دانشکده مهندسی

پایان نامه کارشناسی ارشد در رشته مهندسی کامپیوتر (نرم‌افزار)

بررسی کاربردهای کشف قوانین وابستگی در بیان ژن

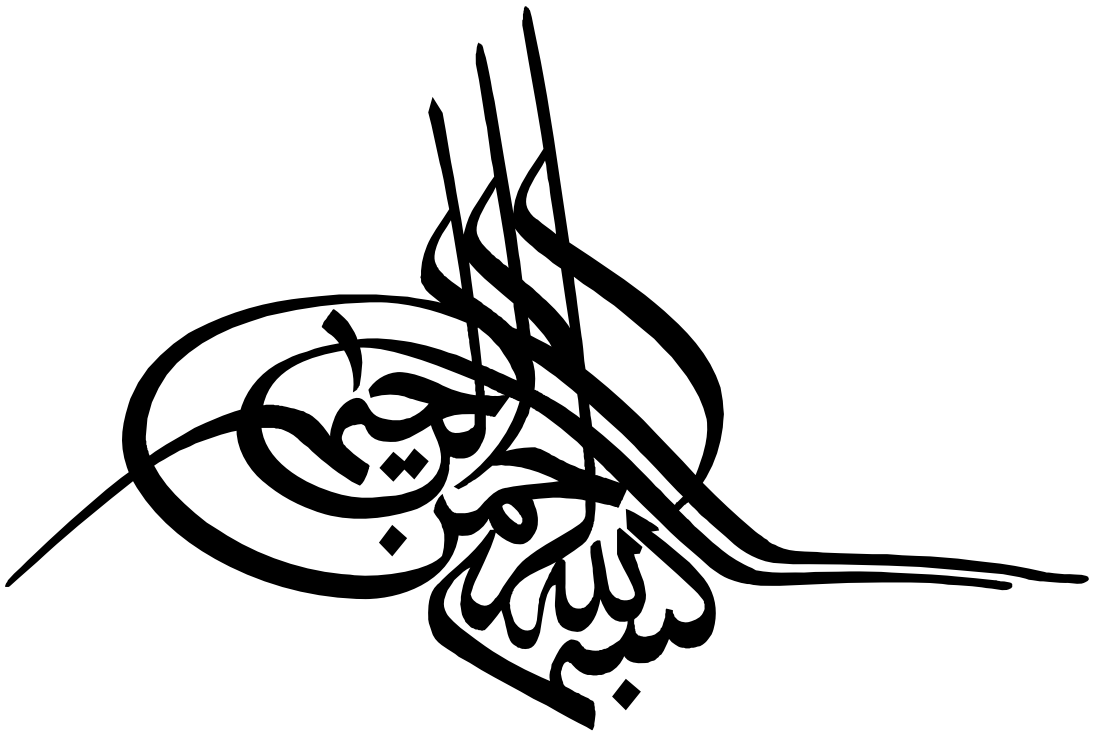
توسط:

مهدی اسدپور

استاد راهنما:

دکتر محمدهادی صدرالدینی

شهریور ۱۳۸۷


بسم الله الرحمن الرحيم

به نام خدا

بررسی کشف قوانین وابستگی در بیان ژن

به وسیله‌ی:

مهدی اسدپور

پایان نامه
ارائه شده به معاونت تحصیلات تکمیلی دانشگاه به عنوان بخشی از
فعالیت‌های تحصیلی لازم برای اخذ درجه کارشناسی ارشد

در رشته‌ی:

مهندسی کامپیوتر - گرایش نرم افزار

از دانشگاه شیراز

شیراز

جمهوری اسلامی ایران

ارزیابی شده توسط کمیته پایان نامه با درجه: عالی

دکتر محمدهادی صدرالدینی، استادیار بخش مهندسی کامپیوتر (رئیس کمیته).....

دکتر علی نیازی، استادیار بخش بیوتکنولوژی دانشکده کشاورزی .....

دکتر منصور ذوالقدری جهرمی، دانشیار بخش مهندسی کامپیوتر .....

شهریور ۱۳۸۷

تقدیم به

خانواده‌ام

و

همه پویندگان این راه

سپاسگزاری

در پایان این رساله بر خود لازم می‌دانم که از زحمات و راهنمایی‌های اساتید بزرگوار دکتر صدرالدینی، دکتر نیازی، و دکتر ذوالقدری جهرمی صمیمانه تشکر نمایم. همچنین از نماینده تحصیلات تکمیلی دانشکده مهندسی، دکتر ابراهیم گشتاسبی راد سپاسگزارم.

به علاوه از کمک‌ها و راهنمایی‌های ارزنده دکتر علیرضا موسوی جراحی، دانشیار دانشکده پزشکی دانشگاه شهید بهشتی نیز قدردانی می‌نمایم.

در انتها نیز بر خود لازم می‌دانم از همراهی دوستانم آقایان انصاری، باوندپوری، شیخ علیشاهی و معدلی، و خانم‌ها آشور ماهانی، پرنیان، شکرپور و عمرانیان تشکر نمایم.

چکیده

بررسی کاربردهای کشف قوانین وابستگی در بیان ژن

بوسیله‌ی:

مهدی اسدپور

بررسی و مطالعات صورت گرفته در زمینه کاربردهای کشف قوانین وابستگی در بیان ژن نشان می‌دهند که این عرصه نیازمند یک الگوریتم موازی کارا به منظور سرعت بخشیدن به فرآیند کشف چنین قوانینی از پایگاه داده‌های بیان ژن می‌باشد. روبرو شدن با حجم عظیم، در حال رشد و توزیع شده داده‌های زیستی، بیوانفورماتیک را نیازمند محاسبه و حافظه بیشتر و بیشتر کرده است. پردازش موازی برای این منظور طراحی شده است به طوریکه هزینه محاسباتی را بین رایانه‌های شخصی همه کاره توزیع می‌کند. کشف قوانین وابستگی یکی از روشهای معروف داده کاوی است که نیازمند زمان محاسباتی فراوان برای وقتی است که بر روی داده‌های بیان ژن بکار گرفته شود. این حجم عظیم محاسباتی به دلیل تعداد ترکیبات بسیار زیادی است که باید بین این داده‌های با بعد زیاد چک شود.

هدف از این پایان نامه معرفی بسته نرم افزاری RuleGene است که شامل یک الگوریتم سریال (SeqRG) و یک الگوریتم موازی (ParRG) برای کشف قوانین وابستگی بر روی پایگاه داده‌های بیان ژن به شیوه‌ای کارا می‌باشد. الگوریتم سریال SeqRG یک الگوریتم طراحی شده مخصوص است که قابلیت‌هایی مانند بعد زیاد داده‌ها و موازی شدن آسان در طراحی آن مورد توجه ویژه قرار گرفته است. به علاوه، SeqRG از یک الگوریتم فشرده سازی خاص بر روی داده‌های تولید شده در مراحل مختلف الگوریتم استفاده می‌کند. الگوریتم موازی ParRG نیز در این پایان نامه معرفی می‌شود که این الگوریتم در ابتدا پردازنده‌های موازی را به صورت یک درخت دودویی پیکربندی کرده و سپس پایگاه داده‌های بیان ژن را بین پردازنده‌های برگ در این درخت به صورت عمودی تقسیم می‌کند. این پردازنده‌های موازی نیز با استفاده از الگوریتم سریال SeqRG اقدام به جمع آوری قوانین وابستگی داده‌ها بر روی بخشی از داده که در اختیار آنهاست، می‌نمایند. علاوه بر معرفی این الگوریتم‌ها، آنالیز آنها بر اساس درستی و هزینه‌ها، و نتیجه آنها بر روی یک پایگاه داده‌های واقعی بیان ژن آورده می‌شود.

فهرست مطالب

عنوان صفحه

فهرست جداول

عنوان صفحه

فهرست شکل‌ها

عنوان صفحه

فصل اول

مقدمه

۱- مقدمه

با پیشرفت‌های جدید در زمینه بیولوژی و بیوتکنولوژی، حجم عظیمی از داده‌ها تولید و انباشته شده است که خود نیازمند ابزار هوشمند و سریع برای آنالیز و فهم فرایندهای موجود بین آنها می‌باشد.

بیان ژن به فرآیندی گفته می‌شود که در طی آن یک ترتیب DNA یک ژن به ساختارها و توابع یک سلول تبدیل می‌شود. این فرآیند در چندین مرحله صورت می‌گیرد (همانطور که در شکل ۱ آورده شده است):

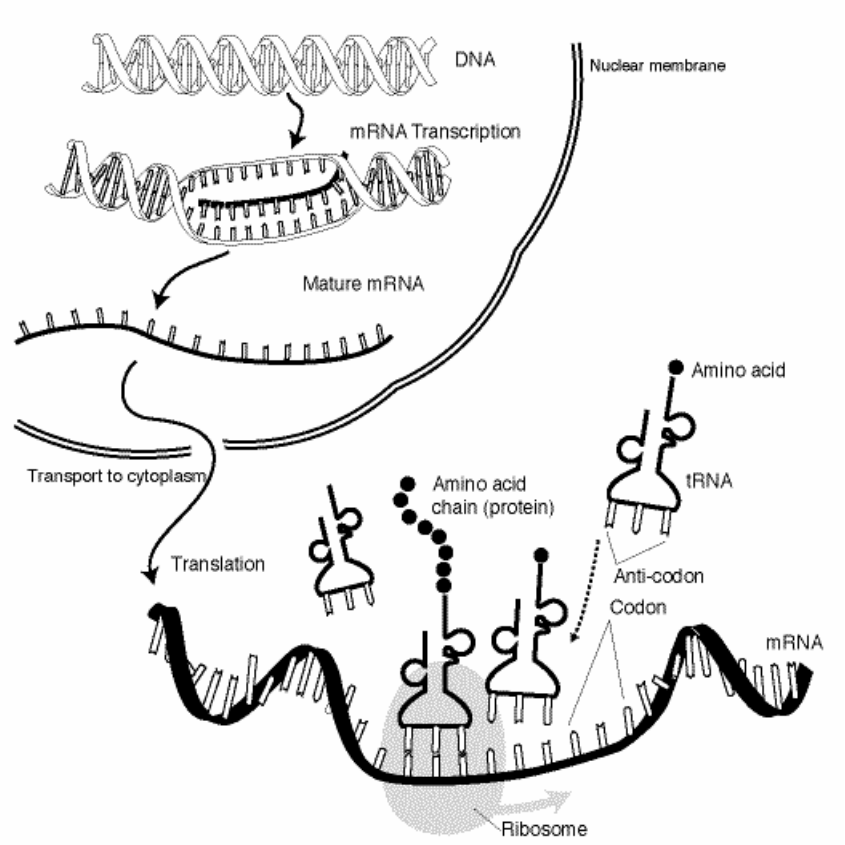
۱. در مرحله اول که به آن رونویسی[1] گفته می‌شود یک DNA به RNA پیغام آور (mRNA) تبدیل می‌شود.

۲. در مرحله دوم که به آن ترجمه[2] گفته می‌شود، نتیجه کار به یک محصول ژنی مانند پروتئین تبدیل می‌شود.

لازم به ذکر است که تعداد غیر طبیعی این محصول ژنی باعث بروز بیماری‌های مختلف در فرد مورد نظر خواهد شد.

خروجی فرآیند بیان ژن مشخص می‌کند که یک ژن در این شرایط بیان شده است یا نه. در نتیجه یک پایگاه داده از ژنهایی که بیان شده‌اند تشکیل می‌شود که همانند جدول ۱، ستون‌های این پایگاه داده ژن‌ها را در بر می‌گیرد و سطرهای آن نمونه‌های تست شده می‌باشند. در این جدول مقدار یک نشان دهنده «بیان شدن ژن» و مقدار صفر نشان دهنده «عدم بیان ژن» می‌باشد. لازم است توجه شود که در اینجا بیشتر علاقه مند به بیان شدن ژن بوده‌ایم نه عدم بیان آن. بنابراین اگر بخواهیم عدم بیان ژن را بررسی کنیم به راحتی می‌توانیم مقدار یک و صفر در این جدول را جا به جا کنیم.

[1] Transcription
[2] Translation



شکل ۱ فرآیند بیان شدن ژن در سلول (برگرفته از Genome Homepage, 2007)

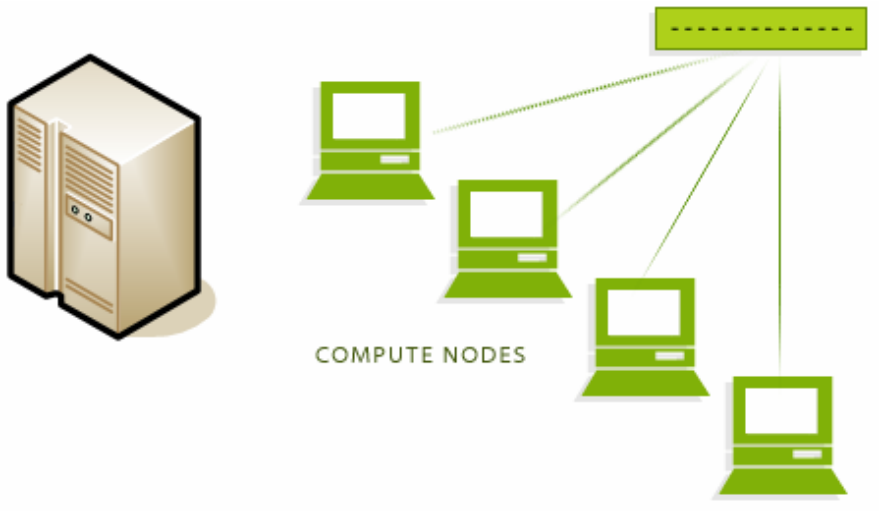
جدول ۱ نمونه‌ای از یک پایگاه داده‌های بیان ژن تستی

	Gene 1	Gene 2	Gene 3	Gene 4	Gene 5
P1	1	1	0	0	0
P2	1	0	0	0	0
P3	0	0	1	0	0
P4	0	0	1	1	0

پایگاه داده‌های بیان ژن شامل اطلاعات اندازه گیری سطح بیان یک ژن خاص در یک شرایط خاص می‌باشند که معمولا توسط روش Microarray محاسبه می‌شوند. از آنالیز این پایگاه داده‌ها می‌توان اطلاعات بیولوژیکی مفیدی بدست آورد مانند اینکه چگونه بیان یک ژن خاص ممکن است در بیان ژن‌های دیگر تاثیر بگذارد و یا اینکه چه ژن‌هایی در اثر یک شرایط

سلولی مشخص بیان می‌شوند. اغلب پایگاه داده‌های بیان ژن در اختیار عموم قرار دارند که از آنجمله می‌توان به ArrayExpress[1] و ExpressDB[2] اشاره نمود.

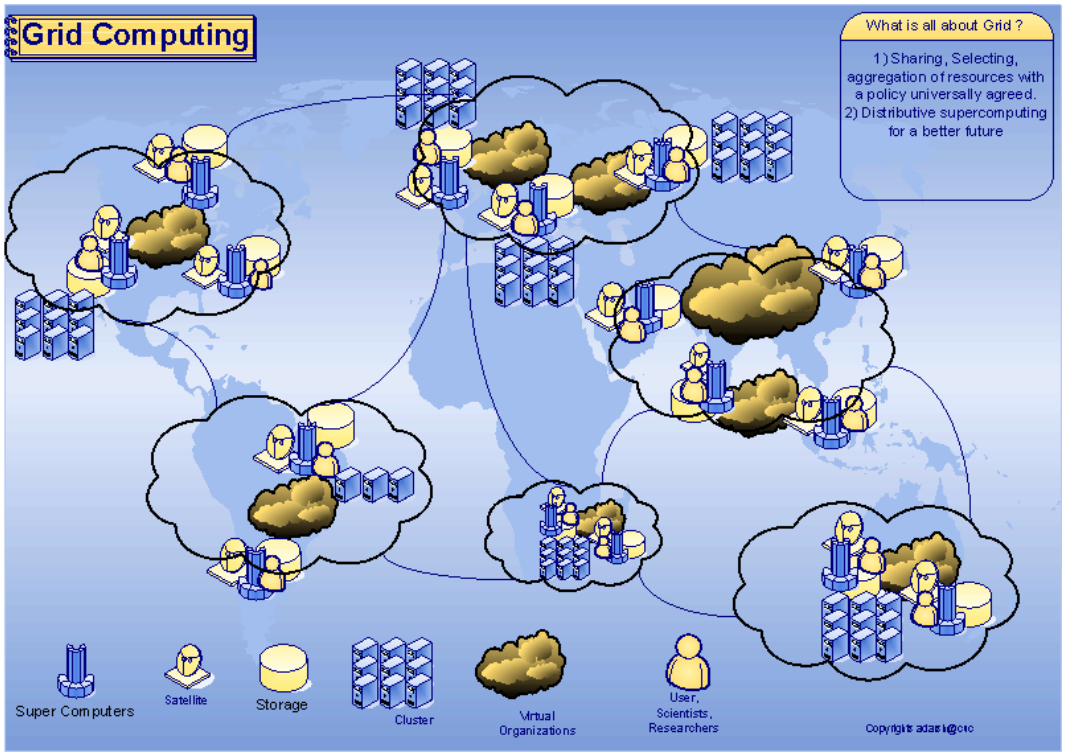
همانطور که گفته شد، روشهای جدید مانند Microarray حجم بسیار زیادی داده بیان ژن[3] (GE) به صورت نمایی تولید می‌کنند. بیوانفورماتیک به منظور حل مسائل عملی ناشی از مدیریت و آنالیز این داده با ساخت الگوریتم‌ها، روش‌ها و فرضیه‌های جدید بنا شده است. بدون شک، هر چه داده افزایش می‌یابد، هزینه محاسبه نیز به همین منوال افزایش می‌یابد. بکارگیری ابررایانه‌ها تنها راه حل ممکن برای این حجم داده‌ها نیست.



شکل ۲ استفاده از رایانه‌های موازی به جای ابررایانه

پردازش موازی و قالب‌های دیگر آن مانند مجاسبه توزیع شده[4] و محاسبه مشبک[5] به منظور چیره شدن بر این مشکلات با پخش پیچیدگی مسئله بین رایانه‌های شخصی عام-منظوره به شیوه «تفرقه بیانداز و حکومت کن» ابداع و معرفی شده‌اند (شکل‌های ۲ و ۳). بنابراین، برای استفاده از این رایانه‌ها در بیوانفورماتیک، ما نیازمند طراحی‌های جدید و موثر هستیم (Trelles, 2001).

[1] http://www.ebi.ac.uk/arrayexpress
[2] http://twod.med.harvard.edu/ExpressDB
[3] Gene Expression
[4] Distributed Computing
[5] Grid Computing



شکل ۳ محاسبه مشبک (برگرفته از CSA Homepage, 2007)

داده کاوی (Data Mining) یکی از روشهای سودمند و امید بخش برای آنالیز داده‌های تولید شده از این طریق است. از بین تکنیک‌های داده کاوی، تکنیک کشف قوانین وابستگی[1] (ARM) به تازگی برای استخراج قوانین موجود بین داده‌های بیولوژیکی مورد توجه بسیار قرار گرفته است. این تکنیک در ابتدا برای آنالیز داده‌های خرید از فروشگاه پیشنهاد شد و به استخراج قوانین موجود بین تراکنش‌ها به قالب $\{X \Rightarrow Y\}$ (X و Y دو مجموعه مجزا از هم هستند) می‌پردازد؛ بدین معنی که در اغلب داده‌هایی که موجودیت X حضور دارد، موجودیت Y هم حضور دارد. به عنوان مثال قانون {شیر ⇒ نان و کره} درداده‌های فروشگاهی به این معنی است که اغلب مشتریانی که «نان و کره» خریده‌اند، «شیر» نیز خریده‌اند. ملاک انتخاب یک قانون بیشتر بودن Support و Confidence آن قانون از یک مقدار کمینه (ورودی از طرف کاربر) است. Support یک قانون تعداد تراکنش‌هایی است که تمام موجودیت‌های یک قانون (در اینجا $X \cup Y$) در آن باشند. Confidence یک قانون به صورت $Support(X \cup Y)/Support(X)$ تعریف می‌شود که نشان دهنده نسبت حضور همزمان Y و X در تراکنشهایی است که X در آن حضور دارد.

[1] Association Rules Mining

روش کشف قوانین وابستگی به جستجو برای یافتن مجموعه‌ای از قوانین می‌پردازد که با فراوانی بالا در داده‌های بیان ژن دیده شده‌اند. این قوانین کشف شده که از لحاظ بیولوژیکی معتبر هستند، به صورت یکی از قالب‌های زیر می‌باشند (نمونه‌ای از این قوانین در شکل ۴ آمده است):

- وابستگی‌های بین ژن‌های مختلف: مانند قانون {Gene C↑} ⇒ {Gene A↑, Gene B↓} که نشان می‌دهد در بیشتر مواقعی که ژن C بیان شده است ژن A نیز بیان شده ولی ژن B بیان نشده است.
- بین بیان ژن و تاثیرات محیطی آن: مانند قانون {Cancer} ⇒ {Gene A↑, Gene B↓} که نشان می‌دهد در بیشتر مواقعی که سلول سرطانی بوده است، ژن A بیان شده ولی ژن B بیان نشده است.

	A	B	C
1	Association rule	Support	Confidence
2	{YHM1}⇒{ARG1,ARG4,ARO3,CTF13,HIS5,LYS1,RIB5,SNO1,SNZ1,YHR029C,YOL118C}	11%	81%
3	{ARO3}⇒{ARG1,ARG4,CTF13,HIS5,LYS1,RIB5,SNO1,SNZ1,YHM1,YHR029C,YOL118C}	11%	89%
4	{ORT1}⇒{ADH5,ARG4,BNA1,CPA2,CTF13,SNO1,SNZ1,YBR047W,YGL117W}	10%	83%
5	{NIT1}⇒{ATR1,BNA1,CPA2,CTF13,LYS1,RIB5,SNO1,SNZ1,SRY1,YBR047W,YHR029C,YOL118C,YPL033C}	11%	80%

شکل۴ نمایی از قوانین کشف شده

نکته مهمی که در مورد تفاوت پایگاه داده‌های بیان ژن و پایگاه داده‌های معمولی باید بدان اشاره کرد اینکه تعداد ستون این داده‌ها بسیار زیاد بوده و معمولا بین ده هزار تا صد هزار هستند. (از آنجا که بین پنجاه تا صد هزار ژن مختلف در بدن موجودات زنده وجود دارد.) در حالیکه تعداد ستون در پایگاه داده‌های معمولی (مثلا تراکنش‌های یک فروشگاه) بسیار کم و بین ده تا صد می‌باشد. (از آنجا که معمولا در سبد خرید یک مشتری فروشگاه به طور متوسط بیش از صد کالا یافت نمی‌شود.) به عنوان نمونه، پایگاه داده ExpressDB دانشگاه هاروارد دارای ۶۴۰۰ ستون است و فقط ۲۱۳ سطر دارد. پس، به منظور، پیدا کردن قوانین مخفی از این پایگاه داده‌ها، روش ARM مجبور است ترکیب‌های بسیار زیادی بین این تعداد زیاد ستون/ژن را مورد بررسی قرار دهد. بنابراین، بکارگیری پردازش موازی در اینجا یک روش اجتناب ناپذیر است (Agrawal and Shafer, 1996). تاکنون، تا آنجا که می‌دانیم، الگوریتم موازی ARM که خاص پایگاه داده‌های بیان ژن باشد پیشنهاد نشده است.

هدف نهایی این پایان نامه پیشنهاد یک روش موازی کارا برای کشف قوانین وابستگی روی پایگاه داده‌های بیان ژن است. این الگوریتم هزینه محاسباتی درگیر را بین پردازنده‌های موازی تقسیم کرده و بعد زیاد داده‌های بیان ژن را به شیوه‌ای کارا مدیریت می‌کند. به منظور پیاده سازی چنین الگوریتمی، ما در ابتدا یک الگوریتم (SeqRG) سریع سریال که به آسانی قابل موازی شدن است را طراحی می‌کنیم و در آن از الگوریتم جدید فشرده سازی خودمان بهره می‌گیریم. الگوریتم SeqRG از این الگوریتم فشرده سازی بر روی داده‌های خروجی مراحل مختلف اعمال کرده و آنرا در یک داده‌ساختار که به راحتی قابل ذخیره و بازیابی است، نگه می‌دارد. با بهره گیری از این ساختار فشرده شده، نه تنها اندازه داده‌های ارسالی بین پردازنده‌های موازی کم می‌شود، بلکه نیازی به الگوریتم‌های پیچیده ذخیره و بازیابی هم نیست. پس از معرفی الگوریتم سریال، به معرفی الگوریتم موازی (ParRG) می‌پردازیم که این الگوریتم پردازنده‌های درگیر را در قالب یک درخت دودویی پیکربندی کرده و پایگاه داده‌های بیان ژن را به صورت عمودی بین گره‌های برگ این درخت تقسیم می‌کند. ما این بسته پیاده سازی شده را RuleGene می‌نامیم. در ادامه، نکات اصلی پیاده سازی چنین بسته‌ای در کنار مشابه-کد[1] آنها توضیح داده می‌شود. در نهایت نیز، الگوریتم‌های آورده شده را آنالیز کرده، کامل بودن آنها را اثبات کرده، و نتایج تجربی حاصل از اجرای این برنامه بر روی یک پایگاه داده واقعی آورده می‌شود[2]. قسمتی از این پایان نامه در قالب یک مقاله چاپ شده است (Asadpour et al., 2007).

ادامه این پایان نامه به صورت زیر است. در بخش بعد، کارهای مربوط انجام شده هم در بخش سریال و هم موازی به صورت خلاصه مرور می‌شوند. بخش ۳ الگوریتم سریال SeqRG در کنار الگوریتم فشرده سازی را معرفی می‌کند. سپس، الگوریتم موازی ParRG را در بخش ۴ با مثال شرح داده می‌شود. بخش ۵ شامل آنالیز الگوریتم موازی و speed-up آن است. در ادامه آن، بخش ۶ نتایج تجربی حاصل از اجرای الگوریتم موازی بر روی یک پایگاه داده‌های واقعی بیان ژن را جمع بندی می‌کند. در نهایت نیز نتیجه‌گیری این پایان نامه آورده شده است.

[1] Pseudo-code

[2] داده‌های مورد نیاز، مستندات و یک برنامه قابل اجرا از این بسته در صفحه خانگی RuleGene موجود می‌باشد.

فصل دوم

کارهای مربوط

۲- کارهای مربوط

به طور کلی، کارهایی تحقیقاتی انجام شده در این زمینه را می‌توان به سه دسته تقسیم کرد:

۱. دسته اول: این دسته شامل کارهایی است که از الگوریتم و برنامه‌های موجود که برای پایگاه داده‌های معمولی طراحی شده‌اند (مانند Apriori و FP-growth) برای پایگاه داده‌های بیان ژن استفاده می‌کنند (Carmona-Saez et al., 2006; Creighton and Hanash, 2002; Becquet et al., 2002). در حقیقت هدف اصلی این کارها نشان دادن سودمندی انجام چنین کاری بوده است و اینکه قوانین کشف شده بسیار مفید می‌باشند. اما به کارایی الگوریتم، بهینه بودن حافظه کامپیوتر مصرفی، و درصد مصرفی از توان پردازنده و زمان اجرا، چندان پرداخته نشده است.

۲. دسته دوم: این دسته شامل کارهایی است که با توجه به مشخصه داده‌های بیان ژن (که تعداد ستون بسیار زیادی دارند) الگوریتم‌های کارا را تهیه و معرفی نموده‌اند (Cong et al., 2004; Jiang and Gruenwald, 2005). در این نوع کارها، بیشتر به بررسی حافظه و زمان اجرای الگوریتم و مقایسه آن با موارد مرتبط پرداخته شده است.

۳. دسته سوم: این دسته شامل کارهایی است که بر روی کم کردن تعداد قوانین تولید شده با اعمال محدودیت و صافی تاکید می‌کنند (Tuzhilin and Adomavicius, 2002). بدلیل تعداد زیاد ستون‌ها در این نوع داده‌ها تعداد قوانین تولید شده در روش ARM بسیار زیاد هستند و حتی برخی از آنها از لحاظ بیولوژیکی درست نمی‌باشند، بنابراین باید با هرس کردن قوانین خروجی به قوانین مناسب‌تری رسید. این دسته به صورت ترکیبی با دو دسته قبل اجرا می‌شود.

جدول شماره ۲ کارهای انجام شده در زمینه اعمال ARM (همراه با اسم الگوریتم مربوط) بر روی داده‌های بیان ژن (همراه با اسم پایگاه داده مربوط) را لیست می‌کند. به علاوه موجود بودن یا نبودن برنامه رایانه‌ای این کارها همراه با توضیحات خاص در آن اشاره شده است.

جدول ۲ بررسی کارهای انجام شده در این زمینه

مرجع	دسته	الگوریتم	داده	در دسترس بودن برنامه	توضیحات خاص
Becque et al., 2002	1	Apriori	SAGE	×	کشف قوانین جدید
Creighton and Hanash, 2002	1	Apriori	Yeast	√	یکی از بهترین کارها در این دسته است.
Carmona-Saez et al., 2006	1	Apriori	چندین پایگاه داده مانند Diauxic shift	√	ترکیب بیان ژن و تفسیر (annotation) ژن
Cong et al., 2004	2	FARMER	۵ پایگاه داده سرطانی مانند ALL-AML	×	به کشف قوانین گروهی می‌پردازد.
Jiang and Gruenwald, 2005	2	BSC-Tree	ExpressDB و Stanford-SMD	×	از یک الگوریتم فشرده سازی نیز استفاده می‌کند.
Tuzhilin and Adomavicius, 2002	3	Apriori	-	×	اعمال محدودیت بعد از تولید شدن قوانین

در ادامه این بخش، به مطالعه و بررسی خلاصه کارهای مربوط در زمینه الگوریتم‌های سریال و موازی کشف قوانین وابستگی خاص داده‌های بیان ژن می‌پردازیم. تاکید اصلی این قسمت بر روی موازی شدن آسان (در مورد الگوریتم‌های سریال) و مناسب بودن برای پایگاه داده‌های با بعد زیاد (در مورد الگوریتم‌های موازی) می‌باشد.

۲-۱- موازی شدن آسان در الگوریتم‌های سریال **ARM**

کار تحقیقاتی متعددی به منظور استفاده از روش‌های سریال ARM برای کشف قوانین وابستگی از پایگاه داده‌های بیان ژن، موجود می‌باشند. اغلب این کارها (Carmona-Saez et al., 2006; Creighton and Hanash, 2002; Tuzhilin and Adomavicius, 2002; Becquet et al., 2002) از الگوریتم‌های موجود (با تغییرات جزئی) مانند الگوریتم معروف Apriori (Agrawal and Srikant, 1994) و بیشتر با هدف معرفی سودمندی این روش ARM برای زیست شناس‌هاست. برخی دیگر (Georgii et al., 2005; Cong et al., 2004; Gyenesei et al., 2007; Jiang and Gruenwald, 2005; Kotala et al., 2001; Georgii et al., 2005) الگوریتم‌های خاص-منظوره‌ای را با توجه به بعد زیاد و یا خواص دیگر این داده‌ها (مانند quantitativity) پیشنهاد می‌کنند. در واقع، فرآیند موازی سازی الگوریتم‌های سریال کنونی که خوب کار می‌کنند آنقدر هم سر راست و آسان نیست؛ زیرا این الگوریتم‌های سریال معمولا داده ساختارهای پیچیده و داینامیک درختی مانند Trie (Bodon, 2003) و BSC-Tree (Jiang and Gruenwald, 2005) برای نگهداری داده‌ها تعریف می‌کنند، که انتقال چنین ساختارهایی از طریق شبکه بسیار مشکل است، و حتی اگر هم منتقل کنیم، مشکل تر وقتی است که بخواهیم به صورت کارا داده ساختار اصلی را در سمت گیرنده بازیابی کنیم.

۲-۲- مناسب بودن الگوریتم‌های موازی موجود برای پایگاه داده‌های بیان ژن

بر روی پایگاه داده‌های تراکنشی[1]، الگوریتم‌های موازی ARM متعددی را می‌توان بررسی نمود (مانند Ashrafi et al., 2004)، اما چنین الگوریتمی بر روی داده‌های بیان ژن وجود ندارد. الگوریتم‌های موجود نمی‌توانند از عهده بعد زیاد پایگاه داده‌های بیان ژن برآیند (Zaki, 1999) و معمولا برای کار با پایگاه داده‌های با تعداد کم ستون/بعد طراحی شده‌اند. تاکید اصلی این الگوریتم‌ها بر روی بهینه کردن هزینه دسترسی یا خواندن از یک پایگاه داده‌هاست. دسترسی به یک پایگاه داده، مطمئنا، یک پروسه زمان-بر و هزینه-بری است اما وقتی در مورد پایگاه

[1] Transactional databases