بسمه تعالی

Refining core string edits and alignment

مقدمه

دربیوانفورماتیک ویاعلم زیست شناسی هم تراز کردن توالی به روشهای مرتب کردن توالی های DNAوRNAوپروتئین گفته می شود به طوریکه مکانهای مشابه بین توالی ها را مشخص نماید.

این مکانهای مشابه میتواند نشانگر ارتباط عملکردی،ساختاری ویاتکاملی بین توالی ها باشد.

توالی های هم تراز شده رامعمولا به شکل سطرهایی زیر هم درون یک ماتریس نشان می دهیم.

درصورت لزوم دربرخی مکانهای توالی فاصله ایجاد واضافه می کنیم تادرچند ستون پی درپی کاراکترهای یکسان زیر هم قرار گیرند.

رویکردهای محاسباتی برای هم تراز کردن توالی ها:1- ترازبندی سراسری (Global Alignment)

2- ترازبندی محلی (Local Alignment)

ترازبندی سراسری

محاسبه ی هم ترازی سراسری،شکلی از بهینه سازی سراسری است که به هم ترازی فشار می آورد تادرکل طول توالی های مورد جستجو گسترده شود .درواقع تلاش می کند هر پس ماندی را در هر توالی هم تراز کندکه این برای وقتی بهترین کاربرد را دارد که اندازه ی دورشته(توالی)یکسان باشد.

ترازبندی محلی

هم تراز کردن محلی،نواحی مشابه درون توالی های بلند راکه معمولا در طول توالی بسیار متفاوتند،تشخیص می دهد.درواقع هم تراز کردن محلی که نشان دهنده ی وضعیت فاصله هاست به نحوی که توالی ها به اندازه ی کافی مشابه باشند.

مثال

Global FTFTALILLAVAV

F--TAL-LLA-AV

Local FTFTALILL-AVAV

--FTAL-LLAAV--

حال به مساله خود می پردازیم.

الگوریتم Needleman_wunsch

مرحله 1-تعریف مساله

الگوریتم Needleman_wunschیک ترازبندی سراسری روی دو رشته s1 وs2 می باشدکه این الگوریتم در تراز بندی DNAهاوپروتئین ها در علم زیست شناسی رایج است.

این الگوریتم اولین الگوریتم کاربردی دربرنامه ریزی پویا برای مقایسه ی توالی هاست.(درعلم زیست شناسی)

ماتریس مشابهت (Similarity matrix)

زمانی که می خواهیم میزان مشابهت دو کاراکتررابررسی کنیم به ماتریس sمراجعه می کنیم که این میزان مشابهت دراین ماتریس مقدار دهی شده است.

مثلا(a,b)s مقدار مشابهت کاراکترهای a,bاست.

	A	G	C	T	_
A	10	−1	−3	−4	−5
G		7	−5	−3	−5
C			9	0	−5
T				8	−5
_					−5

:

مثلا برای رشته های ترازبندی شده ی روبرو:

AGACTAGTTAC

CGA − − −GACGT

S(A,C)+S(G,G)+S(A,A)+S(C,−)+S(T.−)+S(A,−
)+S(G,G)+S(T,A)+S(T,C)+S(A,G)+S(C,T)=

−3+7+10 –(3*5)+7 – 4+0− 1+0=1

برای یافتن ماکزیمم میزان مشابهت در تراز بندی ما نیاز به یک ماتریس Vداریم که هردرایه درسطر iام وستون jام این ماتریس با V(i,j)نشان داده می شودواز طرفی سطرهای ماتریس بیانگر رشته ی s2وستونهای آن بیانگر کاراکترهای رشته ی s1میباشد.

که دراین صورت می دانیم باتوجه به اینکه رشته ی s1دارای nکاراکترورشته ی s2دارای mکاراکتراست میزان فضای حافظه O(mn)است.

مرحله 2-راه حل بازگشتی

همانطور که قبلادرارایه ی edit disstanceذکر شده

$V(0,j)=\sum_{1\leq k\leq j} s(-,s2(k))$

$V(i,0)=\sum_{1\leq k\leq i} s(s1(k),-)$

ورابطه بازگشتی

$V(i,j)=\max[v(i-1,j-1)+s(s1(i),s2(j)),$

$v(i-1,j)+s(s1(i),\_),$

$v(i,j-1)+s(\_,s2(j))]$

مرحله 3-محاسبه یبیشترین مقدار مشابهت ها[(Vmn)ترازبندی بهینه]

شبه کد برای محاسبه ی ماتریس v:

1. For i=0 to length(s1)
2. V(i,0) ← v(i-1,0) + s(_,s1(i))
3. For j=0 to length(s2)
4. V(0,j) ← v(0,j-1) + s(s2(j),_)
5. For i=1 to length(s1)
6. For j=1 to length(s2)
7. (
8. Match ← v(i-1,j-1) + s(s1(i),s2(j)
9. Delete ← v(i-1,j)+s(s1(i),_)

```
10.  Insert ← v(i,j-1)+s(_,s2(j))
11.  V(i,j) ←  max(Match, Delete, Insert)
12.  )
```

بامحاسبه ی ماتریس Vmnمی توان ماکزیمم مقدار رااز ترازبندی های ممکن بدست آورد.

برای محاسبه ی تراز بندی از آخرین خانه سمت راست Vmnشروع به حساب کردن می کنیم ومقدارها را باسه مقدار ممکن که ازسه عمل (Match, Delete, Insert)بدست می آید،مقایسه می کنیم.

مرحله 4-ساختن ترازبندی بهینه

شبه کدبرای محاسبه ی s1,s2ترازبندی شده بااستفاده از ماتریس v(محاسبه مسیر بهینه)

```
1.  Alignment  s1 ← “  “
2.  Alignment  s2 ← “  “
3.  i ←  length(s1)
4.  j ←  length(s2)
5.  while(i>0   and   j>0)
6.  {
7.      Score  ←  v(i,j)
8.      Score diag  ← v(i-1,j-1)
9.      Score up    ← v(i,j-1)
10.     Score  left  ← v(i-1,j)
11. If(score ==  score diag + s(s1(i),s2(j)))
12. {
13. Alignment  s1  ←   s1(i)+alignment s1
14. Alignment  s2  ←   s2(j)+alignment s2
```

15. i ← i−1

16. j ← j−1

17.}

18.Else if (score == score left+ s(s1(i),_))

19.{

20. Alignment s1 ← s1(i) + alignment s1

21. Alignment s2 ← "_" + alignment s2

22. i ← i−1

23.}

24.Other wise (score == score up+ s(_,s2(j)))

25.{

26.Alignment s1 ← "_" + alignment s1

27.Alignment s2 ← s2(j)+alignment s2

28. j ← j−1

29. }

30. }

31.While (i>0)

32.{

33. Alignment s1 ← s1(i)+alignment s1

34. Alignment s2 ← "_" + alignment s2

35. i ← i−1

36.}

37. While (j>0)

38.{

39.Alignment s1 ← "_" + alignment s1

40.Alignment s2 ← s2(j)+alignment s2

41.j ← j−1

42.}

مرحله 5-بهبودبرنامه

همانطور که مشاهده شد الگوریتم بالابه o(mn)فضا وهمچنین به o(mn)زمان نیاز دارد،

برای بهبود برنامه می خواهیم فضای o(mn)رابه o(min(m,n))کاهش دهیم.یعنی فضارا به یک فضای خطی کاهش دهیم.

الگوریتم هیشبرگ فضارابه o(min(m,n)) کاهش میدهد ولی زمان را تقریبا به دو برابر افزایش می دهد.

حال مساله ی جدیدی به وجود آمده است که بااستفاده ازبرنامه ریزی پویابه صورت زیر ارائه می شود.

مرحله 1-تعریف مساله

الگوریتم هیشبرگ می تواند به عنوان یک نسخه ی تقسیم وغلبه

(divide and conquer)ازالگوریتم needleman_wunschمعرفی شود.

الگوریتم هیشبرگ برای محاسبات زیستی رایج است تا بهترین تراز بندی سراسری بین توالی هایDNAوپروتئین هارامحاسبه کند.

دورشته ی زیر موجود می باشد:

S1_____$n-i$.....n

S2_____$m-j$ ……m

تعریف:برای هر رشته s،وجودداردs^rکه به معکوس رشته اشاره می کند.

مثال:

S=abc ⇒ s^r=cba

درنتیجه داریم:

$s1^r$…….i_____n

$s2^r$……j_____m

مرحله 2-راه حل بازگشتی

حال برای حل این مساله نیاز به شکستن مساله به زیر مساله هاست.

برای این کارابتدا مساله رابه دوزیر مساله تبدیل می کنیم،بدین صورت که یکی ازرشته ها مثلا رشته ی اول رانصف میکنیم،پس در ادامه ی کار به دنبال یک مقدار بهینه برای شکستن رشته ی دوم می باشیم(K)

S1_____n/2_____n

S2_____k_____m

بنابراین kبهینه مقدار Vبهینه یعنی vماکزیمم رابه ما می دهد.

برای بدست آوردن مقادیرv،قسمت اول مساله رامی توان از جدول vمحاسبه کرد ولی در مورد تکه ی دوم چون حدپایین آنرا نداریم نمی توان محاسبه نمود.به همین دلیل از معکوس آن استفاده میکنیم.یعنی V^r

بنابراین از فرمول بازگشتی زیر استفاده می کنیم:

$$V(m,n)=\max\left[\, v(n/2,k) + v^{r}(n/2,m-k) \,\right]$$

$$0\le k\le m$$

پس مساله به دو زیرساختاربهینه v,v^{r}تبدیل شد که بابهترین مقدار kکه همان k^{*}است بدست می آید.

اثبات بهینه بودن زیرمساله ها:

اگر k^{*}درتکه ی اول مساله یعنی قسمت vباشد{n/2_____s1}

S2_____k*

آنگاه زیر ساختاراول که بهینه است درمورد زیر ساختار دوم هم اگر فرض کنیم که یک kبهتری در آن موجود می باشد که مقدار vراماکزیمم می کند ،در نتیجه نهایتا یک v(m,n)بدست می آید که مقدار بیشتری از v max دارد که با حکم ما در تناقض استدر نتیجه فرض خلف باطل است.

در مورد قسمت دوم اثبات هم دقیقا به همین ترتیب ،بااین تفاوت که kبهینه را در تکه ی دوم مساله فرض می کنیم و...

****

سپس مساله را به همین ترتیب میشکنیم،یعنی در هر بار رشته ی اول را به دو قسمت تقسیم کرده(n/2→n/4→n/8→...) تا به ابتدای رشته برسیم ،سپس از نقطه ی(0و0)تا (n,m) در هر مرحله kبهینه را بدست می آوریم واز این طریق مسیر بهینه ی ما ساخته می شود.

مشخص است که با این روش دیگر نیازی به نگه داشتن ماتریس vبه طور کامل نیست وماتنها به خود درایه وسطر قبل از آن احتیاج داریم وبقیه ی سطرهای قبل از آن دورریز می شود (یعنی به جز خود درایه ی مورد نظر به درایه ی قبلی آن در ستون قبلی ودرایه ی قطری قبلی ودرایه ی بالای آن در سطر قبلی نیاز است ،بنابراین علاوه بر خود سطر تنهابه سطر قبلی نیاز است)درنتیجه فضا به فضای خطی کاهش می یابد.

مرحله ی3و4-محاسبه وساختن مسیربهینه

V(s1,s2)

n=length(s1) ,m=length(s2)

If n<=1 or m<=1

Output alignment (s1,s2) using Needleman_wunsch

Else

Mid=floor(n/2)

S1_left = s1(1,mid)

S1_right =s1(n,n-mid)

Max=-∞

For k=1 to k=m{

S2_left=s2(1,k)

S2_right=s2(m,m−k)

F=v(s1_left , s2_left)+ v(s1_right ,s2_right)

If F>max

Max=F}

k^*=index max{ v(s1_left , s2_left)+ v(s1_right ,s2_right)}

Return max

تحلیل زمان:

در تحلیل فضایی گفته شد چون بعضی از اطلاعات درهرمرحله دورریز می شود فضای ما به فضای خطی کاهش می یابد ،ولی باید توجه داشت چون همه ی محاسبات باید انجام شود واین بار دو ماتریس v,v^rبایدمحسبه شود پس زمان افزایش می یابد:

تعداد زیر مساله ها :

n/2 2m

$n/2^2$ 2m

.

$\rightarrow 2m\ log_2^n$

.

.

$n/2^i$ $2m$

بنابراین:

$n/2 \times 2m + n/2^2 \times 2m + \ldots + n/2^i \times 2m$

$o(2m log_2^n)$

$o(m)$

$\Rightarrow\Rightarrow o(m^2 log_2^n)$

****

تحلیل زمانی که در کتاب ذکر شده:

$cnm \rightarrow cnm/2 \rightarrow cnm/4 \rightarrow \ldots$

در مرحله iام از بازگشت $\Rightarrow cnm/2^{i-1}$

ازطرفی تعداد فراخوانی ها $=\log_2 n$

$$\sum_{i=1}^{\log_2 n} cnm/2^{i-1} = cnm \sum_{i=1}^{\log_2 n} 1/2^{i-1} \quad (*)$$

تصاعد هندسی:

$a(1-q^n)/1-q = (1-1/2^{\log_2 n})/1/2 = 1-2^{\log_2 n^{-1}}/1/2 = 1-1/n/1/2 = 2(n-1)/n$

$\Rightarrow\Rightarrow$

$(*) = cnm*2(n-1)/n = 2cnm-2m \leq 2cnm \quad \Rightarrow o(nm)$