From Sung §4.1-4.4,§4.6



differences. What do we do?

- We know that we can use Smith-Waterman
- Lets assume we have it as two 3x10⁹ base sequences:
 - The computation time and memory are on the order of 9x10¹⁸
 - Computing each cell of the table would take 52 days*
 - The table would need 250 XB to store**
 - Then there is traceback....

Let's assume (though we have not gotten to the details of how to do it yet) that we have both my and Dr. Salamah's genome and we want to find the

> * assuming 2GHz, 1 core, and that each cell takes only one cycle ** as integers with no overhead



One other caveat:

- That would only account for SNPs and indels

As we discussed theres also structural changes that wouldn't be found

So what do we do?

similar sequence.

Most specialized tools for this problem follow 3 basic steps: aligned.

if the anchors cover the whole sequence.

Well, we know genomes are **very** similar, so they will share regions of highly

- 1.Identify potential anchor points -- ideally these points can be found quickly and will provide a limited number of locations to investigate further
- 2.Identify groups of anchors that are co-linear and non-overlapping -these will provide further evidence that there is a region that should be
- 3.Close any gaps between anchors -- this will complete the entire alignment



1.Identify potential anchor points



2.Identify groups of anchors that are co-linear and non-overlapping



3.Close any gaps between anchors



3.Close any gaps between anchors

Lets start with what an anchor is!

- The simplest version is the *k*-mer Basically you chop both sequences into all of its k length substrings
- (overlapping)
- You use all locations with the same k-mer as a potential anchor Its fast normally since you can use hashing to do this lookup (we will talk about this later)
- But has a tradeoff: small k gives lots of matches but its faster, large k is slow but more precise to a point.

Lets break that phrase down:

- They are matches, meaning the same substring in both occur in both sequences
- They are unique, meaning the substring only occurs once in each sequence
- They are maximal, meaning the characters to the left and right don't match

More formally:

Definition 4.1 Given two genomes A and B, a Maximal Unique Match (MUM) substring is a common substring of both A and B of length longer than a specified minimum length d (commonly d=20 is used) such that • it is maximal, that is, it cannot be extended in either direction without

- incurring a mismatch; and
- it is unique in both sequences.

An example, lets say d=3:

S = ACGACTCAGCTACTGGTCAGCTATTACTTACCGC#T = ACTTCTCTGCTACGGTCAGCTATTCACTTACCGC\$

Whats the algorithm we just used for that?

Brute Force:

- for every position *i* in A and *j* in B:

 - check to ensure $|P| \ge d$
 - check if *P* is unique in both genomes

• find the longest common prefix (call it P) of A[i....n] and B[j....m]

Brute Force:

- for every position *i* in A and *j* in B:

 - check to ensure $|P| \ge d$
 - check if *P* is unique in both genomes

O(mn)

• find the longest common prefix (call it P) of A[i....n] and B[j....m]

Brute Force:

- for every position *i* in *A* and *j* in *B*:

 - check to ensure $|P| \ge d$
 - check if *P* is unique in both genomes

O(mn)

• find the longest common prefix (call it P) of A[i....n] and B[j....m]

What tools do we have that we could use?

Using suffix trees:

- Build a generalize suffix tree for A and B
- from B) check if $A[i-1] \neq B[i-1]$, if so report as a MUM

Using suffix trees:

- Build a generalize suffix tree for A and B
- from B) check if $A[i-1] \neq B[i-1]$, if so report as a MUM



d = 1

Using suffix trees:

- Build a generalize suffix tree for A and B
- from B) check if $A[i-1] \neq B[i-1]$, if so report as a MUM



d = 1

Using suffix trees:

- Build a generalize suffix tree for A and B
- from B) check if $A[i-1] \neq B[i-1]$, if so report as a MUM



d = 1

Using suffix trees:

- Build a generalize suffix tree for A and B
- from B) check if $A[i-1] \neq B[i-1]$, if so report as a MUM



d = 1

Using suffix trees:

- Build a generalize suffix tree for A and B
- from B) check if $A[i-1] \neq B[i-1]$, if so report as a MUM



d = 1

Using suffix trees:

- Build a generalize suffix tree for A and B
- from B) check if $A[i-1] \neq B[i-1]$, if so report as a MUM



d = 1

Using suffix trees:

- Build a generalize suffix tree for A and B
- from B) check if $A[i-1] \neq B[i-1]$, if so report as a MUM



d = 1

Using suffix trees:

- Build a generalize suffix tree for A and B
- from B) check if $A[i-1] \neq B[i-1]$, if so report as a MUM



d = 1

Using suffix trees:

- Build a generalize suffix tree for A and B
- from B) check if $A[i-1] \neq B[i-1]$, if so report as a MUM



d = 1

Using suffix trees:

- Build a generalize suffix tree for A and B
- from B) check if $A[i-1] \neq B[i-1]$, if so report as a MUM



d = 1

Using suffix trees:

- Build a generalize suffix tree for A and B
- from B) check if $A[i-1] \neq B[i-1]$, if so report as a MUM



d = 1

Using suffix trees:

- Build a generalize suffix tree for A and B
- from B) check if $A[i-1] \neq B[i-1]$, if so report as a MUM



d = 1

Using suffix trees:

- Build a generalize suffix tree for A and B
- from B) check if $A[i-1] \neq B[i-1]$, if so report as a MUM



d = 1

Using suffix trees:

- Build a generalize suffix tree for A and B
- from B) check if $A[i-1] \neq B[j-1]$, if so report as a MUM



d = 1

Using suffix trees:

- Build a generalize suffix tree for A and B
- from B) check if $A[i-1] \neq B[i-1]$, if so report as a MUM



d = 1

Using suffix trees:

- Build a generalize suffix tree for A and B
- from B) check if $A[i-1] \neq B[i-1]$, if so report as a MUM



d = 1

Using suffix trees:

- Build a generalize suffix tree for A and B
- from B) check if $A[i-1] \neq B[j-1]$, if so report as a MUM



d = 1

Using suffix trees:

- Build a generalize suffix tree for A and B
- from B) check if $A[i-1] \neq B[j-1]$, if so report as a MUM



d = 1

Using suffix trees:

- Build a generalize suffix tree for A and B
- from B) check if $A[i-1] \neq B[j-1]$, if so report as a MUM



d = 1

 Mark (list) all internal nodes with exactly 1 child from each sequence • For each marked node (lets say the children are labeled *i* from A and *j*



Algorithm says the MUMs are: Т CG

Using suffix trees:

- Build a generalize suffix tree for A and B
- from B) check if $A[i-1] \neq B[i-1]$, if so report as a MUM

What about running time?

- building the tree: O(m+n)
- checking prefixes: again bounded by internal nodes, O(m+n)

 Mark (list) all internal nodes with exactly 1 child from each sequence • For each marked node (lets say the children are labeled *i* from A and *j*

• marking nodes: bounded by number of internal nodes, so O(m+n)

Similar genomes will not only share unique sequences, but also preserve order of these sequences, therefore we can identify the similarity (not necessarily the alignment) by finding the longest common sequence of MUMs from the two sequences.

This is the basis of the original MUMmer program (v1).



MUMmer

LCS: Dynamic Programming

lets let *V*[*i*,*j*] be the length of the longest common subsequence between *P*[1....*i*] and *Q*[1....*j*]

And define a function δ such that $P[i] = Q[\delta(i)]$

$$V[i, j] = \max \begin{cases} V[i - 1, j] \\ 1 + V[i - 1, \delta(i)] \end{cases}$$

// P[i] is not involved in the LCS - 1] $j \ge \delta(i)$ // P[i] is involved in the LCS

LCS: Dynamic Programming



 $V[i,j] = \max \begin{cases} V[i-1,j] \\ 1 + V[i-1,\delta(i)-1] & j \ge \delta(i) \end{cases}$



245831697

V	0	1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0	(
1	0	0	0	0	0	0	1	1	-
2	0	1	1	1	1	1	1	1	-
3	0	1	1	1	1	2	2	2	
4	0	1	2	2	2	2	2	2	
5	0	1	2	3	3	3	3	3	
6	0	1	2	3	3	3	3	4	2
7	0	1	2	3	3	3	3	4	2
8	0	1	2	3	4	4	4	4	2
9	0	1	2	3	4	4	4	4	Ę



- The idea is to sparsity the DP using a two key observations: • going across the rows, only increment by 1
 - we only need to compute new values in a narrow region



(1,1), (2,2), (3,3), (7,4), (9,5)



Let *j*' be the smallest integer greater than $\delta(i)$ such that $V[i-1,\delta(i)-1]+1 < V[i-1,j']$.

In that case we can see that $V[i,j] = \begin{cases} 1 + V[i-1,\delta(i)-1] & \delta(i) \le j \le j'-1 \\ V[i-1,j] & \text{otherwise} \end{cases}$

Therefore, we construct a new row (i) by 1.copying the tuples from *i*-1 3.insert ($\delta(i), V[i-1, \delta(i)-1]+1$).



- 2.delete all tuples (*j*, V[*i*-1,*j*]) where $j \ge \delta(i)$ and V[*i*-1,*j*] $\le V[i-1,\delta(i)-1]+1$
 - (1,1), (2,2), (3,3), (7,4), (9,5) (1,1), (2,2), (3,3), (7,4), (9,5) Step 2 (1,1), (2,2), (3,3), (9,5)(1,1), (2,2), (3,3), (4, 3+1), (9,5)(1,1), (2,2), (3,3), (4,4), (9,5)



If we store the tuples as a binary search tree: search, insert, & delete are *O*(log *n*) time each.

Since we can insert at most *n* tuples (one per row), and each tuple can only be deleted once. Therefore those operations are at most *O(n* log *n)* time total.

Step 1: identify all MUMs. This can be done in O(n) time assuming the sequences are of length *n*.

m << n. This takes $O(m \log m)$ time.

used here).

Original MUMmer

Step 2: employ LCS on the MUMs. Lets call the number of MUMs m, here

Step 3: employ another alignment algorithm to fill small gaps (various tools)

These improvements changed the following:

- Improved memory consumption using better implementations of STs
- Using a simple ST, rather than a generalized tree: build the tree on one them stream the other over it to find MUMs
- •Implementing clustering, since there may be major structural changes, don't just find one LCS, split the problem down and find groups of LCSs
- Relaxing uniqueness

MUMmer2 & MUMmer3

A concept used often in (computational) biology is visualization to get a general idea of what the data means.

In genome alignment (and many other large alignment problems as we will see) we use a dot plot.

- Each position in the graph represents a location in the two genomes.
- A dot represents a match of a certain length surrounding those locations

Dot Plots



and itself. The dot plot on the left was generated using window size 9 and allowing no mismatches. The dot plot on the right was generated using window size 9 and allowing at most 1 mismatch.

Dot Plot

Human vs. Chimpanzee

																	1.101				32.1
i da la	11 an an 17 19 a		1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1	1.5	19 - A - CA	12 114	d (1997)	1111		2. S.	1	1.1.1.1.1.0	S. 77 S.	- 640	17	19-34	1.13	1.1	·	- 19.Ž.	
2.2.3.	distan di	소설 나라 것	- 1. 1	1. 1		198414	1. 21	12.1		da As	2744 - E		¹	18.4		語い	(5)4°)**	244	4.4	in 1 min	1
	M 1 35 - 15	8 C ()	1.1.1.1.1.1.1.1	St. 1911	1990 o	en én co	전환율성	17 10 10	1.15-1			$t : \cdots : y$		100	193	お茶		29-6		SALE REAL	12.2
1 -2. 1 2.2	Frank Kert 1 - 1	151 2213			- El		a 1 1 1 1		1 - 115	-11 - 2-3	- 186 - 12	é i ser el	1 211	Second -	45	1.1.11		111	17-E-1	11.1	29.84
No. 1		100100		1.4827 2.11	21'	1 1 10 1 16	e. D. Let.		L.V. A PEA	ant isa	a	11.0 - 10	1.1.	1111	74.2573	U NA	17	i i A	15 8. 1	1.15	REAL
				1.1.1			_											1		-	
	1. 	1	· · · · · · · · · · · · · · · · · · ·		24							1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1							- 2-4 -	2.00	
	1. A. 1. A. 1. A. A.	이상 나이지?	9 N N		S. 1	144		1. ⁴ 1 4	1.1	1 12	Ag . 1	13.1			1.1417	12	100	200	2.5 1		
<u></u>	i in the second second	1130	<u>- 1- 41, - 12</u>	<u></u>		NR 14 413	2 (1 (1)) 			Providence and a second					- 41-42 	13.4	- <u>1</u> 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1				4.9.1
	1. 1. 1.	*13 ()		de side	1 A		이 옷을 가슴?	나라한 문		1.1	ju - j		<u>A</u> ROS	1	1.00		E. 1		1	ait 4.0	
				1								2				<u> </u>	<u></u>			-	- <u>-</u>
김 문		이 같은 것이 있다.	1.19.14							11.01	i ar c		14	14 34			4.3	1			
·	2.8.19					ri. Erz	97 A.A.A.A.A.A.A.A.A.A.A.A.A.A.A.A.A.A.A.	1.1	1.12.14		14. 1	i i i i i iii	1.1	1.000	1.111			-	1 4	27. J.	
- 1 M	Mertinet .	de de la		1997 - 1947 - 1947 - 1947 - 1947 - 1947 - 1947 - 1947 - 1947 - 1947 - 1947 - 1947 - 1947 - 1947 - 1947 - 1947 -	uller di		F. LT	1.1.1	10.945	18 G.C		1.4.48		1	문문	1655			승규는	417 NO	
					in in			1			i sie a			1				•	·	in an	
, e	1	and the second s	er still still	9 9 Y 19 1	P6 22 3	e the e	1.		1. 1. 1. 1. 1.	San Lord	e ne e je			1	1.1.1	() · · ·	late e	1.1.1	Sec. 3.		2
			· . · · · · · ·					·*· · ·		1 - 1 - 1 - M - M		1.13 ⁴ - 1. 2.1 - 2	1.1	100	199.4%	: : : : : : : : : : : : : : : : : : :	-		1		11
1	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	· · · · ·	1 1 1 1	5 155 - L S		1.11.1		- 1 h h	1	d e staden d	· · ·	1973 - P			2.13	· · ·			· Ę :-	1.1	
					-1 ₂		÷ .:	-			: •		1				-:		. :		· · .
	11 S.C. (17 S			1.1.14		1.1.1	N. 16 1.	1		· · · ·	1.1.1	2	1		1.1	1.51				1.1.1	12
5		12.14	· · · ·	1.7 14	1			1 1 1 1	1.11	1.2.12.1	E 182	114 7	1		1.					128	1.23
e è cue	A		• • • • • • •	1.1.1				• • •			1	1 1 3		:				•		the Le	1
		1. 2. 1. 1. 1. 1.	i de la server	1.11.11.11.11	5	10.14 6	,		·	1	1			1 1	1.1.1.1.1	1.45	TING		· · ·	674.,87p	·
		100 100 100 100 100 100 100 100 100 100		1.1.1		11 2 2 2 2 2 2		AL	- r - 1939		- 1999 - 1	1212 0		*	112	1.1	·** · · ·			ini aib	- 1
	1. 1. 1. 1. 1.	1. 1. 1.					1.11.11	1	1.191	=/			- 1	1 1	<\$.	4.11	11		1 2 1	a	1.1
	A Star &	91 A. S.		·	See	i dag di se	1.1.1.1	145 21	3 2 200	1 24.1	1 16 2	13 3	54	: :	11.2	10.90	÷.	61	113.4	teri ena	32 6
				1	1	1 2 3 4	12			1 1 1		11. 1			2.2		28.4		. 1		
	**										5.4	1.1					· ; .]			16	
1.1.1.1	Contraction of the second	and the la			4.11. 1. 1	Section of the	1 F 1	1.1.1	1.1.30	ene ingi	in al la	1. 1.		200		1	in in			10.000	1.1
	1	and the second second											14 14		and and	2 <u>-2-2-</u> 2-2-2-			2 U	1 (m) 2 (m)	
		1994 C 1			49 g		1. 1. j.		1.1	이제 가슴			- 11		1.683	19.58		si.	0.1	10 ju	
	1.121		÷ · ···.		· · · · · ·				1.000	47	for the second	1.14			· · · 4.2	1.4					
	1.			1.1.1.1																-	
		1						10.1			· · · · ·					- G. (1	з.		10 11	n. 1. 300	
· · · .					19 - Ale			1		1.1.1.1	÷ .		1.1	3.00	1.2		5.1.4	1	1.	1.10	
4 4 4	1 18 18 18 18 18 18 18 18 18 18 18 18 18			1210	19.2	11 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1		1. 19		1.00					1.1.1.1.1	14 (M)					
								111	17 230	1 1 1		2	1		1.1	· · ·				5100	
			1	· · · · ·			1. 1.	1	· · · · · · ·	1. A			20		1.1						·
					10-1. 	• • • • • •	1								1.1.5				11		100
e - 11.	1 4 5 4	1 2 1 2 11			. (. · · ·		4 Ng 4 Mg		A 589		* - (*) *	e	1.1	1.1.1.1	1.1	11.01		. 1	1.5		1.1.1
	Bride Star		a te start	1.14		in the fire	n in jeus	1.1.1	e di senata	19190	12.	1.5	1.1	1.	1440	ca fe	a		141	ini és	Se
1 1	1. 1. 1. 2. 3.	St. 9. 1	11.1.1.1	. 11 14	1		112.191	4-5-54	10.000		1 M. 1	··· 2 . ··			411-	t à às	515	1400	1.1		11
						1			1 - 1 - 1 - 1 - 1 				1.5						1.1.1		· · · ·
		1.1		1	1		de Asiele	1 Sc			i per l			1. C.					51.		
	14 M 1	4 1 mm		- 11 - 11-11 -	1.1	1 1		's '- '		1. 19					1.1	1 a 1	÷		1	=	11-12
	1	. 1. 1			1		- 14 - 1					and the state				1	<u>, 1</u> , 1		1.		
1 1 1 1		and a subject of														1.1				101 - 34D	
12:12	1 Sec. 1997	3 . :	(A. (1999)	a marker	2. 1	- 2::::	1.12	47.1	1.44	1.1.1.1.1	: 8 a	1.11.19	34.0		1.1	1. 1-	-	1.1	1.1	En mo	3272
	1 18 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	1 1		S. C.	· · · ·								1.1	1	1.03	·					1 1
1 -	Sec. 20.	1 I I		N	in the state				A. 19	1.10					1.4	1	11			1.11	1.1
<u> </u>	1.1.1.1.1.1.1				2							1.0.1			1.			1.1		2.00	
· .		•	1.11	1.1.1			(1993) (M	41.4	1.1.1			1.2. 11	12		1.11	-	•			-	
	a set in a	1.1	1 1 1 1	1.		and the second		1. S. S. S.		1.2.0		· · · .	1.1	· ·	1.0	: ÷ ;		· .		· · · ···	· · · ·
· .		1	No.								1.12		1		113	3.1	1.1	1.4			
1.1		111	1	1.1.24						×	- 13 - I		1.	1.1	1.197	e 4				in ta	
		1 1 1	1	1 1 11				11			1.1.1				14	1 1				E. 1.1	1.1
1 5	1. 1. 2. 1. 1.	1. 1 M					1.1.2.2	5	1.000		1.44	A 141	- <u>-</u>	11						يني وين اين وين	19
-		111					· · ·	· · ·			- 14 X			3				·			
1.14	2 2 Carl 12	1. 45	1. 1. 1.	$(1, \frac{1}{2}) = 0$	Se	e i gingit i	a president	1.1.1.1	e città	ورافر معداه	- N		5.52	2.40	14 .00	19.20	-12 C	4.	10	4.64	1.
	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	A second					1944 - E. I.							1	1.1	- 1					
1.1.1	1.1.1.1.1.1.1	1 5. 2. 1	· · · ·	1999 (B)	- St	10 35	2 31 1	1. 18 A.	100		1: 2: 2	i de trader	11 .	. 11-1	12.1	11	a 40		14		1.
1	1 Barris	Anne		1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1	A	111		5. 4 4 2	1 356					10			al.	1.50	. : . :	1000	
•••••	1. 1. 1.	1		1. 1 fell	and a s		1 1 I I I I I I I I I I I I I I I I I I	1		1 164	1.64			1	1.32	1.1	1.1	· · .	E P	· · · · · · · · · · · · · · · · · · ·	1.1
	a state state	1. Sec.	· · · · · ·		State of	. hele .					·			11		1.			· ·	.u.	4.
	मा लोहेले. जोव	· R. Christ		1.1.1.1.	Second Second		1 11 2	1. 22. 22	· · · · · · · · · · · ·	14 million	- 111	111 12	11	1.31	110	6 15	2114	· · · ·	18.4	の方	
· · ·	- 1.Y /	1. 1. 1. 1.		· · · · · i ii	4111	to See	8 . 1 . 1 · ·	1811	100	l'a sele	: i	1 d	12	1	1 1 1 1	12.1	18		: : :		•• •
	·	10 100		1		* G .	4 1 1 1 1	. f.			111		11 -	5 g	- 11	1	. 27				· · ·
· ·	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	2			d'	the line		144.4	- 18 A - 1		1 1 1					1.25	2.0	•		21-01	
	2	3	4	5	6	7	x	8	9	10	11	12	13	14	15	16	17	18	20 V	19	22 21
					-		r*	-						- 1							

https://genomevolution.org/wiki/index.php/File:Master_7029_7071.CDS-CDS.blastn.dag.go_c4_D20_g10_A5.aligncoords.gcoords_ct0.w1200.png