جستجو در پایگاه داده

مسعود صدیقین

۱۲ بهمن ۱۳۹۳

چکیده

حجم داده‌ها در پایگاه‌های داده‌های زیستی به سرعت در حال افزایش است. همچنین تعداد پرسش‌ها نیز با سرعت زیادی رو به افزایش می‌باشد. برای مثال حدود ۴۰۰۰۰ پرسش در یک روز انجام می‌شود. بنابراین نیاز به روش‌های کارایی است تا بتوان در پایگاه‌های داده جستجو انجام داد. در این جلسه به بررسی این روش‌ها می‌پردازیم.

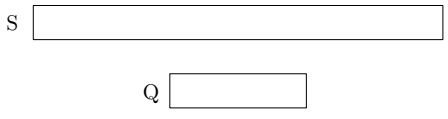
۱ تعریف مساله

مساله‌ی جستجو در پایگاه داده به طور کلی به این شکل می‌باشد:

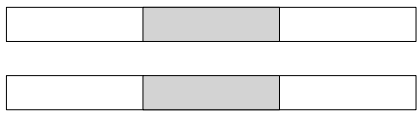
تعریف ۱ *جستجو در پایگاه داده:* *یک پایگاه داده D از رشته‌های ژنوم و یا پروتئینی در اختیار داریم. با داشتن یک پرسش مانند Q می‌خواهیم رشته‌ای مانند S را درون D بیابیم که به Q نزدیک‌تر است.*

می‌توان برای نزدیک بودن، دو معنی در نظر گرفت:

- End-space free



- Local alignment



همچنین برای مقایسه‌ی دو الگوریتم جستجو می توانیم معیارهای زیر را در نظر بگیریم:

- Sensitivity: به این معنی که الگوریتم می‌تواند جواب درست را پیدا کند.
- Specificity: به این معنی که الگوریتم کمتر دچار خطا شده و جواب نادرست را نشان می‌دهد.

در ابتدای کار تنها به مساله تطابق محلی [۱] خواهیم پرداخت. روش‌های موجود برای این کار به شرح زیر است:

۱. جستجوی تمام و کمال: که می‌توان برای جستجو در هر رشته از الگوریتم Smith-Waterman استفاده نمود.

۲. روش‌های مکاشفه ای [۲] : روش‌های مکاشفه‌ای متعددی برای این کار وجود دارد که از جمله‌ی آن می توان به روش‌های $FASTA, BLAST/BLAT, PatternHunter$ اشاره نمود.

۳. روش‌های فیلتر کردن و بهبود دادن: مانند روش‌های $LSH, QUASAR$ اشاره کرد.

۴. روش $BWT - SW$

از میان روش‌های فوق، الگوریتم $Smith - Waterman$ حساس‌ترین الگوریتم بوده و البته نیاز به محاسبات $O(|Q||D|)$ دارد که با توجه به حجم پایگاه داده و تعداد پرسش‌ها، عملی نمی‌باشد. بنابراین به سراغ روش‌های مکاشفه‌ای می‌رویم.

۲ الگوریتم‌های $FASTA$ و $BLAST$

۱.۲ تاریخچه

در جدول زیر، تاریخچه مختصری از این دو الگوریتم ارائه شده است:

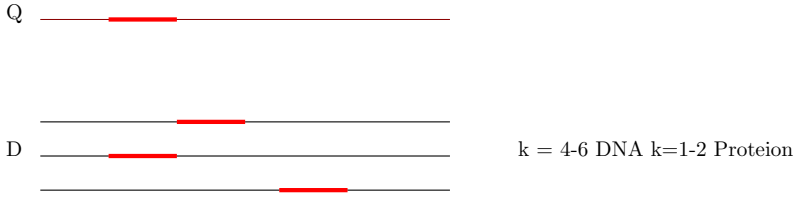
Year	Algorithm	Author	-
1985	FASTP	Limpman and Pearson	Global gapped alignment
1988	FASTA	Pearson and Limpman	Local gapped alignment
1990	BLAST1	Altschul et al	Local ungapped alignment
1996	WU-BLAST2	Gish	Local gapped alignment
1997	NCBI-BLAST2	Altschul et al	Local gapped alignment

۲.۲ روش FASTA

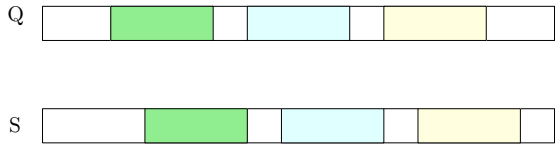
این روش بر پایه آن بنا شده است که اگر بتوان دو رشته را با یکدیگر همردیف نمود و همردیفی دو رشته امتیاز بالایی دارد، آنگاه زیررشته‌های مشترکی بایستی بین آن‌ها موجود باشد. این الگوریتم دارای ۵ مرحله می‌باشد:

[۱] Local Alignment
[۲] Heuristic

۱. در مرحله‌ی اول، ابتدا $Hot-spot$ ها را پیدا می‌کنیم. یک $Hot-spot$ در حقیقت یک $k-mer$ است که بین Q و D مشترک است.

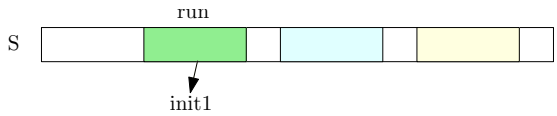


۲. در این مرحله بهترین run را پیدا می‌کنیم. به ازای هر رشته در D ، ۱۰ تا انتخاب می‌گردد. برای امتیاز هی، به $Hot-spot$ ها امتیاز مثبت داده می‌شود و به فاصله میان $Hot-spot$ ها امتیاز منفی که مقدار آن با طول این فاصله، افزایش می‌یابد.



Same distances

۳. مقدار امتیاز واقعی که بر حسب ماتریس امتیازدهی می‌باشد را برای مکان‌های به دست آمده محاسبه می‌کنیم. بهترین آن‌ها امتیاز $init1$ را دارد. این امتیازدهی تنها بر پایه ماتریس امتیازدهی بوده و بنابراین gap را در نظر نمی‌گیرد. (فواصل یکی می‌باشند)



۴. برای هر رشته در D ، آن محل‌هایی که دارای امتیازی پایین‌تر از یک مقدار مخصوص می‌باشند را بیرون می‌ریزیم. برای بقیه آن‌ها سعی می‌کنیم که با اجازه دادن $indel$ ، آن‌ها را همردیف کنیم. بهترین امتیاز در بین ترکیب‌های مختلف از طریق برنامه نویسی پویا قابل محاسبه است.

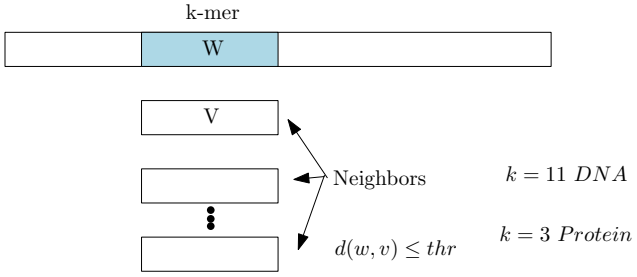
۵. رشته‌هایی که $initn$ آن‌ها کمتر از یک مقدار مشخصی هست را بیرون می‌ریزیم و برای سایرین، الگوریتم $banded\ Smith-Waterman$ را اجرا می‌کنیم و امتیاز بهینه را به دست می‌آوریم رشته‌های باقیمانده را برا حسب Opt ردیف می‌نماییم.

به طور کلی این روش از روش $Smith-Waterman$ سریعتر است. اما دارای حساسیت کمتری می‌باشد.

۳.۲ روش $BLAST$

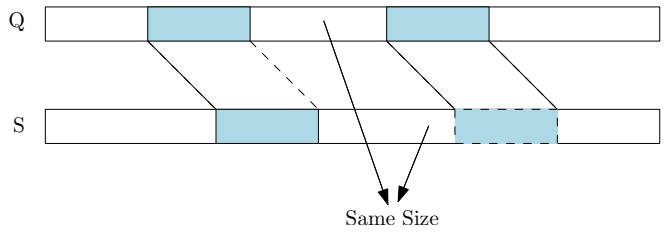
این روش مخفف Basic Local Alignment Search Tool است. این روش نسبت به روش $FASTA$ دارای سرعت بیشتری است. اما میزان حساسیت آن کمتر است. همانطور که در بخش تاریخچه به آن اشاره شد، الگوریتم $BLAST1$ در سال ۱۹۹۰ ارائه شد که در آن شباهت های محلی بدون گپ محاسبه می‌شود. همچنین در الگوریتم $BLAST2$ که در سال ۹۶ و ۹۷ ارائه شد، گپ نیز در نظر گرفته می‌شود. این الگوریتم نیز شامل ۴ مرحله می‌باشد:

۱. **پردازش پرسش:** برای هر مکان از پرسش، تمام $k-mer$ هایی که دارای امتیازی بالاتر از یک مقدار آستانه می باشند را می‌یابیم. به این‌ها همسایه‌های رشته می‌گوییم:



۲. پایگاه داده را اسکن کرده و Hit هایی که با رشته‌های همسایه به دست می‌آیند را پیدا می‌کنیم. هر Hit شامل دقیقا یک مکان در رشته پرسش و یک مکان در پایگاه داده می‌باشد.

۳. برای آنکه یک Hit را بسط دهیم، بایستی یک Hit دیگر در فاصله‌ی کمتر از A از آن موجود باشد، که در آن $A = ۴۰$ است. این Hit بایستی $Diagonal$ باشد.



۴. این Hit ها را انقدر گسترش بدون گپ می دهیم تا امتیاز نواحی اضافه شده از یک مقدار خاص مانند X کمتر گردد. از بین آن‌هایی که گسترش داده شده اند، آن‌هایی را انتخاب می‌کنیم که دارای مجموع امتیاز بالاتر یا مساوی S داشته باشند. به این نواحی HSP [۳] می‌گوییم.

۵. برای HSP های به دست آمده، گسترش با گپ [۴] را انجام می‌دهیم. البته به شرطی که از یک میزان آستانه بیشتر باشند.

در حقیقت این روش، یک روش بهبود یافته از الگوریتم $Smith - Waterman$ است که در آن از وسط یک Hit شروع کرده و به دو طرف با استفاده از برنامه نویسی پویا حرکت می کنیم. هر زمان که میزان از یک امتیاز خاص کمتر گردید، متوقف می‌شویم.

۴.۲ $NCBI - BLAST$

$BLAST$ برنامه‌ای است که به طور متداول برای جستجو در بانک‌های اطلاعاتی مورد استفاده قرار می‌گیرد. که هدف آن بدینگونه است که رشته‌های مشابه یک پرسش را بتواند بیابد. برنامه $BLAST+$ در $NCBI$ شامل برنامه‌های زیر می‌باشد:

[۳] High Scoring Segment
[۴] gapped extension

- BLASTp
- BLASTn
- TBLASTn
- BLASTx
- TBLASTx

همچنین ابزارهایی مانند $makeblastdb$ و $blastdbcmd$ وجود دارند. می توان با اجرای $help-$ متوجه عملکرد این برنامه‌ها شد. مثال:

%makeblastdb -help

۱.۴.۲ $makeblastdb$

این برنامه رشته‌ها را تبدیل به فرمت شناخته شده برای $BLAST$ می‌کند. به عنوان مثال:

%makeblastdb -in swissprot -dbtype prot -parse -seqids

تعریف هر کدام از موارد بالا به این صورت است:

- $-dbtype\ prot$: تعیین می‌کند که رشته‌های پروتئینی را در بانک اطلاعاتی را می‌خواهیم $index$ کنیم. اگر بخواهیم نوکلئوتید ها را $index$ کنیم از $nucl$ استفاده می‌کنیم.
- $-in\ Swissprot$: بخش $-in$ تعیین می‌نماید که پایگاه داده با $FASTA$ فرمت در فایل $Swissprot$ به عنوان ورودی می‌باشد.
- $-parse-seqids$: خط تعریف در فرمت $FASTA$ را $Parse$ می‌نماید. اگر این $option$ را در اختیار نماییم، آنگاه توسط دستور $blastdbcmd$ می‌توانیم رشته ها را خارج نماییم.

پس استفاده از دستور $makeblastdb$ یک سری فایل با فرمت‌های $.pnd, .psi, .phr, ..pni, ..$ تولید می‌کند.

۲.۴.۲ $blastdbcmd$

این برنامه برای خارج نمودن رشته‌ها از پایگاه داده به کار می‌آید. پارامترهای مهم آن به شرح زیر است:

- db: نام پایگاه داده
- $dbtype$: نوع داده که می‌تواند پروتئین $(prot)$ یا نوکلئوتید $(nucl)$ باشد.
- $entry$: نام رشته‌های مورد نظر که با حرف ، از هم جدا می‌شود.
- $range$
- $strand$:plus/minus
- out: نام فایل خروجی

به عنوان مثال می‌توانید دستور زیر را در نظر بگیرید:

blastdbcmd -entry ABL1-HUMAN -db siwssprot -range 1-20

$blastp, blastn, etc$ ۵.۲

پارامترهای مهم برای برنامه‌های $blastp, blastn, tblastn, tblastx$ عبارتند از:

- $outfmt$: فرمت خروجی
- $query$: فایل ورودی
- db: اسم پایگاه داده
- out: فایل خروجی
- $word - size$: عدد صحیح بزرگتر یا مساوی دو برای الگوریتم $word\ finder$
- $num - descriptions$
- $num - alignments$
- seq

به عنوان نمونه می‌توانید دستور زیر را در نظر بگیرید:

blastp -query bcrabl-human.fa -db swissprot -out bcrable.blastp

$PHYLIP$ ۶.۲

یک پکیج می باشد که در آن بیش از ۳۰ برنامه برای ساختن درخت تکامل وجود دارد. در این پکیج فرمت ورودی به شکل $PHYLIP$ می‌باشد که برنامه $Clustalx$ آن را تولید می‌نماید.