به نام خداوند بخشنده و مهربان

درس مقدماتی بر بیوانفورماتیک

تمرین دوم

استاد درس: دکتر زینعلی

نام دانشجو:

روزبه قاسمی ۹۵۳۱۴۲۴

فروردین ۱۳۹۹

سوال اول

در این سوال ابتدا به مقایسه روش های BLAST و FASTA می‌پردازیم و در نهایت در رابطه با روش SW نیز بحث خواهیم کرد:

BLAST و FASTA دو برنامه هستند که به کاربر اجازه می‌دهد توالی پرس و جو خود را با توالی های موجود در پایگاه داده های موجود مقایسه کند و شباهت ها را بررسی کند. هدف اولیه FASTA مقایسه تنها توالی پروتئین بود اما نسخه اصلاح شده این نرم افزار هم مقایسه توالی پروتئین و DNA را تسهیل می‌کند. اگرچه FASTA نرم افزار خوبی است و نرخ پوشش بهتری برای همولوگ ها دارد، اما اکثر مردم از ابزار تراز BLAST استفاده می کنند زیرا محبوبیت بالاتری دارد و نتایج دقیق تری نسبت به FASTA تولید می کند. همچنین، ابزار BLAST با توجه به نیاز کاربر قابل تغییر است و می‌تواند انتخاب‌گری بیشتری داشته باشد. اگر بخواهیم عمیق‌تر به بررسی این دو روش بپردازیم می‌توان به این موضوع اشاره کرد که مرحله‌ی دانه پاشی(Seeding) این دو روش با هم متفاوت است. از لحاظ الگوریتمی نیز، روش BLAST با تشکیل لیستی از کلمات به دنبال توالی مناسب می‌گردد در حالی که روش FASTA با ایجاد Hash Table و امتیازدهی انطابق های مناسب را پیدا می‌کند. همچنین طبق نکته‌ای که استاد ذکر کردند، روش FASTA در صورت استفاده از توالی با طول کم و روش BLAST در صورت استفاده از طوالی با طول زیاد مناسب تر از دیگری است و در نهایت می‌توان گفت که مهم‌ترین تفاوت این دو روش این است که روش BLAST معمولاً برای همترازسازی انطباق های محلی بهینه بدون ایجاد Gap به کار می‌رود در صورتی که FASTA برای پیدا کردن شباهت میان توالی ها با شباهت کمتر به کار می‌رود.

حال به بررسی SW به صورت مختصر می‌پردازیم. الگوریتم اسمیت-واترمن[1] یا به اختصار SW انجام دادن یک هم‌ترازسازی توالی محلی به کار گرفته می‌شود و برای مشخص کردن مناطق مشابه بین دو توالی اسید نوکلئیک یا پروتین استفاده می‌شود. به جای در نظر گرفتن تمام توالی این الگوریتم سعی می‌کند که با در نظر گرفتن بخش‌های مختلف با همه طول های ممکن میزان شباهت را بهینه کند. این روش چون که اوردر زمانی بسیار بالایی دارد به صورت تئوری برای توالی های کم خوب عمل می‌کند اما اگر دیتابیس بزرگی داشته باشیم تقریباً غیر ممکن است که بشود از این روش استفاده کرد!

[1] Smith-Waterman Algorithm

سوال دوم

در این سوال برای هر کدام از توالی های داده شده، مقایسه را انجام می‌دهیم سپس بر اساس نتایج بدست آمده می‌گوییم کدام توالی شبیه ترین توالی به توالی هدف است.

طبق فرض سوال گفته شده که Ktups را یک در نظر بگیریم:

توالی اول:

Query Sequence: ACTCCCGTTAAAGCACA

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
A	C	T	C	C	C	G	T	T	A	A	A	G	C	A	C	A

Hash Table for Query:

T	C	A	G
3	2	1	7
8	4	10	13
9	5	11	
	6	12	
	14	15	
	16	17	

Target Sequence 1: TTACCCATGGCGCAATTACTG

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
T	T	A	C	C	C	A	T	G	G	C	G	C	A	A	T	T	A	C	T	G

Target Table 1:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
T	T	A	C	C	C	A	T	G	G	C	G	C	A	A	T	T	A	C	T	G
2	1	-2	-2	-3	-4	-6	-5	-2	-3	-9	-5	-11	-13	-14	-13	-14	-17	-17	-17	-14
7	6	7	0	-1	-2	3	0	4	3	-7	1	-9	-4	-5	-8	-9	-8	-15	-12	-8
8	7	8	1	0	-1	4	1			-6		-8	-3	-4	-7	-8	-7	-14	-11	
		9	2	1	0	5				-5		-7	-2	-3			-6	-13		
		12	10	9	8	8				3		1	1	0			-3	-5		
		14	12	11	10	10				5		3	3	2			-1	-3		

همانطور که در جدول بالا نشان داده شده است، طولانی ترین رشته هایی که گپ ایجاد نشود، **3** است. در نتیجه به این معناست که آنها در صورت شیفت دادن به اندازه اعدادشان یک سری exact match ایجاد می‌شود که exact match های آن به شرح زیر است:

ACT , CCC , TTA , GCA

توالی دوم:

از توالی اولی، جدول Hash table برای رشته Query را داریم پس دیگر نیاز به محاسبه دوباره نداریم.

Target Sequence 2: TATACTCCCGTTAAATACCCA

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
T	A	T	A	C	T	C	C	C	G	T	T	A	A	A	T	A	C	C	C	A

Target Table 2:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
T	A	T	A	C	T	C	C	C	G	T	T	A	A	A	T	A	C	C	C	A
2	-1	0	-3	-3	-3	-5	-6	-7	-3	-8	-9	-12	-13	-14	-13	-16	-16	-17	-18	-20
7	8	5	6	-1	2	-3	-4	-5	3	-3	-4	-3	-4	-5	-8	-7	-14	-15	-16	-11
8	9	6	7	0	3	-2	-3	-4		-2	-3	-2	-3	-4	-7	-6	-13	-14	-15	-10
	10		8	1		-1	-2	-3				-1	-2	-3		-5	-12	-13	-14	-9
	13		11	9		7	6	5				2	1	0		-2	-4	-5	-6	-6
	15		13	11		9	8	7				4	3	2		0	-2	-3	-4	-4

همانطور که در جدول بالا نشان داده شده است، طولانی ترین رشته هایی که گپ ایجاد نشود، **12** است:

ACTCCCGTTAAA

توالی سوم:

از توالی اولی، جدول Hash table برای رشته Query را داریم پس دیگر نیاز به محاسبه دوباره نداریم.

Target Sequence 3: CGCGCAAATTGGCACATTCGA

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
C	G	C	G	C	A	A	A	T	T	G	G	C	A	C	A	T	T	C	G	A

Target Table 3:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
C	G	C	G	C	A	A	A	T	T	G	G	C	A	C	A	T	T	C	G	A
1	5	-1	3	-3	-5	-6	-7	-6	-7	-4	-5	-11	-13	-13	-15	-14	-15	-17	-13	-20
3	11	1	9	-1	4	3	2	-1	-2	2	1	-9	-4	-11	-6	-9	-10	-15	-7	-11
4		2		0	5	4	3	0	-1			-8	-3	-10	-5	-8	-9	-14		-10
5		3		1	6	5	4					-7	-2	-9	-4			-13		-9
13		11		9	9	8	7					1	1	-1	-1			-5		-6
15		13		11	11	10	9					3	3	1	1			-3		-4

همانطور که در جدول بالا نشان داده شده است، طولانی ترین رشته هایی که گپ ایجاد نشود، **5** است:

GCACA

نتیجه گیری:

همانطور که دیدیم، توالی دوم در این پایگاه داده، بیشترین شباهت را با توالی کوئری ما داشت!

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
T	A	T	A	C	T	C	C	C	G	T	T	A	A	A	T	A	C	C	C	A
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	
			A	C	T	C	C	C	G	T	T	A	A	A	G	C	A	C	A	

سوال سوم

الف)

همانطور که در صورت سوال گفته شد قرار است دو الگوریتم پیشنهادی را با هم از نظر دقت و زمان اجرا مقایسه کنیم. اولین الگوریتم یعنی الگوریتم Star یک بار هم‌ترازی[۲]رخ می‌دهد و سپس عملیات ادغام[۳]انجام می‌شود اما در الگوریتم حریصانه[۴]به تعداد K-1 بار هم‌ترازی رخ می‌دهد. از آنجایی که هزینه زمانی هم‌تراز کردن نسبت به ادغام کردن، بیشتر است پس الگوریتم حریصانه سرعت کمتری دارد، زیرا همانطور که گفته شد دارای پیچیدگی زمانی بیشتری است اما این الگوریتم دقت بیشتری نسبت به الگوریتم Star دارد زیرا بهترین امتیاز بین هم‌ترازی‌ها را در نظر می‌گیرد.

ب)

الگوریتم ClustalW به gap اهمیت نمی‌دهد و بر اساس فاصله[۵]عمل می‌کند چون که می‌دانیم gap به معنای گذاشتن یک آمینواسید دلخواه در آن ناحیه است. از طرفی هم جریمه های gap قابل تنظیم هستند پس اجازه‌ی رخ دادن gap کمتری در مناطق حفاظت شده[۶]می‌دهد و این روش بر اساس درخت فیلوژنتیک[۷]می‌تواند به ترتیب فواصل کمتر را ادغام کند که Mismatch کمتری دارند و در نتیجه Mismatch در حالت کلی کاهش پیدا می‌کند پس بهترین Match را برای توالی های انتخاب شده محاسبه می‌کند. اما الگوریتم Star پس از بدست آورن مرکز توالی ها به صورت کاهش از لحاظ امتیاز و شباهت به هم‌ترازی چندگانه اضافه می‌شود و MSA با امتیاز بالایی تولید می‌شود. این درحالی هست که در الگوریتم ClustalW به جواب بهینه تری می‌رسیم پس الگوریتم ClustalW مناسب‌تر است.

پایان

[2] Alignment
[3] Merge
[4] Greedy
[5] Distance
[6] Conserved Domain
[7] phylogenetic tree