualizaci a diferenčním analýzám vícedimenzionálních matic počtů. Funkce **DESeqDataSetFromMatrix** vytvoří objekt pro ukládání vstupních hodnot, mezivýpočtů a výsledků analýz. Následně jsou funkcí **estimateSizeFactors** odhadnuty velikostní faktory metodou poměru mediánů. Funkce **counts** spočítá a vypíše normalizovaná data, kde jsou do řádků uloženy jednotlivé geny a do sloupců jsou uloženy hodnoty jednotlivých vzorků. [28]

Funkce pro měření času výpočtů, a tedy doby trvání algoritmu, byly z knihovny **pracma** ve verzi 2.3.8. Uplynulý čas byl měřen funkcemi **tic** a **toc**. Jedná se o implementovaný časovač z MATLABU. Na začátku měření času ve funkci tic je nastaven parametr gcFirst na FALSE, aby se čas měřil od tohoto příkazu. Oproti tomu je u funkce toc nastaven parametr echo jako TRUE pro vypsání uplynulého času. [60]

Pomocí funkcí **KernSec** a **KernSur** z knihovny **GenKern** ve verzi 1.2-60 byly vypočítány odhady hustoty jádra, které byly následně využity pro výpočet odhadu vzájemné informace. Funkce KernSec byla využita k odhadu hustoty jádra pomocí Gaussovských jader pro jednu genovou expresi. Když jsme počítali odhad hustoty pomocí Gaussovských jader mezi dvěmi genovými expresemi, tak jsme použili funkci pro výpočet odhadu hustoty jádra mezi dvěma proměnými KernSur. [61]

Funkce **tsbootstrap** z knihovny **tseries** ve verzi 0.10-50 byla využita k bootstrapu časových řad genové exprese, které sloužily k vytvoření náhodných sítí pro výpočet prahu pro filtraci vzájemné informace. Byl zde nastaven typ bootstrapu na stacionární a blokový. [62]

6.2 Popis navrženého algoritmu

Jako vstupní data algoritmu byla využita data genové exprese, která byla získána z RNA-Seq experimentů. Experimenty byly prováděny tak, že genová exprese byla měřena v různých časových bodech. Ty mohou být stanoveny po pravidelných časových intervalech, případně při konkrétních experimentech mohou být nastaveny tak, že časové body korelují s významnými událostmi v buňce nebo organismu. Stejným způsobem byly navrhnuty časové body na datech, která také byla použita k testování algoritmu z experimentu *Clostridium beijerinckii* NRRL B-598 [16].

S ohledem na typ dat, která byla použita (data genové exprese v časové řadě) byl navržen následující algoritmus, který vychází z algoritmu TimeDelay-ARACNE [45]

a využívá jeho základní kroky, jako jsou zjištění počáteční změny genové exprese, výpočet vzájemné informace, bootstrap časových řad a opětovný výpočet vzájemné informace k zjištění parametrů k nastavení filtrace vzájemné informace a odstranění nerovnosti dat. Oproti tomuto algoritmu je námi navržený algoritmus rozšířen o určení typu hrany (určujeme, jestli se jedná o aktivaci +1, nebo o inhibici -1). Jako vstupní data genové exprese algoritmus využívá několik dat z několika experimentů, ze kterých počítá odhad hustoty. Data genové exprese nemodelového organismu jsou ze čtyř experimentů o šesti časových bodech a všechna jsou použita k navržení genové regulační sítě organismu.

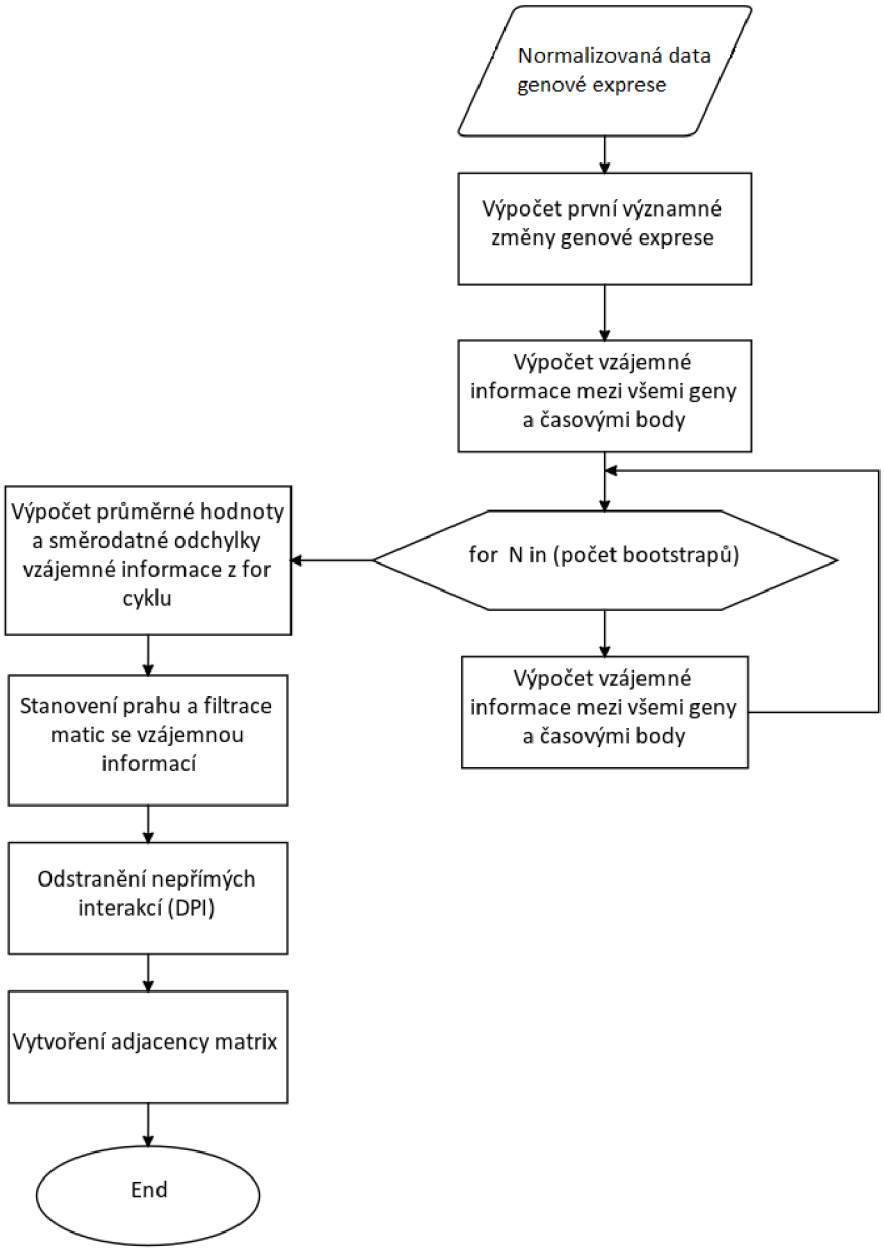
6.2.1 Podoba vstupních dat

Je důležité, aby vstupní data byla před výpočtem genové regulační sítě předzpracována. Proces předzpracování dat je popsán v kapitole 4. Předzpracování dat. Výsledkem předzpracování jsou normalizované matice počtů (viz. příloha A.2). Jako vstup do našeho algoritmu jsou tyto matice upraveny a převedeny na list, který má rozměry matice počtů. Do jednotlivých buněk jsou následně ukládány normalizované hodnoty genové exprese pro jednotlivé geny v určitých časových bodech. Každý řádek obsahuje hodnoty genové exprese jednoho genu a sloupce listu obsahují hodnoty rozdělené podle časových bodů. Jednotlivé buňky tedy obsahují vektor normalizovaných hodnot genové exprese daného genu v určitý časový bod.

Tab. 6.1: Podoba vstupních dat genové exprese do algoritmu

- v řádcích jsou hodnoty genové exprese pro jednotlivé geny
- ve sloupcích jsou genové exprese jednotlivých genů v časový bod T
- každá buňka obsahuje hodnoty genové exprese genu v časovém bodě (v tomto příkladu jsou hodnoty ze třech experimentů)

	T1	T2
gen 1	c(0.6665114, 0.6927804, 0.7025546)	c(0.3257748, 0.6137, 0.2670111)
gen 2	c(0.1272186, 0.1735119, 0.0965407)	c(0.1218223, 0.0954328, 0.0066399)
gen 3	c(0.3550646, 0.3080785, 0.4955336)	c(0.3464115, 0.2815777, 0.4120065)
gen 4	c(0.7745716, 0.5990709, 0.6290043)	c(0.7229108, 0.6576991, 0.633102)



Obr. 6.1: Diagram algoritmu

6.2.2 Počáteční změna exprese genu

První částí algoritmu je označení časových bodů, kde byla zaznamenána významná změna genové exprese. Tento krok zavádíme proto, abychom snížili výpočetní náročnost algoritmu. Ta souvisí s výpočtem odhadu vzájemné informace. Pokud uvažujeme, že bychom počáteční časový bod snížili o jednu hodnotu, tak budeme moci

vynechat výpočet vzájemné informace mezi genovou expresí v časovém bodě a všemi ostatními genovými expresemi (s vyjímkou nesplněných podmínek, viz 6.2.3 Výpočet MI), vynechá se tedy téměř N (počet genů) x T (počet časových bodů) nepotřebných výpočtů. Tímto krokem vynecháváme časové body, kde se genová exprese příliš nelišila a tudíž zde genová exprese pravděpodobně nebyla řízena a tím pro nás není důležitá k tvorbě genové regulační sítě. První významná (a tedy pravděpodobně i řízená) změna genové exprese tedy nastává až po splnění podmínky popsané ve vzorci 6.1.

K tomu, aby bylo možné tento krok provést, je nutné si nejdříve stanovit prahové hodnoty, kdy je pro nás změna genové exprese (jak její nárůst, či pokles) významná. V literatuře [45, 63] stanovují tento práh na hodnotu větší než 1,2-násobek první naměřené genové exprese pro horní práh a pro dolní práh hodnoty menší než násobek 0,83 či 0,7 prvního měření, přičemž v [45] dopočítávají druhou prahovou hodnotu dle vzorce takto $\tau_{up} = 1/\tau_{down}$.

Při použití v našem algoritmu jsme ideální prahovou hodnotu zvolili na základě testování takto: $\tau_{down} = 0,6$ a τ_{up} dopočítané podle vzorce, což byla nižší hodnota než doporučovala literatura. Testování probíhalo na základě porovnávání F-skóre.

Hledáme tedy nejmenší časový bod (initial change of expression, IcE) genu, kde proběhla výrazná změna exprese podle vzorce:

$$IcE(g_a) = argmin(\frac{g_a^0}{g_a^i} \geq \tau_{up} \ or \ \frac{g_a^i}{g_a^0} \leq \tau_{up}) \tag{6.1}$$

Významnou změnu genové exprese zaznamenáme pro všechny experimenty (tedy pro jednotlivé hodnoty v buňkách listu) samostatně. Získáme tím matici s počtem řádků shodných s počtem genů a počtem sloupců odpovídajícím počtu experimentů. Na každém řádku vybereme nejmenší časový bod. Zabráníme tím tomu, abychom nevynechali významnou změnu genové exprese při výpočtu vzájemné informace.

Nesmíme zapomenout, že gen A ovlivňuje gen B, pouze pokud je počáteční změna exprese genu A $IcE(g_A) \leq IcE(g_B)$. Tato podmínka nám zajišťuje následnost regulací.

6.2.3 Výpočet MI

Jádrem algoritmu je výpočet vzájemné informace. Dle literatury byl zvolen výpočet vzájemné informace pomocí odhadu Gaussovského jádra [44, 45]. Výpočet vzájemné informace byl proveden pomocí vzorce:

$$MI^{\kappa}(X,Y) = \bar{p}(g_a^i, g_b^{i+\kappa}) log \frac{\bar{p}(g_a^i, g_b^{i+\kappa})}{\bar{p}(g_a^i) \cdot \bar{p}(g_b^{i+\kappa})} \tag{6.2}$$

Abychom mohli MI vypočítat, je nutné vypočítat odhad hustoty Gaussovského jádra $\bar{p}(x)$ pro jeden gen a následně ještě s úpravou vzorce na dvě dimenze podle [64]. K výpočtům odhadu hustoty $\bar{p}(x)$ a $\bar{p}(xy)$ byly použity funkce z knihovny GenKern, konkrétně pro odhad hustoty jednoho genu $\bar{p}(x)$ KernSec a $\bar{p}(xy)$ KernSur. Implementaci do jazyka R uvádíme v příloze B.

Vzájemnou informaci vypočítáme vždy mezi genem A v určitém časovém bodě a mezi všemi ostatními geny ve všech časových bodech, pokud jsou splněny určité podmínky. Podmínky jsou zde nastaveny zejména proto, abychom co nejvíce snížili počet výpočtů vzájemné informace a tím výpočetní náročnost algoritmu.

Tab. 6.2: Příklad listu vzájemných informací pro čtyři geny ve třech časových bodech

- každému jmenovanému genu náleží počet řádků vzájemné informace odpovídající počtu genů a počet sloupců podle počtu časových bodů

	T1			T2			T3			
gen 1	0	0	0	0	0	0	0	0	0	gen 1
	0	0	0	0	0	0	0	0	0.062	gen 2
	0	0	0	0	0	0	0	0	0.068	gen 3
	0	0	0	0	0	0	0	0	0.939	gen 4
gen 2	0	0	0	0	0	0.850	0	0	0.132	gen 1
	0	0	0	0	0	0	0	0	0	gen 2
	0	0	0	0	0.352	0.392	0	0	1.709	gen 3
	0	0	0	0	0.187	1.583	0	0	1.686	gen 4
gen 3	0	0	0	0	0	0.466	0	0	0.585	gen 1
	0	0	0	0	2.480	3.564	0	0	1.709	gen 2
	0	0	0	0	0	0	0	0	0	gen 3
	0	0	0	0	0.366	0.467	0	0	0.485	gen 4
gen 4	0	0	0	0	0	0	0	0	0.311	gen 1
	0	0	0	0	0.088	0.300	0	0	0.482	gen 2
	0	0	0	0	0.617	0.366	0	0	0.467	gen 3
	0	0	0	0	0	0	0	0	0	gen 4

K podmínkám, které zabraňují výpočtům nepotřebné vzájemné informace, patří:

1. Výpočet vzájemné informace mezi jedním samotným genem v určitých časových bodech (self regulace). Výpočty vzájemné informace mezi samotnými geny vedly k velké chybovosti algoritmu (velké množství falešně pozitivních prvků). To je způsobeno tím, že je malé množství genů, které se sami regulují, avšak vzájemná informace mezi samotným genem je vysoká, a tudíž často hodnocena jako významná interakce. Pokud bychom chtěli uvažovat self regulace,

tak by bylo vhodné zapojit do algoritmu informace o takovýchto genech, jako jsou například databáze self regulačních genů.

2. Gen mohl být regulován nebo být regulátorem, tudíž zde byla počítána vzájemná informace až od časového bodu, který byl určen, jako první významná změna genové exprese daného genu v prvním kroku algoritmu.
3. V neposlední řadě jsme uvažovali, že vzájemná informace byla počítána pouze mezi genovou expresí konkrétního genu v určitý časový bod a ostatními geny v týž stejný časový bod, nebo v časových bodech následujících. Výpočet vzájemné informace byl počítán i ve stejném časovém bodě mezi všemi geny. Tento krok byl počítán, jelikož mezi jednotlivými časovými body mohla proběhnout významná genová exprese. Často jsou mezi jednotlivými časovými body rozdílné časové intervaly podle povahy experimentu, či mohou být časové intervaly velké a regulace zde mohla být v tomto intervalu a změna genové exprese byla vidět jen v jednom časovém bodě.

Tímto způsobem byla vzájemná informace vypočítána a uložena do matice. Pro každý gen v každém časovém bodě byla vytvořena tato matice a ta byla následně uložena do listu. V tabulce 6.2 můžeme vidět vzhled listu s vypočítanými vzájemnými informacemi ve třech časových bodech pro čtyři geny. Vidíme, že vzájemná informace se nepočítá mezi genem samotným v určitých časech. Zároveň je zde vidět, že gen 1 měl významou změnu genové exprese oproti počáteční genové expresi ve 3. časovém bodě T3, gen 2, 3 a 4 v T2. Tyto časové body, kde proběhla význačná změna genové exprese, ovlivnily, na jakých poziciích byla vzájemná informace počítána, jak můžeme v tabulce pozorovat.

6.2.4 Filtrace

Získané hodnoty vzájemné informace se nyní musí filtrovat, abychom dále mohli pracovat pouze se vzájemnou informací mezi geny, které se mezi sebou regulují.

Důležité pro filtraci je stanovit její prahovou hodnotu. Tu stanovíme poté, co pomocí bootstrapu vytvoříme nové časové řady. Byl použit blokový stacionární bootstrap, který je vhodné použít pro časové řady. Pro generování nových časových řad byla využita funkce tbootstrap z knihovny R tseries. [45, 65]. Počet nově vytvořených časových řad pomocí bootstrapů a jejich vliv na výsledné F-skóre byl testován.

Pro každou časovou řadu byly vypočítány hodnoty vzájemné informace, jak již bylo popsáno. Liší se zde pouze podmínky, kdy je vzájemná informace počítána. Zachovány zde byly podmínky o výpočtu vzájemné informace mezi jedním samým genem (opomíjí se self-regulace) a o zachování následnosti v časové řadě. První významnou změnu genové exprese zde neuvažujeme, jelikož se snažíme získat průměr-

nou hodnotu vzájemné informace mezi všemi geny, která není významná pro tvorbu genových regulačních sítí.

Ze všech hodnot vzájemné informace z bootstrapovaných časových řad byla zjištěna jejich průměrná hodnota a směrodatná odchylka. Tyto hodnoty použijeme k nastavení prahu tímto $I_0 = \mu + \alpha \cdot \sigma$ s nastavením $\alpha = 0,05$. [45] Nastavení této hodnoty také bylo testováno.

Tab. 6.3: Příklad tabulky po filtrování vzájemné informace pro čtyři geny ve třech časových bodech

	T1			T2			T3		
gen 1	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	1
gen 2	0	0	0	0	0	1	0	0	0
	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	1	0	0	1
gen 3	0	0	0	0	0	0	0	0	0
	0	0	0	0	1	1	0	0	1
	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0
gen 4	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0
	0	0	0	0	1	0	0	0	0
	0	0	0	0	0	0	0	0	0

Po filtrování všechny zbylé hodnoty vzájemné informace, které nebyly vyfiltrovány, postoupily do další části algoritmu a mají větší informační přínos pro tvorbu genových regulačních sítí. Díky těmto výsledkům můžeme vytvořit matici, do které bude uložena závislost mezi geny. Pokud se tedy dané geny ovlivňují, tak je do této matice uloženo číslo 1. Matice zde má již rozměry výsledné adjacency matrix. V této části algoritmu byla také do pomocné matice uložena hodnota vzájemné informace mezi dvěma geny. Hodnot vzájemných informací mezi dvěma geny mohlo být i více. Ale abychom mohli v poslední části algoritmu určit, jestli daná regulace byla aktivací, nebo regulací, tak zde uložíme tu vzájemnou informaci, která vycházela z nejmenšího časového bodu genu, který byl regulátorem a regulovala druhý gen taktéž v co nejmenším (tedy prvním, kde byla zaznamenána vzájemná informace)

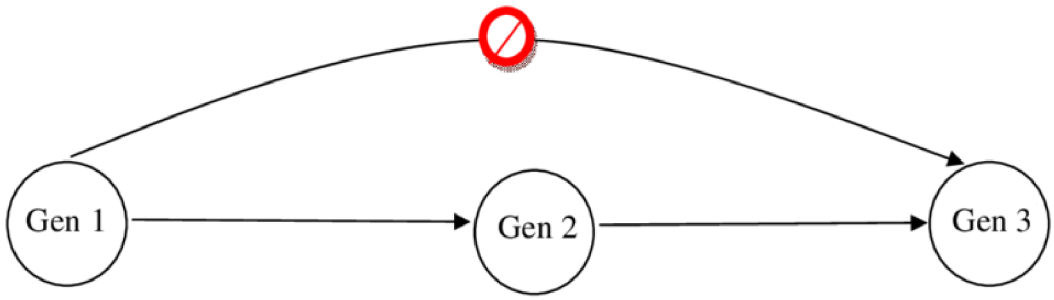
časovém bodě. Také zde byly uloženy tyto časové body k dalšímu výpočtu adjacency matrix.

Tab. 6.4: Příklad tabulky všech interakcí

	gen 1	gen 2	gen 3	gen 4
gen 1	0	0	0	1
gen 2	1	0	0	1
gen 3	0	1	0	0
gen 4	0	0	1	0

6.2.5 Nerovnost zpracování dat

Předposlední částí algoritmu je nerovnost zpracování dat (DPI). Nerovnost je způsobena vytvořením hran mezi geny podle hodnoty vzájemné informace, která je vyšší než nastavený práh. Hodnota vzájemné informace vyšší, než prahová hodnota, nám udává, že tyto dva geny na sebe působí, avšak regulace zde neprobíhá přímo. To znamená, že po výpočtu vzájemné informace nám vznikly hrany mezi geny, které na sebe působí skrz nějaký třetí gen. Pro lepší pochopení fungování nerovnosti zpracování dat je její fungování ilustrováno na obrázku 6.2.



Obr. 6.2: Princip fungování DPI [66]

Kontrola a odstranění nerovnosti zpracování dat provádíme porovnáním velikosti vzájemné informace, která je mezi jednotlivými geny z pozorované trojice genů. Kontrolujeme-li, jestli gen G1 reguluje gen G3 přímo, tak nejdříve nalezneme gen G2, který je regulován genem G1 a zároveň je regulátorem genu G3. Mezi všemi těmito geny jsou interakce ohodnocené vzájemnou informací.

$$MI(G1, G3) > max(MI(G1, G2), MI(G2, G3)) \tag{6.3}$$

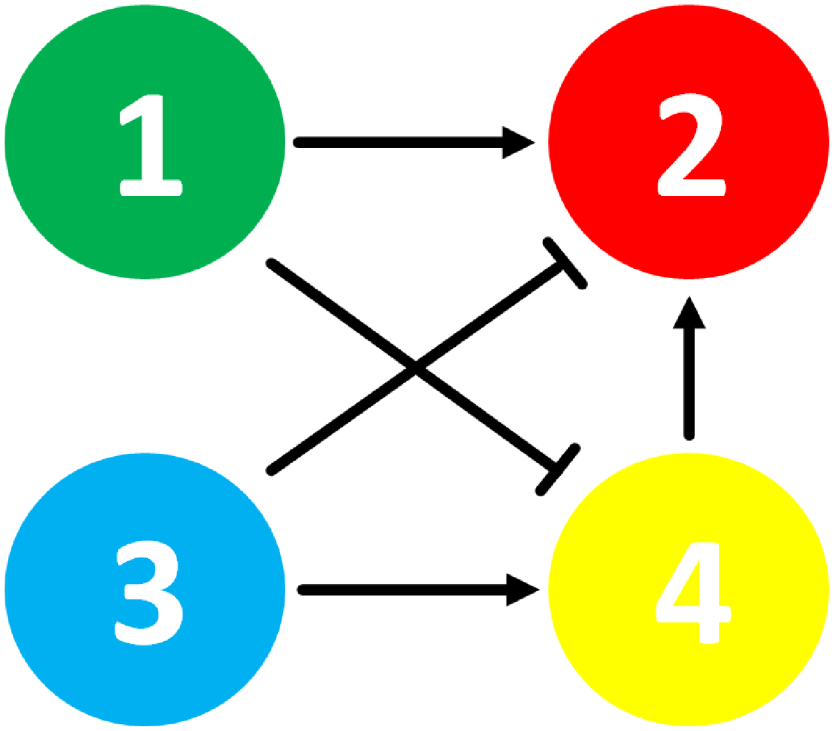
Dle rovnice stanovujeme, že interakce mezi geny G1 a G3 je přímá, pokud je hodnota vzájemné informace mezi geny G1 a G3 větší než větší hodnota vzájemná informace interakcí mezi geny G1, G2 nebo G2, G3. Pokud podmínka splněna není, tak je interakce mezi geny G1 a G3 považována za nevýznamnou a je odstraněna. [44]

6.2.6 Vytvoření adjacency matrix

Poslední částí algoritmu je námi navržený přístup k vytvoření adjacency matrix (matice sousedství). Tato matice již nese informace o genové regulační síti. V adjacency matrix bude zapsáno, jestli je daný gen regulátorem jiného genu a zda jej aktivuje (1) nebo inhibuje (-1). Mezi geny, kde není žádná interakce se zapíše 0. Jestli dva geny spolu interagují, máme zapsané v matici s vypočítanou vzájemnou informací, která je již vyfiltrována a jsou z ní odstraněny hrany, které jsou zde mezi geny, které spolu interagují nepřímo. K této matici jsme si vytvořili ještě další matici, ve které jsou uloženy časové body, kde je uložena informace o časových bodech, mezi kterými byla vypočítána vzájemná informace mezi regulátorem a regulovaným genem. Tyto informace použijeme, abychom z count table (matice počtů) zjistili, že regulátor před uloženým časem měl určitou hodnotu, která klesla, či stoupla do následující časového bodu. To stejné zjistíme i pro regulovaný gen. Pokud genová exprese regulátoru klesla a regulovaného genu taktéž klesla, případně obě genové exprese stouply, tak jsme tuto hranu označili jako aktivaci. Pokud genová exprese regulátoru stoupla a genová exprese regulovaného genu klesla, či naopak, tak jsme tuto hrano označili jako inhibici.

Tab. 6.5: Příklad adjacency matrix

	gen 1	gen 2	gen 3	gen 4
gen 1	0	0	0	-1
gen 2	1	0	0	1
gen 3	0	-1	0	0
gen 4	0	0	1	0



Obr. 6.3: Genová regulační síť vytvořená podle tabulky 6.5

6.3 Výstupy algoritmu

Výstupem algoritmu je adjacency matrix (matice sousedství). Jedná se o matici o velikosti NxN, kde N je počet genů. Každý řádek a každý sloupec je označen názvem, případně jiným identifikátorem genu. Pořadí sloupce, který značí jeden gen je totožné i pro pořadí řádku. Každý řádek nám říká, že gen, o kterém nese daný řádek informace, je regulátorem a reguluje geny, které jsou označeny v příslušném sloupci. Oproti tomu ve sloupcích můžeme pozorovat informace o regulovaných genech a na řádcích máme uložené informace o jejich regulátorech.

Pokud je gen regulátorem, tak je na jeho řádku zapsaná hodnota +1, je-li aktivátorem, anebo hodnota -1, jedná-li se o inhibitor. Hodnota 1/-1 je vždy zapsaná na pozici sloupce, který symbolizuje regulovaný gen. Pokud gen z řádku nereguluje gen ze sloupce, tak je na této pozici zapsaná 0. Ukázka adjacency matrix je v příloze C.

Přestože adjacency matrix má symetrické názvy sloupců a řádků, tak její obsah není symetrický kolem diagonály. To je způsobeno tím, že genové regulační sítě nám dávají informaci o směru regulace. Jeden gen tedy může regulovat jiný gen, aniž by byl regulovaným genem ovlivněn. To ovšem nemusí být pravidlem a geny na sebe mohou působit z obou směrů.

7 Diskuse výsledků

V této kapitole bude provedeno zhodnocení navrženého algoritmu, jeho úskalí a oproti tomu i jeho přínosy. Dále je zde uvedeno, co bylo na algoritmu testováno, jaké parametry byly optimalizovány a jakým způsobem byla úspěšnost algoritmu vyhodnocována.

Testování probíhalo na dvou datasetech. Nejprve byla využita *in silico* data dostupná z DREAM4 [18, 19, 20] s cílem nastavit parametry a vyhodnotit přesnost algoritmu. DREAM4 totiž obsahuje nejen data o genové expresi, ale také zlatý standard navržené sítě. Druhým datasetem byla data naměřené genové exprese pro *Clostridium beijerinckii* NRRL B-598 s cílem vyhodnotit náročnost algoritmu na reálných datech.

7.1 Způsob testování kvality získané genové regulační sítě

Kvalita testované sítě byla hodnocena pomocí F-skóre. Tato metrika je často používána k měření přesnosti sítě a díky tomu je možné fungování našeho algoritmu porovnat s výsledky z dostupných výzkumů a článků [67, 68]. F-skóre nám dává informaci o funkčnosti sítě z globálního pohledu a je možné pomocí této metriky porovnávat rozdílné sítě o různých rozměrech a případně i sítě různých typů. Dává nám tedy pouze jeden pohled na vytvořenou síť. Existují zde samozřejmě i jiné možnosti hodnocení menších částí sítě, jako například pomocí motivů. Avšak pro nás je celkový pohled na celou síť přínosný při porovnávání sítí při optimalizacích a také, jak bylo zmíněno, při porovnávání s články. [67]

K výpočtu F-skóre potřebujeme zjistit nejdříve hodnoty matice záměn (confusion matrix), konkrétně jsou pro nás důležité počty správně určených pozitivních prvků (TP, hrany, které byly správně nalezeny), falešně pozitivních prvků (FP, označené hrany, které ve skutečné síti neexistují) a falešně negativních prvků (FN, počet nenalezených interakcí v síti).

Tyto hodnoty (TP, FP, FN) získáme při porovnání vytvořené genové regulační sítě s daty zlatého standardu, případně s náhodně vytvořenou sítí.

Z těchto získaných hodnot nejdříve vypočítáme přesnost (p, precision) a výtěžnost (r, recall) genové regulační sítě podle vzorců [68]:

$$p = \frac{TP}{TP + FP} \tag{7.1}$$

$$r = \frac{TP}{TP + FN} \tag{7.2}$$

Z hodnot přesnosti a výtěžnosti můžeme vypočítat hodnotu F-skóre: [68]:

$$F = \frac{2pr}{r+p} \tag{7.3}$$

F-skóre může nabývat hodnot od 0 po 1. Pokud získáme hodnotu F-skóre 1, jedná se o dvě sítě, které jsou totožné. Oproti tomu při získání hodnoty 0, se jedná o sítě, které nemají jedinou správně označenou interakci. Proto se při optimalizačních úpravách snažíme dosáhnout hodnot blízkých nule, tedy snažíme se hodnotu F-skóre zvýšit.

Navržený algoritmus získává, kromě odhadnutí existence interakce a jejího směru, také informaci o typu interakce (v adjacency matrix nejsou pouze +1/0, ale -1/0/+1). Proto při výpočtu F-skóre vypočítáme F-skóre zvlášť pro aktivace 1 a následně pro inhibice -1. Obě získané hodnoty zprůměrujeme a získáme výsledné F-skóre.

Pro zjištění, zda-li navržený algoritmus správně nalézá interakce mezi geny a jejich směr, ale nesprávně je zařazuje k aktivacím nebo inhibicím, porovnáváme hodnotu F-skóre vypočítané pouze pro hrany a jejich směr a F-skóre pro hrany, jejich směr a typ interakce.

7.2 Nastavované parametry

K optimalizaci parametrů algoritmu pro zpřesnění výstupní genové regulační sítě byla použita hodnota F-skóre, kdy byla snaha o její maximalizaci.

Největší kapitolou v nastavování parametrů je nastavení počtu bootstrapů. Proto se jeho nastavením zabýváme v samostatné kapitole 7.3 Výpočetní náročnost.

Práh počáteční změny genové exprese

Dále zde byl nastavován práh pro nalezení indexu počáteční změny exprese genu (Kap. 6.2.2). Podle literatury [45] jsem znali přibližné hodnoty, které byly používány i v jiných algoritmech a výpočet pro stanovení druhé meze (byla stanovena mez dolní a horní mez byla dopočítána).

Tab. 7.1: Tabulka nastavení prahu počáteční změny genové exprese

τ_{down}	0,83	0,7	0,6	0,5
F-skóre	0,12	0,16	0,16	0,14
čas [min]	32	22	21	20

V tabulce 7.1 je vidět, jak se měnilo F-skóre s nastavením prahu, kdy nejvyšších hodnot bylo dosaženo při nastavení τ_{down} na hodnotu 0,7 a 0,6. Obecně rozdíly mezi

nastavením prahu nejsou příliš velké. Do algoritmu byl zvolen práh $\tau_{down} = 0,6$, jelikož při jeho použití bylo dosaženo nejvyšší hodnoty a zároveň je tím častěji počáteční změna exprese označena při pozdějším časovém bodě a tím se sníží počet výpočtů vzájemné informace. Následně byl dopočítán horní práh $\tau_{up} = 1,67$.

Parametry výpočtu vzájemné informace

Výpočet vzájemné informace je stěžejním bodem celého algoritmu. Je počítána podle vzorce 6.2. Pro její výpočet je nutné vytvořit odhady hustoty jádra. Ty byly vypočítány pomocí implementovaných funkcí v jazyku R (viz. kapitola 6.1.1 Použité knihovny). Důležitým vstupem do těchto funkcí je šířka pásma, která je nastavována pomocí funkce dpik z knihovny GenKern [61]. Zde bylo podle literatury nastaveno Gaussovo jádro [45], dále bylo testováno použití jiného nastavení měřítka, ale základní parametr "minim" se ukázal jako nejpřínosnější. "minim" vybírá menší ze zbylých dvou přístupů, které lze k odhadu škály také nastavit. Poslední parametr této funkce, který byl nastaven a testován je level, který byl stanoven na hodnotu deset. Kdy byly testovány tři sítě a s jejím nastavením se zlepšilo F-skóre, ovšem také se prodloužil výpočetní čas.

Nastavení hladiny filtrace

Filtrace probíhala jednoduchou horní propustí, kdy práh byl nastaven součtem průměru vzájemné informace z náhodné sítě a její směrodatnou odchylkou násobenou parametrem α. Nastavení parametru bylo také testováno, ovšem nebylo dosažono lepších výsledků, než při užití hodnoty 0,05, která byla nalezena v liteatuře [45].

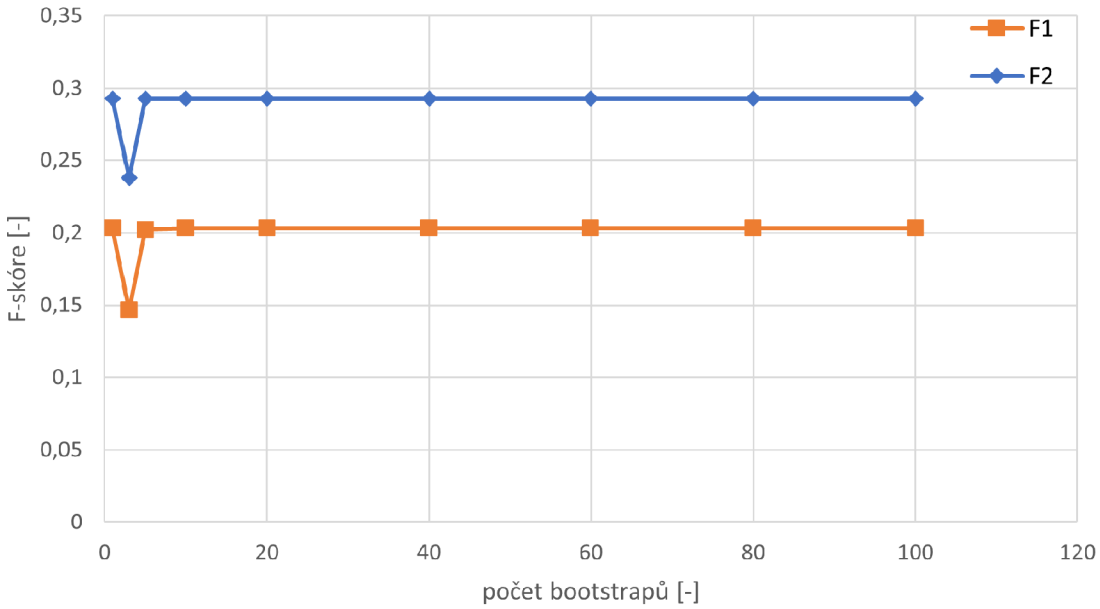
7.3 Výpočetní náročnost

Velkým problémem při navrhování a testování algoritmu byla výpočetní náročnost. Při navrhování genových regulačních sítí se využívá velké množství dat. Například pro experimenty *Clostridium beijerinckii* NRRL B-598 mající 5 442 genů [69], který zahrnuje 6 časových bodů musíme vypočítat přibližně miliardu odhadů vzájemné informace jen pro list vzájemných informací, který dále zpracováváme. Přesný počet se může lišit v závislosti na datech podle splněných podmínek (viz. podkapitola 6.2.3 Výpočet MI), které nám slouží zejména pro co největší snížení počtu provedených výpočtů odhahu vzájemné informace.

Problém ovšem nastává záhy, kdy k nastavení parametrů filtrace používáme bootstrap a vytváříme si nové náhodné sítě, ze kterých vypočítáváme směrodatnou

odchylku a průměrnou hodnotu. Při každém tomto bootstrapu musíme opět vypočítat téměř stejný počet odhadů vzájemné informace jako v prvním kroku. Tím se výpočetní náročnost N-krát znásobí (N je počet bootstrapů).

Při nastavování parametrů algoritmu byl počet bootstrapů stanoven po testování na datech DREAM4 challenge. Použita byla data s deseti geny, u kterých byla měřena genová exprese při 21 časových bodech. Pro porovnání zlepšení sítě bylo použito F-skóre. Na obr. 7.1 je graf závislosti F-skóre na počtu provedených bootstrapů a na obr. 7.2 je graf závislosti provedených bootstrapů na čase.

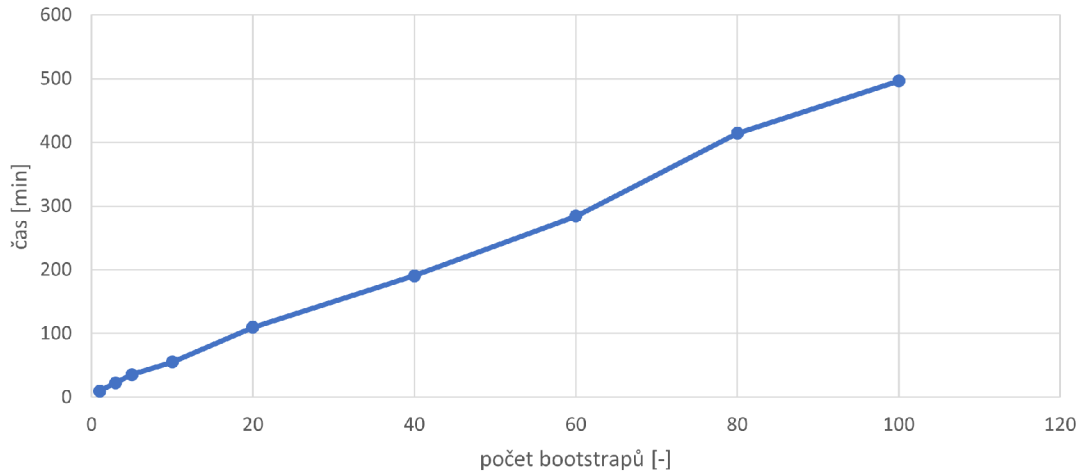


Obr. 7.1: Graf závislosti F-skóre na počtu provedených bootstrapů

Čas, který je zapotřebí k výpočtu genové regulační sítě s konkrétním počtem bootstrapů je závislý od použitého hardwaru (pro toto nastavení bylo použito zařízení s procesorem Intel CORE i5-7200U, CPU4, 8 GB RAM, 2.50GHz s grafickou kartou NVIDIA GEFORCE 940MX).

Grafy na obrázcích 7.1 a 7.2 byly vytvořeny pomocí Microsoft Excel.

Změny bootstrapů byly testovány na datech z DREAM4 challenge o velikosti 10 genů a 21 časových bodů. Na obr. 7.2 je vidět, jak počet bootstrapů prodlužuje čas, který je potřeba k tomu, aby algoritmus vytvořil adjacency matrix. Vidíme, že čas se přibližně přímo úměrně prodlužuje s narůstajícím počtem použitých bootstrapů. Při použití 20 bootstrapů nám výpočet adjacency matrix pro data genové exprese o velikosti listu 10 x 21 trvá přibližně hodinu a 40 minut. Dále k tomu, abychom získali adjacency matrix s použitím 100 bootstrapů potřebujeme asi osm hodin a 15 minut s parametry hardwaru uvedených výše.



Obr. 7.2: Graf závislosti provedených bootstrapů na čase

Oproti tomu zlepšení F-skóre zde nebylo žádné, jak můžeme pozorovat na obr. 7.1. Řada F1 nám ukazuje F-skóre při určování typu hrany a vyšší hodnoty z řady F2 je F-skóre pokud uvažujeme pouze interakci mezi geny bez jejího typu. Při použití tří bootstrapů zde byl pozorován pokles F-skóre, avšak u všech ostatních otestovaných bootstrapů zde byla hodnota F-skóre totožná. Pravděpodobně zde není přínos velkého počtu bootstrapů pro síť s malým počtem pozorovaných genů. Na sítě o více genech by pravděpodobně počet bootsrtapů měl větší vliv. Avšak jak můžeme pozorovat na obr. 7.2, už u takto malého počtu genů je výpočet poměrně časově náročný s přibývajícím počtem bootstrapů. Pro celé sítě, které jsou i několikaset násobně větší se stejným způsobem prodlouží i čas pro jejich výpočet. Proto byl počet bootstrapů stanoven na tři, jelikož pro malé sítě rozdíl není příliš znatelný (pokles na hodnotě tří bootstrapů byl pravděpodobně náhodný) a pro velké sítě by větší parametr velmi prodloužil výpočetní čas.

7.4 Kvalita vypočítaných genových regulačních sítí

Parametry algoritmu byly nastaveny vzhledem k výše uvedenému testování následovně: $\tau_{down} = 0,6$ a $\tau_{up} = 1,67$ a parametry funkce dpik scale="minim", level=10. Parametr α pro nastavení hladiny filtrace byl zvolen na 0,05 a počet bootstrapů byl tři s přihlédnutím na časovou náročnost.

Výsledné F-skóre bylo pro testovacídataset DREAM4 naměřeno 0,2362 pouze pro stanovení interakcí mezi geny a 0,1486 při určování typu interakcí. Doba výpočtu byla s těmito nastavenými parametry přibližně 20 minut. Pro testovací data-

set nemodelového organismu *Clostridium beijerinckii* NRRL B-598 neexistuje zlatý standard, a proto nebylo možné navrženou síť porovnat a tedy vypočítat F-skóre. Výpočetní doba je na výše zmíněném hardwaru více než deset dní.

Přesnost sítě tedy není příliš vysoká a výpočetní doba je dlouhá. Přesněji a rychleji pracují některé další algoritmy, které jsou porovnávány s navrženým algoritmem v kapitole 7.5 Porovnání s dostupnými algoritmy. Nepřesnosti mohly být způsobeny použitými daty z RNA-Seq. Snažíme se totiž využívat k určení typu interakce změny genové exprese v čase. Pro tento účel by nejspíš lépe posloužila singel cell analýza, která reálně analyzuje pouze jednu buňku [70]. Dobu výpočtu nejvíce ovlivňuje výpočet vzájemné informace, jehož použití bylo zvoleno podle literatury [45].

7.5 Porovnání s dostupnými algoritmy

Navržený algoritmus využívá základní stavební prvky jaké jsou použity u algoritmu time-delay ARACNE [45], který je rozšířením algoritmu ARACNE [44]. Podle těchto kapitol byl zvolen výpočet vzájemné informace.

Ovšem vyjma základních kroků jsou jednotlivé kroky výpočtu genové regulační sítě upraveny.

Při výpočtu vzájemné informace v našem algoritmu vypočítáváme vzájemnou informaci mezi všemi geny s různým časovým zpožděním a všechny získané hodnoty ukládáme pro další zpracování. Oproti tomu Time-delay ARACNE vypočítává vzájemnou informaci mezi dvěma geny s určitým časovým zpožděním a všechny hodnoty vzájemné informace mezi dvěma geny s jedním zpožděním sečte. Z nich následně vybere pro dané geny hodnotu vzájemné informace vypočítané se zpožděním, která je nejvyšší.

Při filtraci tedy Time-delay ARACNE filtruje už pouze maximální hodnoty vzájemné informace. V našem algoritmu se filtrují všechny hodnoty a teprve až z vyfiltrovaných hodnot vybíráme vzájemnou informaci, která bude ovlivňovat výsledný vzhled adjacency matrix a to tím způsobem, že vybereme první nevyfiltrovanou hodnotu z časové řady. Chceme tím docílit určité souslednosti dějů v organismu, abychom mohli určit typ interakcí mezi geny. Zároveň v našem algoritmu ukládáme i informaci o časovém zpoždění a informaci z jakého časového bodu interakce genu vycházela.

Dále v Time-delay ARACNE autoři používají DPI dvakrát, kdy nejdříve použijí DPI pro jedno časové zpoždění mezi geny a následně po výpočtu různých zpoždění. V našem algoritmu používáme DPI pouze jednou mezi všemi geny a časovými zpožděními a dbáme na to, aby hrany byly odstraněny pouze v případě, že hrany mezi geny časově navazují.

Asi největší změnou je návrh zjištění typu interakce (-1/+1), který nebyl v porovnávaných algoritmech použit. Pomocí ARACNE zjišťujeme pouze intrerakce mezi geny bez jejich směru. Time-daly ARACNE je rozšířen o rozlišení směru genové interakce.

Pro porovnání algoritmů, jejich výpočetní náročnosti a úspěšnosti bylo využito dat z DREAM4 challenge s deseti geny. Navržený algoritmus byl porovnáván s algoritmy C3NET [71] a s algoritmy MRNET, CLR a ARACNE implementovanými v knihovně minet [72]. Výsledné hodnoty jsou zapsané v tabulce 7.2. F-skóre bylo počítáno pouze pro interakce mezi geny bez určení jejich typu, jelikož to tyto algoritmy neumožňují.

Tab. 7.2: Porovnání výpočetní náročnosti a F-skóre mezi algoritmy pro odvozování genových regulačních sítí

	F-skóre	čas [s]
navržený algoritmus	0,24	1346 (22 min)
C3NET	0,44	2,57
MRNET	0,31	0,82
CLR	0,76	0,77
ARACNE	0,69	0,68

Můžeme vidět, že námi navržený algoritmus má mnohem delší výpočetní čas, než ostatní porovnávané algoritmy, které vytváří sítě do několika sekund. Nejvyšší přesnost potom byla pozorována u algoritmu CLR, kde i výpočetní čas byl téměř nejnižší. Hodnota F-skóre byla u navrženého algoritmu nejnižší z provedených měření a tudíž i nejhorší.

Přínos navrženého algoritmu tedy můžeme pozorovat v tom, že jako jediný z těchto algoritmů zjišťuje typ interakce mezi geny.

Závěr

Cílem práce bylo vytvořit základní přehled o laboratorních technikách používaných k zisku genové exprese a vytvoření základního povědomí o genové expresi. Dalším krokem poté bylo vytvoření matice počtů a její normalizace na reálných datech. K tomuto tématu byl také vytvořen základní přehled používaných metod.

Ovšem hlavním tématem práce bylo vytvoření genových regulačních sítí za použití vzájemné informace. Teoreticky toto téma včetně dalších metod odvozování genových regulačních sítí bylo popsáno v páté kapitole.

V rámci práce byl vytvořen algoritmus pro tvorbu genové regulační sítě, který byl následně implementován v jazyku R. Jeho podrobný popis je v šesté kapitole. Algoritmus využil základních kroků Time-delay ARACNE, které byly doplněny a upraveny pro potřeby výpočtů. Hlavním zamýšleným přínosem navrženého algoritmu oproti jiným dostupným algoritmům byl rozdíl v určování typu interakce mezi geny (viz. kapitola 7.5).

Algoritmus byl testován a porovnáván s dostupnými algoritmy. Porovnávána byla výpočetní náročnost a kvalita sítě, která byla měřena pomocí F-skóre. K tomu, aby bylo možné algoritmy porovnávat, musely být opomenuty typy interakcí mezi geny. Ovšem navržený algoritmus měl oproti jiným dostupným algoritmům mnohem horší výsledky jak v kvalitě sítě, tak zejména ve výpočetní náročnosti a době výpočtu.

Ke zlepšení by v budoucí práci mohlo vést využití existujících algoritmů, které mají větší přesnost, k určení interakce mezi geny. A až mezi těmito objevenými hranami začít zjišťovat typ interakce. To by mohlo vést ke zrychlení výpočtu a větší přesnosti.

Dále potom v algoritmu zcela opomíjíme self-regulace, kterých není mnoho a vnášely nám do výsledné adjacency matrix velkou chybovost, jelikož hrany mezi tímtéž genem měly vysokou hodnotu vzájemné informace. Jejich doplnění do algoritmu by opět mohlo vést ke zlepšení přesnosti, avšak bylo by nutné zapojit do výpočtu více informací, než pouze data z RNA-Seq.

Literatura

[1] DOUG CHUNG, D.-W. a K.G. LE ROCH. *Genome-Wide Analysis of Gene Expression.* Encyclopedia of Biological Chemistry (Second Edition). Second Edition. Waltham: Academic Press, 2013, s. 369-374. ISBN 978-0-12-378631-9.

[2] LI, Gene-wei a X. Sunney XIE. *Central dogma at the single-molecule level in living cells.* Nature. 2011 Jul 20;475(7356):308-15. doi: 10.1038/nature10315. PMID: 21776076; PMCID: PMC3600414.

[3] *The central dogma.* Genius [online]. Genius Media Group, 2022 [cit. 2022-03-20]. Dostupné z: https://genius.com/Biology-genius-the-central-dogma-annotated

[4] GIBNEY, E. a C. NOLAN. *Epigenetics and gene expression.* Heredity 105, 4–13, 2010. https://doi.org/10.1038/hdy.2010.54

[5] DJEBALI, S., C. DAVIS, A. MERKEL et al. *Landscape of transcription in human cells.* Nature 489, 101–108, 2012. https://doi.org/10.1038/nature11233

[6] MERCATELLI, Daniele et al. *Gene regulatory network inference resources: A practical overview.* Biochimica et Biophysica Acta (BBA)-Gene Regulatory Mechanisms, 2020. 1863.6: 194430.

[7] YANG, B., Y. XU, A. MAXWELL, W. KOH, C. ZHANG a P. GONG. *MICRAT: a novel algorithm for inferring gene regulatory networks using time series gene expression data.* BMC Systems Biology. 2018, 12. ISSN 17520509. doi:10.1186/s12918-018-0635-1

[8] JAFARI, Mina, Behnam GHAVAMI a Vahid SATTARI. *A hybrid framework for reverse engineering of robust Gene Regulatory Networks.* Artificial Intelligence In Medicine [online]. 2017, 79, 15-27. ISSN 09333657. doi:10.1016/j.artmed.2017.05.004

[9] KIZHAKKETHIL YOUSEPH, A.S., M. CHETTY a G. KARMAKAR. *PCA based population generation for genetic network optimization.* Cogn Neurodyn 12, 417–429 (2018). https://doi.org/10.1007/s11571-018-9486-0

[10] BRUGGEMAN, F. J. a H. V. WESTERHOFF. *The nature of systems biology.* TRENDS in Microbiology, 2007. 15.1: 45-50.

[11] NATHAN, Sheila. *Transcriptome profiling to understand host-bacteria interactions: Past, present and future.* SCIENCEASIA, 2020. 46.5: 503-513.

[12] MATOUŠKOVÁ, Petra. *Stanovení genové exprese.* Hradec Králové, 2018. Habilitační práce. Univerzita Karlova, Farmaceutická fakulta.

[13] YAMAMOTO, Mikio et al. *Use of serial analysis of gene expression (SAGE) technology.* Journal of immunological methods, 2001. 250.1-2: 45-66.

[14] FARRELL JR, Robert E. *RNA Methodologies: laboratory guide for isolation and characterization.* Academic Press, 2009. ISBN 978-0-12-374727-3

[15] VESELINYOVÁ, Dominika et al. *Selected In Situ Hybridization Methods: Principles and Application.* Molecules, 2021. 26.13: 3874.

[16] PATAKOVA, Petra et al. *Acidogenesis, solventogenesis, metabolic stress response and life cycle changes in Clostridium beijerinckii NRRL B-598 at the transcriptomic level.* Scientific reports, 2019. 9.1: 1-21.

[17] SEDLAR, Karel et al. *Transcription profiling of butanol producer Clostridium beijerinckii NRRL B-598 using RNA-Seq.* BMC genomics vol. 19,1 415. 30 May. 2018. doi:10.1186/s12864-018-4805-8

[18] MARBACH D., T. SCHAFFTER, C. MATTIUSSI a D. FLOREANO. *Generating Realistic In Silico Gene Networks for Performance Assessment of Reverse Engineering Methods.* Journal of Computational Biology, 2009. 16(2):229–239. [infoscience.epfl.ch/record/128148]

[19] STOLOVITZKY G., R.J. PRILL a A. CALIFANO. *Lessons from the DREAM2 Challenges.* In Stolovitzky G, Kahlem P, Califano A, Eds, Annals of the New York Academy of Sciences, 2009. 1158:159–95.

[20] STOLOVITZKY G., D. MONROE a A. CALIFANO. *Dialogue on ReverseEngineering Assessment and Methods: The DREAM of High-Throughput Pathway Inference.* In Stolovitzky G and Califano A, Eds, Annals of the New York Academy of Sciences, 2007. 1115:11–22. ¨

[21] EWELS, Philip et al. *MultiQC: summarize analysis results for multiple tools and samples in a single report.* Bioinformatics, 2016. 32.19: 3047-3048.

[22] LI, Heng et al. *The sequence alignment/map format and SAMtools.* Bioinformatics, 2009. 25.16: 2078-2079.

[23] LIAO, Y., G. K. SMYTH a W. SHI. *featureCounts: an efficient general purpose program for assigning sequence reads to genomic features.* Bioinformatics, 2014. 30.7: 923-930.

[24] HARVARD CHAN BIOINFORMATICS CORE. *Hbctraining DGE_workshop* [online]. 2021 GitHub, [cit. 27. 12. 2021]. Dostupné z URL: <`https://github.com/hbctraining/DGE_workshop`>

[25] ZHAO, S., Z. YE a R. STANTON. *Misuse of RPKM or TPM normalization when comparing across samples and sequencing protocols.* Rna, 2020. 26.8: 903-909.

[26] ROBINSON, M. D. a A. OSHLACK. *A scaling normalization method for differential expression analysis of RNA-seq data.* Genome biology, 2010. 11.3: 1-9.

[27] SMID, Marcel et al. *Gene length corrected trimmed mean of M-values (GeTMM) processing of RNA-seq data performs similarly in intersample analyses while improving intrasample comparisons.* BMC bioinformatics, 2018. 19.1: 1-13.

[28] LOVE, M.I., W. HUBER a S. ANDERS. *Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2.* Genome Biology, 2014 15, 550. doi: 10.1186/s13059-014-0550-8.

[29] BARBOSA, Sara et al. *A guide to gene regulatory network inference for obtaining predictive solutions: Underlying assumptions and fundamental biological and data constraints.* Biosystems, 2018, 174: 37-48.

[30] *DNA Sequencing Costs: Data from the NHGRI Genome Sequencing Program (GSP)* [online]. Bethesda: Wetterstrand KA [cit. 2021-11-13]. Dostupné z: <`www.genome.gov/sequencingcostsdata`>.

[31] KABIR, M., N. NOMAN a H. IBA. *Reverse engineering gene regulatory network from microarray data using linear time-variant model.* BMC bioinformatics, 2010. 11.1: 1-15.

[32] WU, Jun et al. *Large scale gene regulatory network inference with a multi-level strategy.* Molecular Biosystems, 2016. 12.2: 588-597.

[33] ROY, S. a P.H. GUZZI. *Biological network inference from microarray data, current solutions, and assessments.* In: Microarray Data Analysis. Humana Press, New York, NY, 2015. p. 155-167.

[34] HILL, S. M. et al. *Inferring causal molecular networks: empirical assessment through a community-based effort.* Nature methods, 2016. 13.4: 310-318.

[35] DUGGAN, David J. et al. *Expression profiling using cDNA microarrays.* Nature genetics, 1999. 21.1: 10-14.

[36] USADEL, Björn et al. *Co-expression tools for plant biology: opportunities for hypothesis generation and caveats.* Plant, cell environment, 2009. 32.12: 1633-1651.

[37] SHANNON, Claude Elwood. *A mathematical theory of communication.* The Bell system technical journal, 1948, 27.3: 379-423.

[38] ZHANG, Xiujun et al. *Inferring gene regulatory networks from gene expression data by path consistency algorithm based on conditional mutual information.* Bioinformatics, 2012. 28.1: 98-104.

[39] FAITH, Jeremiah J. et al. *Large-scale mapping and validation of Escherichia coli transcriptional regulation from a compendium of expression profiles.* PLoS biology, 2007. 5.1: e8.

[40] BUTTE, Atul J. a I. S. KOHANE. *Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements.* In: Biocomputing 2000. 1999. p. 418-429.

[41] DIMITRAKOPOULOS, Georgios N. et al. *A clustering based method accelerating gene regulatory network reconstruction.* Procedia Computer Science, 2014. 29: 1993-2002.

[42] RESHEF, David N. et al. *Detecting novel associations in large data sets.* science, 2011. 334.6062: 1518-1524.

[43] VILLAVERDE, Alejandro F. et al. *MIDER: network inference with mutual information distance and entropy reduction.* PloS one, 2014. 9.5: e96732.

[44] MARGOLIN, Adam A. et al. *ARACNE: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context.* In: BMC bioinformatics. BioMed Central, 2006. p. 1-15.

[45] ZOPPOLI, P., S. MORGANELLA a M. CECCARELLI. *TimeDelay-ARACNE: Reverse engineering of gene networks from time-course data by an information theoretic approach.* BMC bioinformatics, 2010. 11.1: 1-15.

[46] INUKAI, Sachi, K.H. KOCK a M. L. BULYK. *Transcription factor–DNA binding: beyond binding site motifs.* Current opinion in genetics development, 2017. 43: 110-119.

[47] NAKATO, Ryuichiro a Katsuhiko SHIRAHIGE. *Recent advances in ChIP-seq analysis: from quality management to whole-genome annotation.* Briefings in bioinformatics, 2017. 18.2: 279-290.

[48] ALTENHOFF, Adrian M. a Christophe DESSIMOZ. *Phylogenetic and functional assessment of orthologs inference projects and methods.* PLoS computational biology, 2009. 5.1: e1000262.

[49] SINGH, Harmanjit a Richa SHARMA. *Role of adjacency matrix adjacency list in graph theory.* International Journal of Computers Technology, 2012. 3.1: 179-183.

[50] VALIENTE, G. *Adjacency Maps and Efficient Graph Algorithms.* Algorithms, 2022. 15, 67. https://doi.org/10.3390/a15020067

[51] SZABO, Fred. *The linear algebra survival guide: illustrated with Mathematica* Academic Press, 2015. ISBN 978-0-12-409520-5.

[52] REVERTER A. a E.K.F. CHAN. *Combining partial correlation and an information theory approach to the reversed engineering of gene co-expression networks.* Bioinformatics, Volume 24, Issue 21, 2008. 2491–2497. https://doi.org/10.1093/bioinformatics/btn482

[53] ROY, S., D.K. BHATTACHARYYA a J.K. KALITA. *Reconstruction of gene co-expression network from microarray data using local expression patterns.* BMC Bioinformatics 15, S10, 2014. https://doi.org/10.1186/1471-2105-15-S7-S10

[54] *The R Project for Statistical Computing* [online]. [cit. 2022-04-28]. Dostupné z: https://www.r-project.org/

[55] *Available CRAN Packages By Name.* Cran.r-project.org [online]. [cit. 2022-05-04]. Dostupné z: https://cran.r-project.org/web/packages/available_packages_by_name.html

[56] GENTLEMAN, Robert. *R programming for bioinformatics.* Chapman and Hall/CRC, 2008. https://doi.org/10.1201/9781420063684

[57] *RStudio.* [online]. [cit. 2022-04-28]. Dostupné z: https://www.rstudio.com/

[58] LIAO, Y., G.K. SMYTH a W. SHI. *The R package Rsubread is easier, faster, cheaper and better for alignment and quantification of RNA sequencing reads.* Nucleic Acids Research, 2019. 47, e47. doi: 10.1093/nar/gkz114.

[59] LIAO, Y., G.K. SMYTH a W. SHI. *featureCounts: an efficient general-purpose program for assigning sequence reads to genomic features.* Bioinformatics, 30(7):923-30, 2014.

[60] BORCHERS W. H. *pracma: Practical Numerical Math Functions.* R package version 2.3.8., 2022. https://CRAN.R-project.org/package=pracma

[61] LUCY, D. a R. AYKROYD. *GenKern: Functions for generating and manipulating binned kernel density estimates.* R package version 1.2-60, 2013. https://CRAN.R-project.org/package=GenKern

[62] TRAPLETTI, A. a K. HORNIK. *tseries: Time Series Analysis and Computational Finance.* R package version 0.10-50, 2022.

[63] ZOU, M. a S.D. CONZEN. *A new dynamic Bayesian network (DBN) approach for identifying gene regulatory networks from time course microarray data.* Bioinformatics, 2005. 21.1: 71-79.

[64] STEUER, Ralf et al. *The mutual information: detecting and evaluating dependencies between variables.* Bioinformatics, 2002. 18.suppl_2: S231-S240.

[65] HÄRDLE, W., J. HOROWITZ, a J.-P. KREISS. *Bootstrap methods for time series.* International Statistical Review, 2003. 71.2: 435-459.

[66] ZARNEGAR, Armita. *Gene Regulatory Network Discovery Using Heuristics.* 2010. PhD Thesis. University of Ballarat.

[67] ALTAY, G. a F. EMMERT-STREIB. *Structural influence of gene networks on their inference: analysis of C3NET.* Biology Direct, 2011. 6.1: 1-16.

[68] MEYER, P. E., F. LAFITTE a G. BONTEMPI. *minet: AR/Bioconductor package for inferring large transcriptional networks using mutual information.* BMC bioinformatics, 2008. 9.1: 1-10.

[69] *NCBI: Clostridium beijerinckii NRRL B-598 chromosome, complete genome* [online]. [2022-05-03]. Dostupné z: https://www.ncbi.nlm.nih.gov/nuccore/CP011966.3?report=gbwithpartslog$=seqview

[70] SCHMID, Andreas et al. *Chemical and biological single cell analysis.* Current opinion in biotechnology, 2010. 21.1: 12-20.

[71] ALTAY, G. a F. EMMERT-STREIB. *Structural influence of gene networks on their inference: Analysis of C3NET.* 2010. URL: http://cran.r-project.org/web/packages/c3net/index.html

[72] MEYER, P.E., F. LAFITTE a G. BONTEMPI. *MINET: An open source R/-Bioconductor Package for Mutual Information based Network Inference.* BMC Bioinformatics, 9, 2008. http://www.biomedcentral.com/1471-2105/9/461.

Seznam symbolů a zkratek

CLR kontextová pravděpodobnost příbuznosti

cDNA komplementární DNA

FISH fluorescenční in situ hybridizace

GRN genová regulační síť

ChIP chromatin immunoprecipitation

MI vzájemná informace

PCR polymerázová řetězová reakce

RMC relativní molární koncentrace

mRNA mediátorová RNA

rRNA ribosomální RNA

sRNA malá regulační RNA

PPI protein-protein interkace

RNA-Seq RNA-sekvenování

RPKM/FPKM „čtení/fragmentů na kilobázi exonu na milion mapovaných čtení/fragmentů“

RT-PCR PCR s revezní transkripcí

SAGE sériová analýza genové exprese

TEP transkripční expresní profil

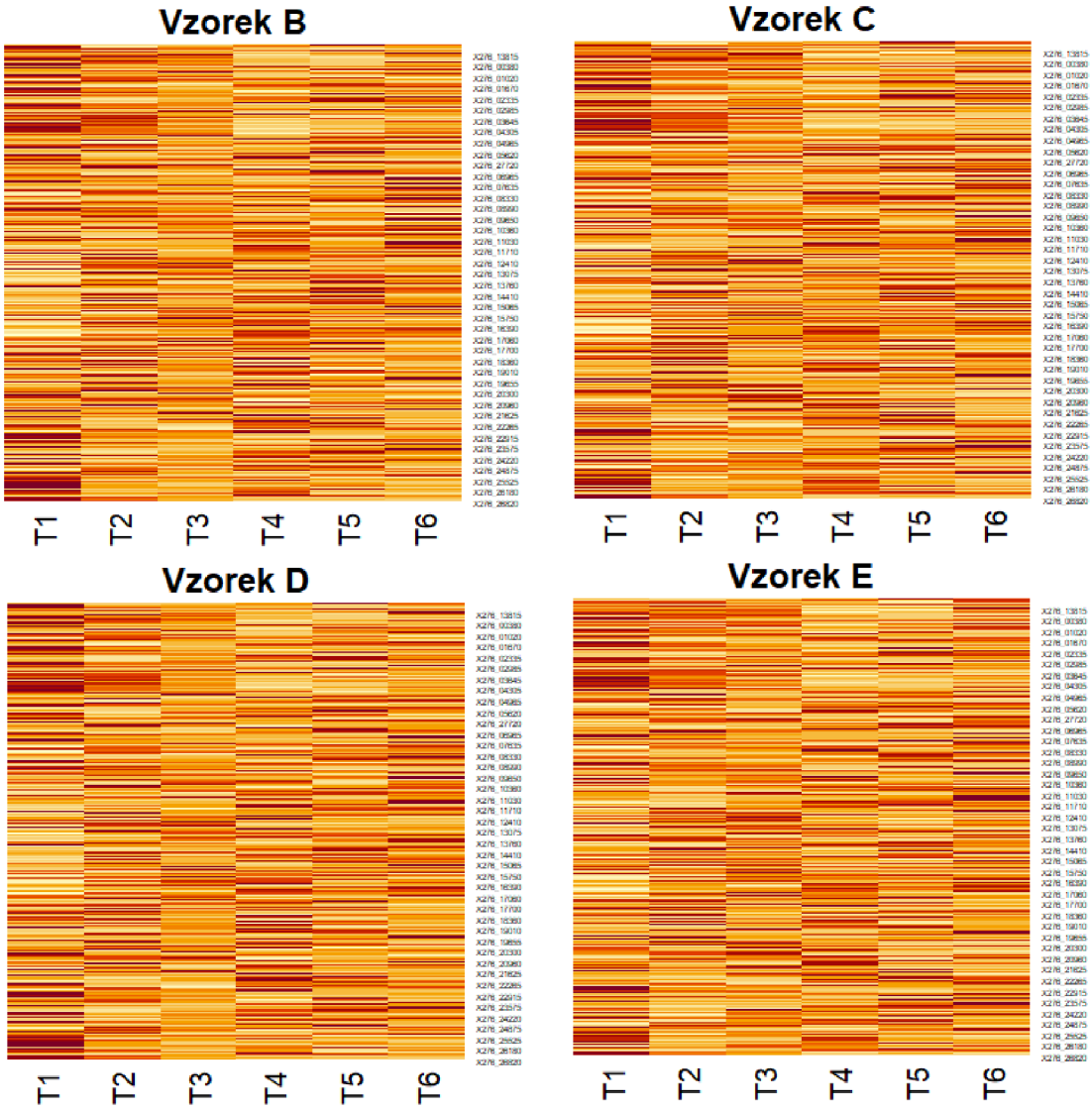
TMM oříznutý průměr M hodnot

TPM „přepis na miliony“

Seznam příloh

A Předzpracovaná data

A.1 Heatmapy



Obr. A.1: Heatmapy normalizovaných dat ze 4 experimentů s *Clostridium beijerinckii* NRRL B-598

A.2 Ukázka části tabulky normalizovaných dat genové exprese

Tab. A.1: Ukázka části tabulky normalizovaných dat genové exprese ze 4 časových bodů 1 experimentu *Clostridium beijerinckii* NRRL B-598

	B1_sorted.bam	B2_sorted.bam	B3_sorted.bam	B4_sorted.bam
X276_26820	2989,26	2413,63	1601,91	1146,04
X276_26815	2045,79	1619,35	1031,09	792,13
X276_26810	444,74	418,38	238,11	196,76
X276_26805	4328,24	2466,72	1639,85	1275,50
X276_26800	1373,12	905,77	563,66	442,06
X276_26795	9923,97	7348,12	4886,56	3220,49
X276_26790	16225,72	12428,10	9774,22	6896,24
X276_26785	38,91	22,30	29,15	23,00
X276_26780	13,50	12,74	12,65	8,94
X276_26775	27699,91	25534,73	13698,43	18122,80
X276_26770	139,77	60,53	72,59	94,12
X276_26765	0,79	0,00	0,00	2,13
X276_26760	145,33	90,26	98,44	108,60
X276_26755	418,53	441,74	399,79	307,48
X276_26750	957,77	999,22	1039,89	1098,34
X276_26745	99,27	107,25	124,28	199,74
X276_26740	50,83	16,99	29,15	68,14
X276_26735	146,92	38,23	57,19	84,32
X276_26730	137,39	41,41	68,74	67,29
X276_26725	212,04	50,97	84,69	102,64
X276_26720	178,69	104,06	116,03	87,30
X276_26715	1184,11	902,59	841,37	840,26
X276_26710	1456,51	982,23	870,52	852,61
X276_26705	38,12	167,78	650,55	871,77
X276_26700	8,74	15,93	21,45	27,26

B Implementace výpočtu vzájemné informace

Výpis B.1: Implementace výpočtu vzájemné informace v jazyce R

```
xgs <- 50
ygs <- 50

init_expresion <- unlist(count_table_Time[list_count_i,...
list_count_j])
if(std(init_expresion) == 0){
  init_expresion[1] <- init_expresion[1] + 1e-100
}

bandx <- tryCatch(dpik(init_expresion, scalest = "minim",...
level = 10L, kernel = "normal"), error = function(err) 0.2)
Px <- KernSec(init_expresion, xgridsize = xgs,...
xbandwidth = bandx)$yden
Px <- Px/sum(Px)

actual_expresion <- unlist(count_table_Time[i,y])
if(std(actual_expresion) == 0){
  actual_expresion[1] <- actual_expresion[1] + 1e-100
}

bandy <- tryCatch(dpik(actual_expresion, scalest = "minim",...
level = 10L, kernel = "normal"), error = function(err) 0.2)
Py <- KernSec(actual_expresion, xgridsize = ygs,...
xbandwidth = bandy)$yden
Py <- Py/sum(Py)
Pxy <- KernSur(init_expresion, actual_expresion,...
xgridsize = xgs, ygridsize = ygs, xbandwidth = bandx,...
ybandwidth = bandy)$zden
Pxy <- Pxy/sum(Pxy)

MI <- 0
for (MI_i in 1:xgs) for (MI_j in 1:ygs) {
  tmp <- Pxy[MI_i, MI_j] *...
  log2(Pxy[MI_i, MI_j] / Px[MI_i] / Py[MI_j])
  if (tmp != "NaN")
    MI <- MI + tmp
}
```

C Ukázka části výsledné adjacency matrix

Tab. C.1: Ukázka výsledné adjacency matrix

	X276_ 26820	X276_ 26815	X276_ 26810	X276_ 26805	X276_ 26800	X276_ 26795	X276_ 26790	X276_ 26785	X276_ 26780	X276_ 26775
X276_ 26820	0	0	0	0	0	1	0	1	0	0
X276_ 26815	0	0	0	0	0	-1	0	1	0	1
X276_ 26810	0	0	0	0	0	0	0	1	0	0
X276_ 26805	0	0	0	0	0	-1	0	0	0	1
X276_ 26800	0	0	0	0	0	-1	0	0	1	1
X276_ 26795	0	0	0	0	0	0	1	0	1	0
X276_ 26790	0	0	0	0	0	1	0	0	0	0
X276_ 26785	-1	0	1	0	1	0	0	0	0	0
X276_ 26780	0	0	0	0	1	0	0	-1	0	0
X276_ 26775	0	1	0	1	1	-1	0	0	0	0

D Seznam elektronických příloh

- /Pirkl_Petr_DP_prilohy/hodnoceni.R
  - implementované hodnocení GRN pomocí F-skóre v jazyku R
- /Pirkl_Petr_DP_prilohy/normalizovane_pocty.R
  - implementovaný algoritmus pro zisk a normalizaci count table z .BAM souborů v R jazyku
- /Pirkl_Petr_DP_prilohy/normalized_counts.csv
  - normovaná count table genové exprese *Clostridium beijerinckii* NRRL B-598 pro 4 experimenty, připravená k nahrání do program.R
- /Pirkl_Petr_DP_prilohy/program.R
  - implementovaný navržený algoritmus pro tvorbu GRN v jazyku R