Aplikace procesu dolování dat v biologii - genetice

David ZEMAN, Doktorský studíjní program (1)
UIFS, FIT, VUT v Brně
E-mail: zemand@fit.vutbr.cz

Abstrakt:

Tato kompilace se zabývá využitím procesu získávání znalostí v oblasti genetiky. Snaží se postihnout celý proces a nastínit problémy jednotlivých fází. Představuje stručně dostupná řešení, různé varianty a trendy ve vývoji. Obsahuje definici základních problémů v oblasti genetiky a použitelnost procesu získávání znalostí při jejich řešení. Blíže se zaměřuje na proces dolování dat a využití shlukování. Nabízí základní dělení shlukovacích algoritmů a popisuje princip nejznámějších a nejpoužívanějších zástupců těchto skupin.

1 Úvod

Markantní rozmach molekulární biologie, ke kterému došlo za posledních 30 let přinesl další směr, kterým lze zkoumat živé organismy. Objevení technologie rekombinace DNA v sedmdesátých a osmdesátých letech umožnil testovat a objevovat jednotlivé geny i v běžných laboratořích. Pokročilé techniky zpracování DNA umožnili zabývat se celým genetickým materiálem jednotlivých organismů. Nejlépe je vidět asi pokrok na konceptech představených nedávno, které se snaží mapovat celý lidský genom.

Jako nejtěžší úkol se jeví hledání funkce příslušného genu. Již dlouho se ví, že funkce genu *in vivo* (v organismu) je dána 3D strukturou. Tato struktura je poměrně lehce detekovatelná pomocí rentgenové krystalografie. Umožňuje zkoumat geny na úrovni atomů a je nejpoužívanější metodou pro určování proteinů. Více v [1]. Je vyvíjena značná snaha kombinovat informace o prostorovém uspořádání s dalšími poznatky pocházející z tradičních biochemických měření, pomocí technik pro analýzu genetických řetězců nebo získaných pomocí mikročipů.

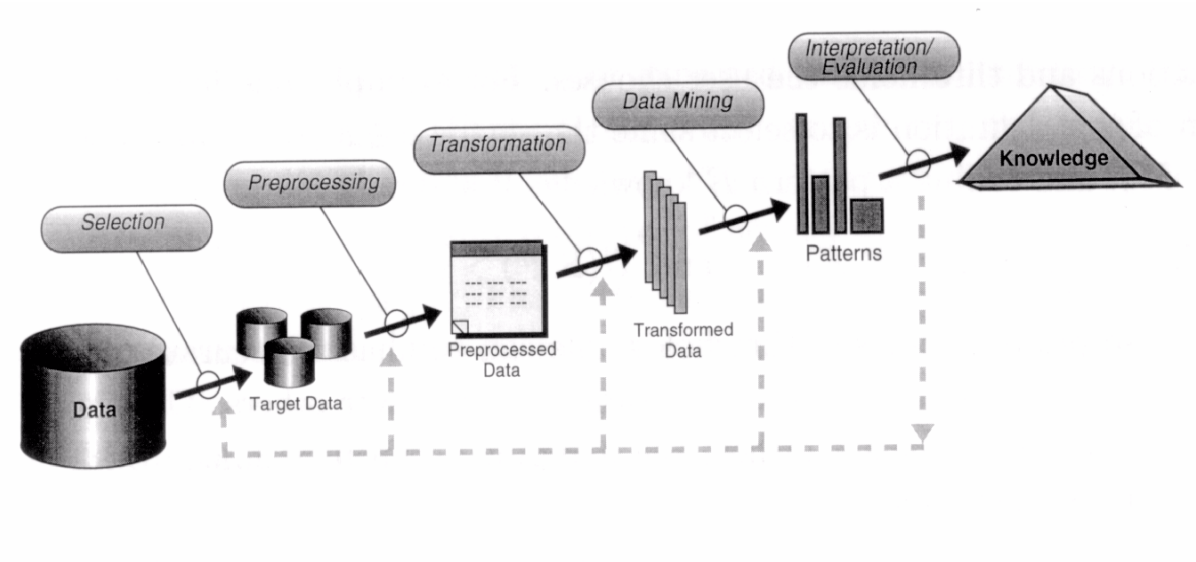
Prostorové uspořádání je pouze jeden z údajů, které lze uchovávat. Dalšími typy mohou být genomy, což jsou sekvence DNA a jejich pozice. Proteomy – plný počet proteinů. Metabolické cesty znázorňující biochemické reakce obsahující četné proteiny, malé molekuly a vzájemné interakce. Další informace se mohou týkat opteronů atd.

2 Proces získávání znalostí

Vidíme, že v oblasti genetiky je možné získat nespočetné množství dat, se kterými je možné pracovat mnoha různými způsoby. Aplikace data miningu na tato data nás může posunout dál a přinést další užitečné informace. Může ulehčit práci s těmito daty nebo lépe tato data pochopit a interpretovat.

Tento proces se nazývá *získávání znalostí (knowledge discovery)*. Užším pojmem vyjadřujícím konkrétní aplikaci algoritmů na množinu dat se nazývá *dolování dat (data mining).* Celý proces se stává stále častějším námětem výzkumných prací a projektů, což je zapříčiněno daleko větší schopností efektivně generovat a uchovávat obrovské množství dat, než tomu bylo dříve. S tím vyvstává potřeba dalšího zdokonalování procesu vhodného ukládání, vyhledávání a zpracování, zejména kvůli jejich objemnosti a rozličné struktuře informaci. Obecný postup práce s daty a jejich použití v procesu získávání znalostí lze zjednodušeně popsat následovně :

- **“čištění“ dat** (data cleaning) – odstranění nekonzistentních dat.
- **sjednocení dat** (data integration) – sjednocení datových zdrojů v celek.
- **transformace dat** (data transformation) – převod dat na takovou podobu, která vyhovuje metodě pro dolování.
- **dolování dat** (data mining) – aplikace vybraného algoritmu za účelem získání cílených dat.
- **vyhodnocení** (pattern evaluation) – dolovací algoritmus může vrátit značně objemné množiny dat, vzorků, pravidel, proto je třeba vybrat pouze to nejnutnější, popřípadě zhodnotit užitečnost, novost atd.
- **prezentace** (knowledge presentation) – předání výsledků uživateli za použití nejrůznějších vizualizačních a reprezentačních technik a metod.



Obr. 1: **Proces získávání znalostí**

Jednotlivé body celého procesu získávání znalostí, tak jak na sebe navazují, jsou přehledně uvedeny v obrázku 1. Organizace kapitol v této práci se snaží též dodržet toto schéma. Práce zabývající se aplikací toho procesu v oblasti genetiky a biologie obecně je čím dál více. Samozřejmě je potřeba tento postup vhodně upravovat dle požadovaných výsledků, konkrétních potřeb a možných vstupů. Možných úprav a zdokonalování je možné dosáhnout v každém bodě, které jsou zmíněny v předešlém odstavci. Tomu odpovídá i rozmanitost článků a knih zabývající se touto tematikou. Trendy ve výzkumu ukazují, že tato problematika je velice důležitá a je snaha hledat možná vylepšení ve všech bodech tohoto procesu.

Následující kapitola pojednává o fázi přípravy dat pro aplikaci konkrétních algoritmů. Ukazuje různé způsoby zvýšení efektivity v této části procesu získávání znalosti. Následuje kapitola představující multi-relační dolování dat, jako nástupce klasického přístupu relačního dolování. Kapitola 5 popisuje konkrétní dolovací nástroje a seznamuje zejména s oblastí shlukování. Kapitola 6 se zabývá interpretací výsledků a ukazuje důležitost tohoto kroku na příkladu správy asociačních pravidel. Poslední kapitola se snaží zmapovat obecně problémová místa celého procesu získávání znalostí.

3 Preprocessing

Již na počátku celého procesu se dá docílit výrazného zrychlení a to vhodnou přípravou dat, výběrem vhodné databáze apod. Vědci jsou si vědomi toho, že čím více dat použijí zaráz, tím kvalitnější výsledky mohou dostat, což se týká nejen množství, ale i různých typů dat. Pro

databáze obsahující několik typů dat nebo dokonce pro daný problém všechny, je zdůrazňována nutnost použití nových multi-relačních nástrojů.

Biologové se snaží pomocí získaných vědomostí ovlivnit chování biologických procesů, například vývojem nových farmak. Z toho důvodu musí být testovány miliony směsí *in vitro* (ve zkumavce) a *in vivo* (v těle organismu). Účele je zjistit, zda se naváží na požadovaný protein nebo zda mají cílený efekt. Zjištěním, že daná látka má požadovaný efekt však proces nekončí. Látka totiž může vykazovat jiné nepřípustné parametry, například vysokou toxicitu, může mít příliš krátkou dobu rozpadu nebo naopak příliš dlouhou. Nemusí proniknout přes střevní stěnu do krevního oběhu apod. Je tedy potřeba testovat řadu dalších faktorů. Důsledkem je však další nárůst jak velikosti tak rozmanitosti dat.

K značnému nárůstu různorodosti dochází také mezi jednotlivými systémy pro správu dat. Při tvorbě takovýchto systémů nebo jen souborů s biologickými daty neexistuje žádný standard nebo referenční postup. Proto každý, kdo s těmito daty pracuje, vytváří vlastní, stále většinou nový, systém včetně databáze svých poznatku a tento systém upravuje dle svých potřeb. V poslední době se sice začínají objevovat zejména v oblasti mapování lidského genomu referenční databáze, zatím však nijak velkého významu.

3.1 Integrace zdrojů dat

Proto vznikají různé systémy pro integraci heterogenních dat z různých zdrojů. Kostru takového systému představuje [1]. Snaží se představit systém, jehož má umožnit efektivněji zodpovídat některé otázky biologů díky provázání nově nabytých informací o strukturálních vlastnostech zkoumaných prvků s jinými typy znalostí. Jiné typy znalostí se nazývají derivované. Autoři se snaží vytvořit integrovaný systém dovolující ukládání a zpracování biologických dat. Vytvořili také kaskádní systém, který dovoluje aktualizovat data mezi mapovanými databázemi. Mezi používané databáze patří PDB – The Protein Data Bank, databáze které uchovává 3D souřadnice molekul. Mezi databáze s derivovanými daty patří PRINTS nebo PROSITE databáze, které se zaměřují na klasifikaci proteinů, následné vytváření rodin a jejich domén.

Integrace takovýchto heterogenních databází je velice složitá z hlediska transformací schémat každé z databází. Každá informace musí být identifikována. Globální schéma by mělo být dostatečně obecné, aby bylo možné pracovat se všemi heterogenními datovými modely. V takovýchto situacích se omezení relačních databází stávají problematická.

Autoři si zvolili XML jazyk jako meta-jazyk pro výměnu dat mezi různými datovými zdroji. XML data padají do třídy semistrukturovaných dat, jelikož se nepodřizují přísnému schématu. Jako dotazovací jazyk byl autory zvolen XQL jazyk. Pro práci s databázemi se zdál nejvhodnější

systém SODA, což je client-server systém vhodný pro správu XML informací. Dále je detailněji popsáno mapování dat a transformace do požadovaného XML formátu. Funkčnost navrženého systému je ilustrována na problému hledání strukturálních charakteristik aminokyselin, kdy autoři využívají dat z již zmíněných databází PDB a PROSITE.

3.2 Indexování databází

Důležitá je také struktura databáze a způsob, jakým k datům přistupujeme. DNA sekvence můžeme zjednodušeně definovat jako značně dlouhé řetězce skládají se ze čtyř písmen A, C, G, T. Proteiny (bílkoviny) využívající abecedu 20ti symbolů jsou překlady úseků DNA. Při překladu se využívají překladové tabulky, kdy každé tři písmena DNA jsou přeložena na jednu aminokyselinu (AA – amino acid).

Délka DNA sekvencí se měří v bázových párech (bp – base pairs) a většinou stačí uvést pouze jednu bázi, jelikož druhá je komplementární (A je komplementární s G, C je komplementární s T). Kvůli obrovské velikosti genomu organismů se spíše používají jednotky Gbp – Gigabase pairs nebo Mbp Megabase pairs. Jako příklad lze uvést, že savci mývají v průměru 3 Gbp dlouhý genom. Velikosti databází zmapovaných genomů organismů na internetu se pohybují v desítkách Gbp v závislosti na druhu databáze. Hledání sekvencí DNA se z počátku dělo sekvenčně s využitím několika filtrovacích technik, které umožňovali vyloučit některé oblasti a tím zrychlit prohledávání. Později se začal využívat velký výpočetní výkon. Příkladem může být Sanger centrum se 400 počítači. I tento výkon však nezaručuje rychlé nalezení správné odpovědi. V dnešní době se zvyšuje poptávka po mechanismech, které by umožnili systémy na hledání sekvencí DNA daleko více dostupnější a více nezávislé na výpočetním výkonu. Proto autoři [12] navrhují nový systém indexování databáze, který umožňuje indexovat rozsáhlé databáze a urychlit vyhledávání v nich. Řada stávajících algoritmů pro indexování nemůže být využita, jelikož DNA nelze dělit na slova. Ze stejného důvodu nelze využít prefixové indexování. Jelikož jde o podobnostní vyhledávání, tedy nejde o přesné nalezení shodného vzorku, algoritmy založené na B-stromech, q-gramové algoritmy apod. též nejsou vhodné. Navrhované řešení je založeno suffixovém indexování, které je upraveno pro rozsáhlé databáze, a využívá suffixový stromů. To jsou komprimované digitální stromy obsahující všechny suffixy daného řetězce. Kořen stromu je vstupní bod a startovací index každého suffixu je uložen v listu. Každý suffix může být jednoduše přečten průchodem stromu od kořene k listu. Při tvorbě stromu jsou přidávány terminální značky a suffixové linky, které výrazně zvyšují efektivnost využití suffixového stromu.

Důležitým bodem je též efektivní implementace podobnostního porovnávání. Již bylo uvedeno, že nejde o přesné vyhledávání podřetězců v řetězcích, ale hledání podobných

sekvencí. Techniky zabývající se touto problematikou bychom mohli rozdělit na dynamické programování, automaty a filtrovací techniky. V kontextu genetiky se jako nejvhodnější jeví dynamické programování. To zahrnuje výpočet matice, která má jako jednu dimenzi text a jako druhou dimenzi hledaný vzorek. Použitím hodnotící funkce, která odměňuje shodu písmen a trestá neshodu, lze změřit podobnost dvou řetězců.

Známým systémem, který řeší tuto problematiku, je BLAST a jeho varianty [13]. BLAST pracuje ve čtyřech krocích. V prvním kroku se vytváří dotazovací slova (hledané sekvence) délky *k*. V druhém kroce se prochází databáze a vyhledávají se v řetězcích úseky, které lze párovat se zadanými slovy tak, že jejich skóre je větší, než definovaný práh. Tyto páry se označují jako semínka a jsou uložena do vyhledávací tabulky. Na toto sekvenční procházení databáze používá BLAST konečný automat. Následuje krok, kdy se pracuje se semínky. Ta jsou rozšiřována na delší podobné segmentové páry s větším skóre. Nakonec se provádí srovnání řetězců, které jsou nejvíc podobné dotazovanému slovu. Ve [13] se lze setkat s variantou BLAST++, která na základě testů dosahuje daleko lepších časových výsledků. Toho dosahuje tím, že pracuje s několika dotazovacími slovy zároveň, s kterými zachází jako s jedním virtuálním dotazem. Tím se snaží upravit postup, kdy dochází k vyhledávání semínek, který je slabým místem algoritmu BLAST.

3.3 Zpracování obrazu

Do fáze preprocesingu nepatří pouze správa uložených dat, ale také zpracování obrazu. To je problém řešený zejména v rámci oboru počítačové vidění, což je vědní disciplína, která se snaží technickými prostředky alespoň částečně napodobit složitý proces lidského vidění. Při vyhodnocení vizuální informace však není důležité jen zrakové ústrojí, ale také inteligence člověka, která umožňuje v praxi aplikovat dlouho nabývané zkušenosti a znalosti o okolním světě. Výzkum počítačového vidění se snaží o napodobení řešení analogických úloh.

Ve zpracovaní obrazu můžeme rozlišit několik logických kroků. Na nejvyšší úrovni jde o proces pochopení obsahu, kde řešená úloha má často základní rysy problémů řešených v oblasti umělé inteligence popř. expertních systémů. Základem je však dokonalá spolupráce s jednotlivými kroky zpracování obrazu na nižších úrovních. Cílem nižší úrovně počítačového vidění je analyzovat vstupní dvojrozměrná obrazová data číselného charakteru a najít kvalitativní symbolickou informaci potřebnou pro vyšší úroveň.

Postup zpracovávání a rozpoznávání obrazu reálného světa se daří rozložit do posloupnosti základních kroků:

- Snímání, digitalizace a uložení obrazu v počítači
- Předzpracování

- Segmentace obrazu na objekty
- Popis objektů
- Porozumění obsahu obrazu nebo jeho klasifikace

Fázi segmentace obrazu a transformací dat do podoby, která je vhodná pro dolování, se obecně zabývá závěrečná část práce [8] a ukazuje na možnosti použití shlukování při této činnosti a demonstruje je na příkladech. My se nebudeme detailněji zabývat touto oblastí a se shlukováním se seznámíme blíže v kontextu dolovacích úloh.

4 Multi-relační data mining

Z podobných důvodů, jaké byly řečeny v úvodu kapitoly 3, tedy obecně velké množství a rozmanitost dat, se objevuje snaha přesunout se k multi-relačnímu data miningu. Jako multi-relační data mining (MRDM) se označuje proces získávání znalostí z relačních databází obsahujících vícenásobné vztahy. Do tohoto pole výzkumu patří také dolování dat nad vysoce složitými daty. Tato oblast se zaměřuje na spojení znalostí z oblastí jako jsou ILP (indukční logické programování), KDD (knowledge data discovery – získávání znalostí), strojové učení, relační databáze. Pro dolování dat je typické hledání vzorků nad jednou relací v databázi. Pro mnoho aplikací je však shromáždění dat do jedné tabulky neřešitelný problém, navíc pokud realizováno, vede k ztrátě informací. Multi-relační dolování umožňuje pracovat s daty bez nutnosti transformovat je nejdříve do jedné tabulky. Současné MRDM systémy umožňují typické dolovací úlohy jako je asociační analýza, klasifikace, shlukování, pravděpodobnostní modely a regrese.

Známou úlohou aplikace ILP na molekulární data je program GOLEM, který umožňuje modelovat aktivity a vztahy v požadovaných strukturách. Jako další systém můžeme uvést PROGOL, který byl užit k získávání informací o strukturách látek, které jsou mutagenní. Zde se ukázala nutnost přímého použití multi-relačních systémů, jelikož převedení dat na vektory rysů vedly k vytvoření vzorků obsahujících každý milióny rysů. Více informací lze nalézt ve [2], která obsahuje také odkazy na výsledky testů srovnávajících klasické RDM a MRDM systémy.

Článek [10] přináší podrobnější pohled na problematiku z hlediska ILP. Popisuje vazby mezi relačními databázemi a logickým programováním. Zavádí základní pojmy predikátové logiky. K pojmům frekventované množiny a asociační pravidla zavádí pojmy frekventované relační množiny a relační asociační pravidla. Frekventované relační množiny jsou datalogové dotazy a relační asociační pravidla jsou pak odvozeny z těchto dotazů. Na tomto principu pracuje i známý ILP systém WARMR, sloužící k detekci molekulárních podřetězců. WARMR vylepšuje klasický apriori algoritmus, jehož hlavní výhodou je generování frekventovaných

množin. Od Apriori algoritmu se liší ve hledání četnosti výskytu jednotlivých dotazů (stanovení frekvencí) a v generování kandidátních dotazů.

Asi nejvíce diskutovaným směrem použití MRDM v genetice je aplikace pravděpodobnostních modelů za účelem vytvořit genový regulační systém s využitím jak klasických dat tak dat z mikročipů.

Genová exprese je dvoukrokový proces, ve kterém je gen přepsán do mRNA a následně je tato mRNA přeložena na protein. Pomocí mikročipů se v rámci prvního kroku měří míra tvorby mRNA. Je to proces jednodušší než měřit, do jaké míry je jsou produkovány odpovídající proteiny. Víme, že když exprese jednoho genu (tedy zahájení přepisu a následná tvorba odpovídajícího proteinu) se zvedne, může ten proces ovlivnit expresi ostatních genů včetně sama sebe. Takový vliv je zvláště patrný, jestliže gen kóduje tzv. transkripční faktor. Transkripční faktor je protein, který se váže na podřetězec DNA před gen a podněcuje nebo potlačuje odstartování transkripce. Podřetězec kde se váže transkripční faktor se nazývá vazební místo transkripčního faktoru. Jestliže dva geny mají stejné profily expresí, pak je pravděpodobné, že jsou kontrolovány stejnými transkripčními faktory a stejná vazební místa v sekvencích, které jim předchází.

Na pravděpodobnostní modely lze nahlížet jako na Bayesovu síť, kde proměnné této sítě korespondují s jednotlivými poli databáze. Známá je práce pana Segala, který se zabývá touto tematikou. Odkazy na jeho práci lze nalézt v [2]. Ten ve své aplikaci uchovává jednu tabulku pro geny, jednu tabulku pro experimenty a jednu tabulku pro data z měření genové exprese na mikročipech. Každé měření je vždy pro jeden gen v rámci jednoho experimentu a za jedné „sady“ určitých podmínek. Pak měření genové exprese v databázovém schématu zachytává vícenásobné vztahy , které existují mezi geny a experimenty. V rámci Bayesovi sítě je aplikován EM (expectation – maximization) algoritmus, který se zároveň učí, které geny jsou ovlivňovány kterými transkripčními faktory a shlukuje experimenty to skupin. Celý systém by nebyl tolik složitý, kdybychom znali všechna data. Právě příslušné vazby genů a transkripčních faktorů a příslušnost jednotlivých experimentů do skupin však neznáme.

Několik aplikací již bylo vytvořeno se stejnými funkcemi ovšem vždy jak měření genové exprese tak hledání vazebních míst transkripčních faktorů bylo vždy oddělené. Systém Segala ukazuje, že spojení dvou různých typů dat a práce s nimi zároveň umožňuje dosáhnout lepších výsledků.

Ve [2] lze nalézt diskuse i k jiným oblastem použití MRDM jako je extrakce informací z textu nebo analýza sekvencí.

5 Dolování dat

Dolování dat je část celého procesu získávání znalostí, kde jsou aplikovány konkrétní algoritmy. Předpokládají se již čistá data, která byla předzpracována, a je možné se již zabývat dolováním požadovaných znalostí jako jsou již dříve zmiňovaná asociační pravidla, příslušnost do tříd nebo shluků.

5.1 Definice problému

DNA mikročipy umožnili monitorovat současně dosažené úrovně expresí tisíce genů. Důležitým úkolem je často identifikovat nejen úroveň exprese jednotlivých genů, ale i množinu genů, u kterých dochází k expresi zároveň, tedy ovlivňují navzájem svoji tvorbu. Dále pak se hledají vzory genových expresí. Množiny genů, které dosahují stejné míry exprese ve stejný čas, odpovídá stejný vzor. Zatímco vzor genových expresí charakterizuje společný trend ve vývoji úrovně exprese pro skupinu genů. Jednodušeji řečeno je vzor šablonou, od kterého se míra exprese pro různé geny v rámci jedné skupiny liší minimálně. Obecně tyto závislosti hledáme, jelikož geny, u kterých dochází k expresi zároveň, patří do stejné funkční kategorie. Vzory pak charakterizují důležité cellulární procesy a poukazují, jak dochází během procesu k regulaci genové exprese v buňkách. Ideální pro hledání těchto závislostí je použití shlukovacích algoritmů, kde jednotlivé shluky reprezentují funkčně podobné skupiny genů.

Blíže bychom mohli říci, že biologové chtějí provést například řadu experimentů měřící úroveň genové exprese buněk nebo množiny buněk za různých okolních podmínek ovlivňujících tyto úrovně. Snaží se porozumět, jak je míra genové exprese ovlivněna druhem tkáně, stářím organismu, terapeutickými faktory (např. léky) nebo okolím. Z výpočetního hlediska je úroveň exprese reálné číslo. Proto výsledek jednoho experimentu je pole reálných čísel. Množina genů na mikročipu je určena potřebami biologů a zůstává stejná pro všechny experimenty. Dostáváme tedy matici, kde každý prvek reprezentuje úroveň exprese určitého genu v určitém experimentu. Mimoto je zajímavé sledovat, jak se úroveň exprese genů mění od běžných hladin exprese v tělech organismů. Je tedy zajímavější uchovávat odchylky od normálu než uchovávat absolutní hodnoty naměřené na čipu. Zde se objevuje potřeba normalizace a výsledkem může být matice obsahující tři hodnoty. Tyto tři hodnoty vyjadřují, zda se úroveň exprese zvýšila, snížila nebo zůstala nezměněna. Nutno podotknout, že v některých aplikacích se lze setkat s více než třemi hodnotami, ovšem jde spíše o výjimečný jev. S bližší specifikací problémů týkajících se analýzy mikročipů se lze setkat v [5, 7, 9].

Okrajově bych se chtěl zmínit také o normalizaci. I tato část sebou přináší řadu problému a vyžaduje pozornost, pokud má být výsledný systém efektivní. Data z mikročipů obsahují dle [6] dva druhy chyb, náhodné a symetrické. Normalizace je pak běžným postupem,

jak minimalizovat symetrické chyby. Může jít o globální normalizaci, log-transformaci, regresní normalizaci apod.

Symetrické chyby jsou konstantní mezi vzorky z jednoho experimentu, ale mezi experimenty se liší. Symetrické chyby v rámci jednoho experimentu můžeme považovat za intra-experimentální odchylky. Většinou máme však data z různých experimentů a u každého experimentu odlišné symetrické chyby. Odtud pak změny mezi těmito odchylkami lze definovat jako inter-experimentální chyby. Většina normalizačních postupů má však problémy vypořádat se právě s inter-experimentálními chybami. Proto Fung [6] zavádí kolizní faktory (IF – impact factors), aby bylo možné měřit odchylky mezi jednotlivými třídami vzorků v trénovací množině a heterogenními daty získané z různých experimentů. Následně představují zabudování těchto IF do klasifikátorů pro klasifikaci heterogenních vzorků dat (dat z různých experimentů) a jejich testování na vzorku dat pro detekci rakoviny plic. Následující kapitola představuje nový přístup při dolování znalosti. Pak již jsou představeny jednotlivé skupiny shlukovacích algoritmů.

5.2 Explorativní dolování

Novým přístupem v této oblasti je interaktivní explorativní způsob [3]. Navrhovaný systém GeneX (Gene Explorer) pracuje ve dvou krocích.

Krok jedna zahrnuje extrakci informací o vztazích mezi geny. Ty jsou uloženy do atrakčního stromu (attraction tree). Taktéž shromažďuje informace o shlucích v datech. Jakmile je jednou tento strom vybudován, není již nutné pracovat s původní množinou dat.

Krok dvě je krokem, kdy dochází k indexaci vzorů a genů. Geny jsou seřazeny ve vytvořeném indexovaném seznamu. Seznam je tvořen tak, že geny odpovídající stejnému vzoru zůstávají blízko sebe i v tomto seznamu. Následně může být generován indexový graf vzorů. Na x-ové ose jsou jednotlivé geny seřazené stejně jako v seznamu. Na y-ové ose jsou hodnoty jednotlivých vzorů. Jestliže má následující část seznamu stejný vzor, první gen vykazuje vysokou hodnotu („puls“) a následující geny mají nízkou hodnotu . Index graf tak nabízí intuitivní a informativní nástroj zobrazující výsledek shlukování.

Třetím krokem je interakce s uživatelem. Uživatel může pracovat s jednotlivými pulsy a zanořovat se v grafu na nižší úrovně a tak zjišťovat jak se jednotlivé pulsy rozpadají na menší. Lze tak rekurzivně prozkoumat skupiny genů se stejnou mírou exprese.

Autoři také představili koncept na rozšíření toho systému. Jde o úpravu tohoto systému tak, aby mohl pracovat i s jinými typy dat, jako jsou vzorky z interakcí proteinů nebo výsledky spektrometrie.

5.3 Shlukování

Shlukování je proces sdružování dat do skupin disjunktních tříd zvaných shluky. Proces sdružování je založen na maximální podobnosti uvnitř shluku a minimální podobnosti mimo shluky. V tomto se liší od klasifikace, nebo-li rozdělení do tříd, kdy předem máme vzorky, kde známe příslušnost do jednotlivých tříd. Prvním krokem je pak trénovací proces, kdy se na základě těchto vzorků učíme charakterizovat jednotlivé třídy, tak abychom byli schopni do nich zařadit nové vzorky. Obecně můžeme shlukovací algoritmy rozdělit na tři kategorie – dělící metody, hierarchické metody a metody založené na vzorech. Dělící metody mohou být dále rozděleny z hlediska předpokládání shluků a optimalizace na K-means algoritmy a jeho derivace, SOM algoritmy a rozšíření, grafové algoritmy a algoritmy založené na modelech. Hierarchické metody mohou být rozděleny na aglomerativní a divizivní podle toho, jak je budována hierarchická struktura.

Při shlukování se můžeme setkat s celou řadou rozličných typů proměnných. Mohou to být intervalové, binární, nominální, ordinální nebo smíšené proměnné. Důležitou vlastností při shlukování je schopnost vyjádřit odlišnost nebo nějak postihnout rozdíl mezi hodnotami dvou objektů v množině dat. Nebo bychom mohli naopak říci, že se snažíme postihnout nějakým způsobem míru shody mezi objekty. Proto vznikají různé koeficienty vyjadřující vzdálenost mezi objekty datové skupiny. Pro intervalové proměnné jsou to například Eukleidovská, Minkowského nebo Manhattanská vzdálenost. Pro binární proměnné jsou to např. Jaccardův, Michenerův, Sokalův koeficient a mnoho dalších. Tato problematika vyjádření míry podobnosti by vydala na samostatnou práci, proto nejdůležitějším shrnutím je, že jsme téměř ve všech případech schopni vyjádřit míru podobnosti mezi objekty a to, i když jsou popsány proměnnými různých typů. Detailněji se lze s touto problematikou seznámit v [8].

5.3.1 K-means algoritmus a jeho derivace (dělící algoritmy)

K-means algoritmy by se dali nazvat K-středové. Na začátku je třeba specifikovat počet shluků *K*. Algoritmus pak dělí množinu dat do K disjunktních shluků. Kritérium příslušnosti ke shluku je vzdálenost od těžiště. Algoritmus byl odzkoušen na databázi naměřených genových expresi [3]. Popis algoritmů lze nalézt ve [3, 8, 11]. Ovšem K-means algoritmus má řadu nevýhod. Jednou z nich je nutnost zadat počet shluků. To je často velmi složité předpokládat. Navíc tento algoritmus není odolný proti šumu a přiděluje každý prvek v databázi k nějakému shluku. To může znamenat značné posunutí těžiště shluku a následně špatnou interpretaci výsledků. Snaha o překonání těchto nevýhod vedla ke vzniku řady variant tohoto jinak velice populárního algoritmu a tak vznikly algoritmy K-medoids, Adapt_Cluster a řada dalších. Tyto algoritmy využívají pravděpodobností při určování, do kterého shluku se prvek zařadí.

Obecně lze říci, že tyto metody vyžadují vstupní parametry. Navíc pokud je algoritmus spuštěn, chová se jako „černá skříňka“. Proces shlukování je bez jakékoliv interakce s uživatelem.

Zajímavou variantu lze najít v [11]. Jde o Fuzzy C-Means shlukovací algoritmus. Představme si, že máme množinu X s n objekty, kde každý objekt je popsán m atributy. Mějme též množinu C s k třídami. Hlavním rysem fuzzy c-means algoritmu je, že každému objektu x z X přiřazuje algoritmus hodnost vyjadřující členství $\mu(x,c)$, takový, že $\mu(x,c)$: X x C $\rightarrow$ [0,1]. Algoritmus se zastaví, jestliže změna hodností všech objektů se sníží pod definovaný práh. Pro měření vzdáleností se používá Eukleidovská vzdálenost. Algoritmus pracuje velice špatně pokud se tvary shluků výrazně odlišují od sférických tvarů. Tento problém se snaží řešit Kohenovy sítě.

5.3.2 SOM a jeho rozšíření

Self-Orgainzing Maps byly vynalezeny Kohenem na základě jednovrstvých neuronových sítí. Též se nazývají topologicky organizované neuronové sítě nebo Kohenovy mapy. Jde o jednoduchou soutěživou plně propojenou síť. Po přiložení vstupního vektoru začnou neurony mezi sebou soutěžit až zvítězí jediný. Ten představuje shluk, do kterého síť zařadila vstupní vektor. Učení spočívá v modifikaci vah vítězného neuronu a jeho topologických sousedů.

SOM algoritmus byl aplikován např. v případě studii hematologické diferenciace. Vzory 1 036 lidských genů byly mapovány na 6 x 4 SOM. Po provedení shlukování byly geny zorganizovány do biologicky relevantních shluků, které pomohly stanovit nové hypotézy. Příklady některých těchto hypotéz jsou uvedeny v [3]. Důležitou stránkou SOM je, že umožňují uživateli ovlivnit strukturu a uspořádat podobné vzory jako sousedy na výstupních neuronech. Tato výhoda ulehčuje vizualizaci a interpretaci výsledků. Bohužel stejně jako K-Means algoritmy uživatel musí specifikovat počet shluků. Tyto nedostatky se opět snaží řešit varianty SOM jako je příklad Fuzzy ART (Fuzzy Adapitve Resonance Theory) apod. Další nevýhodou je, že počet shluků, které může síť rozlišit, se přímo vztahuje k počtu neuronů v síti. Jelikož je tento počet předdefinován, formování nově potřebných nových shluků nemusí být povoleno. S tímto problémem se snaží vypořádat GCS (Growing Cell Structure Networks). Jde o variantu Kohenových sítí s proměnlivou topologii, takže není třeba požadovat po uživateli zadávání počtu neuronů. Výhodou je také použití minimálního množství konstant a vyloučení časově závislých proměnných. Možnost přerušit učící proces a následně ho kdykoliv opět nastartovat dělá z této varianty Kohenových sítí účinný nástroj pro tvorbu dynamicky se učících systémů. V [11] lze nalézt srovnávací studii, kdy jsou vzájemně porovnány GCS, Fuzzy c-means, základní Kohenovy sítě, K-means a Fuzzy Kohenovy sítě. Vítězně z této pětice vyšel Fuzzy

Kohenova síť, která bohužel není v práci nijak detailněji popsána. Algoritmy K-means a Kohenova síť vykazovaly nejhorší výsledky.

5.3.3 Grafové algoritmy

U algoritmů využívajících grafů G(V,E) je každý gen reprezentován vrcholem $v \in V$. Například v systému CLICK je pár genů $x,y \in V$ spojen hranou $e(x,y) \in E$ s váhou odpovídající podobnosti na základě vzorů x a y. Problém shlukování je takto převeden na problémy známé z teorie grafů. Algoritmus HCS (Highly Connected Subgraph) rekurzivně dělí graf pomocí řezů na množinu vysoce propojených částí. Každá tato část je považována za shluk. Na algoritmu HCS je postaven i systém CLICK. Síla grafových shlukovacích algoritmů je v dobrém matematickém základu teorie grafů, přesto vyžadují řadu úprav. Například některé geny patřící do stejného shluku jsou spojeny značným množstvím genů „prostředníků“, které v tomto shluku budou též, ač tam nepatří.

5.3.4 Algoritmy založené na modelech

Tyto algoritmy využívají při formování shluků pravděpodobnostních rozložení. Bylo implementováno pár příkladů využívajících Gaussovo rozložení. Ukázalo se však, že tento typ rozložení není vhodný pro data, kde proměnnou je čas. Je to proto, že pak je nahlíženo na časové vzorky jako na neuspořádanou množinu vzorků dat a dochází k ignoraci závislosti na čase. Byli přestaveny jiné modely využívajících kubických spline křivek nebo B-spline křivek k zachycení exprese genů. Pro popis časových závislosti byly využity například i skryté Markovovy modely. Výhodou těchto algoritmů je, že pracují s pravděpodobnostmi vyjadřující náležitost vzorku dat k danému shluku. Jelikož daný gen může participovat ve více shlucích, je vyjádření pomocí pravděpodobností velice vhodné. Nevýhodou je naopak, že se spoléhají na to, že množina dat odpovídá danému rozložení.

5.3.5 Aglomerativní metody

Princip těchto metod je ve vytváření stromové struktury shluků. Nejdříve je nazíráno na každý datový prvek jako na samostatný shluk a v každém kroku se spojí nejbližší dva shluky. Algoritmus pracuje dokud není vytvořen jediný shluk. Zástupcem těchto metod může být systém UPGMA (Unweighted Pair Group Method with Arithmetic Mean), který graficky reprezentuje data rozdělená do shluků. Výhodou těchto metod je právě jednoduchá grafická reprezentace shluků. Díky této grafické reprezentaci je právě i systém UPGMA velice oblíbený u biologů. Bohužel i tyto metody nejsou odolné vůči šumu. Spojování je prováděno na základě informací, které má algoritmus k dispozici jen v daném kroku a nikdy se nedívá „zpět“. A tak špatná rozhodnutí o dělení na začátku nejdou již v průběhu celého procesu napravit. Navíc

hierarchické shlukování obecně vrací pouze stromovou strukturu množiny dat, zvanou dendrogram, a je velmi těžké rozhodnout, kdy se má algoritmus zastavit. S dalším podrobnějším dělením skupiny těchto algoritmů se lze seznámit v [8].

Vylepšení těchto metod ubíhá různými cestami. Vznikla řada variant, jako je Filtrovací shlukování, které se snaží učinit algoritmus více odolný proti šumu. Jiná vylepšení se naopak snaží uživateli napomoci při tvoření shluku vhodnou visulaizací dendrogramu. Touto cestou jde HCE (Hierarchical Clustering Explorer). Nutno podotknout, že jde jen o jakousi zástěrku, jelikož algoritmus stále nedokáže rozhodnout, kdy přestat dále tvořit dendrogram.

O radikální vylepšení se snaží autoři [4] a představují nový algoritmus PAH (probabilistic abstraction hierarchies). Shrnují současné nedostatky této třídy algoritmů, zejména právě řadu menších vylepšení, které nevedou k výraznému posílení využitelnosti těchto algoritmů. Autoři pro každý uzel stromu vytvářejí třídu, které odpovídá specifický pravděpodobnostní model (CPM – class probabilistic model). Algoritmus je odolnější vůči lokálním extrémům a navržený koncept hierarchii sbližuje jednotlivé CPM, pokud jsou si podobné. To vede ke zvýšení robustnosti z hlediska šumu i z hlediska výběru množiny dat pro tvorbu hierarchické struktury. Algoritmus se tedy současně snaží optimalizovat tři věci: přidělování dat do shluků, modely asociované se shluky a hierarchickou strukturu. Unikátnost vidí autoři právě v druhém a třetím bodě, kdy dochází k viditelné redukci citlivosti na zašumělá data a náchylnosti k lokálním maximům. Postup je demonstrován na práci s daty týkající se genové exprese, proteinových sekvencí a HIV proteázy.

5.3.6 Divizivní metody

Jsou velice podobné aglomerativním metodám, ovšem postupují opačně. Na začátku algoritmu existuje jeden shluk, který obsahuje celou množinu dat. Iterativně dělí shluky, dokud každý shluk neobsahuje jeden objekt z datové množiny nebo není splněno nějaké kritérium ukončení. Příkladem může být DAA (deterministic-annealing algorithm). DHC algoritmus využívá prostorové reprezentace. Tedy geny se stejnou mírou exprese vytváří v prostoru oblasti o větších hustotách. Datové prvky (geny) v „centrech“ shluků jsou nositeli vzorů genové exprese, které jsou sdíleny dalšími prvky v této husté oblasti. Algoritmus je závislý na zadání dvou vstupních parametrech. Prahu minimální podobnosti a minimálního počtu prvků ve shluku. Řada systémů využívá této myšlenky jako HCS, SOTA. Systémy se jeví vhodné pro analýzu dat se složitou strukturou shluků, ovšem stejně jako algoritmus DHC trpí nutností zadat vstupní hodnoty.

5.3.7 Metody založené na vzorcích

Pozor, úvodem bych chtěl říci, že zde je třeba odlišovat kdy jde o vzory genové exprese a kdy jde o vzorky, čímž se myslí například vzorky nasbírané v nějakém časovém období.

Algoritmy zmiňované dříve jsou příklady tzv. globálního shlukování. Buněčný proces ovšem probíhá v čase. Vzorky dat tedy nejsou jen množina, ale závisí v jakých časových úsecích byly odměřeny a kolik jich máme. Objevují se algoritmy, které se snaží zachytit i tyto závislosti. Cheng a Church představili koncept dvojshlukování (biclustering). Postup je založen na heuristických metodách, tedy nelze zaručit, že dvojshluk zachytí všechny kompletní množinu odpovídajících dat.

Bližší popis některých zmíněných metod a odkazy na jejich autory lze nalézt v [3, 8]. Zde jsme si uvedli jen ty nejznámější a nejvíce propracované skupiny shlukovacích algoritmů. V [8] lze nalézt další algoritmy jako je Fuzzy shlukování, shlukování založené na simulovaném žíhání apod. Článek se též zabývá reprezentací shluků a diskutuje možnosti jako je reprezentace pomocí těžiště, pomocí uzlů ve stromové struktuře nebo pomocí logických výrazů.

6 Postprocessing

Postprocessingem bychom mohli nazvat tu část, kdy dochází ke zpracování výsledků, které obdržíme aplikací některého z dolovacích algoritmů. Významnost tohoto kroku vzrostla s příchodem nových experimentálních technik jako je například metoda mikročipů. Tato metoda dovoluje biologům měřit expresi až 40 000 genů na jednom čipu. Pokud vezmeme problém měření genové exprese, který jsme nastínili v úvodu kapitoly 5, a budeme se zabývat hledáním vztahů mezi změnami hladin exprese jednotlivých genů, lze z takovéhoto počtu testů vygenerovat až stovky miliónů asociací v závislosti na nastavení vstupních parametrů.

Samozřejmě, že daleko menší množina asociací přináší nějakou zajímavou informaci. Většina pravidel je irelevantních nebo vyjadřují již známou informaci. Cílem je pak oddělit skupinu užitečných pravidel od těch ostatních. Jednou cestou je omezit generování asociací využitím některých dolovacích technik, které umožňují zadávání omezení. Sem patří i omezování vstupních dat podle jistých pravidel. Například Berrar [5] zúžil skupinu genu z 1500 na 20 a i počet testovaných léků. Ovšem omezení celého problému na 20 genů a 20 léků může vést k opomenutí některých velice důležitých vazeb a nenalezení velmi důležitých asociací (zaváděním omezení se okrajově zabývá článek [8]). Proto se autoři [5] snaží snížit počet asociačních pravidel právě ve fázi postprocesingu tím, že nabízí biologům interaktivní nástroje pro správu obrovského množství asociační pravidel. Těmito nástroji je sada operátorů pro sdružování, filtrování a prohlížení a inspekci dat.

Systém umožňuje definovat šablony a tak se přímo zaměřit na pouze na cílenou část vygenerované sady asociací. Šablony mohou být následujícího tvaru:

RulePart **HAS** *Qunatifier* **OF** $C_1,C_2\ldots, C_N$ [**ONLY**]

Kde *RulePart* může být TĚLO, HLAVA, nebo PRAVIDLO a specifikuje část pravidla, na které se omezení aplikuje. Množina $C_1,C_2\ldots, C_N$ reprezentuje seznam genů (případně úroveň exprese), které mají být porovnány. *Qunatifier* je parametr určující, kolik genů z množiny $C_1,C_2\ldots, C_N$ má asociace obsahovat. Může nabývat konkrétních hodnot, může to být rozsah nebo může být zastoupen výrazy ALL, ANY, NONE. Příklady takových to šablon mohou vypadá následovně.

RULE HAS (ANY) OF G1,G5,G7
Všechna pravidla obsahující nejméně jeden z genů G1, G5, G7.
HEAD HAS (ANY) OF [DNA_Repair]
Všechna pravidla, které obsahují opravný gen v hlavičce pravidla.

Příklady šablon mohou být použity samostatně nebo je možné je kombinovat. Autoři nabízí též řadu předdefinovaných operátorů. Například biology může zajímat, jestli skupina genů ovlivňuje jinou. Příkladem by pak mohla být šablona :

POSSIBLE_INFLUENCE(GeneSet1, GeneSet2) =
BODY HAS (ANY) OF GeneSet1 **AND**
HEAD HAS (ANY) OF GeneSet2 **OR**
BODY HAS (ANY) OF GeneSet2 **AND**
HEAD HAS (ANY) OF GeneSet1

Autoři vybavily systém také možností sdružovat pravidla do skupin a možností procházet tyto skupiny. Využívají přitom svůj dříve implementovaný systém pro měření podobnosti. Ten upravili tak, že se vytváří stromová struktura. Dle autorů by měla být hlavní výhodou jejich postupu možnost naprosté kontroly granularity. Tedy možnost kontrolovat strukturu a velikost shluků. Druhou výhodou je upravené sdružování pravidel do skupin. Kdy skupiny se vytváří pouze na základě struktury pravidel a nejsou ovlivněny celou množinou vygenerovaných asociací. Tímto nedostatkem trpí řada shlukovacích algoritmů, kdy zařazení vygenerovaných asociací (v tomto případě, jinak platí obecně pro jakýkoliv typ znalosti) do shluků závisí také na

ostatních vygenerovaných asociacích. Třetí výhodu vidí autoři na rozdíl od jiných metod, kde je nesnadná interpretace shluků z důvodu rozdílné struktury asociací uvnitř shluku, v jednoduchosti popisu jednotlivých tříd pomocí agregačních pravidel, které mohou přímo reprezentovat shluk. Poslední výhodou je, že daný systém je schopen pracovat s velkým počtem atributů, jak numerických tak kategorických. To je dosaženo použitím vyhledávacích tabulek pro ukládání hierarchické struktury a struktur podobných hashovacím tabulkám pro ukládání agregačních pravidel.
Bohužel u posledních dvou výhod, tedy reprezentace shluků pomocí agregačních pravidel a dosažení dobré škálovatelnosti je v [5] popsáno velmi povrchně.

7 Závěr

Celá tato práce si kladla za cíl zmapovat využití procesu získávání znalostí v genetice. Řada dolovacích nástrojů již našla uplatnění v mnoha biologických experimentech a populárnost těchto technik stále roste zejména díky novým experimentálním postupům, jako je využití mikročipů. Tyto techniky dovolují testovat naráz obrovské množství dat a dolovací algoritmy jsou vhodným nástrojem pro jejich zpracování.

Proces získávání znalostí není jednoduchý a jeho efektivní implementace vyžaduje pozornost ve všech fázích procesu. Jak vhodnou přípravu dat, tak vhodné zpracování výsledků a následnou interpretaci.

Práce [11] prezentuje srovnávací testy pěti shlukovacích algoritmů. Dále popisuje algoritmus C5.0, založený na rozhodovacím stromě, který pomáhá v interpretaci a validaci výsledků. Nehledě na výsledky shlukovacích analýz bylo demonstrováno, jak důležité je fáze postprocesingu. Velice úspěšný algoritmus C5.0 dokázal, jak ukazují experimenty, detekovat úspěšně rozdílné typy leukémií jen na základě 12 genů. Zatímco dřívější testy vyžadovaly testovat 50 genů.

7.1 Nedostatky současných algoritmů

V jednotlivých kapitolách této práce popisuji etapy celého procesu dolování. Celkově lze shrnout problematické oblasti procesu získávání znalostí do těchto bodů a to nejen v kontextu genomických databází:

- **Manipulace s různými typy dat**
  Očekává se, že systém pro získávání znalostí bude schopen efektivně dolovat nad různými typy dat. Navíc se již považuje za běžné, že databáze obsahuje komplexní typ dat, jako jsou strukturovaná data a komplexní datové objekty,

hypertextová a multimediální data, prostorová a temporální data atd. Systém, který má obstát, by měl být schopen pracovat nad takovýmto typem dat. Nicméně rozmanitost datových typů a rozdílné cíle činí nereálným existenci dolovacího systému, který by byl schopen pracovat nad celou škálou typů dat. Na složitosti též přidává fakt, že v multimediálních aplikacích se často kombinuje několik výrazových prostředků.

- **Účinnost a měřitelnost dolovacích algoritmů**
  Abychom mohli efektivně extrahovat informace z obrovského množství dat v databázích, algoritmy musí být efektivní a měřitelné. To znamená, že čas pro běh algoritmu musí být předpovídatelný a přijatelný pro databáze jakékoliv velikost. Algoritmy s exponenciální nebo dokonce polynomiální složitostí nemají šanci na praktickou realizaci.
- **Schopnost vyjádření výsledků dolování**
  Získaná znalost by měla vystihovat obsah databáze a měla by být užitečná pro určité aplikace. Výjimky by měly být též obslouženy. Tyto požadavky motivují ke studiu jak kvalitativně ohodnotit získanou znalost, vyjádřit míru její zajímavostí a spolehlivosti.
- **Vyjádření různých druhů výsledků dolování**
  Z obrovského množství dav lze získat různé druhy znalostí. Lze na ně také pohlížet z různých pohledů a prezentovat je v rozličné podobě. To vyžaduje schopnost vyjádřit jak dotaz tak získanou znalost jazykem vyšší úrovně nebo pomocí grafického uživatelského rozhraní. Tím bude zároveň umožněno klást dotazy a přímo porozumět nalezené znalosti i laikům.
- **Interaktivní získávání znalostí na více abstrakčních úrovních**
  Jelikož lze předpokládat, že můžeme najít další zajímavé znalosti, než pouze ty, které jsme očekávali, měla by být vypracována možnost dotazování na vysoké úrovni. Interaktivní dolování by mělo být též podpořeno, takže uživatel může upravit svoji žádost k systému a flexibilně prohlížet dat a výsledky.
- **Získávání znalostí z různých zdrojů**
  Široce přístupné lokální i globální sítě včetně internetu spojují mnoho datových zdrojů a formují tak obrovskou heterogenní distribuovanou síť. Získávání znalostí z takovéto sítě s různými datovými sémantikami přináší novou výzvu a zároveň problém. Dolování dat zde muže pomoci odkrýt zákonitosti v takovýchto databázích, kde jednoduché dotazovací systémy by selhaly. Navíc značná

velikost databází, široká distribuce a výpočetní složitost některých dolovacích technik motivuje k vývoji paralelních a distribuovaných dolovacích algoritmů.

- **Bezpečnost**

  Tam, kde se lze na data dívat z mnoha úhlů a na různých abstrakčních úrovních, vyvstává problém s ochranou dat a obranou proti zásahům do soukromí. Je důležité pro další rozvoj vědět, kdy může dolování vést k porušení bezpečnostní politiky a jaká opatření tomu mohou zabránit.

- **Směšná podpora**

  Zatím bylo učiněno velmi málo ve vývoji specifických algoritmů pro multimediální data. Nejvšestrannější algoritmy byly vyvinuty pro numerická data, většinou obyčejné soubory nebo relační databáze, a jsou velmi těžko adaptovatelné na obrazové databáze. Nástroje pro dolování obrazových dat by měli být flexibilní a rychle přizpůsobitelné na jinou doménu znalostí a to i v případech vyšší specializace. Měla by být zahrnuta možnost ad-hoc dolování.

- **Nedostatečné zkušenosti**

  V mnoha věcech spoléhá obor dolování dat na příbuzné vědní disciplíny, kde se ovšem také setkává s nedostatkem podpory. Jsou to obory strojové učení,umělá inteligence, databáze, statistika, velmi náročné výpočty, vizualizace. Mezioborové zkušenosti jsou velmi důležité. Je velmi málo odborníků s praxí hned z několika takových oborů. Velmi důležité je také informovanost vlastníků dat, že nad svými daty vůbec mohou provádět dolování a může jim pomoci vyřešit jejich problémy.

Všimněme si, že některé body si kolidují. Například cíl protekce a ochrany dat může být ve sporu s potřebou interaktivního dolování víceúrovňových znalostí z různých úhlů pohledu.

7.2 Zhodnocení

Je třeba podotknout, že narozdíl od dolování dat v business sféře, je oblast získávání znalostí v kontextu genomických, obrazových či audio databází stále mladým odvětvím. Zájem o zpracování obrovského množství dat žene kupředu vývoj v této oblasti a proces získávání znalostí se stává nutností.

Pro dosažení maximální efektivity se předpokládá společné řešení problémů s obory počítačového vidění, strojového učení, umělé inteligence a počítačové grafiky. Kvalitní předzpracování obrazu, extrakce rysů stejně jako následná prezentace výsledků jsou základem úspěchu při dolování obrazových dat. Problémy podobnostního vyhledávání, ukládání dat, shlukování, klasifikace či detekce objektů jsou aktuálními tématy.

Oblast genetiky spolu s oblasti predikce počasí, analýzou nákupního košíku nebo zpracovaní snímků vesmíru byly doposud jediné oblasti, kdy byl proces dolování dotáhnut do podoby profesionálních komerčních systémů. Praxe v těchto oborech však ukazuje na daleko širší použití téměř v jakémkoli oboru.

8 Použitá literatura

[1] Shui W. M., Wong R. K., Graham C. G., Lee L. K., Church W. B..: A new approach to protein and function analysis using semi-structured databases, University of New South Wales,University of Sydney (8 stran)

[2] Page D., Craven M.: Biological Applications of Multi-Relational Data Mining, University of Wisconsin (8 stran)

[3] Jian D., Pei J., Zhang A.: Towards Interactive Exploration of Gene Expression. State University of New York at Buffalo (11 stran)

[4] Segal E., Koller D.: Probabilistic Hierarchical Clustering for Biological Data, Stanford University. (8 stran)

[5] Tuzhilin A.,Adomavicius G.: Handling Very Large Numbers of Association Rules in the Analyis of Microarray Data, New York University (8 stran)

[6] Yung B. Y. M., Ng V. T. Y.: Classification of Heterogeneous Gene Expression Data, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, 2002 (9 stran)

[7] Fayyad U., Haussler D., Stolorz P.: Mining Scientific Data, Communications of ACM, Listopad 1996/Vol. 39, No. 11 (7 stran)

[8] Jain A. K., Murty M. N., Flynn P. J.: Data Clustering: Review, Michigan State University (52 stran)

[9] Piatetsky-Shapiro G., Tamayo P.: Microarray data mining: Facing the Challenges (4 strany)

[10] Džeroski S.: Multi-Relational Data Mining, Jožef Stefan Institute, Ljubljana, Slovenia (14 stran)

[11] Granzow M., Berrar D., Dubitzky W., Schuster A., Azuaje F.J., Eils R.: Tumor Classification by Gene Expression Profiling: Comparison and Validation of Five Clustering Methods, German Cancer Research Center, Heidelberg, Germany (6 stran)

[12] Hunt E., Atkinson P. M., Irving R.W.: Database indexing for large DNA and protein sequence collections, Dept. Of Computer Science, University of Glasgow, Glasgow (14 stran).

[13] Wang H., Ong T.,Ooi B. Ch., Tan K.: BLAST++: A Tool for BLASTing Queries in Batches, Dept. of Computer Science, National University of Singapore,Singapore (8 stran)