مجله دانشکده پزشکی اصفهان **مقاله پژوهشی**
سال سی‌ام/شماره ۲۰۸/هفته دوم آذرماه ۱۳۹۱
تاریخ دریافت: ۹۱/۵/۲۰ **تاریخ پذیرش:** ۹۱/۷/۲۶

بایکلاسترینگ سری‌های زمانی همبسته در داده‌های میکروآرایه

حسین تقی زاد[۱]، دکتر علیرضا مهری دهنوی[۲]، دکتر مجید محمد بیگی[۳]

چکیده

مقدمه: با شناخته شدن توالی ژنوم‌های مختلف، گام منطقی بعدی یافتن عملکرد و تنظیم آن‌ها می‌باشد. جهت دسته‌بندی ژن‌ها در آزمایشگاه مواردی همچون توصیف رفتار ژن، عوامل کنترل‌کننده‌ی بیان ژن و تعامل پروتئین بررسی می‌شوند. انتظار می‌رود ژن‌هایی که با مکانیسم مشابهی تنظیم می‌شوند، دارای الگوی بیان یکسانی باشند.

روش‌ها: در این مقاله، یک روش خاص خوشه‌بندی به نام بایکلاسترینگ را برای داده‌های میکروآرایه به دست آمده از بیماران مبتلا به MS (Multiple sclerosis) معرفی می‌کنیم. از دیدگاه بیولوژیکی، بایکلاسترهای تنظیم‌کننده‌ی ژنی شامل ژن‌هایی است که در چندین نقطه از زمان تحت چندین شرایط رفتار مشابهی دارند. با شناسایی این بایکلاسترها، پی بردن به مکانیسم‌های تنظیمی که باعث این رفتار مشترک می‌شوند ممکن می‌شود.

یافته‌ها: ما از فرمت تغییریافته‌ی الگوریتم ISA (Iterative signature algorithm) برای استخراج پروفایل‌های هم‌بیان ژن از داده‌های میکروآرایه استفاده کردیم. روش KNN (K-nearest neighbor) در ترکیب با ISA، الگوریتمی ارائه کرد که منجر به یک روش مطلوب برای به دست آوردن مجموعه‌ی همبسته‌ای از ژن‌های همسان در داده‌های میکروآرایه شد.

نتیجه‌گیری: این الگوریتم بر روی دو نوع داده‌ی سنتز شده و داده‌ی واقعی (اطلاعات بیماران مبتلا به مولتیپل اسکلروز) اعمال شد و نشان داد که تفاوت بارزی بین بایکلاسترهای استخراج شده در مقایسه با ISA وجود دارد؛ هر چند که بهره‌وری این روش بر روی داده‌ی سنتز شده و داده‌ی مبتلایان به مولتیل اسکلروز نشان داده شد، اما برای هر نوع داده‌ی دیگری نیز قابل استفاده خواهد بود.

واژگان کلیدی: میکروآرایه، بایکلاستر، سری‌های زمانی، همبستگی

مقدمه

داده‌های با ابعاد بالا به عنوان مشخصه‌ی کلی انواع داده‌ها در نظر گرفته شده‌اند. یادگیری بدون نظارت نقش مهمی در استخراج این داده‌های با ابعاد بالا به منظور پیدا کردن ساختارهای معنی‌دار در آن‌ها دارد (۱). یکی از این انواع داده‌ها، میکروآرایه‌ی DNA است که ما را قادر به مشاهده و نظارت هزاران ژن به طور هم‌زمان می‌کند (۲). در کاربرد خاصی، روش میکروآرایه در محدوده‌ی گسترده‌ای از تجزیه و تحلیل ژنوم مانند شناسایی ژن و کشف دارو با موفقیت قابل اجرا است. این روش به درک درستی از فرایندهای سلولی و شبکه‌های رونویسی منجر می‌شود (۳). الگوی بیان ژن یک سلول یا بافت، ساختار و عملکرد آن را تعیین می‌کند. بیان ژن، یک فرایند پویا است که ممکن است به صورت گذرا یا دایمی تغییر کند. بنابراین، قادر است تغییرات آنی و دایم در حالت بیولوژیکی سلول‌ها و بافت‌ها را منعکس کند (۴).

بیان ژن حاوی اطلاعات با ارزش در شبکه‌های

[۱] کارشناس ارشد، گروه فیزیک و مهندسی پزشکی، دانشکده‌ی پزشکی، دانشگاه علوم پزشکی اصفهان، اصفهان، ایران
[۲] دانشیار، گروه فیزیک و مهندسی پزشکی، دانشکده‌ی پزشکی، دانشگاه علوم پزشکی اصفهان، اصفهان، ایران
[۳] استادیار، گروه مهندسی پزشکی، دانشکده‌ی فنی و مهندسی، دانشگاه اصفهان، اصفهان، ایران
نویسنده‌ی مسؤول: دکتر علیرضا مهری دهنوی
Email: mehri@med.mui.ac.ir

بیولوژیکی، حالات سلولی و ژنی است. یکی از اهداف تجزیه و تحلیل بیان ژن، این است که چگونه یک ژن بر ژن دیگر در همان شبکه‌ی ژنی تأثیر می‌گذارد. یکی دیگر از اهداف ممکن است چگونگی بیان ژن‌ها در سلول‌های سالم و آسیب‌دیده باشد. کاربرد عملی پروفایل بیان ژن کنترل سرطان و بیماری‌های عفونی است (۵). ایده‌ی اصلی در این مطالعات شامل تعریف و شناسایی روند آسیب‌شناسی مربوط به نوع بیماری و پیش‌بینی بیماری است. یکی از کارهای انجام شده در آزمایشگاه خوشه‌بندی ژن‌ها است (۶) و مورد استفاده‌ی آن در مواردی از قبیل توصیف رفتار ژن و عوامل کنترل بیان ژن است. روش‌های خوشه‌بندی استاندارد مانند K-means (۷)، سلسله مراتبی (۸) و طیفی (۹) برای این داده‌ها ارائه شده‌اند. با این حال، فرضیه‌ی در نظر گرفته شده برای این روش‌ها این است که ژن‌ها در همه‌ی شرایط بیان شده‌اند (۱۰). این نظریه همیشه صدق نمی‌کند؛ چرا که بسیاری از ژن‌ها در برخی از شرایط بیان نمی‌شوند.

در روش جدید ارائه شده توسط Cheng و Church، بایکلاسترینگ برای تجزیه و تحلیل بیان ژن معرفی شد (۱۱). الگوریتم آن‌ها مسأله‌ی بایکلاسترینگ را به عنوان یک مسأله‌ی بهینه‌سازی معرفی کرده است و امتیازی برای هر بایکلاستر نامزد در نظر می‌گیرد و روش‌های هوشمندی را برای حل مسأله‌ی بهینه‌سازی محدودشده توسط این تابع امتیاز تعریف می‌کند. Cheng و Church به طور ضمنی فرض کردند که جفت (ژن، شرط) در یک بایکلاستر خوب دارای سطح بیان ثابت می‌باشد. پس از حذف سطر، ستون و زیر ماتریس متوسط، سطح باقی مانده باید تا حد ممکن کوچک باشد. بایکلاسترینگ موضوع محبوبی در خوشه‌بندی داده‌های میکروآرایه پس از پیشنهاد Church و Charge (۱۱) بوده و مقالات بسیاری در این زمینه منتشر شده است. در یک مطالعه، نرم‌افزار Samba ارائه شده است که یک روش تئوری گراف در ترکیب با یک مدل داده‌ی آماری می‌باشد (۱۲). در Samba، ماتریس بیان شده به عنوان گراف دو بخشی مدل شده در نظر گرفته می‌شود. یک امتیاز احتمالی برای دسترسی به اهمیت زیر گراف مشاهده شده نیز تعیین می‌شود. الگوریتم زیر ماتریس نگهدار (Order-preserving submatrix یا OPSM) به صورت زیر ماتریسی در نظر گرفته می‌شود که مرتبه‌ی ستون‌های انتخاب‌شده برای تمام ردیف‌های انتخاب‌شده را حفظ می‌کند (۱۳).

در این مقاله ما بر روی یکی از بهترین الگوریتم‌های بایکلاسترینگ تمرکز کردیم که بنا بر مطالعه‌ای (۱۴)، به نام الگوریتم امضای تکرارشونده (Iterative signature algorithm یا ISA) که در سال ۲۰۰۳ پیشنهاد شده بود (۱۵)، طراحی شد. مفهوم بایکلاستر معنی‌دار بر روی ژن‌ها و نمونه‌ها تعریف می‌شود. ژن‌ها در یک بایکلاستر هم‌بیان هستند و بنابراین، برای هر نمونه میانگین بیان ژن بر روی همه‌ی ژن‌های بایکلاستر باید بزرگ باشد و برای هر ژن، بیان ژن میانگین بر روی همه‌ی نمونه‌های بایکلاستر باید قابل توجه باشد. اما آن چه برای این الگوریتم مشکل‌ساز است تمایل آن به سیگنال‌های قوی می‌باشد که صدها بار قبل ازسیگنال‌های ضعیف شناسایی می‌شوند.

الگوریتم ما از ISA به دو روش متفاوت طراحی شد. اول، روش انتخاب نمونه‌ی ژن از روش

نزدیک‌ترین همسایگی (Nearst neighborhood) به جای روش تصادفی انجام شد. برای این کار، ما اولین ژن نمونه را به طور تصادفی انتخاب کردیم، اما K نمونه‌ی دیگر توسط نزدیک‌ترین همسایگی آن ژن انتخاب شدند. این روش مؤثر است؛ چرا که در اولین گام الگوریتم ژن‌های هم‌بیان در نظر گرفته می‌شوند. دوم، معیارهای امتیازدهی در تعریف بایکلاستر تغییر کرد. همان طور که ما در بخش بعدی خواهیم دید، ISA با استفاده از میانگین بیان ژن روی سطرها و ستون‌ها در ماتریس میکروآرایه و سپس تصمیم‌گیری بر اساس این میانگین برای حذف و یا نگه داشتن یک شرط و ژن ویژه کار می‌کند. این کار به این صورت است که ژن‌ها و شرایطی که میانگین آن‌ها کوچک‌تر از آستانه‌ی T_G و T_C باشند باید حذف شوند. اما، ما همبستگی را به عنوان معیار و یا حذف یک ژن یا شرط در نظر گرفتیم. در روش ما، سری‌های زمانی که با میانگین ژن‌ها و یا شرایط همبسته نیستند، حذف شدند. بنابراین، ما با این کار مشکل اصلی ISA که تمایل به سیگنال‌های قوی دارد را حل خواهیم کرد.

روش‌ها

نمای کلی از الگوریتم ISA در شکل ۱ ارائه شده است.

دو نسخه‌ی نرمالیزه شده از ماتریس اصلی بیان ژن مورد استفاده قرار گرفتند. ماتریس E^G دارای سطرهای نرمال با میانگین صفر و واریانس ۱ و ماتریس E^C دارای ستون‌های نرمالیزه شده به طور مشابه است. میانگین بیان ژن از مجموعه‌ی V' در شرط u را با $e^G_{uV'}$ و میانگین بیان از مجموعه‌ی شرایط U' برای ژن v را با $e^C_{U'v}$ نشان می‌دهیم. در واقع، این میانگین‌ها به معنای امتیاز برای شرایط و ژن‌ها خواهد بود.

$ISA(U, V, E, V_{in}, T_G, T_C, m, \epsilon)$:
U: $Conditions$, V: Genes
E: Gene expression matrix
V_{in}: Initial gene set
T_G, T_C: Gene and condition thresholds
m, ϵ: Stopping criteria
Construct a column standadized matrix E^C
Construct a row standardized matrix E^G
Initialize counters $n = 0, n' = 0$
Initialize the current genes set $V' = V_{in}$
Initialize an empty condition set U'
While $(n - n' < m)$ do
Compute score $e^C_{uV'} = \frac{1}{|V'|}\sum_{v\epsilon V'} e^C_{uv}$ for $u \epsilon U$
$U' = \left\{ u\epsilon U : |e^C_{uV'}| > \frac{T_C}{\sqrt{|V'|}} \right\}$
Compute score $e^G_{U'v} = \frac{1}{|U'|}\sum_{u\epsilon U'} e^G_{uv}$ for $v \epsilon V$
$V'' = V'$
$V' = \left\{ v \epsilon V : |e^G_{U'}| > \frac{T_G}{\sqrt{U'}} \right\}$
if $\left(\frac{|V' \backslash V''|}{V' \cup V''|} < \epsilon \right)$ then $n' = n$
$n = n + 1$
Report U', V'

شکل ۱. الگوریتم ISA (Iterative signature algorithm)

یک بایکلاستر B = (U',V') باید در شرایط زیر صدق کند:

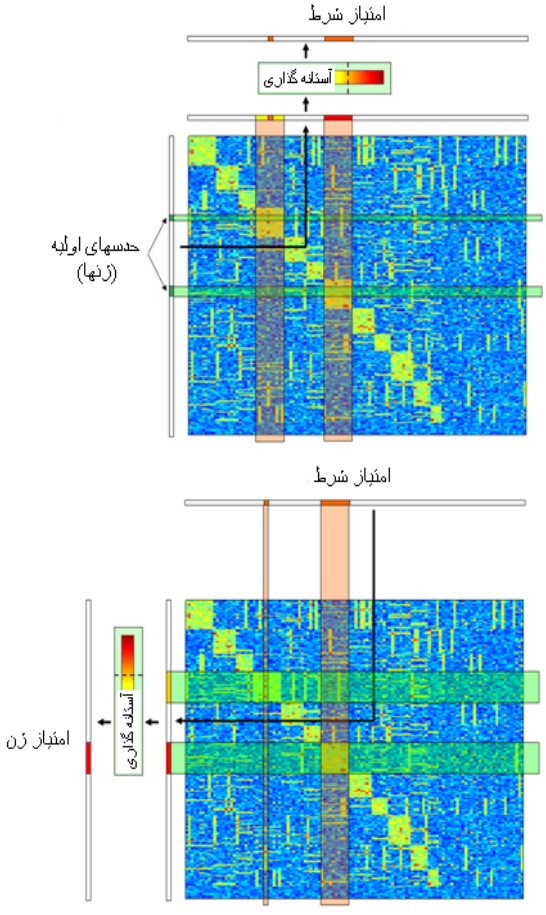
$$U' = \{u \in U : |e^C_{uV'}| > T_C\sigma_C\}$$
$$V' = \{v \in V : |e^G_{U'v}| > T_G\sigma_G\} \quad (۱)$$

در این جا T_G پارامتر آستانه و σ_G انحراف استاندارد از میانگین $V' = \{v \in V : |e^G_{U'v}| > T_G\sigma_G\}$ است که در آن v محدوده‌ی همه‌ی ژن‌های ممکن و U' ثابت است. به طور مشابه T_C و σ_C پارامترهای مربوط برای مجموعه‌ی ستون V' می‌باشد. ایده این بود که

اگر ژن‌ها در V' در شرایط U' هم تنظیم باشند، بیان متوسط آن‌ها باید قابل توجه باشد. استدلال مشابهی برای شرایط U' برقرار است. الگوریتم از مجموعه‌ای دلخواه از ژن‌ها $V_0 = V_{in}$ شروع به کار می‌کند. این مجموعه اغلب ممکن است به طور تصادفی تولید شود. سپس الگوریتم از معادلات به روز رسانی زیر استفاده می‌کند:

$$U_i = \{u \in U : |e^C_{uV_i}| > T_C\sigma_C\}$$
$$V_{i+1} = \{v \in V : |e^G_{U_iv}| > T_G\sigma_G\} \qquad (۲)$$

شکل ۲ نشان می‌دهد که چگونه امتیازدهی برای ژن‌ها و شرایط در نظر گرفته می‌شود.



شکل ۲. نحوه‌ی امتیازدهی ژن‌ها و شرایط

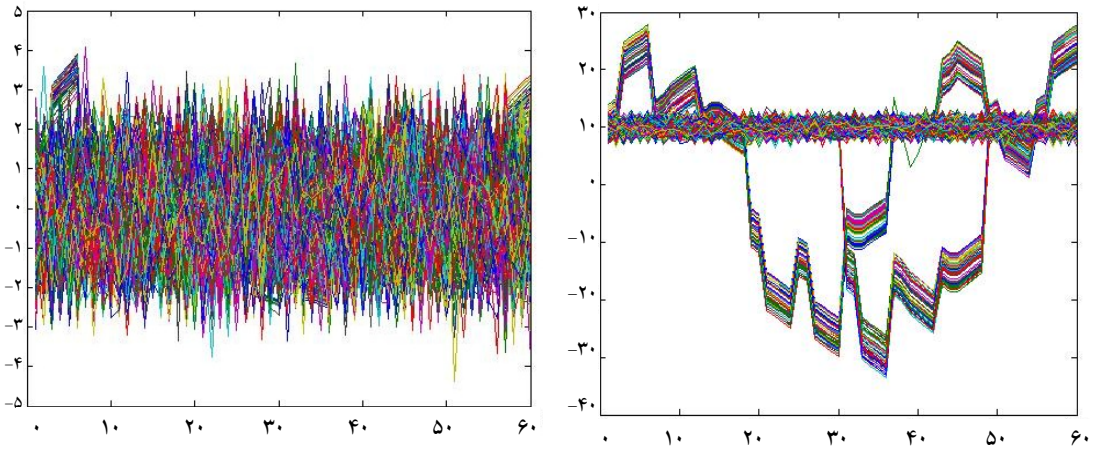
تکرار آن قدر انجام می‌گیرد که رابطه‌ی زیر به ازای nهای کوچک‌تر از یک m برقرار گردد:

$$\frac{|V_{n-i} \setminus V_{n-i-1}|}{|V_{n-i} \cup V_{n-i-1}|} < \epsilon \qquad (۳)$$

بنابراین ISA در یک نقطه‌ی تقریبی ثابت که بایکلاستر در نظر گرفته می‌شود همگرا می‌شود. نقطه‌ی ثابت و واقعی بستگی به هر دو مجموعه‌ی اولیه‌ی V_{in} و پارامترهای آستانه‌ی T_C و T_G دارد. برای تولید مجموعه‌ای از بایکلاسترها، ممکن است ISA با شرایط متفاوت اولیه، از جمله مجموعه‌های شناخته شده‌ای از ژن‌ها یا مجموعه‌های تصادفی و تغییر آستانه به کار گرفته شود. پس از حذف اضافات (نقطه‌ی ثابت که چندین بار ایجاد شده است) مجموعه‌ای از نقاط ثابت را می‌توان به عنوان مجموعه‌ای از بایکلاسترها تجزیه و تحلیل کرد.

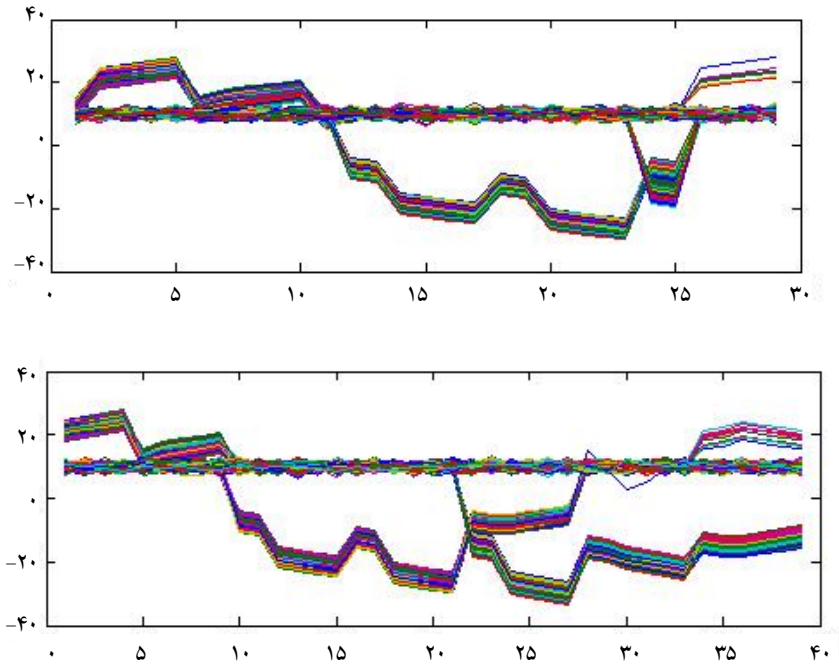
همان طور که توضیح داده شد، الگوریتم تمایل زیادی به سیگنال‌های قوی دارد و میانگین کل برای بایکلاستر مورد انتظار فراتر از سیگنال‌های کوچک خواهد بود. این باعث ایجاد یک مشکل بزرگ است که اجازه نمی‌دهد تا سیگنال‌های کوچک در نظر گرفته شوند. بنابراین، احتمال تشکیل یک بایکلاستر از سیگنال‌های کوچک به شدت کاهش خواهد یافت.

در داده‌ی سنتزشده (شکل ۳)، که در آن چهار بایکلاستر تعبیه شده است، ما ابتدا پیاده‌سازی داده‌ها (شکل ۳-الف) را با الگوریتم انجام دادیم و همان طور که انتظار داشتیم، بایکلاسترها را تا حدودی به ما برگرداند. اما، پس از محدود کردن دامنه‌ی سری‌های زمانی به وسیله‌ی استاندارد کردن آن‌ها (شکل ۳-ب) الگوریتم قادر به تشخیص بایکلاسترهای اصلی حتی پس از تکرارهای متعدد نبود.



۳- الف ۳- ب

شکل ۳. داده‌ی سنتز شده



شکل ۴. یک نمونه بایکلاستر که از الگوریتم ISA (Iterative signature algorithm) به دست آمده است. بعضی از قسمت‌های سری‌های زمانی همبسته شناسایی شده‌اند.

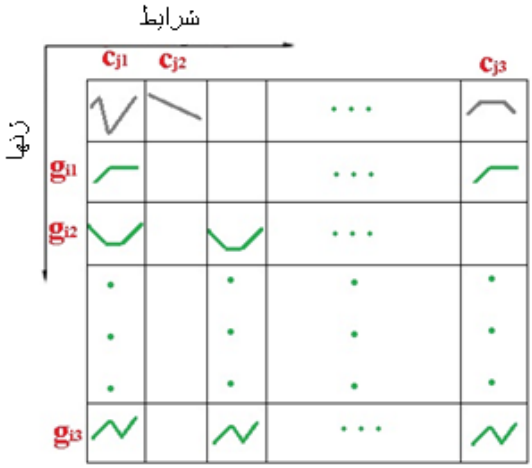
نتایج حاصل از اجرای الگوریتم بر روی شکل ۳- الف در شکل ۴نشان داده شده است.

ما نیازمند روشی هستیم که بتواند سیگنال‌های هم‌تنظیمی در شکل ۳- ب را تشخیص دهد. اگر چه از الگوریتم تکرارشونده‌ی ISA بدین منظور استفاده خواهیم کرد، اما این کار شامل دو تغییر عمده است. اول این که ما تمام نمونه‌های اولیه را بر خلاف ISA، به طور تصادفی انتخاب نکردیم و به جای آن ما فقط یک نمونه به طور تصادفی در هر مرحله تکرار انتخاب کردیم و مابقی نمونه‌ها با استفاده از روش نزدیک‌ترین

همسایگی (Nearest neighborhood) به سیگنال انتخاب شده، انتخاب شدند. این کار ما را قادر ساخت تا برای سیگنال‌های ضعیف اهمیتی در حد سیگنال‌های قوی قایل بشویم. دوم این که ما مقایسه‌ی میانگین بایکلاستر با سیگنال‌های دیگر را از طریق روش همبستگی انجام دادیم. بر خلاف ISA، الگوریتم ما از فاصله‌ی همبستگی برای تعیین این که یک سیگنال می‌تواند در یک بایکلاستر قرار بگیرد استفاده می‌کند.

داده‌های میکروآریه که دو بعدی در قالب ماتریس هستند برای پیاده‌سازی مورد استفاده قرار گرفتند. سطرهای این ماتریس شامل ژن‌ها و ستون‌های آن دارای شرایط آزمایش یا نمونه می‌باشند. درایه‌های ماتریس، شامل سیگنال و یا سری‌های زمانی حاصل از آزمایش هستند. در ISA، داده دارای یک نقطه‌ی نمونه برای هر یک از شرایط است و نقاط موجود در یک وضعیت با یکدیگر مقایسه می‌شوند. اما، داده‌های ما متشکل از سری‌های زمانی به جای نقاط نمونه برای هر شرط بود. این نوع داده، دارای برخی مزیت‌ها است. اول این که ما اطلاعات بیشتری برای هر یک از شرایط تجربی داریم و این واقع‌بینانه به نظر می‌رسد تا این که تنها یک نقطه برای شرایط وجود داشته باشد. دوم این که ما می‌توانیم انواع مختلفی از تکنیک‌های مقایسه‌ی سری‌های زمانی را در تجزیه و تحلیل اعمال کنیم. شکل ۵ نمونه‌ای از این نوع داده را نشان می‌دهد. هر درایه در این ماتریس، شامل یک سری زمانی است که الگوی بیان ژن مربوط به هر شرط را نشان می‌دهد. هدف ما استخراج سری‌های زمانی مشابه در این ماتریس می‌باشد. به عنوان مثال، با توجه به شکل ۵، سری‌های زمانی در ردیف g_{i3} و در شرایط C_{j3}، C_{j2} و C_{j1} دارای قالب یکسان هستند.

از آن جا که تعداد ژن‌ها در داده‌های میکروآرایه‌ی واقعی بیشتر از تعداد شرایط است، بنابراین در نظر گرفتن همه‌ی ژن‌ها برای تجزیه و تحلیل در گام اول کار طاقت‌فرسایی بود. اما در نظر گرفتن همه‌ی شرایط امکان‌پذیر بود. اول، ما یک ژن را به صورت تصادفی انتخاب کردیم و با استفاده از روش NN ژن‌هایی از شرط اول را که به ژن انتخابی مشابهت داشتند، انتخاب کردیم. تعداد نمونه‌ی انتخابی (K) متغیر و قابل تنظیم توسط الگوریتم است. در مرحله‌ی بعد، این ژن تمام شرایط موجود در داده را استخراج کرد و به صورت یک زیر ماتریس در آمد. الگوریتم ما تکرارشونده بود و ما این کار را برای هر شرطی، چندین بار انجام دادیم.



شکل ۵. ماتریس میکروآرایه

در این جا ما نیاز به یک معیار داشتیم تا در تعیین یک زیر ماتریس به عنوان یک بایکلاسترعمل کند. از یک روش آماری ساده برای هرس کردن زیرماتریس اولیه استفاده کردیم. برای هر زیر ماتریس، دو میانگین آماری تعریف شد. برای هر ژن، میانگین همه‌ی شرایط در زیر ماتریس را پیدا کردیم. این کار منجر به ایجاد

یک ستون جدید به نام "ستون میانگین" شد. برای حذف ستون‌های نامناسب در زیر ماتریس، هر یک از ستون‌ها را با ستون میانگین مقایسه کردیم. در این مقاله ما از فاصله‌ی همبستگی مقایسه‌ی شباهت (یا تفاوت) استفاده کردیم. این فاصله به صورت $\rho = 1 - r$ تعریف می‌شود که در آن r همبستگی بین ستون میانگین و هر یک از ستون‌های زیر ماتریس است. هنگامی که ρ کوچک‌تر از آستانه‌ی T_c باشد، ستون مربوط حفظ می‌شود؛ در غیر این صورت، آن را حذف می‌کنیم. با این روش ما شرایط نامناسب از داده را حذف کردیم. به طور مشابه، این کار را برای حذف ژن‌های ناهمبسته نیز انجام دادیم. با تعریف "سطر میانگین" که بر روی سطرهای زیر ماتریس گرفته می‌شود، فاصله‌ی همبستگی برای سطرها به دست می‌آید و سطرهایی که دارای ρ کوچک‌تر از آستانه‌ی T_g هستند، ابقا شدند.

الگوریتم

ماتریس بیان ژن از ژن‌های $G = \{g_1, g_2, \dots, g_{Gn}\}$ و شرایط $C = \{c_1, c_2, \dots, c_{Cn}\}$ تشکیل شده است که در آن Gn و Cn به ترتیب تعداد ژن‌ها و شرایط می‌باشد. بردار E^{gC} پروفایل ژن g تحت شرط c است. با استفاده از این نماد، یک سطر e^{gC} شامل همه‌ی شرایط یک ژن منفرد و یک ستون e^{Gc} شامل تمام ژن تحت یک شرایط واحد می‌باشد.

$$e^{gC} = (e^{gc_1}, e^{gc_2}, \dots, e^{gc_{Cn}})$$
$$e^{Gc} = (e^{g_1c}, e^{g_2c}, \dots, e^{g_{Gn}c}) \qquad (۴)$$

میانگین روی همه‌ی ژن‌های زیر ماتریس، G_s، به این صورت تعریف می‌شود:

$$\langle e^{gC} \rangle = \frac{\sum_{g \epsilon G_s} e^{gC}}{|G_s|} \qquad (۵)$$

تعریف مشابهی برای میانگین روی همه‌ی ستون‌های زیر ماتریس C_s در نظر گرفته می‌شود:

$$\langle e^{Gc} \rangle = \frac{\sum_{c \epsilon C_s} e^{Gc}}{|C_s|} \qquad (۶)$$

حال می‌توانیم تعریف ریاضی برای بایکلاستر را ارائه دهیم. برای هر بایکلاستر، فاصله‌ی Pearson هر شرط c نباید از حد آستانه‌ی τ_C تجاوز کند. به طور مشابه، فاصله‌ی Pearson هر ژن g، نباید از حد آستانه‌ی τ_G تجاوز کند. بایکلاستر به شرح زیر تعریف می‌شود:

$$Bi(\tau_G, \tau_C) = \begin{cases} (G_s, C_s) | \begin{matrix} \forall c \epsilon C_s : \frac{1}{|G_s|} \sum_{g \epsilon G_s} \rho\left(e^{gc}, \langle e^{Gc} \rangle\right) < \tau_C \\ \forall c \epsilon G_s : \frac{1}{|C_s|} \sum_{c \epsilon C_s} \rho\left(e^{gc}, \langle e^{gC} \rangle\right) < \tau_G \end{matrix} \end{cases} \qquad (۷)$$

در هر گام تکرار i، یک فیلتر برای حذف ژن‌ها و شرایطی از G^i و C^i که معیارهای بایکلاستر را برآورده نمی‌کنند، اعمال می‌شود. این منجر به ایجاد ژن‌ها و وضعیت‌های جدید G^{i+1} و C^{i+1} برای گام تکرار i + ۱ می‌شود. تکرار تا جایی ادامه پیدا می‌کند که $|G^i| = |G^{i+1}|$ و $|C^i| = |C^{i+1}|$ رخ دهد.

پس پردازش

الگوریتم ما تعداد زیادی از نمونه‌های تصادفی را استخراج کرد. بنابراین اجتناب‌ناپذیر بود که با این روش یک بایکلاستر چندین بار استخراج شود. علاوه بر آن ممکن است یک بایکلاستر بزرگ، حاوی چندین بایکلاستر کوچک باشد که در مراحل تکرارشونده از الگوریتم تشخیص داده شده‌اند. برای حل این مشکل، بایکلاسترها باید در هم ادغام می‌شدند.

در این مرحله، یک روش خوشه‌بندی معمولی برای ادغام بایکلاسترها می‌تواند مفید باشد. اول، مرکز جرم را برای هر یک از بایکلاسترهای به دست آمده تعریف کردیم. برای این کار، میانگین پروفایل ژن‌های موجود در بایکلاستر را به دست آوردیم و آن را به عنوان مرکز جرم بایکلاستر مربوط در نظر گرفتیم.

سپس آن دسته از بایکلاسترها که شرایط یکسان داشتند، با استفاده از الگوریتم خوشه‌بندی K-means خوشه‌بندی و در هم ادغام شدند. خوشه‌بندی بر روی فاصله‌ی Pearson مراکز جرم عمل کرد؛ به طوری که مراکز جرم مشابه در یک خوشه قرار گرفتند.

یافته‌ها

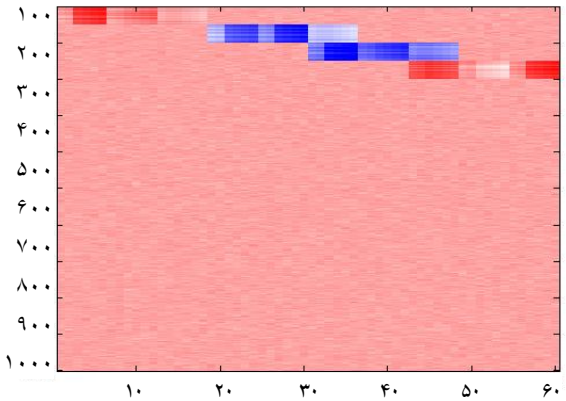
پیاده‌سازی

ما از نرم‌افزار MATLAB R2009a برای اجرای الگوریتم استفاده کردیم. همان طور که پیشتر نیز اشاره شد، ما با استفاده از دو نوع داده‌ی میکروآرایه برای پیاده‌سازی استفاده کردیم. یکی از آن‌ها داده‌ی سنتز شده بود که در آن ما چندین سری زمانی همبسته قرار دادیم. این نوع داده از آن جهت که کارایی الگوریتم را در پیدا کردن بایکلاستر نشان می‌دهد، مفید است. داده‌ی دوم میکروآرایه واقعی بود که از بیماران گونه‌ی انسان مبتلا به بیماری مولتیپل اسکلروز (MS یا Multiple sclerosis) گرفته شد (۱۶).

داده‌ی سنتزشده

این داده که دارای چهار مجموعه‌ی سری زمانی همبسته است، شامل ۱۰۰۰ ژن در ۱۰ شرط می‌باشد. هر یک از این شرایط از ۶ نقطه‌ی زمانی تشکیل شده است. بنابراین، ما با یک ماتریس ۶۰ × ۱۰۰۰ که نمای کلی آن در شکل ۶ نشان داده شده است، کار کردیم.

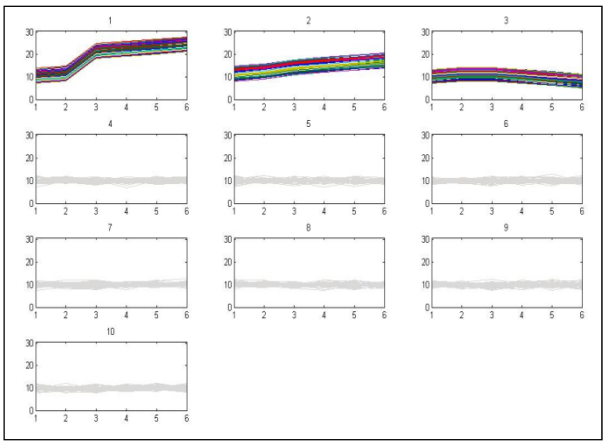
این تصویر توسط جعبه‌ی ابزار پردازش تصویر در نرم‌افزار MATLAB ساخته شده است و چهار بایکلاستر موجود در آن به وضوح دیده می‌شوند. نقشه رنگ (Color map) خاصی (از آبی به قرمز) برای نشان دادن بهتر بایکلاسترها استفاده شده است. برای این کار، درایه‌های ماتریس در فاصله‌ی [۱+، ۱-] نرمالیزه شده‌اند. ۱- به رنگ آبی نسبت داده شده است و ۱+ به رنگ قرمز. همان طور که در بخش روش‌ها اشاره شد، ما علاقمند به پیدا کردن سری‌های زمانی همبسته که در شکل ۳- ب نهفته شده‌اند، بودیم. این سری‌ها در الگوریتم ISA غیر قابل استخراج بود. بنابراین ما، داده‌ی سنتز شده‌ی استانداردشده را برای پیاده‌سازی در نظر گرفتیم.



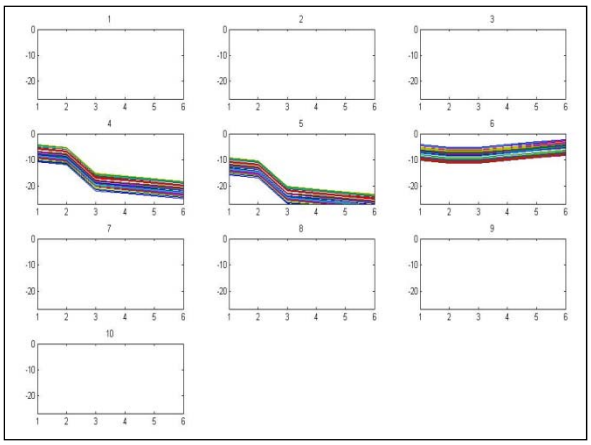
شکل ۶. تصویر داده‌ی سنتزشده با بایکلاسترهای موجود در آن

برای این داده‌ها ما آستانه‌ی سطر T_G را برابر با ۰/۱ و آستانه‌ی ستون T_C را برابر ۰/۱ در نظر گرفتیم. حجم نمونه‌ی k در روش نزدیک‌ترین همسایه ۲۰ در نظر گرفته شد. پس از اعمال الگوریتم به داده، بایکلاسترهای پیش‌بینی‌شده به ترتیب در شکل‌های ۷-۱، ۷-۲، ۷-۳ و ۷-۴ استخراج شدند که با انتظارات ما نیز سازگار است. برای راحتی، ما هر یک از ده حالت را به طور جداگانه در نظر گرفتیم و سری‌های زمانی را در هر شرط نشان دادیم.

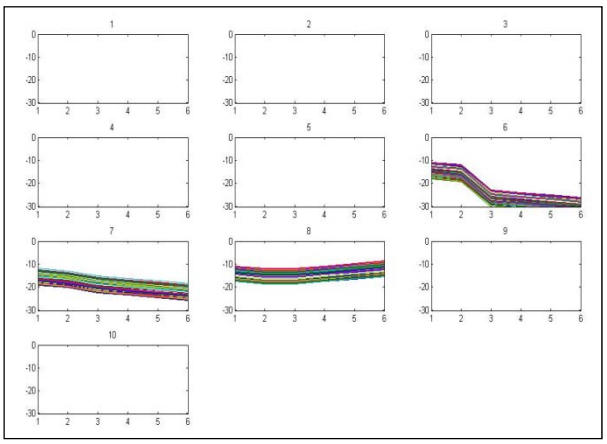
همان طور که از شکل مشخص است، چهار بایکلاستر که هر یک شامل سه سری زمانی همبسته هستند، شناسایی شده است. جدول ۱ تعداد ژن‌ها در هر بایکلاستر را نشان می‌دهد.



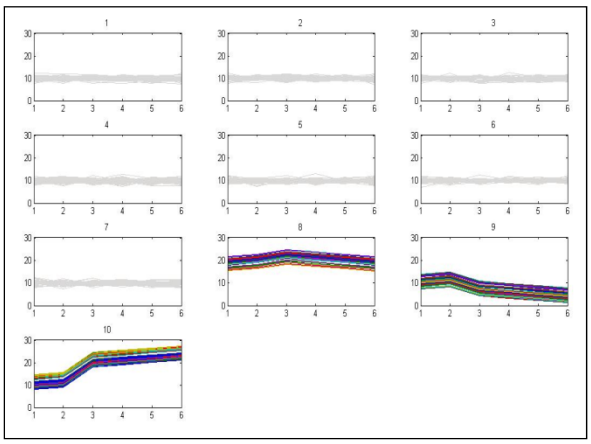
شکل۷-۱. بایکلاستر ۱



شکل ۷-۲. بایکلاستر ۲



شکل ۷-۳. بایکلاستر ۳



شکل ۷-۴. بایکلاستر ۴

شکل ۷. بایکلاسترهای استخراج‌شده در داده‌ی سنتزشده بعد از اعمال الگوریتم. شرایطی که در بایکلاستر قرار نگرفته‌اند به رنگ خاکستری در تصویر نشان داده شده‌اند.

جدول ۱. تعداد بایکلاسترها و ژن‌های استخراج شده در داده‌ی سنتز شده

تعداد بایکلاستر	تعداد ژن‌ها
۱	۴۷
۲	۵۰
۳	۴۸
۴	۵۰

این اعداد نیز با آن چه کـه مـا در داده‌ی سنتزشـده تعبیه کرده بودیم سازگار بودند. این داده نشان می‌دهد که الگوریتم پیشنهادی ما قادر به شناسایی و اسـتخراج سری‌های زمانی همبسته در داده‌های میکروآرایه می‌باشد. بنـابراین، الگـوریتم مـی‌توانـد بـه عنـوان روشـی بـرای بایکلاستر کردن داده‌های میکروآرایه به کار گرفته شود.

داده‌ی بیماری *Multiple sclerosis*

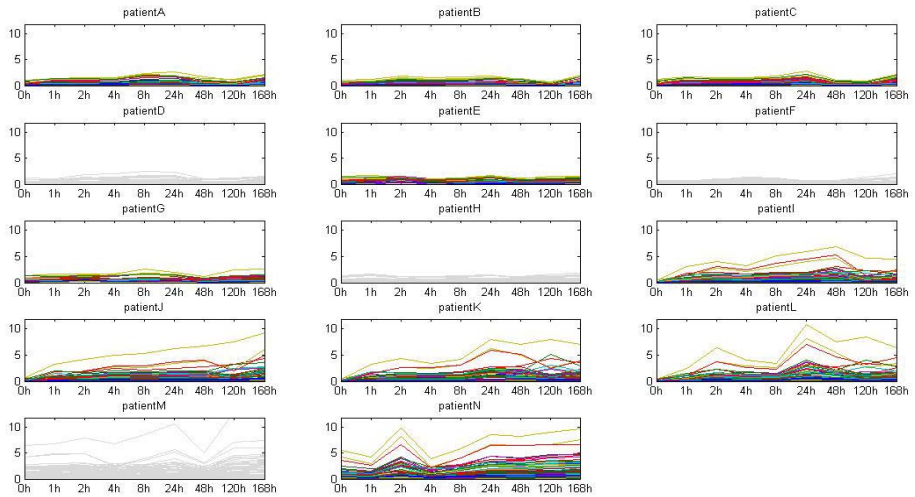
این مجموعه داده در یک دوره از مطالعه‌ی داروشناسی که تجزیه و تحلیل بیماران مبتلا به MS را در پاسخ به درمان IFN-β (Interferon beta) در نظر گرفتـه بـود، ایجاد شد. خون ۱۴ بیمار مبتلا به MS آزمایش شـد و اندازه‌گیری‌ها قبل از درمان و همچنین بعد از ۱، ۲، ۴، ۸، ۲۴، ۴۸، ۱۲۰ و ۱۶۸ ساعت بعد از درمان صـورت گرفـت. ایـن داده از مقالـه‌ای کـه توسـط Weinstock-Guttman و همکاران منتشر شد به دست آمد (۱۶). داده دارای ۲۹۲۰ ژن (سـطر) و ۱۴ شـرایط

(ستون) است. هر یک از شرایط دارای ۹ نقطه‌ی زمانی است و در نتیجه، اندازه‌ی ماتریس آن ۱۲۶ × ۲۹۲۰ است. الگوریتم با مقادیر پارامتری مشابه با آن چه برای داده‌ی سنتزشده تعیین شده بود اعمال شد. نتیجه در جدول ۲ و در شکل‌های ۸-۱، ۸-۲ و ۸-۳ نشان داده شده است.

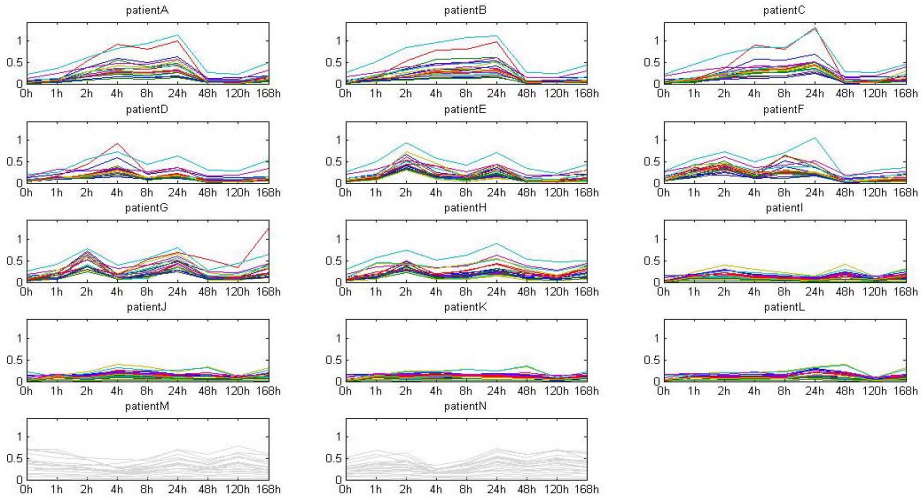
در این شکل‌ها، ما تنها ۳ تا از ۹ بایکلاستر را نشان داده‌ایم. در این داده، شرایط را با توجه به تعداد بیماران نامگذاری کردیم. از آن جا که تعداد بیماران ۱۴ نفر بود، آن‌ها را به ترتیب از A تا N نشان داده‌ایم.

جدول ۲. تعداد بایکلاسترها و ژن‌های استخراج شده در داده‌ی بیماران مبتلا به MS

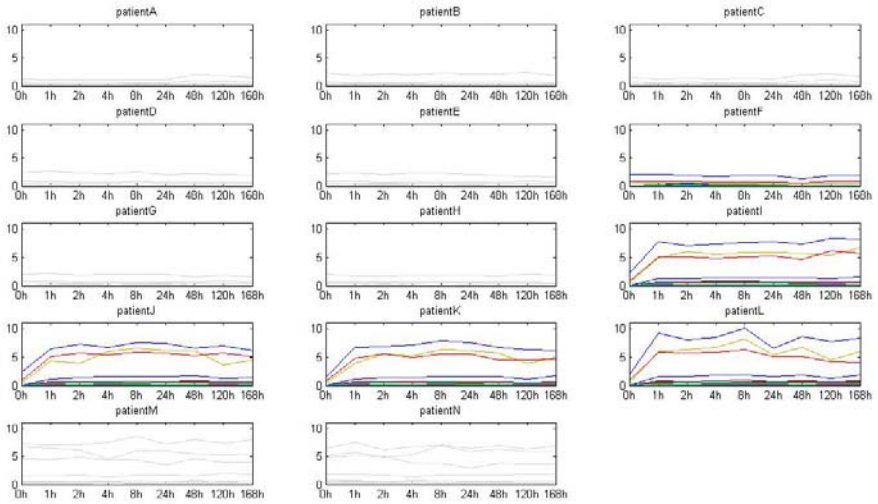
تعداد بایکلاستر	تعداد ژن‌ها
۱	۶۴
۲	۲۰
۳	۱۱
۴	۱۰۸۷
۵	۸۷
۶	۲۰۳
۷	۶۱۶
۸	۴۵۷
۹	۱۷۷



شکل ۸-۱. بایکلاستر ۱



شکل ۸-۲. بایکلاستر ۲



شکل ۸-۳. بایکلاستر ۳

شکل ۸. سه بایکلاستر از نه بایکلاستر استخراج‌شده از داده‌ی مبتلایان به MS (Multiple sclerosis).
شرایطی که در بایکلاستر قرار نگرفته‌اند به رنگ خاکستری در تصویر نشان داده شده‌اند.

بحث

پیدا کردن تنظیم‌کننده‌های ژنی روشی برای تحلیل داده‌های میکروآرایه است. داده‌های میکروآرایه در برگیرنده‌ی هزاران بیان ژنی هستند که در قالب ماتریسی با سطر و ستون‌هایی که ژن‌ها و شرایط بیان آن‌ها را در بر دارد، بیان می‌شود. از دیدگاه ژنتیکی، ژن‌های دارای بیان مشابه با فاکتورهای رونویسی یکسانی فعالیت می‌کنند. با پیدا کردن چنین ژن‌هایی کنترل آن‌ها از لحاظ بیان راحت‌تر است. این امر در مباحثی همچون کلونینگ (Cloning)، مهندسی ژنتیک، ژن‌درمانی و کاربردهای صنعتی نظیر تولید مقادیر زیادی از یک پروتئین به کار برده می‌شود.

ما در این تحقیق به دنبال ارائه‌ی یک روش منسجم و کاربردی در پیدا کردن ژن‌هایی با این خاصیت بودیم. تکنیک‌هایی که به دنبال یافتن بیان ژن‌های مشابه در داده‌های میکروآرایه به کار گرفته می‌شوند، تحت عنوان بایکلاسترینگ مطرح می‌شوند.

سلول‌ها بیان ژن‌های خود را به منظور پاسخ‌دهی مناسب به محرک‌ها هم تنظیم می‌کنند. آن‌ها این ژن‌ها را برای پاسخ‌دهی به استرس سازماندهی می‌کنند. این سازمان پاسخ را می‌توان در مجموعه‌ی داده‌های میکروآرایه به صورت سری‌های زمانی چندگانه مشاهده کرد. ما یکی از الگوریتم‌های بایکلاسترینگ اخیر، ISA، را که به عنوان یکی از موفق‌ترین الگوریتم‌های بایکلاسترینگ شناخته شده بود، تعمیم و توسعه دادیم. الگوریتم پیشنهادی قادر به گرفتن الگوهای پاسخ پیچیده است.

قابلیت الگوریتم در استخراج این الگوهای پاسخ به وسیله‌ی مجموعه‌ی داده‌های مختلفی ارزیابی شد. با استفاده از مجموعه‌ی داده‌ی سنتز شده، نشان دادیم که با وجود ماهیت تصادفی آن، نتایج الگوریتم تا حدودی پایدار هستند. الگوریتم قادر به شناسایی و استخراج سری‌های زمانی جاسازی شده در داده بود. این الگوریتم را می‌توان با تنظیم پارامترهای یکسان برای τ_G,τ_C و تعداد تکرارها به مجموعه‌ی داده‌های مختلف زیستی اعمال کرد. تمایل ویژه‌ی ISA برای سیگنال‌های

قوی را می‌توان از طریق تغییرات کوچک در انتخاب ژن و روش اندازه‌گیری فاصله جبران کرد. این امر منجر به ایجاد مجموعه‌ای جامع از بایکلاسترها می‌شود که طیف وسیعی از پاسخ‌های استرس را در بر می‌گیرد.

تفاوت در الگوهای پاسخ بیماران در داده‌ی مبتلایان به MS بیشتر مورد بررسی قرار گرفت تا تفاوت در محرک‌ها. بایکلاسترهای مبتلایان به MS گویای تفکیک روشنی از بیماران به دو گروه متمایز بود که پاسخ‌های متفاوتی به یک محرک یکسان می‌دهند. این تفاوت‌ها می‌تواند حاوی اطلاعات ارزشمندی در مورد حالت بیماری، پیشرفت بیماری و مکانیسم‌های نظارتی مربوط باشند.

References

1. Beltrame F, Papadimitropoulos A, Porro I, Scaglione S, Schenone A, Torterolo L, et al. GEMMA - A Gridenvironment for microarray-management and analysis in bonemarro-wstemcellsexperiments. Future Generation Computer Systems 2007; 23(3): 382-90.
2. Ben-Dor A, Chor B, Karp R, Yakhini Z. Discovering local structure in gene expression data: the order-preserving submatrix problem. J Comput Biol 2003; 10(3-4): 373-84.
3. Bergmann S, Ihmels J, Barkai N. Iterative signature algorithm for the analysis of large-scale gene expression data. Phys Rev E Stat Nonlin Soft Matter Phys 2003; 67(3 Pt 1): 031902.
4. Cheng Y, Church GM. Biclustering of expression data. Proc Int Conf Intell Syst Mol Biol 2000; 8: 93-103.
5. Ciaramita M, Gangemi A, Ratsch E, Saric J, Rojas I. Unsupervised Learning of Semantic Relations for Molecular Biology Ontologies. IOS Press 2008; 91-104.
6. Eisen MB, Spellman PT, Brown PO, Botstein D. Cluster analysis and display of genome-wide expression patterns. Proc Natl Acad Sci U S A 1998; 95(25): 14863-8.
7. Ho SY, Hsieh CH, Chen HM, Huang HL. Interpretable gene expression classifier with an accurate and compact fuzzy rule base for microarray data analysis. Biosystems 2006; 85(3): 165-76.
8. Lee JM, Sonnhammer EL. Genomic gene clustering analysis of pathways in eukaryotes. Genome Res 2003; 13(5): 875-82.
9. Ng AY, Jordan MI, Weiss Y. On spectral clustering: Analysis and an algorithm. Advances in neural information processing systems 2002; 2(849): 56.
10. Prelic A, Bleuler S, Zimmermann P, Wille A, Buhlmann P, Gruissem W, et al. A systematic comparison and evaluation of biclustering methods for gene expression data. Bioinformatics 2006; 22(9): 1122-9.
11. Sandvik AK, Alsberg BK, Norsett KG, Yadetie F, Waldum HL, Laegreid A. Gene expression analysis and clinical diagnosis. Clin Chim Acta 2006; 363(1-2): 157-64.
12. Schena M, Shalon D, Davis RW, Brown PO. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. Science 1995; 270(5235): 467-70.
13. Tanay A, Sharan R, Kupiec M, Shamir R. Revealing modularity and organization in the yeast molecular network by integrated analysis of highly heterogeneous genomewide data. Proc Natl Acad Sci U S A 2004; 101(9): 2981-6.
14. Tavazoie S, Hughes JD, Campbell MJ, Cho RJ, Church GM. Systematic determination of genetic network architecture. Nat Genet 1999; 22(3): 281-5.
15. Troyanskaya O, Cantor M, Sherlock G, Brown P, Hastie T, Tibshirani R, et al. Missing value estimation methods for DNA microarrays. Bioinformatics 2001; 17(6): 520-5.
16. Weinstock-Guttman B, Badgett D, Patrick K, Hartrich L, Santos R, Hall D, et al. Genomic effects of IFN-beta in multiple sclerosis patients. J Immunol 2003; 171(5): 2694-702.

Journal of Isfahan Medical School Original Article

Vol. 30, No. 208, 2nd week, December 2012 Received: 10.8.2012 Accepted: 17.10.2012

Biclustering of Coherent Time Series in Microarray Data

Hossein Taghizad MSc[1], Alireza Mehridehnavi PhD[2], Majid Mohammadbeigi PhD[3]

Abstract

Background: After recognition of sequences of different genomes, the next logical step is the discovery of their function and regulation. To classify genes in the laboratory, factors such as the behavior of genes, gene expression control and protein interactions have been reviewed. It is expected that genes with similar regulation mechanisms have the same expression patterns.

Methods: In this paper, we introduce a special way of clustering, called biclustering, for microarray data obtained from multiple sclerosis (MS) patients. From a biological perspective, gene regulatory modules consist of genes that have similar behaviors at different points of time under several conditions. By identifying these modules, the recognition of the regulatory mechanisms that are the common causes of genes behaviors might be conceivable.

Findings: We used a modified format of iterative signature algorithm (ISA) to extract co-expressed gene profiles from microarray data. The combination of K-nearest neighbor (KNN) algorithm and ISA provides a helpful algorithm which results in an outstanding and optimum way to obtain similar genes in microarray data.

Conclusion: The algorithm was performed on a synthetic as well as a real database (MS patients' data), and showed a pronounced difference between the extracted modules in contrast to ISA. Although we showed our method's efficiency over synthetic and MS data, it will be usable for any other kinds of data. In other words, our method is based on a series of logical and statistical methods rather than data-based methods.

Keywords: Microarray, Biclustering, Time series analysis, Correlation of data

[1] Department of Medical Physics and Engineering, School of Medicine, Isfahan University of Medical Sciences, Isfahan, Iran
[2] Associate Professor, Department of Medical Physics and Engineering, School of Medicine, Isfahan University of Medical Sciences, Isfahan, Iran
[3] Assistant Professor, Department of Biomedical Engineering, School of Engineering, University of Isfahan, Isfahan, Iran
Corresponding Author: Alireza Mehridehnavi PhD, Email: mehri@med.mui.ac.ir