Sequence Alignment

Michael Schatz

Bioinformatics Lecture 2
Quantitative Biology 2011

Exact Matching Review

Where is GATTACA in the human genome?
E=183,105

Brute Force
(3 GB)

```
BANANA
BAN
 ANA
  NAN
   ANA
```

Naive

Slow & Easy

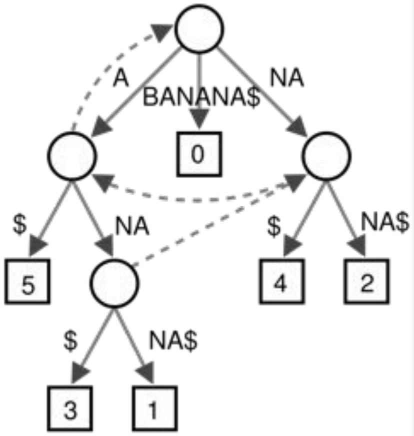
Suffix Array
(>15 GB)

6	$
5	A$
3	ANA$
1	ANANA$
0	BANANA$
4	NA$
2	NANA$

Vmatch, PacBio Aligner

Binary Search

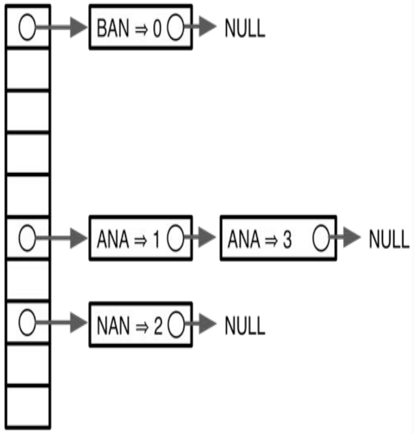
Suffix Tree
(>51 GB)



MUMmer, MUMmerGPU

Tree Searching

Hash Table
(>15 GB)



BLAST, MAQ, ZOOM, RMAP, CloudBurst

Seed-and-extend

In-exact alignment

- Where is GATTACA *approximately* in the human genome?
  - And how do we efficiently find them?

- It depends…
  - Define 'approximately'
    - Hamming Distance, Edit distance, or Sequence Similarity
    - Ungapped vs Gapped vs Affine Gaps
    - Global vs Local
    - All positions or the single 'best'?

  - Efficiency depends on the data characteristics & goals
    - Smith-Waterman: Exhaustive search for optimal alignments
    - BLAST: Hash based homology searches
    - MUMmer: Suffix Tree based whole genome alignment
    - Bowtie: BWT alignment for short read mapping

Searching for GATTACA

- Where is GATTACA *approximately* in the human genome?

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

Match Score: 1/7

Searching for GATTACA

- Where is GATTACA *approximately* in the human genome?

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

Match Score: 7/7

Searching for GATTACA

- Where is GATTACA *approximately* in the human genome?

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

Match Score: 1/7

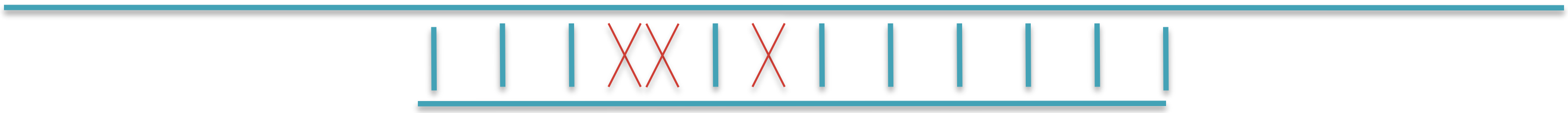
Searching for GATTACA

- Where is GATTACA *approximately* in the human genome?

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

Match Score: 6/7 <- We may be very interested in these imperfect matches
Especially if there are no perfect end-to-end matches

Hamming Distance



- Metric to compare sequences (DNA, AA, ASCII, binary, etc…)
  - Non-negative, identity, symmetry, triangle equality
  - How many characters are different between the 2 strings?
    - Minimum number of substitutions required to change transform A into B

- Traditionally defined for end-to-end comparisons
  - Here end-to-end (global) for query, partial (local) for reference

[When is Hamming Distance appropriate?]

- Find all occurrences of GATTACA with Hamming Distance ≤ 1

[What is the running time of a brute force approach?]

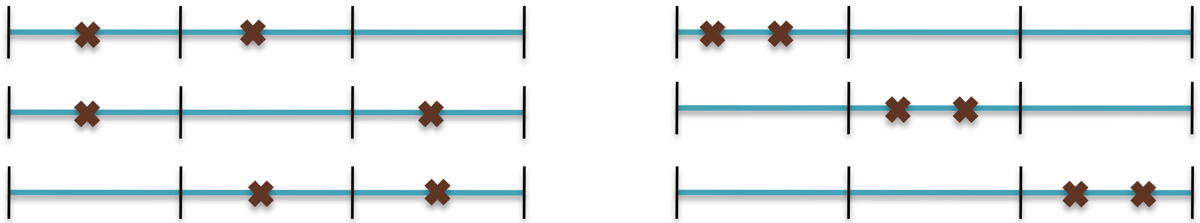
Seed-and-Extend Alignment

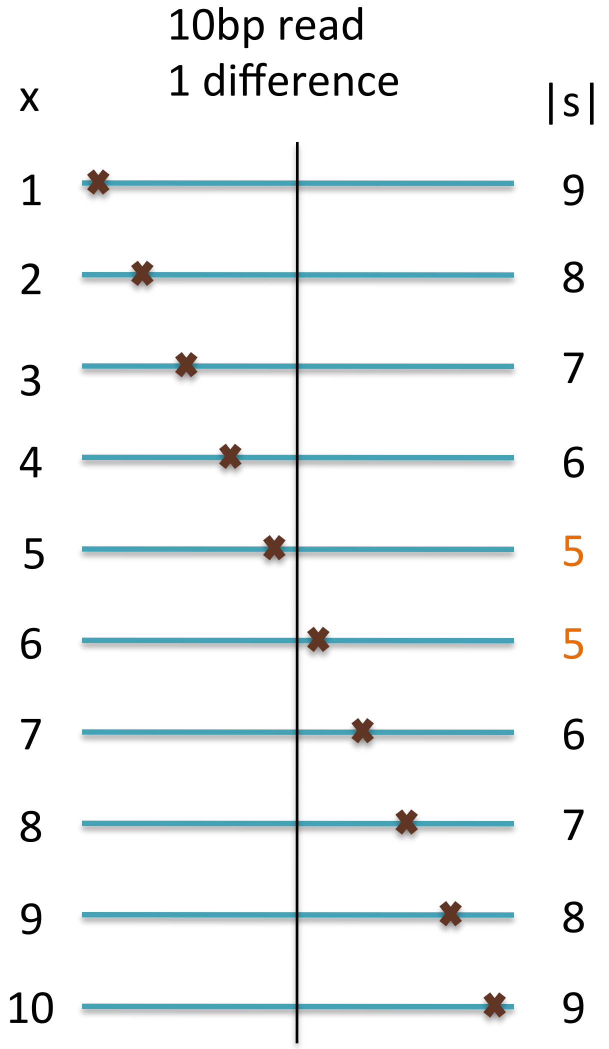
Theorem: An alignment of a sequence of length *m* with at most *k* differences ***must*** contain an exact match at least *s=m/(k+1)* bp long

(*Baeza-Yates* and Perleberg, 1996)

Proof: Pigeon hole principle

K=2 pigeons (differences) can't fill all K+1 pigeon holes (seeds)





- Search Algorithm
  - Use an index to rapidly find short exact alignments to seed longer in-exact alignments
    - RMAP, CloudBurst, …
  - Specificity of the seed depends on length
    => See Lecture 1
  - Length s seeds can also seed some lower quality alignments
    - Won't have perfect sensitivity, but avoids very short seeds

Hamming Distance Limitations

- Hamming distance measures the number of substitutions (SNPs)
  - Appropriate if that's all we expect/want to find
    - Illumina sequencing error model
    - Other highly constrained sequences

- What about insertions and deletions?
  - At best the indel will only slightly lower the score
  - At worst highly similar sequences will fail to align

Example Alignments

```
ACGTCTAG
||*****^
ACTCTAG-
```

- Hamming distance=5
  - 2 matches, 5 mismatches, 1 not aligned

Example Alignments

```
ACGTCTAG
^**|||||
-ACTCTAG
```

- Hamming distance = 2
  - 5 matches, 2 mismatches, 1 not aligned

Example Alignments

```
ACGTCTAG
||^|||||
AC-TCTAG
```

- Edit Distance = 1
  - – 7 matches, 0 mismatches, 1 not aligned

Global Alignment problem

- Given two sequences, S (length n) and T (length m), find the best end-to-end alignment of S and T.

  [When is this appropriate?]

- Edit distance (Levenshtein distance)
  - ***Minimum*** number of substitutions, insertions and deletions between 2 sequences.
  - Hamming distance is an upper bound on edit distance

- Definition
  - Let D(i,j) be the edit distance of the alignment of S[1...i] and T[1...j].
  - Edit distance of S and T (end-to-end) is D(n,m).

Edit Distance Example

AGCACACA → ACACACTA in 4 steps

AGCACACA → (1. change G to C)
ACCACACA → (2. delete C)
ACACACA → (3. change A to T)
ACACACT → (4. insert A after T)
ACACACTA → done

[Is this the best we can do?]

Edit Distance Example

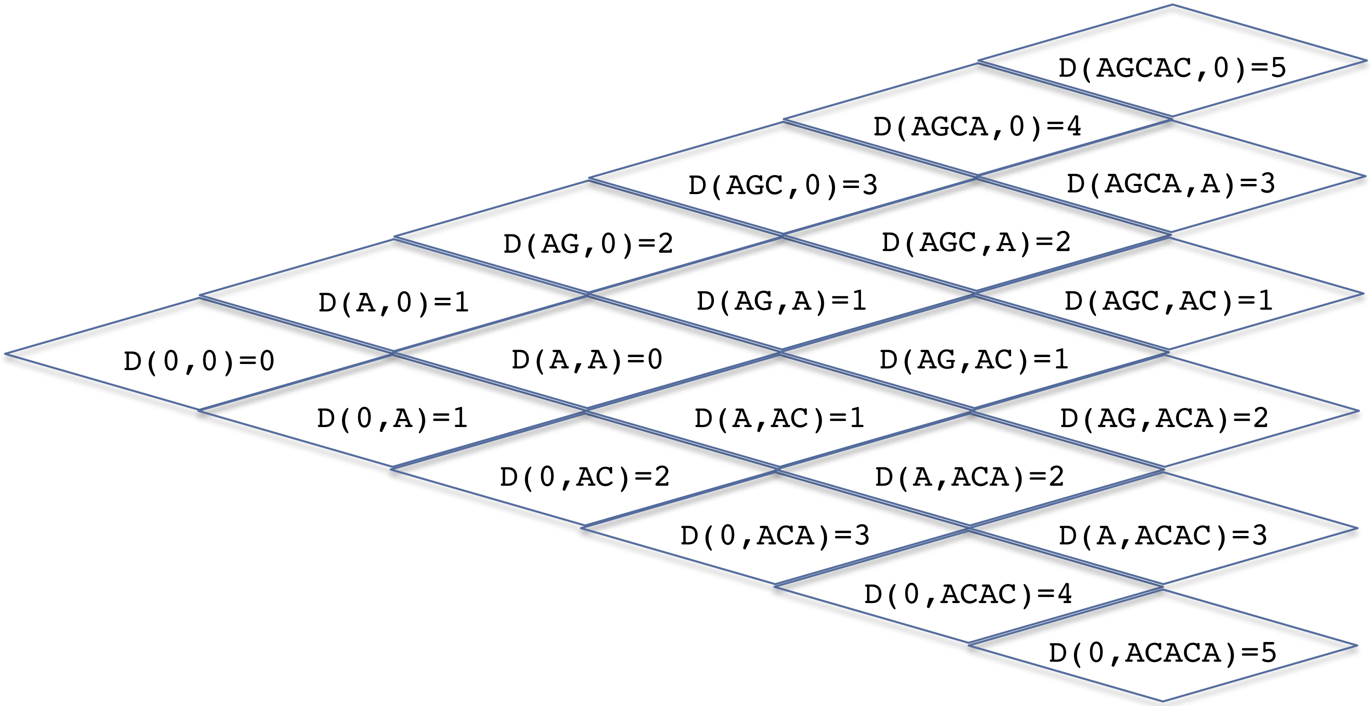
AGCACACA → ACACACTA in 3 steps

AGCACACA → (1. change G to C)
ACCACACA → (2. delete C)
ACACACA → (3. insert T after 3rd C)
ACACACTA → done

[Is this the best we can do?]

Edit Distance Observations

```
D(AGCACACA, ACACACTA) = ?
```



D(AGCAC,0)=5

D(AGCA,0)=4

D(AGC,0)=3 D(AGCA,A)=3

D(AG,0)=2 D(AGC,A)=2

D(A,0)=1 D(AG,A)=1 D(AGC,AC)=1

D(0,0)=0 D(A,A)=0 D(AG,AC)=1

D(0,A)=1 D(A,AC)=1 D(AG,ACA)=2

D(0,AC)=2 D(A,ACA)=2

D(0,ACA)=3 D(A,ACAC)=3

D(0,ACAC)=4

D(0,ACACA)=5

Base cases are easy, seems to be some relation between scores of neighboring cells

Reverse Engineering Edit Distance

```
D(AGCACACA, ACACACTA) = ?
```

Imagine we already have the optimal alignment of the strings, the last column can only be 1 of 3 options:

```
…M          …I          …D
…A          …-          …A
…A          …A          …-
```

The optimal alignment of last two columns is then 1 of 9 possibilities

```
…MM  …IM  …DM        …MI  …II  …DI        …MD  …ID  …DD
…CA  …-A  …CA        …A-  …--  …A-        …CA  …-A  …CA
…TA  …TA  …-A        …TA  …TA  …-A        …A-  …A-  …--
```

The optimal alignment of the last three columns is then 1 of 27 possibilities…

```
…M…      …I…      …D…
…X…      …-…      …X…
…Y…      …Y…      …-…
```

Eventually spell out every possible sequence of {I,M,D}

Reverse Engineering Edit Distance

```
D(AGCACACA, ACACACTA) = ?
```

Imagine we already have the optimal alignment of the strings, the last column can only be 1 of 3 options:

```
…M          …I          …D
…A          …-          …A
…A          …A          …-

D=2         D=2         D=3
```

If these are the edit distances of the three possible optimal alignments up to the last column, the overall optimal alignment must use option M and the final distance is 2+0=2

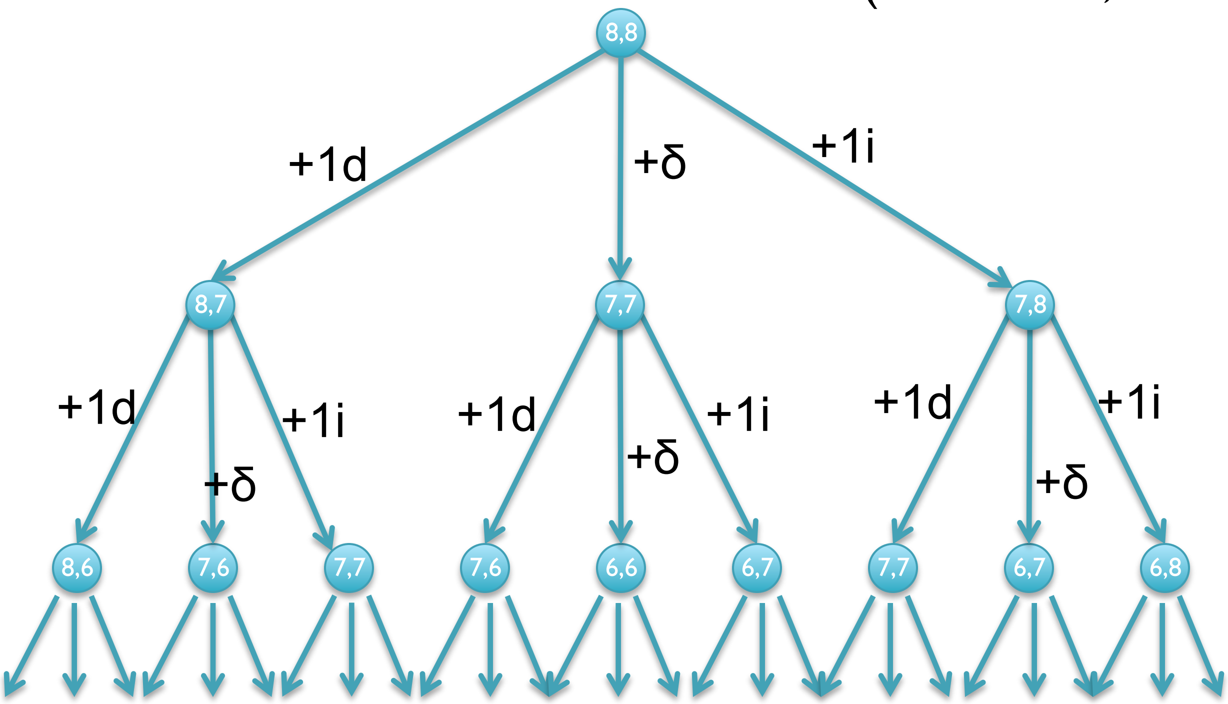
This suggests an overall structure to the computation:

> The edit distance of the overall alignment is computed from the edit distance of three possible alignments with the last column removed…. these values are themselves computed by removing the second to last column, and so forth down to the trivial base case of an empty string.

Recursive solution

- Computation of D is a recursive process.
  - At each step, we only allow matches, substitutions, and indels
  - D(i,j) in terms of D(i’,j’) for i’ ≤ i and j’ ≤ j.

D(AGCACACA, ACACACTA) = min{D(AGCACACA, ACACACT) + 1,
D(AGCACAC, ACACACTA) + 1,
D(AGCACAC, ACACACT) +δ(A, A)}



[What is the running time?]

Dynamic Programming

- We could code this as a recursive function call…

  …with an exponential number of function evaluations

- There are only (n+1)x(m+1) pairs i and j
  - We are evaluating D(i,j) multiple times

- Compute D(i,j) bottom up.
  - Start with smallest (i,j) = (1,1).
  - Store the intermediate results in a table.
    - Compute D(i,j) *after* D(i-1,j), D(i,j-1), and D(i-1,j-1)

Recurrence Relation for D

Find the edit distance (minimum number of operations to convert one string into another) in O(mn) time

- Base conditions:
  - D(i,0) = i, for all i = 0,...,n
  - D(0,j) = j, for all j = 0,...,m

- For i > 0, j > 0:

D(i,j) = min {

D(i-1,j) + 1, // align 0 chars from S, 1 from T

D(i,j-1) + 1, // align 1 chars from S, 0 from T

D(i-1,j-1) + δ(S(i),T(j)) // align 1+1 chars

}

[Why do we want the min?]

Dynamic Programming Matrix

		A	C	A	C	A	C	T	A
	0	1	2	3	4	5	6	7	8
A	1								
G	2								
C	3								
A	4								
C	5								
A	6								
C	7								
A	8								

[What does the initialization mean?]

Dynamic Programming Matrix

		A	C	A	C	A	C	T	A
	0	1	2	3	4	5	6	7	8
A	1	0							
G	2								
C	3								
A	4								
C	5								
A	6								
C	7								
A	8								

$D[A,A] = \min\{D[A,]+1,\ D[,A]+1,\ D[,]+\delta(A,A)\}$

Dynamic Programming Matrix

		A	C	A	C	A	C	T	A
	0	1	2	3	4	5	6	7	8
A	1	0	1						
G	2								
C	3								
A	4								
C	5								
A	6								
C	7								
A	8								

D[A,AC] = min{D[A,A]+1, D[,AC]+1, D[,A]+δ(A,C)}

Dynamic Programming Matrix

		A	C	A	C	A	C	T	A
	0	1	2	3	4	5	6	7	8
A	1	0	1	2					
G	2								
C	3								
A	4								
C	5								
A	6								
C	7								
A	8								

D[A,ACA] = min{D[A,AC]+1, D[,ACA]+1, D[,AC]+δ(A,A)}

Dynamic Programming Matrix

		A	C	A	C	A	C	T	A
	0	1	2	3	4	5	6	7	8
A	1	0	1	2	3	4	5	6	7
G	2								
C	3								
A	4								
C	5								
A	6								
C	7								
A	8								

D[A,ACACACTA] = 7

```
-------A
*******|
ACACACTA
```

[What about the other A?]

Dynamic Programming Matrix

		A	C	A	C	A	C	T	A
	0	1	2	3	4	5	6	7	8
A	1	0	1	2	3	4	5	6	7
G	2	1	1	2	3	4	5	6	7
C	3								
A	4								
C	5								
A	6								
C	7								
A	8								

D[AG,ACACACTA] = 7

```
----AG--
****|***
ACACACTA
```

Dynamic Programming Matrix

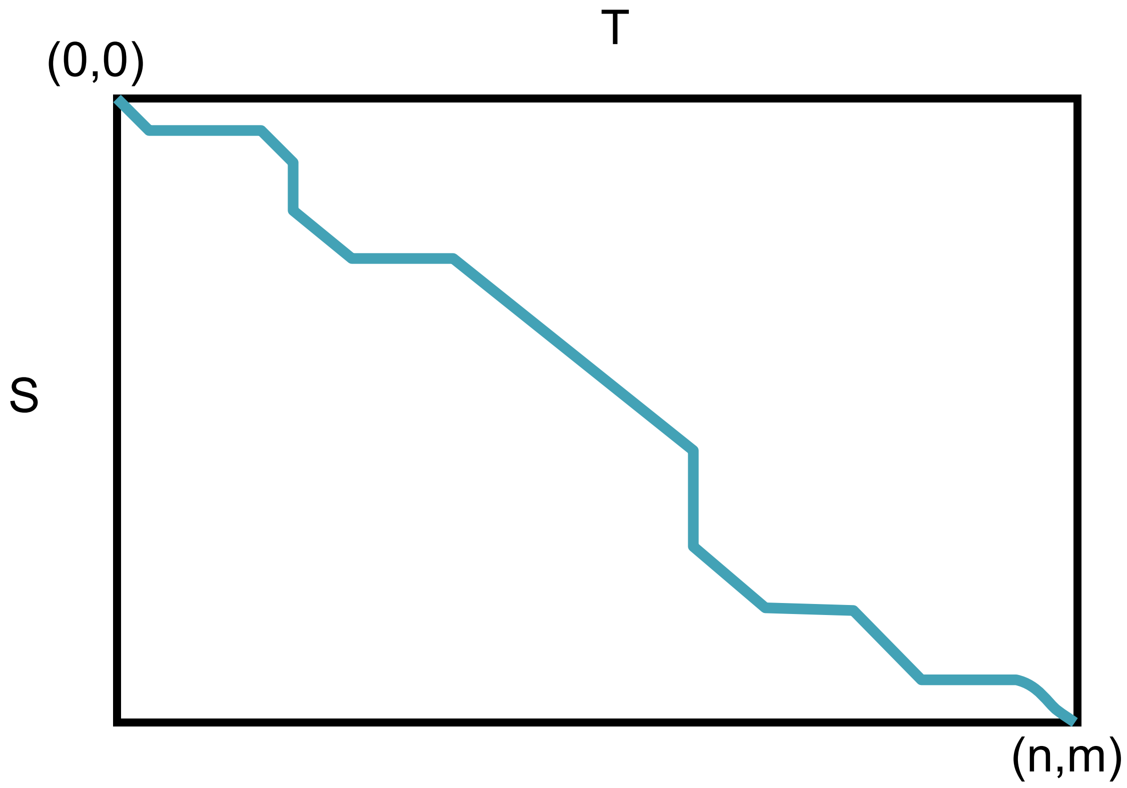
		A	C	A	C	A	C	T	A
	0	1	2	3	4	5	6	7	8
A	1	0	1	2	3	4	5	6	7
G	2	1	1	2	3	4	5	6	7
C	3	2	1	2	2	3	4	5	6
A	4	3	2	1	2	2	3	4	5
C	5	4	3	2	1	2	2	3	4
A	6	5	4	3	2	1	2	3	3
C	7	6	5	4	3	2	1	2	3
A	8	7	6	5	4	3	2	2	2

D[AGCACACA,ACACACTA] = 2

```
AGCACAC-A
|*|||||*|
A-CACACTA
```

[Can we do it any better?]

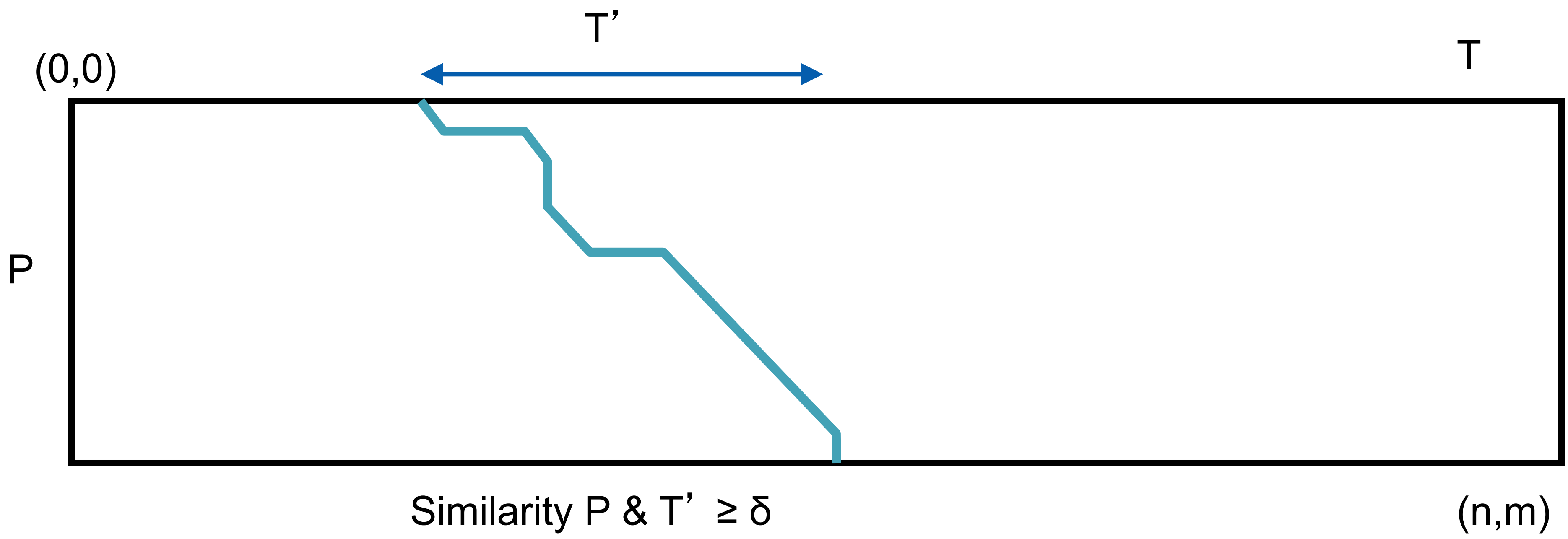
Global Alignment Schematic



- A high quality alignment will stay close to the diagonal
  - If we are only interested in high quality alignments, we can skip filling in cells that can't possibly lead to a high quality alignment
  - Find the global alignment with at most edit distance d: O(2dn)

Nathan Edwards

Searching for GATTACA

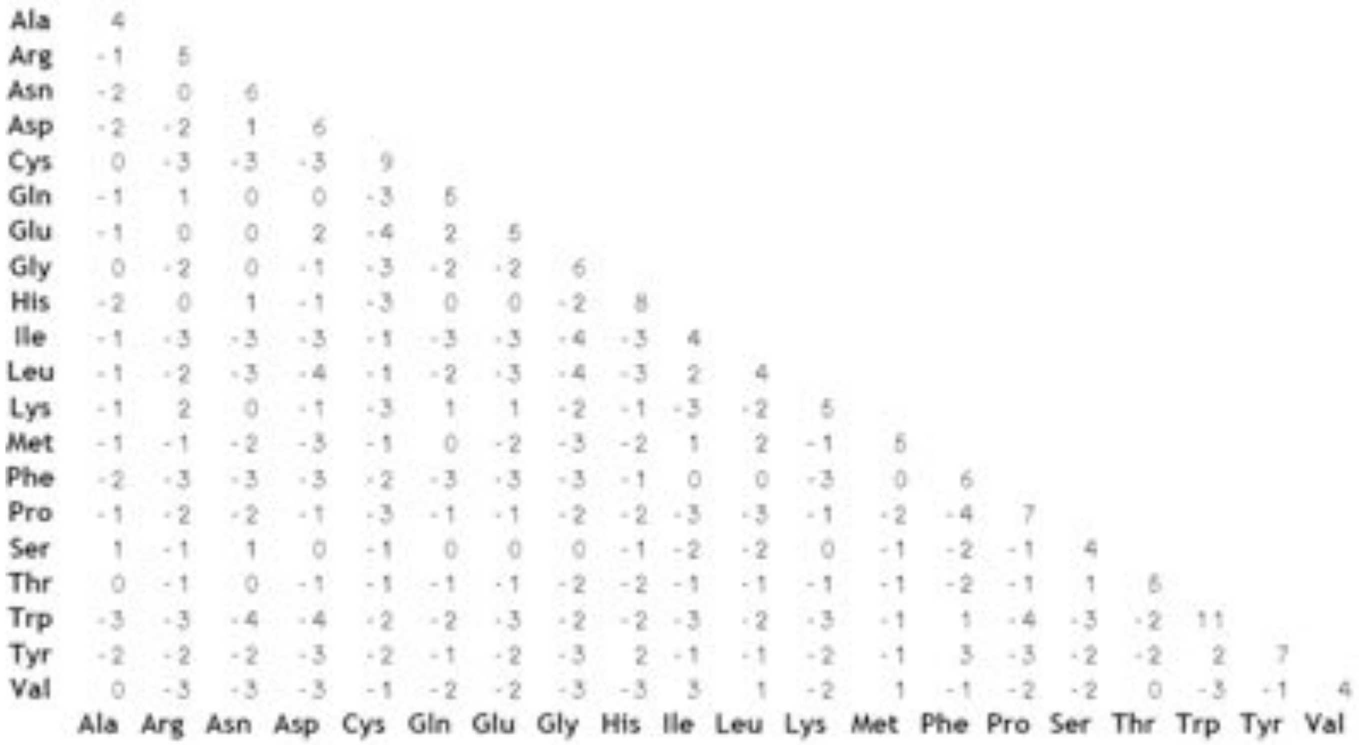


- Don’t “charge” for optimal alignment starting in cells (0,j)
  - Base conds: D(0,j) = 0, D(i,0) = $\Sigma_{k \leq i}$ s(S(k), ‘-’)
- Don’t “charge” for ending alignment at end of P (but not necc. T)
  - Find cell (n,j) with edit distance ≤ δ

Nathan Edwards

Sequence Similarity

- Similarity score generalizes edit distance
  - Certain mutations are much more likely than others
    - Hydrophilic -> Hydrophillic much more likely than Hydrophillic -> Hydrophobic
  - BLOSSUM62
    - Empirically measure substitution rates among proteins that are 62% identical
    - Positive score: more likely than chance, Negative score: less likely



	Ala	Arg	Asn	Asp	Cys	Gln	Glu	Gly	His	Ile	Leu	Lys	Met	Phe	Pro	Ser	Thr	Trp	Tyr	Val
Ala	4																			
Arg	-1	5																		
Asn	-2	0	6																	
Asp	-2	-2	1	6																
Cys	0	-3	-3	-3	9															
Gln	-1	1	0	0	-3	5														
Glu	-1	0	0	2	-4	2	5													
Gly	0	-2	0	-1	-3	-2	-2	6												
His	-2	0	1	-1	-3	0	0	-2	8											
Ile	-1	-3	-3	-3	-1	-3	-3	-4	-3	4										
Leu	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4									
Lys	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5								
Met	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5							
Phe	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	6						
Pro	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	7					
Ser	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	4				
Thr	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5			
Trp	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	11		
Tyr	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7	
Val	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	4

Edit Distance and Global Similarity

$$D(i,j) = \min \{ \\ D(i-1,j) + 1, \\ D(i,j-1) + 1, \\ D(i-1,j-1) + \delta(S(i),T(j)) \\ \}$$

s = 4x4 or 20x20 scoring matrix

$$S(i,j) = \max \{ \\ S(i-1,j) + 1, \\ S(i,j-1) + 1, \\ S(i-1,j-1) + s(S(i),T(j)) \\ \}$$

[Why max?]

Local vs. Global Alignment

- The Global Alignment Problem tries to find the best path between vertices *(0,0)* and (*n*,*m*) in the edit graph.

- The Local Alignment Problem tries to find the best path among paths between **arbitrary vertices** (*i,j*) and (*i'*, *j'*) in the edit graph.

[How many (i,j) x (i',j') pairs are there?]

Local vs. Global Alignment (cont’ d)

- Global Alignment

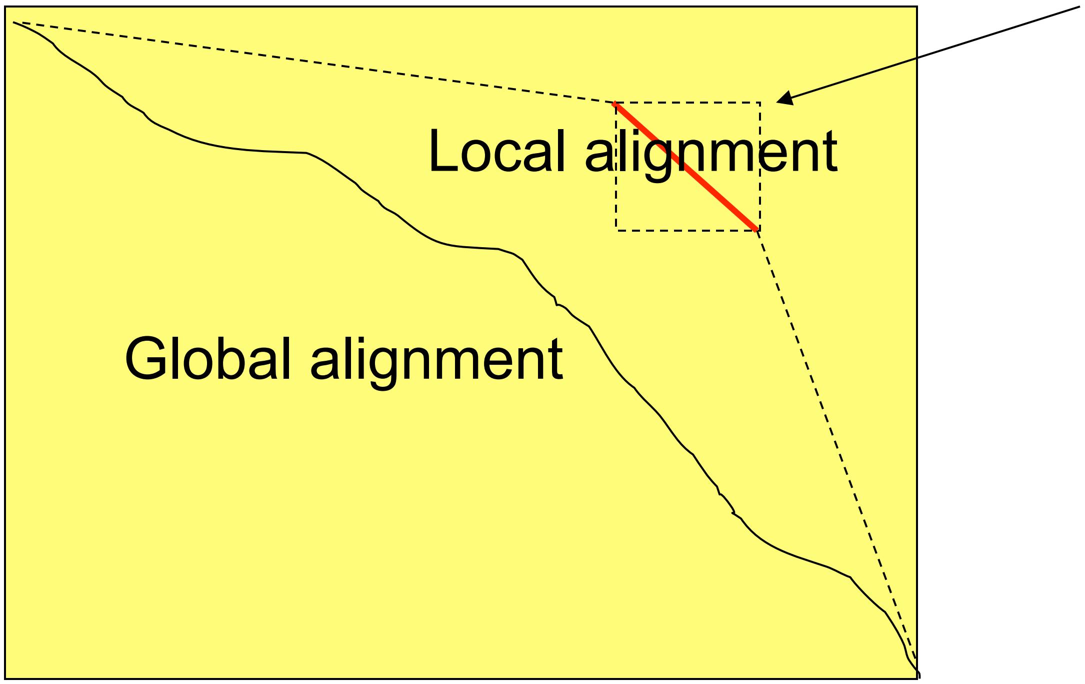
```
--T—-CC-C-AGT—-TATGT-CAGGGGACACG—A-GCATGCAGA-GAC
  |  || |  ||  | | | |||    || | | |  | ||||   |
AATTGCCGCC-GTCGT-T-TTCAG----CA-GTTATG—T-CAGAT--C
```

- Local Alignment—better alignment to find conserved segment

```
                  tccCAGTTATGTCAGgggacacgagcatgcagagac
                     ||||||||||||
aattgccgccgtcgttttcagCAGTTATGTCAGatc
```

bioalgorithms.info

Local Alignment: Example



Compute a “mini” Global Alignment to get Local

The Local Alignment Recurrence

- The largest value of $s_{i,j}$ over the whole edit graph is the score of the best local alignment.

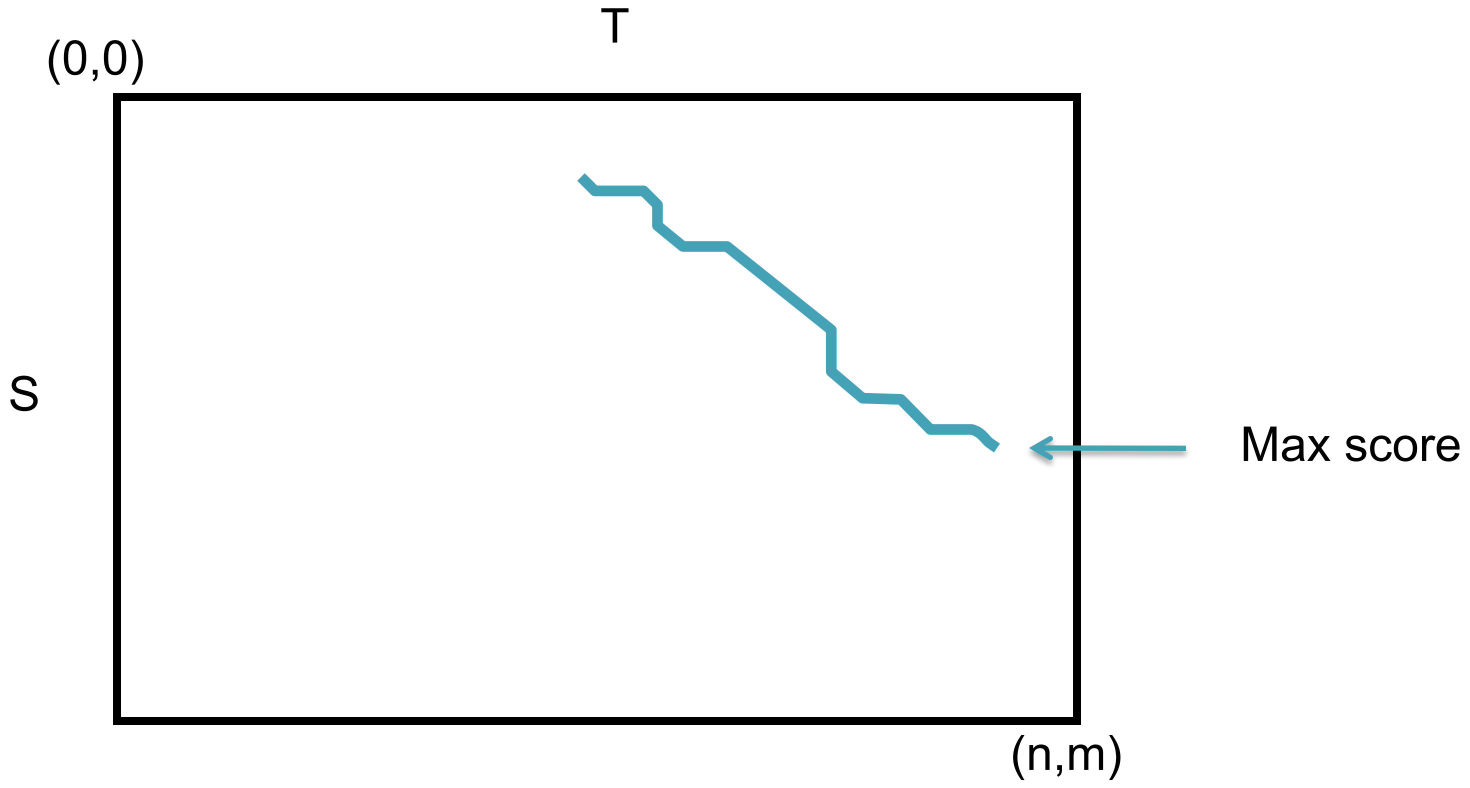
- The recurrence:

$$s_{i,j} = max \begin{cases} 0 \\ s_{i-1,j-1} + \delta\,(v_i,\, w_j) \\ s_{\,i-1,j} + \delta\,(v_i,\, -) \\ s_{\,i,j-1} + \delta\,(-,\, w_j) \end{cases}$$

Power of ZERO: there is only this change from the original recurrence of a Global Alignment - since there is only one “free ride” edge entering into every vertex

bioalgorithms.info

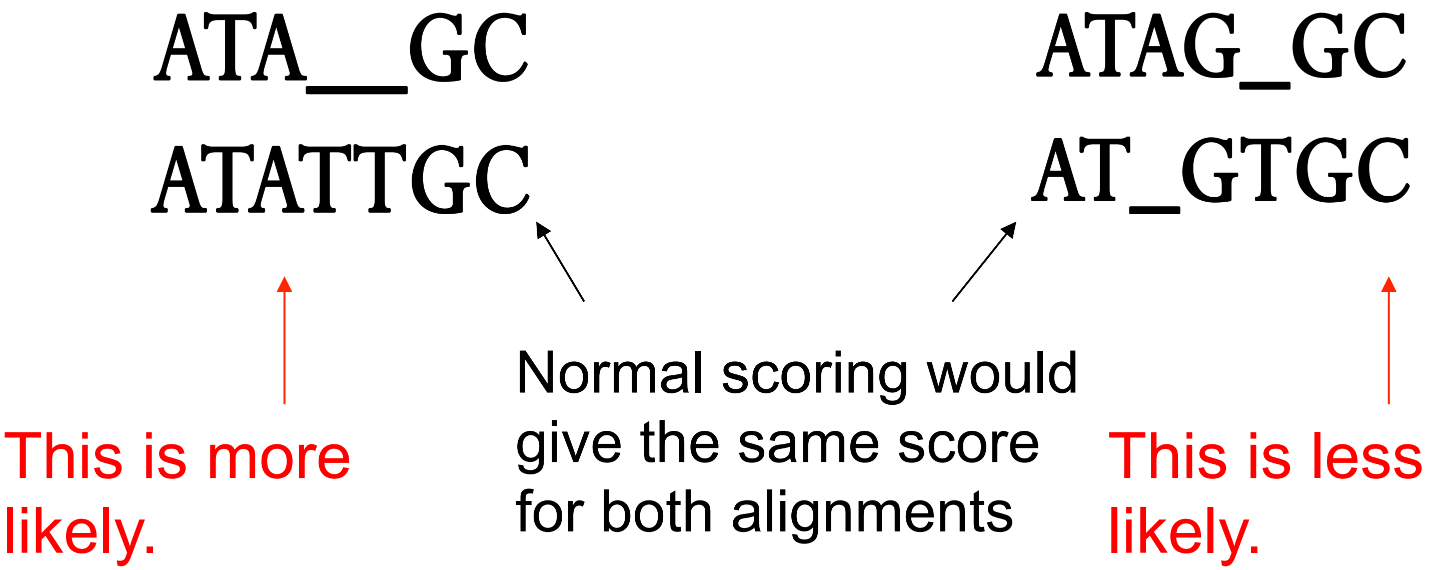
Local Alignment Schematic



Nathan Edwards

Affine Gap Penalties

- In nature, a series of *k* indels often come as a single event rather than a series of *k* single nucleotide events:



ATA__GC
ATATTGC

ATAG_GC
AT_GTGC

This is more likely.

Normal scoring would give the same score for both alignments

This is less likely.

bioalgorithms.info

Accounting for Gaps

- *Gaps*- contiguous sequence of spaces in one of the rows

- Score for a gap of length x is: $-(\rho + \sigma x)$
  where ρ >0 is the gap opening penalty
  ρ will be large relative to gap extension penalty σ

  - Gap of length 1: $-(\rho + \sigma) = -6$
  - Gap of length 2: $-(\rho + \sigma 2) = -7$
  - Gap of length 3: $-(\rho + \sigma 3) = -8$

- Smith-Waterman-Gotoh incorporates affine gap penalties without increasing the running time O(mn)

Break

Basic Local Alignment Search Tool

- Rapidly compare a sequence Q to a database to find all sequences in the database with an score above some cutoff S.
  - Which protein is most similar to a newly sequenced one?
  - Where does this sequence of DNA originate?

- Speed achieved by using a procedure that typically finds "most" matches with scores > S.
  - Tradeoff between sensitivity and specificity/speed
    - Sensitivity – ability to find all related sequences
    - Specificity – ability to reject unrelated sequences

(Altschul et al. 1990)

Seed and Extend

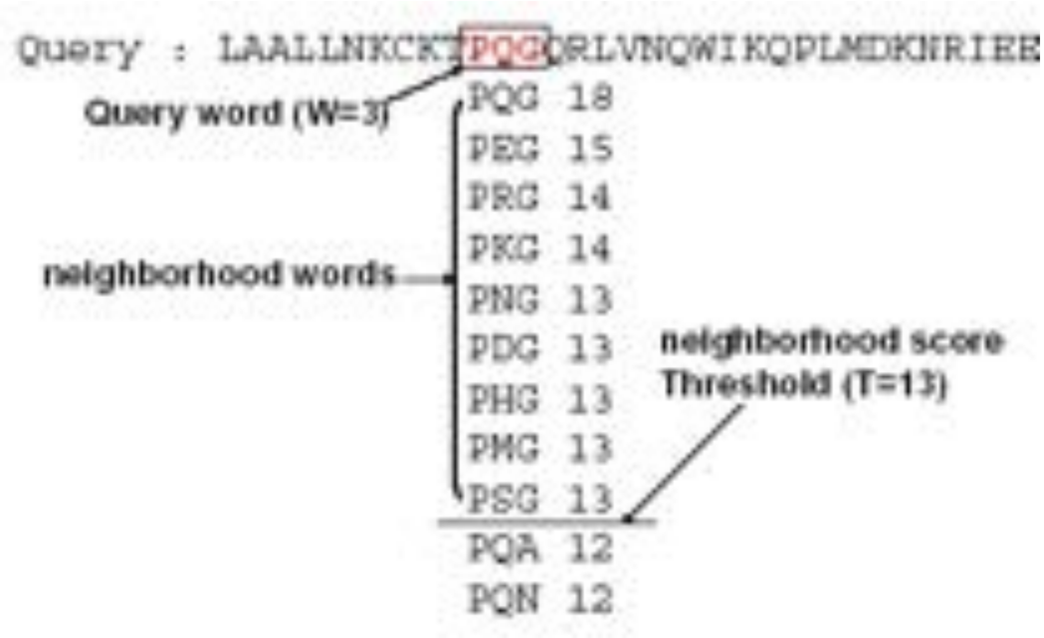
```
FAKDFLAGGVAAAISKTAVAPIERVKLLLQVQHASKQITADKQYKGIIDCVVRIPKEQGV
F  D  +GG AAA+SKTAVAPIERVKLLLQVQ ASK I  DK+YKGI+D ++R+PKEQGV
FLIDLASGGTAAAVSKTAVAPIERVKLLLQVQDASKAIAVDKRYKGIMDVLIRVPKEQGV
```

- Homologous sequence are likely to contain a short high scoring word pair, a seed.
  - Unlike *Baeza-Yates*, BLAST *doesn't* make explicit guarantees

- BLAST then tries to extend high scoring word pairs to compute maximal high scoring segment pairs (HSPs).
  - Heuristic algorithm but evaluates the result statistically.

BLAST - Algorithm -

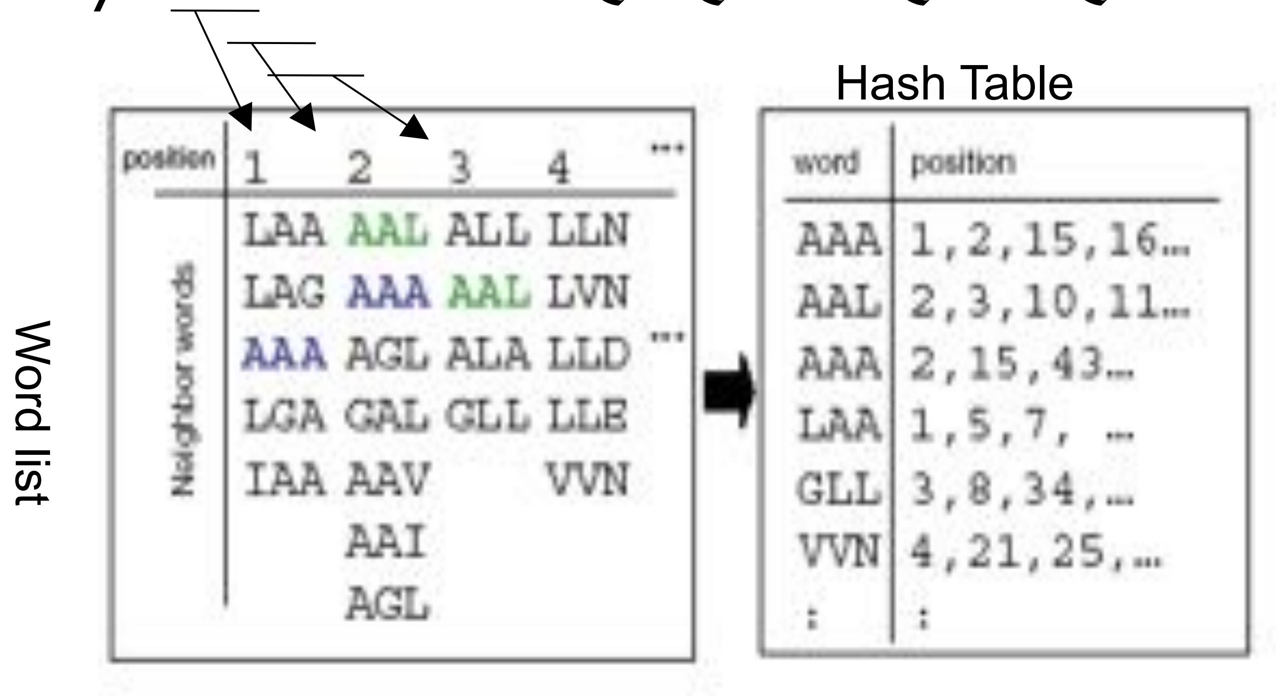
- Step 1: Preprocess Query
  Compile the short-high scoring word list from query.
  The length of query word, w, is 3 for protein scoring
  Threshold T is 13



BLAST - Algorithm -

- Step 2: Construct Query Word Hash Table

Query: LAALLNKCKTPQGQRLVNQWIKQPLMD



Word list

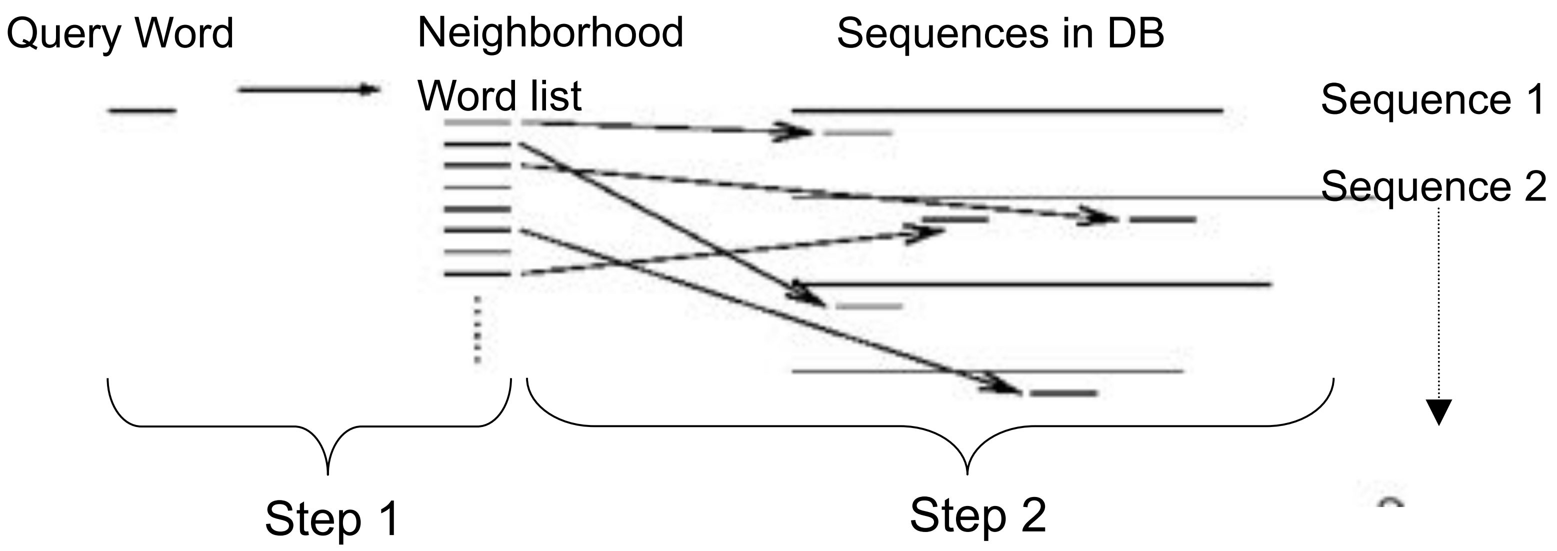
position	1	2	3	4	...
Neighbor words	LAA	AAL	ALL	LLN	
	LAG	AAA	AAL	LVN	
	AAA	AGL	ALA	LLD	...
	LGA	GAL	GLL	LLE	
	IAA	AAV		VVN	
		AAI			
		AGL			

Hash Table

word	position
AAA	1,2,15,16...
AAL	2,3,10,11...
AAA	2,15,43...
LAA	1,5,7, ...
GLL	3,8,34,...
VVN	4,21,25,...
:	:

BLAST - Algorithm -

- Step 3: Scanning DB
  Identify all exact matches with DB sequences



BLAST - Algorithm -

- Step 4 (Search optimal alignment)
  For each hit-word, extend ungapped alignments in both directions.
  Let S be a score of hit-word

- Step 5 (Evaluate the alignment statistically)
  Stop extension when E-value (depending on score S) become less than threshold. The extended match is called High Scoring Segment Pair.

E-value = the number of HSPs having score S (or higher) expected to occur by chance.
→ Smaller E-value, more significant in statistics
Bigger E-value , by chance

E[# occurrences of a string of length m in reference of length L] ~ $L/4^m$

BLAST E-values

The expected number of HSPs with the score at least S is :

$$E = K*n*m*e^{-\lambda S}$$

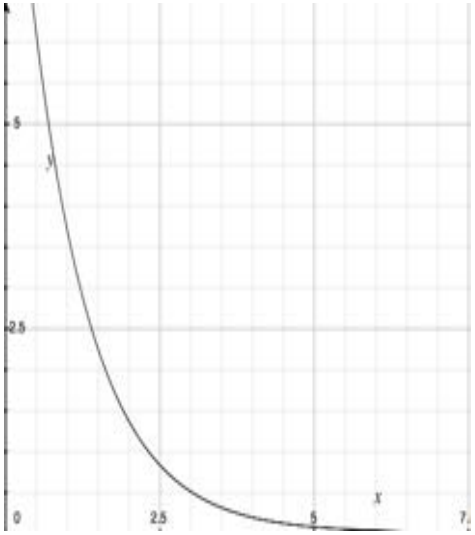
K, λ are constant depending on model

n, m are the length of query and sequence

The probability of finding at least one such HSP is:

$$P = 1 - e^{E}$$

→ If a word is hit by chance (E-value is bigger), P become smaller.



The distribution of Smith-Waterman local alignment scores between two random sequences follows the Gumbel extreme value distribution

Parameters

- Larger values of w increases the number of neighborhood words, but decreases the number of chance matches in the database.
  - Increasing w decreases sensitivity.

- Larger values of T decrease the overall execution time, but increase the chance of missing a MSP having score ≥ S.
  - Increases T decreases the sensitivity

- Larger values of S increase the specificity. The value of S is affected by changes in the expectation value parameter.

Very Similar Sequences

```
Query: HBA_HUMAN Hemoglobin alpha subunit
Sbjct: HBB_HUMAN Hemoglobin beta subunit

Score =  114 bits (285),  Expect = 1e-26
Identities = 61/145 (42%), Positives = 86/145 (59%), Gaps = 8/145 (5%)

Query  2   LSPADKTNVKAAWGKVGAHAGEYGAEALERMFLSFPTTKTYFPHF------DLSHGSAQV 55
           L+P +K+ V A WGKV  +  E G EAL R+ + +P T+ +F  F      D   G+ +V
Sbjct  3   LTPEEKSAVTALWGKV--NVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPKV 60

Query  56  KGHGKKVADALTNAVAHVDDMPNALSALSDLHAHKLRVDPVNFKLLSHCLLVTLAAHLPA 115
           K HGKKV  A ++ +AH+D++    + LS+LH  KL VDP NF+LL + L+  LA H
Sbjct  61  KAHGKKVLGAFSDGLAHLDNLKGTFATLSELHCDKLHVDPENFRLLGNVLVCVLAHHFGK 120

Query  116 EFTPAVHASLDKFLASVSTVLTSKY 140
           EFTP V A+  K +A V+  L  KY
Sbjct  121 EFTPPVQAAYQKVVAGVANALAHKY 145
```

Quite Similar Sequences

```
Query: HBA_HUMAN Hemoglobin alpha subunit
Sbjct: MYG_HUMAN Myoglobin

Score = 51.2 bits (121), Expect = 1e-07,
Identities = 38/146 (26%), Positives = 58/146 (39%), Gaps = 6/146 (4%)

Query  2   LSPADKTNVKAAWGKVGAHAGEYGAEALERMFLSFPTTKTYFPHF------DLSHGSAQV  55
           LS  +   V   WGKV A    +G E L R+F   P T   F  F      D    S  +
Sbjct  3   LSDGEWQLVLNVWGKVEADIPGHGQEVLIRLFKGHPETLEKFDKFKHLKSEDEMKASEDL  62

Query  56 KGHGKKVADALTNAVAHVDDMPNALSALSDLHAHKLRVDPVNFKLLSHCLLVTLAAHLPA  115
           K HG  V  AL   +         +  L+  HA K ++     + +S C++  L +  P
Sbjct  63 KKHGATVLTALGGILKKKGHHEAEIKPLAQSHATKHKIPVKYLEFISECIIQVLQSKHPG  122

Query  116 EFTPAVHASLDKFLASVSTVLTSKYR  141
            +F      +++K L      + S Y+
Sbjct  123 DFGADAQGAMNKALELFRKDMASNYK  148
```

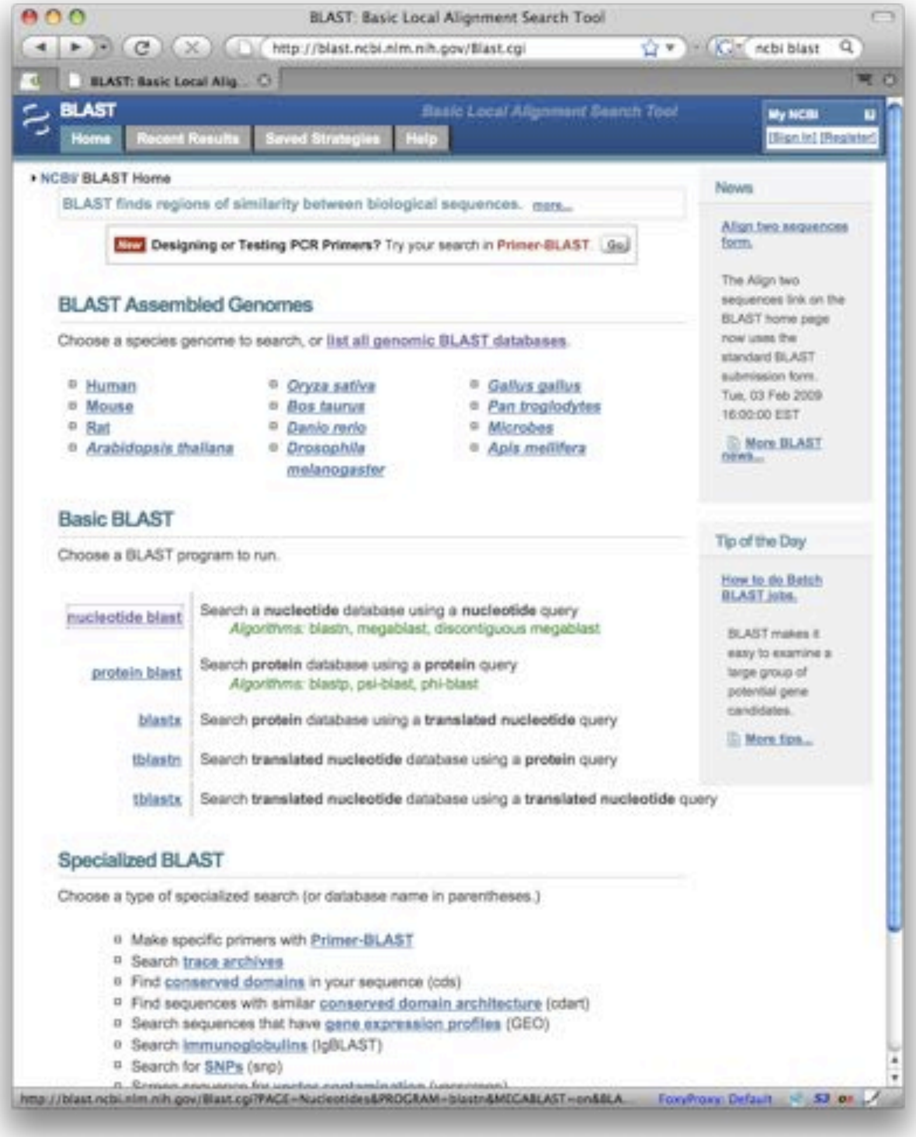
Not similar sequences

```
Query: HBA_HUMAN Hemoglobin alpha subunit
Sbjct: SPAC869.02c [Schizosaccharomyces pombe]

 Score = 33.1 bits (74),  Expect = 0.24
 Identities = 27/95 (28%), Positives = 50/95 (52%), Gaps = 10/95 (10%)

Query  30  ERMFLSFPTTKTYFPHFDLSHGSAQVKGHGKKVADALTNAVAHVDDMPNALSALSDLHAH  89
           ++M  ++P      P+F+ +H  +      + +A AL N   ++DD+  +LSA  D
Sbjct  59  QKMLGNYPEV---LPYFNKAHQISL--SQPRILAFALLNYAKNIDDL-TSLSAFMDQIVV 112

Query  90  K---LRVDPVNFKLLSHCLLVTLAAHLPAEF-TPA  120
           K   L++   ++ ++ HCLL T+   LP++  TPA
Sbjct  113 KHVGLQIKAEHYPIVGHCLLSTMQELLPSDVATPA  147
```

Blast Versions

Program	Database	Query
BLASTN	Nucleotide	Nucleotide
BLASTP	Protein	Protein
BLASTX	Protein	Nucleotide translated in to protein
TBLASTN	Nucleotide translated in to protein	Protein
TBLASTX	Nucleotide translated in to protein	Nucleotide translated in to protein

NCBI Blast


BLAST: Basic Local Alignment Search Tool
http://blast.ncbi.nlm.nih.gov/Blast.cgi
ncbi blast
BLAST: Basic Local Alig...
BLAST
Basic Local Alignment Search Tool
Home
Recent Results
Saved Strategies
Help
My NCBI
[Sign In] [Register]

NCBI/ BLAST Home

BLAST finds regions of similarity between biological sequences. more...

New Designing or Testing PCR Primers? Try your search in Primer-BLAST Go

BLAST Assembled Genomes

Choose a species genome to search, or list all genomic BLAST databases.

- Human
- Mouse
- Rat
- *Arabidopsis thaliana*
- *Oryza sativa*
- *Bos taurus*
- *Danio rerio*
- *Drosophila melanogaster*
- *Gallus gallus*
- *Pan troglodytes*
- *Microbes*
- *Apis mellifera*

Basic BLAST

Choose a BLAST program to run.

nucleotide blast	Search a **nucleotide** database using a **nucleotide** query *Algorithms:* blastn, megablast, discontiguous megablast
protein blast	Search **protein** database using a **protein** query *Algorithms:* blastp, psi-blast, phi-blast
blastx	Search **protein** database using a **translated nucleotide** query
tblastn	Search **translated nucleotide** database using a **protein** query
tblastx	Search **translated nucleotide** database using a **translated nucleotide** query

Specialized BLAST

Choose a type of specialized search (or database name in parentheses.)

- Make specific primers with Primer-BLAST
- Search trace archives
- Find conserved domains in your sequence (cds)
- Find sequences with similar conserved domain architecture (cdart)
- Search sequences that have gene expression profiles (GEO)
- Search immunoglobulins (IgBLAST)
- Search for SNPs (snp)
- Screen sequence for vector contamination (vecscreen)

News

Align two sequences form.

The Align two sequences link on the BLAST home page now uses the standard BLAST submission form. Tue, 03 Feb 2009 16:00:00 EST

More BLAST news...

Tip of the Day

How to do Batch BLAST jobs.

BLAST makes it easy to examine a large group of potential gene candidates.

More tips...

http://blast.ncbi.nlm.nih.gov/Blast.cgi?PAGE=Nucleotides&PROGRAM=blastn&MEGABLAST=on&BLA...

- Nucleotide Databases
  - nr: All Genbank
  - refseq: Reference organisms
  - wgs: All reads
- Protein Databases
  - nr: All non-redundant sequences
  - Refseq: Reference proteins

BLAST Exercise

```
>whoami
TTGATGCAGGTATCTGCGACTGAGACAATATGCA
ACAGTTGAATGAATCATAATGGAATGTGCACTCT
AACCAGCCAATTTGATGCTGGCTGCAGAGATGC
AAGATCAAGAGGTGACACCTGCTCTGAAGAAAG
CACAGTTGAACTGCTGGATCTGCAACTACAGCA
GGCACTCCAGGCACCAAGACAACATCTTTTACA
CCAGCAAACATGTGGATTGATATCTCCTAACAGC
AGTGATTAACAGAGACGACTGCAGGATTTGCTTC
CACAAACAAAAT
```

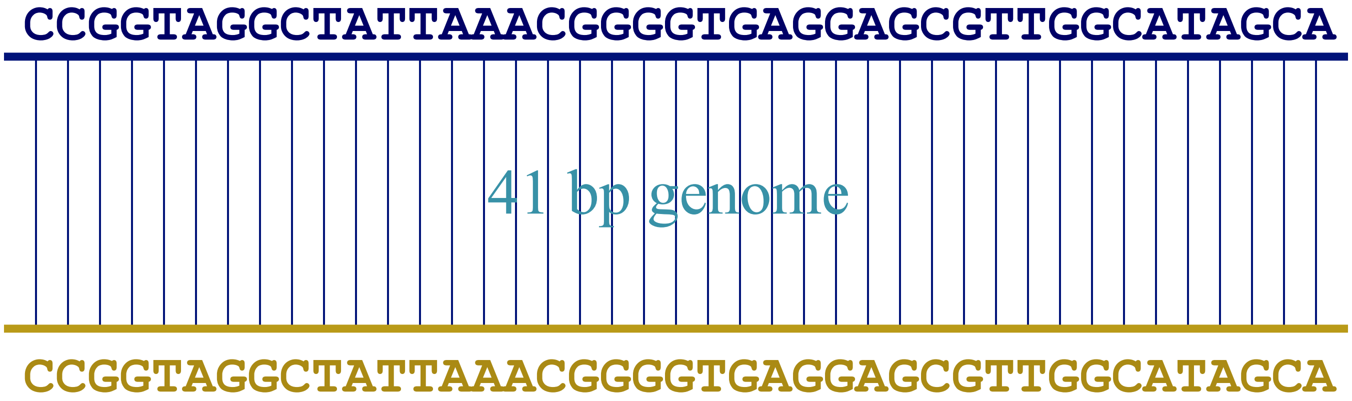
Whole Genome Alignment with MUMmer

Slides Courtesy of Adam M. Phillippy

amp@umics.umd.edu

Goal of WGA

- For two genomes, *A* and *B*, find a mapping from each position in *A* to its corresponding position in *B*



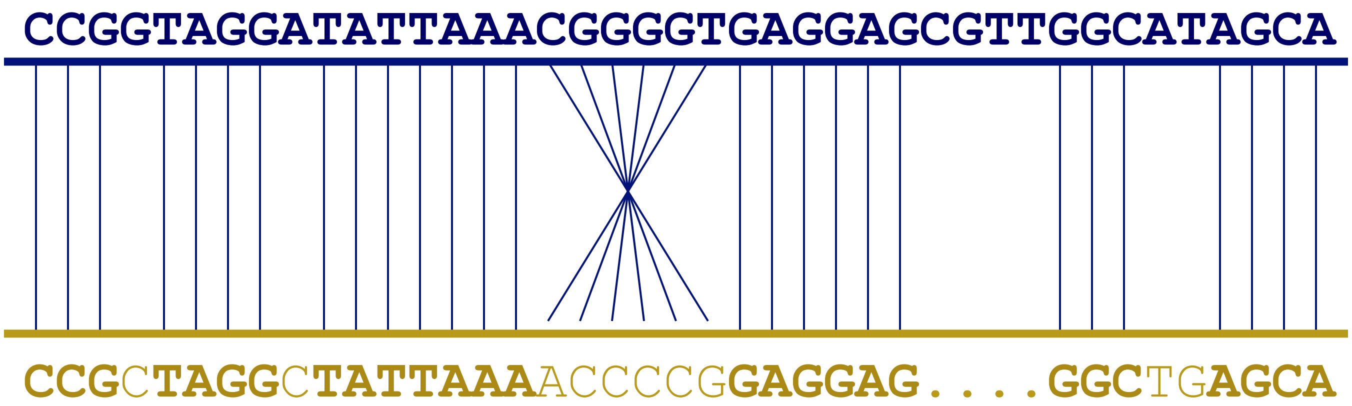
CCGGTAGGCTATTAAACGGGGTGAGGAGCGTTGGCATAGCA

41 bp genome

CCGGTAGGCTATTAAACGGGGTGAGGAGCGTTGGCATAGCA

Not so fast...

- Genome *A* may have insertions, deletions, translocations, inversions, duplications or SNPs with respect to *B* (sometimes all of the above)



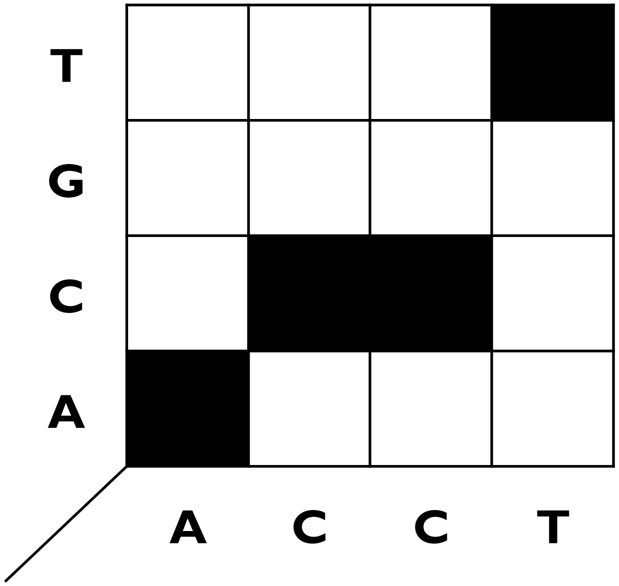
CCGGTAGGATATTAAACGGGGTGAGGAGCGTTGGCATAGCA

CCGCTAGGCTATTAAAACCCCGGAGGAG....GGCTGAGCA

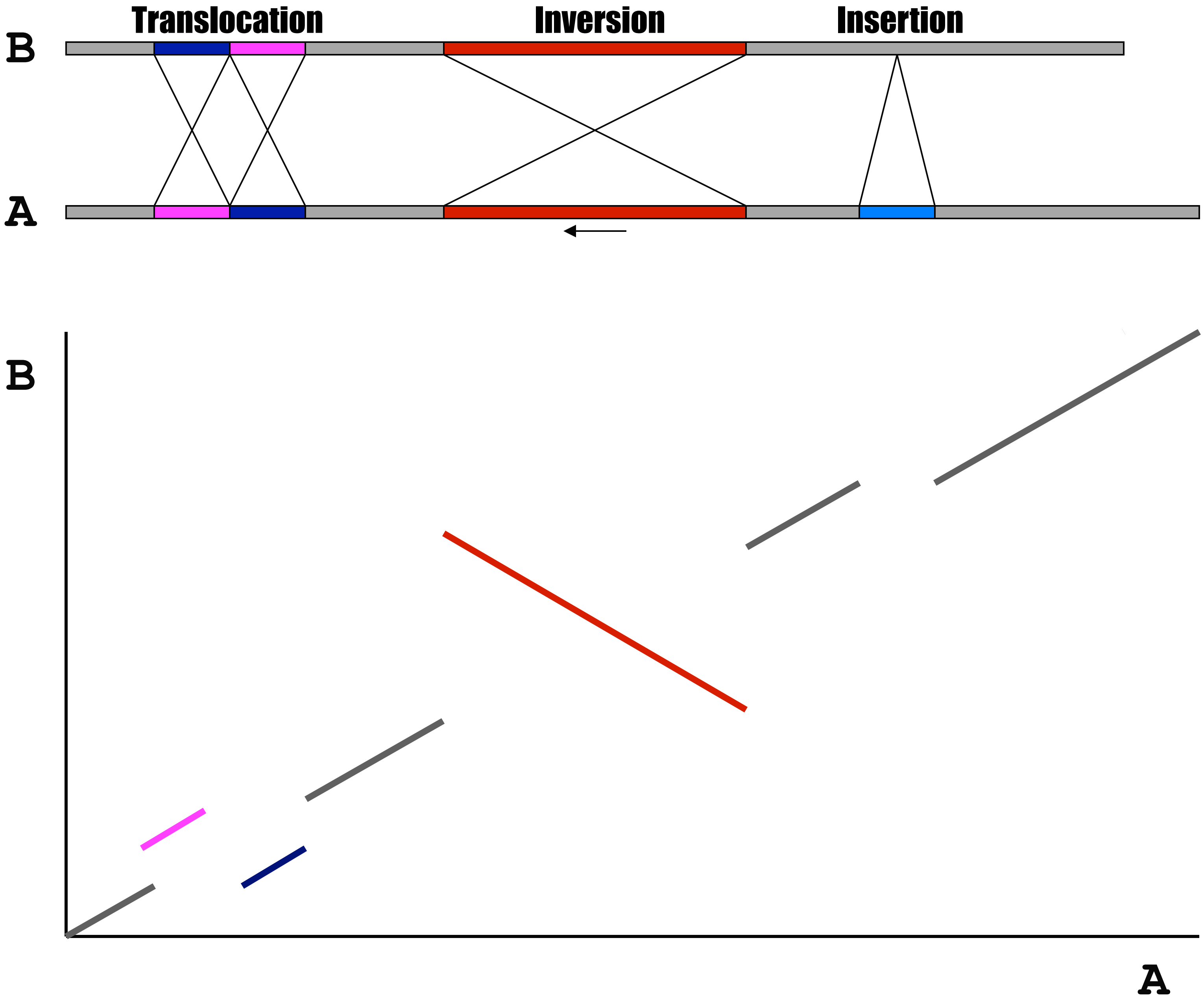
WGA visualization

- How can we visualize *whole* genome alignments?

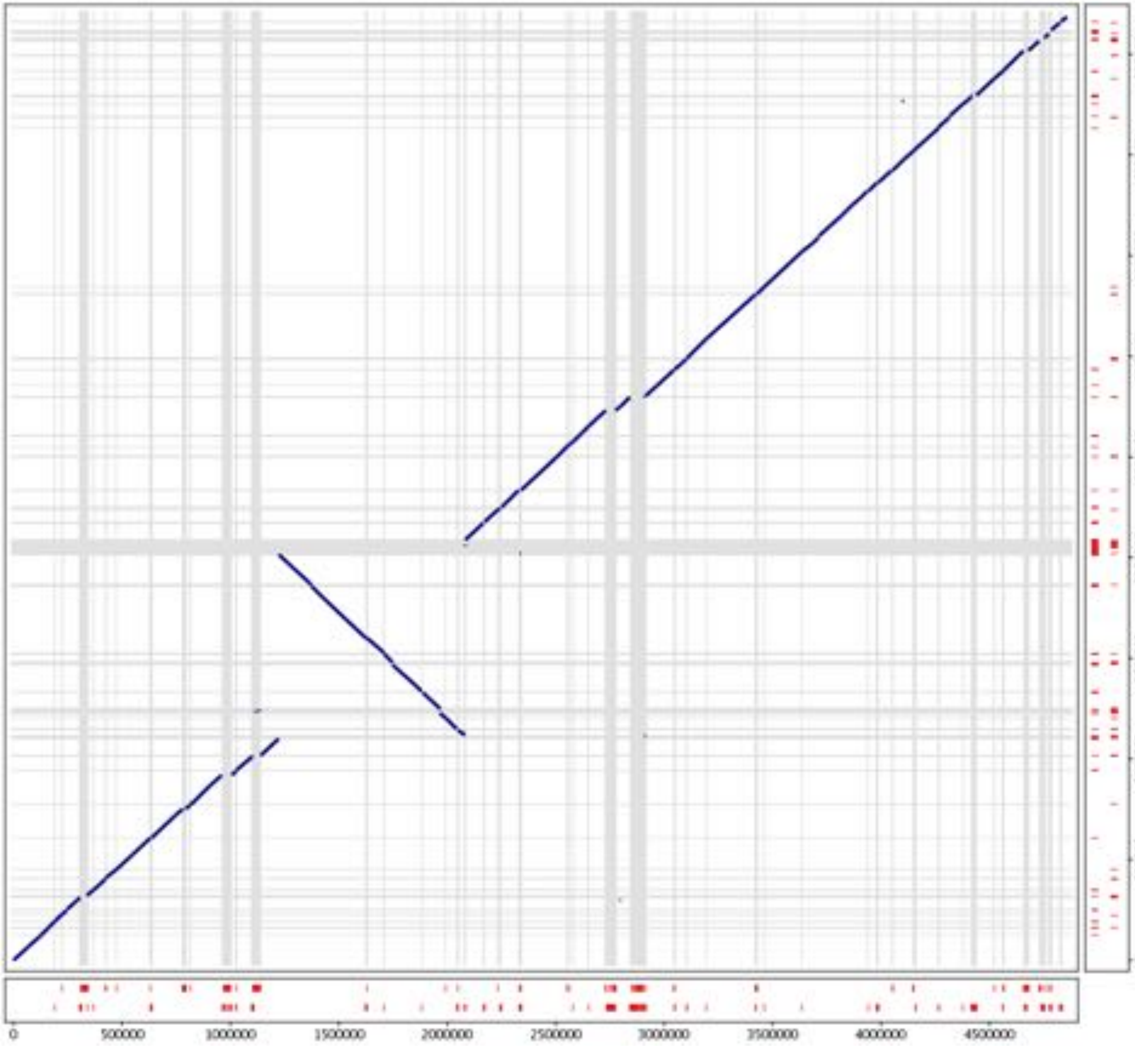
- With an alignment dot plot
  - *N* x *M* matrix
    - Let *i* = position in genome *A*
    - Let *j* = position in genome *B*
    - Fill cell *(i,j)* if A_i shows similarity to B_j



  - A perfect alignment between *A* and *B* would completely fill the positive diagonal



http://mummer.sourceforge.net/manual/AlignmentTypes.pdf


0
500000
1000000
1500000
2000000
2500000
3000000
3500000
4000000
4500000
0
500000
1000000
1500000
2000000
2500000
3000000
3500000
4000000
4500000

MUMmer

- Maximal Unique Matcher (MUM)
  - match
    - exact match of a minimum length
  - maximal
    - cannot be extended in either direction without a mismatch
  - *unique*
    - occurs only once in both sequences (MUM)
    - occurs only once in a single sequence (MAM)
    - occurs one or more times in either sequence (MEM)

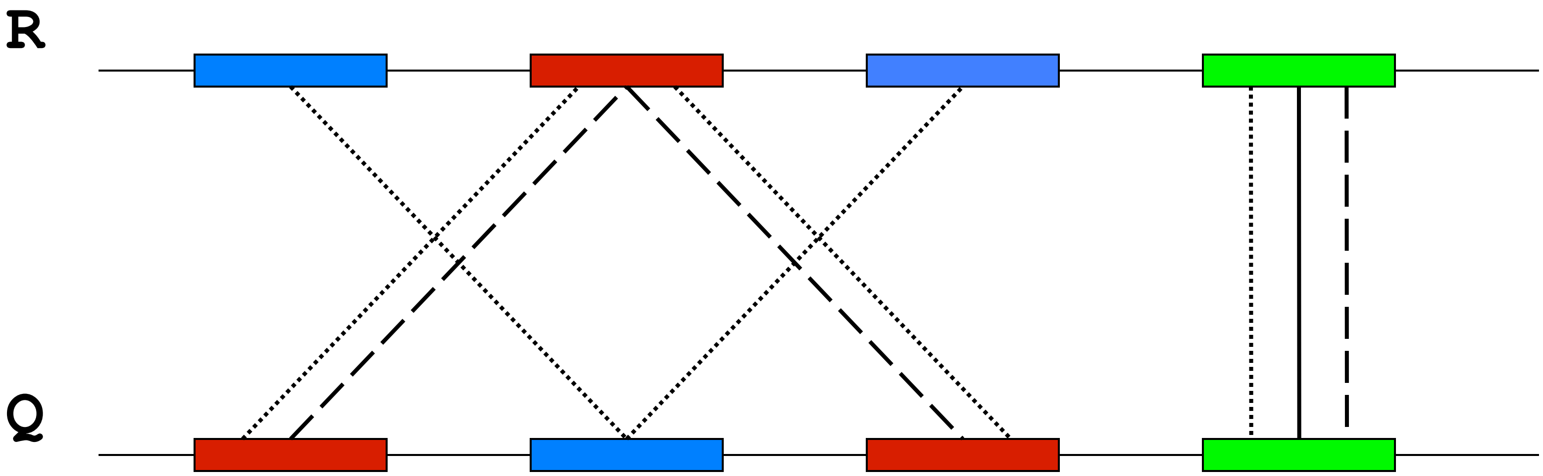
Fee Fi Fo Fum,

is it a MAM, MEM or MUM?

MUM : maximal unique match ────────

MAM : maximal almost-unique match – – – – – – – –

MEM : maximal exact match ················



Seed and Extend

- How can we make MUMs **BIGGER?**

1. Find MUMs
   - using a suffix tree
2. Cluster MUMs
   - using size, gap and distance parameters
3. Extend clusters
   - using modified Smith-Waterman algorithm

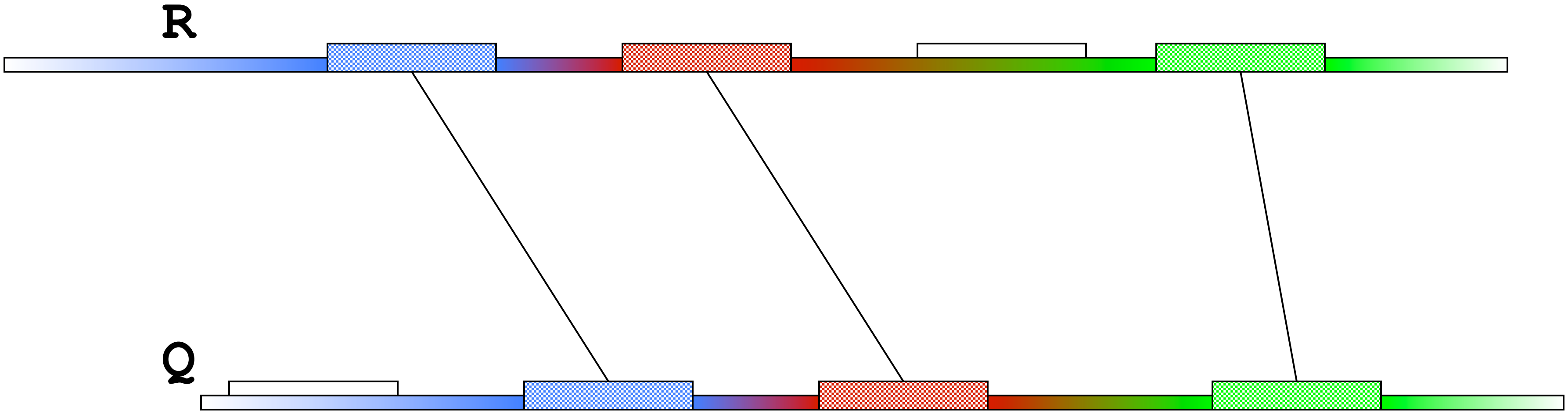
Seed and Extend

visualization

FIND all MUMs

CLUSTER consistent MUMs

EXTEND alignments



WGA example with **nucmer**

- *Yersina pestis CO92* vs. *Yersina pestis KIM*
  - High nucleotide similarity, 99.86%
    - Two strains of the same species
  - Extensive genome shuffling
    - Global alignment will not work
  - Highly repetitive
    - Many local alignments

WGA Alignment

```
nucmer -maxmatch CO92.fasta KIM.fasta
-maxmatch    Find maximal exact matches (MEMs)
```

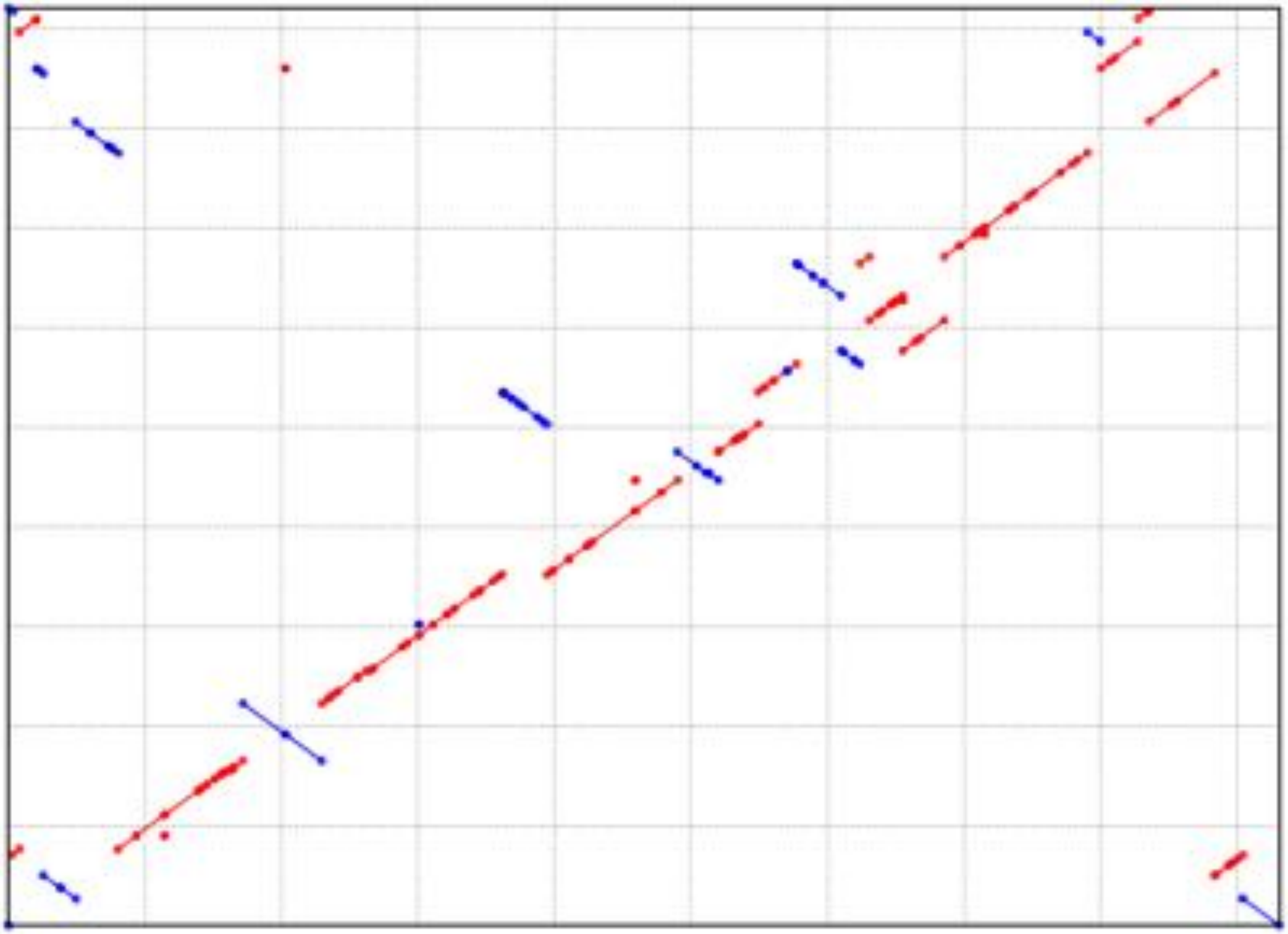
```
delta-filter -m out.delta > out.filter.m
-m  Many-to-many mapping
```

```
show-coords -r out.delta.m > out.coords
-r  Sort alignments by reference position
```

```
dnadiff out.delta.m
Construct catalog of sequence variations
```

```
mummerplot --large --layout out.delta.m
--large      Large plot
--layout Nice layout for multi-fasta files
--x11        Default, draw using x11 (--postscript, --png)
*requires gnuplot
```

See manual at http://
mummer.sourceforge.net/manual



References

- Documentation
  - http://mummer.sourceforge.net
    - » publication listing
  - http://mummer.sourceforge.net/manual
    - » documentation
  - http://mummer.sourceforge.net/examples
    - » walkthroughs

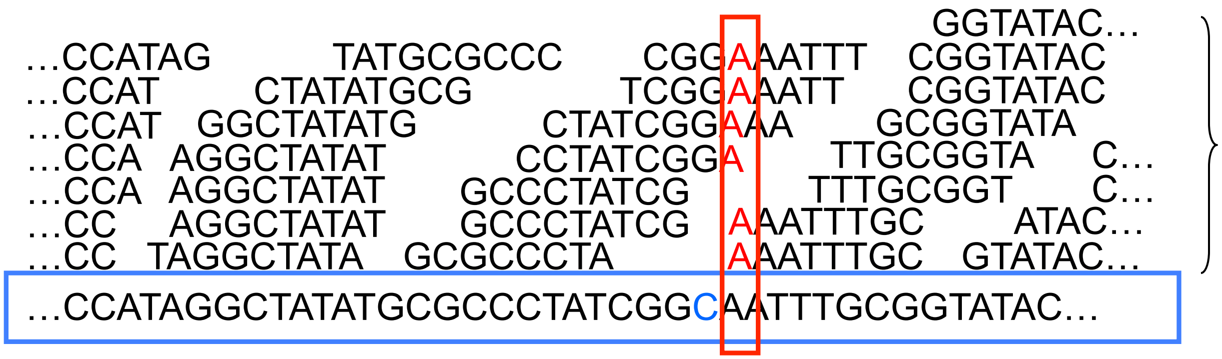
- Email
  - mummer-help@lists.sourceforge.net
  - amp@umiacs.umd.edu

Bowtie: Ultrafast and memory efficient alignment of short DNA sequences to the human genome

Slides Courtesy of Ben Langmead
(langmead@umiacs.umd.edu)

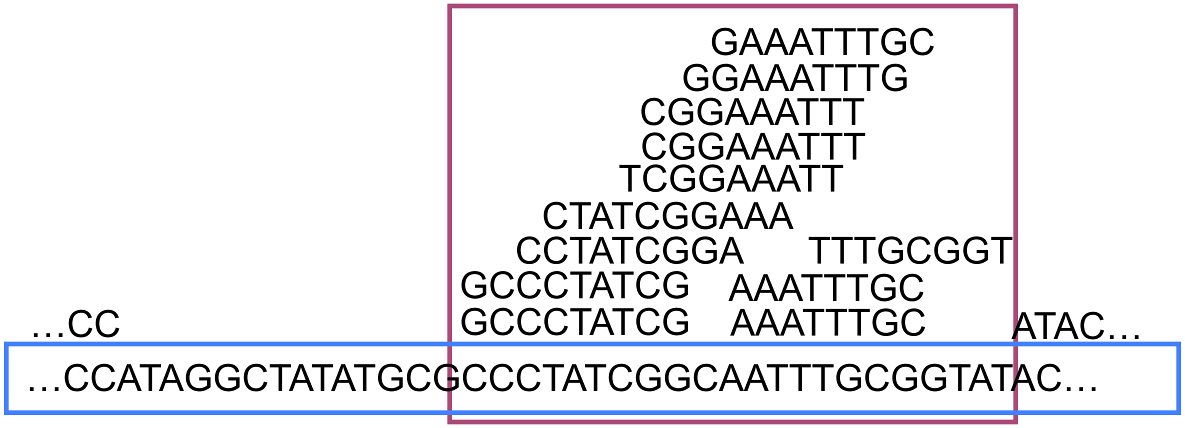
Short Read Applications

- Genotyping: Identify Variations



```
                                    GGTATAC...
...CCATAG    TATGCGCCC   CGGAAATTT CGGTATAC
...CCAT   CTATATGCG     TCGGAAATT  CGGTATAC
...CCAT GGCTATATG    CTATCGGAAA   GCGGTATA
...CCA AGGCTATAT    CCTATCGGA   TTGCGGTA  C...
...CCA AGGCTATAT  GCCCTATCG    TTTGCGGT   C...
...CC  AGGCTATAT  GCCCTATCG AAATTTGC   ATAC...
...CC TAGGCTATA GCGCCCTA    AAATTTGC GTATAC...
...CCATAGGCTATATGCGCCCTATCGGCAATTTGCGGTATAC...
```

- *-seq: Classify & measure significant peaks

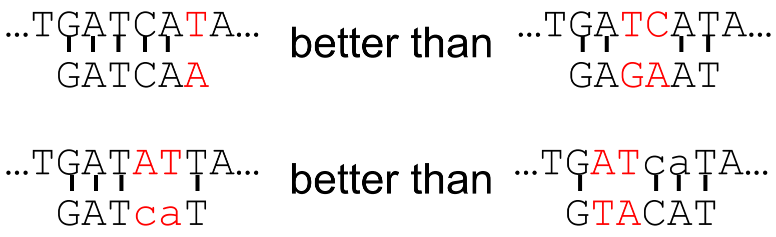


```
                           GAAATTTGC
                          GGAAATTTG
                         CGGAAATTT
                         CGGAAATTT
                        TCGGAAATT
                     CTATCGGAAA
                    CCTATCGGA  TTTGCGGT
                  GCCCTATCG AAATTTGC
...CC             GCCCTATCG AAATTTGC   ATAC...
...CCATAGGCTATATGCGCCCTATCGGCAATTTGCGGTATAC...
```

Short Read Alignment

- Given a reference and a set of reads, report at least one "good" local alignment for each read if one exists
  - Approximate answer to: where in genome did read originate?

- What is "good"? For now, we concentrate on:
  - Fewer mismatches is better
  - Failing to align a low-quality base is better than failing to align a high-quality base

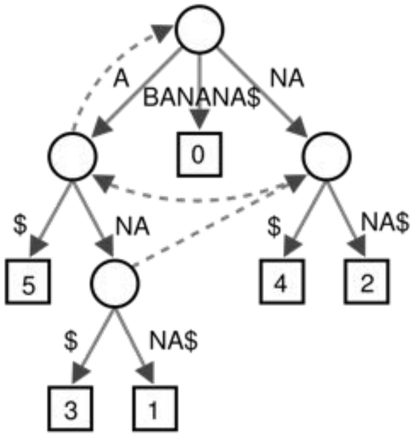


...TGATCATA...
GATCAA
better than
...TGATCATA...
GAGAAT

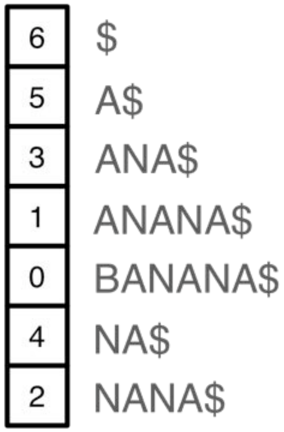
...TGATATTA...
GATcaT
better than
...TGATcaTA...
GTACAT

Indexing

- Genomes and reads are too large for direct approaches like dynamic programming
  - Genome indices can be big. For human:

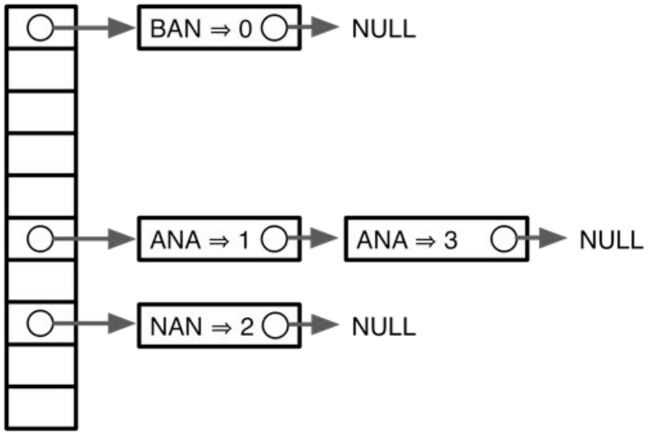


> 35 GBs



6	$
5	A$
3	ANA$
1	ANANA$
0	BANANA$
4	NA$
2	NANA$

> 12 GBs



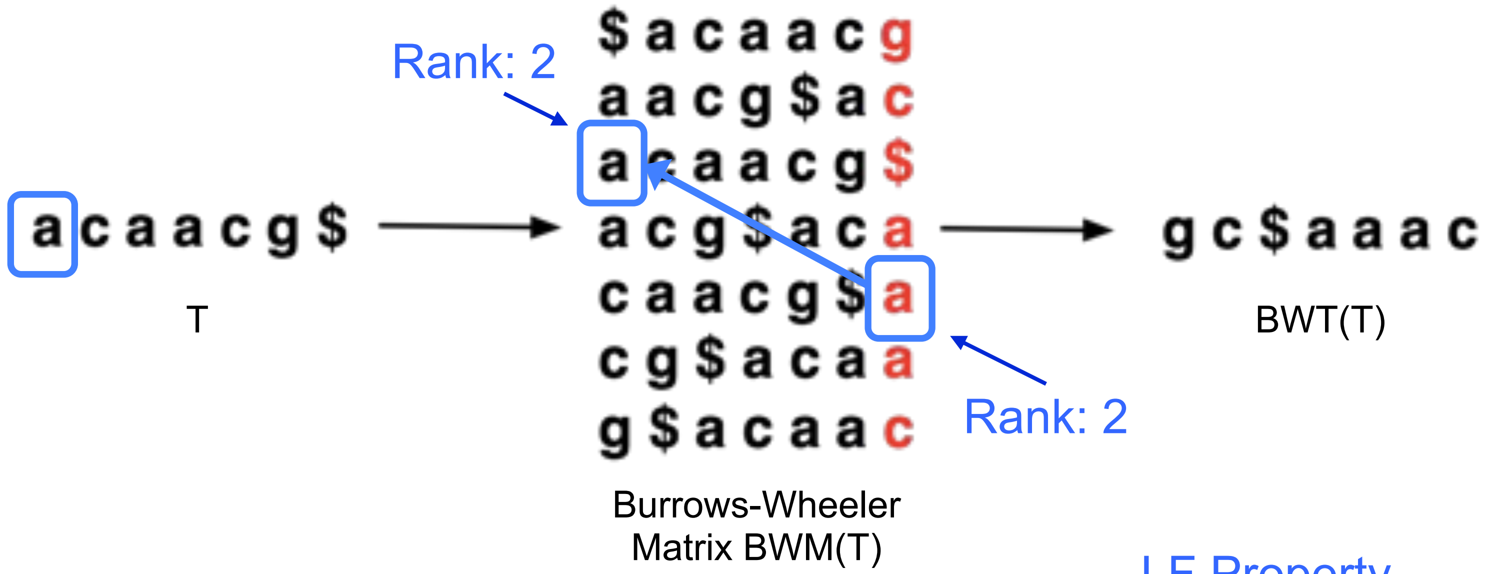
> 12 GBs

- Large indices necessitate painful compromises

1. Require big-memory machine
2. Use secondary storage
3. Build new index each run
4. Subindex and do multiple passes

Burrows-Wheeler Transform

- Reversible permutation of the characters in a text



a c a a c g $

T

Rank: 2

$ a c a a c g
a a c g $ a c
a c a a c g $
a c g $ a c a
c a a c g $ a
c g $ a c a a
g $ a c a a c

Rank: 2

Burrows-Wheeler Matrix BWM(T)

g c $ a a a c

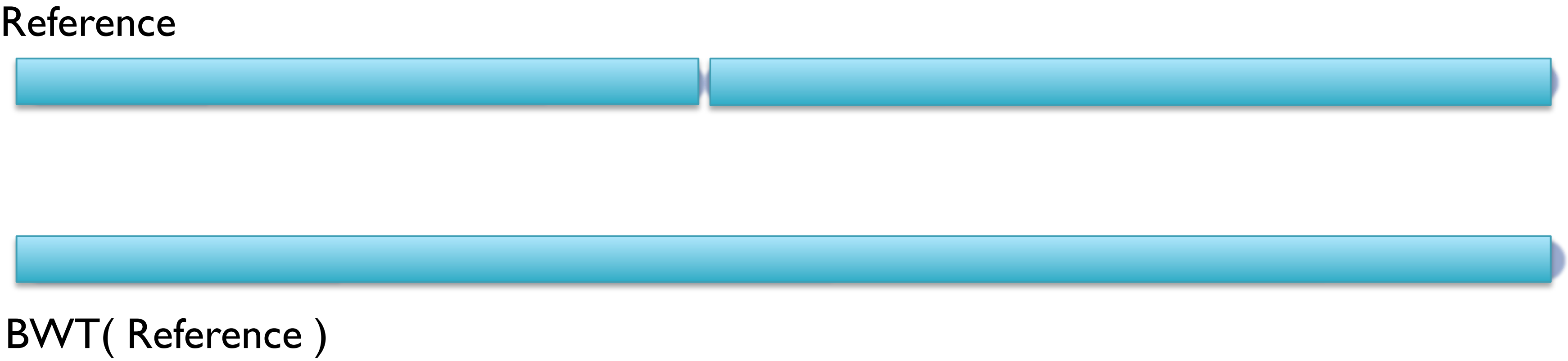
BWT(T)

LF Property implicitly encodes Suffix Array

- BWT(T) is the index for T

A block sorting lossless data compression algorithm.
Burrows M, Wheeler DJ (1994) *Digital Equipment Corporation.* Technical Report 124

Bowtie algorithm



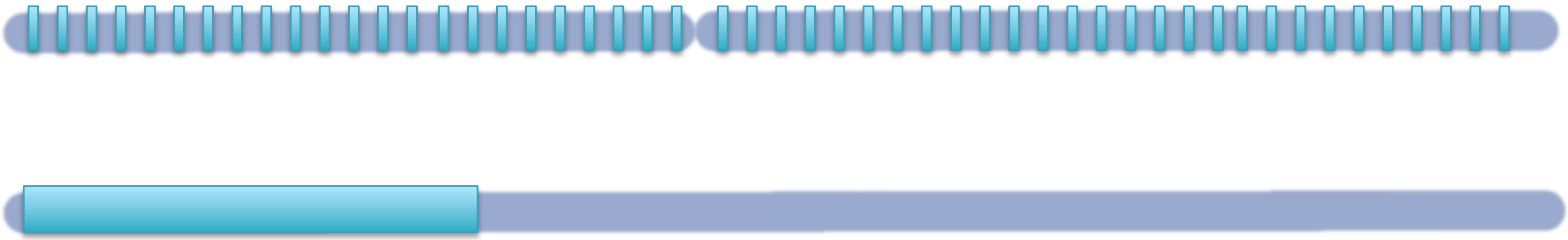
Reference

BWT(Reference)

Query:
AATGATACGGCGACCACCGAGATCTA

Bowtie algorithm

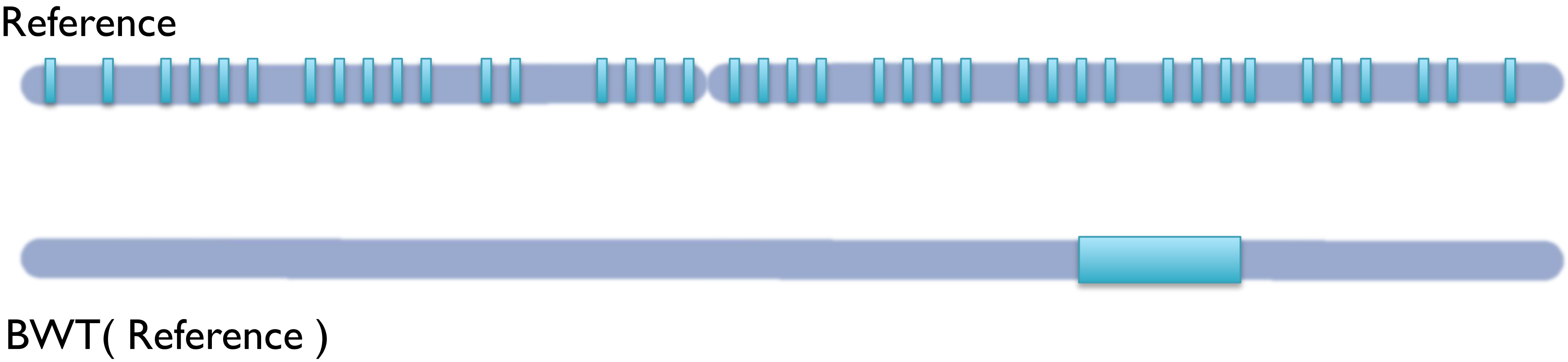
Reference



BWT(Reference)

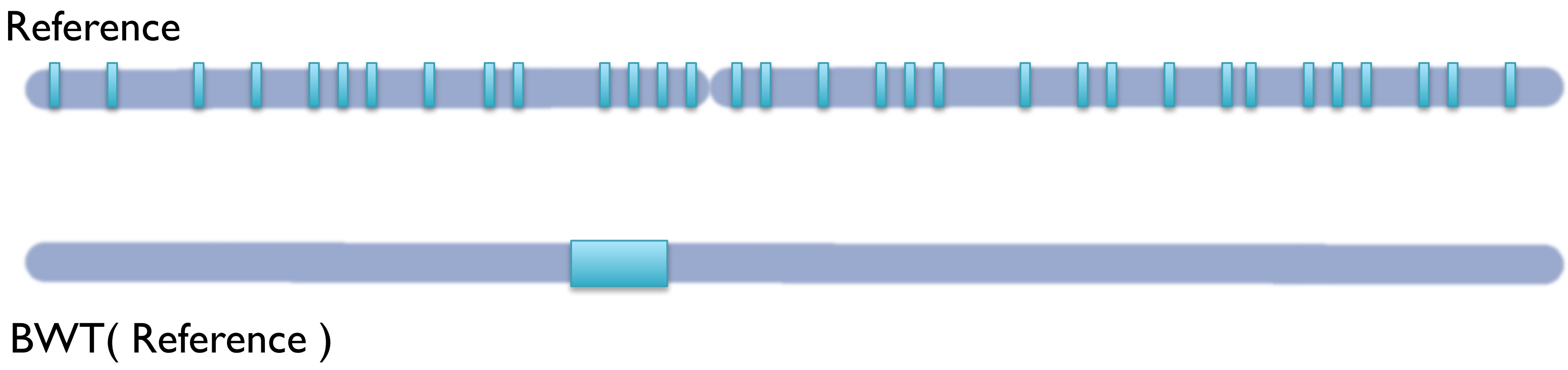
Query:
AATGATACGGCGACCACCGAGATCTA

Bowtie algorithm



Query:
AATGATACGGCGACCACCGAGATCTA

Bowtie algorithm

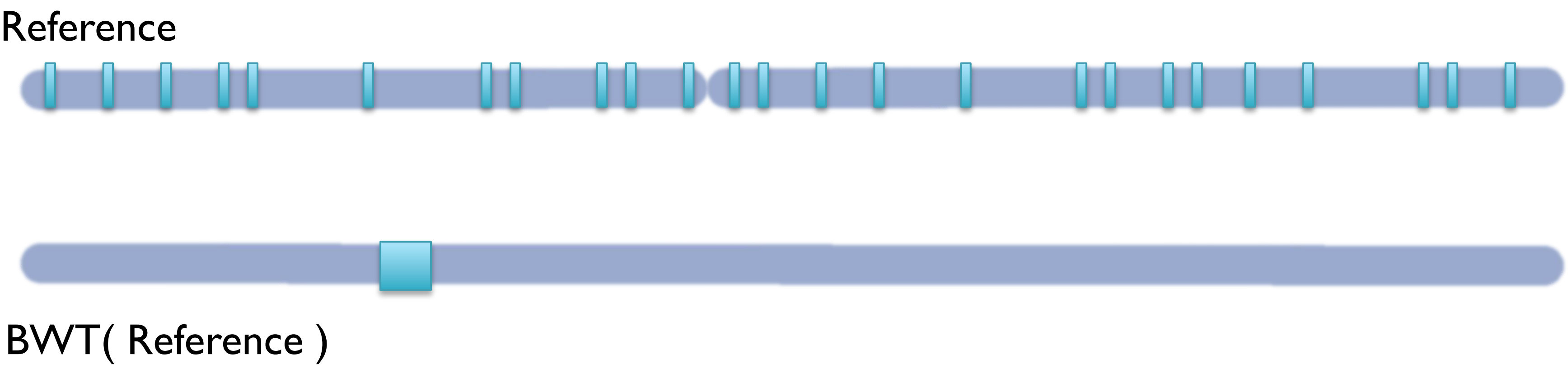


Reference

BWT(Reference)

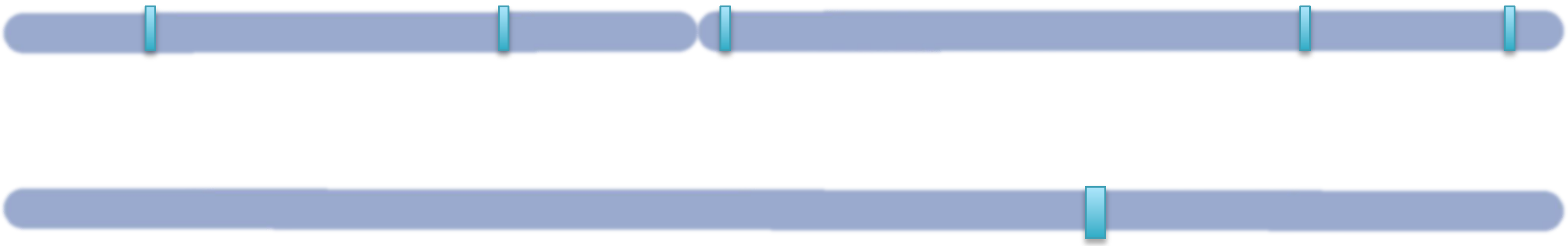
Query:
A A T G A T A C G G C G A C C A C C G A G A T C T A

Bowtie algorithm



Query:
AATGATACGGCGACCACCGAGATCTA

Bowtie algorithm



Reference

BWT(Reference)

Query:
AATGATACGGCGACCACCGAGATCTA

Bowtie algorithm

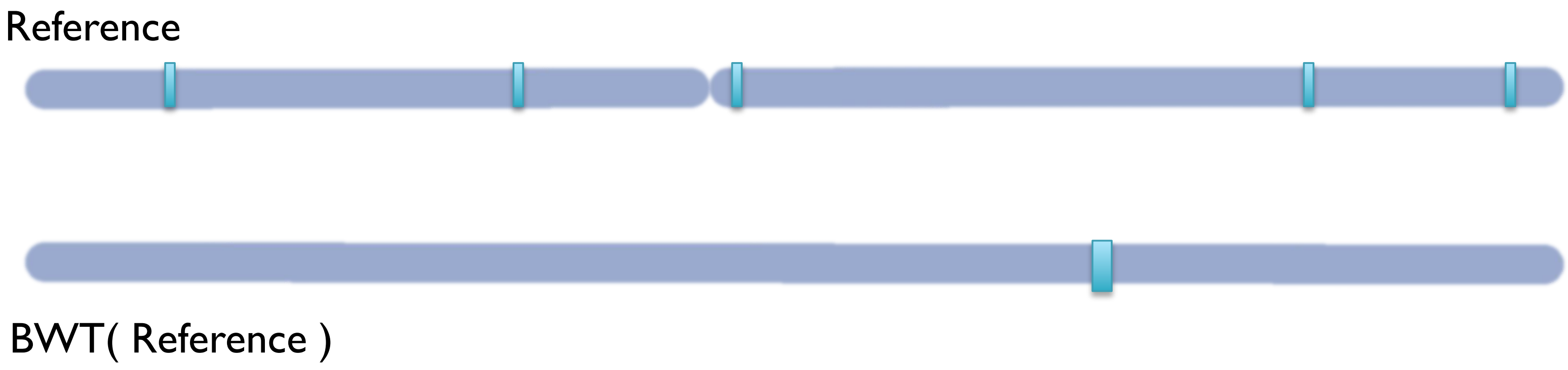
Reference

BWT(Reference)

Query:

AATG ATACGGCGACCACCGAGATCTA

Bowtie algorithm



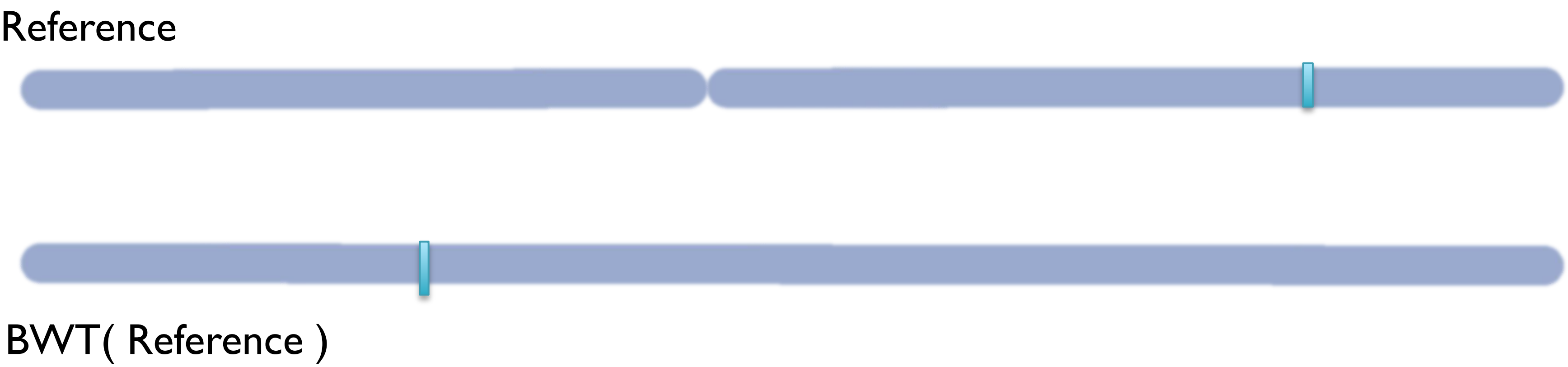
Reference

BWT(Reference)

Query:

AATGTTACGGCGACCACCGAGATCTA

Bowtie algorithm



Reference

BWT(Reference)

Query:

AATGTTACGGCGACCACCGAGATCTA

BWT Short Read Mapping

1. Trim off very low quality bases & adapters from ends of sequences

2. Execute depth-first-search of the implicit suffix tree represented by the BWT
    1. If we fail to reach the end, back-track and resume search
    2. BWT enables searching for good end-to-end matches entirely in RAM
        1. 100s of times faster than competing approaches

3. Report the "best" n alignments
    1. Best = fewest mismatches/edit distance, possibly weighted by QV
    2. Some reads will have millions of equally good mapping positions
    3. If reads are paired, try to find mapping that satisfies both

Mapping Applications

- Mapping Algorithms
  - Bowtie: (BWT) Fastest, No indels => moderate sensitivity
  - BWA: (BWT) Fast, small indels => good sensitivity
  - Novoalign: (Hash Table) Slow, RAM intensive, big indels => high sensitivity

- Variation Detection
  - SNPs
    - SAMTools: Bayesian model incorporating depth, quality values, also indels
    - SOAPsnp: SAMTools + known SNPs, nucleotide specific errors, no indels
  - Structural Variations
    - Hydra: Very sensitive alignment, scan for discordant pairs
    - Large indels: Open Research Problem to assembly their sequence
  - Copy number changes
    - RDexplorer: Scan alignments for statistically significant coverage pileup
  - Microsatellite variations
    - See Mitch!

Sequence Alignment Summary

- Distance metrics:
  - Hamming: How many substitutions?
  - Edit Distance: How many substitutions or indels?
  - Sequence Similarity: How similar (under this model of similarity)?

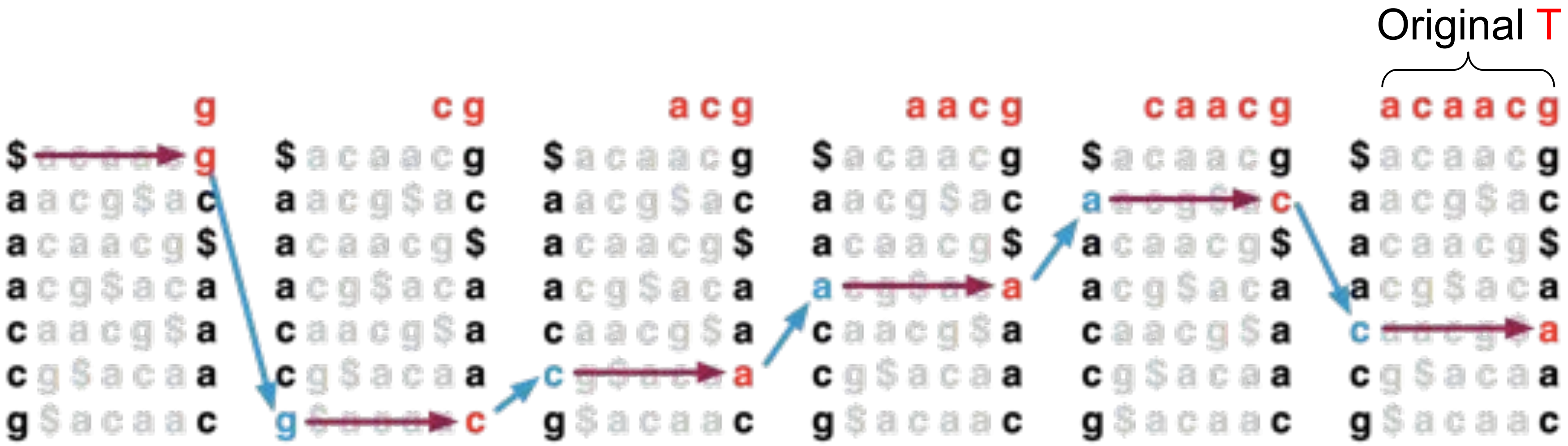
- Techniques
  - Seed-and-extend: Anchor the search for in-exact using exact only
  - Dynamic Programming: Find a global optimal as a function of its parts
  - BWT Search: implicit DFS of SA/ST

- Sequence Alignment Algorithms: Pick the right tool for the job
  - Smith-Waterman: DP Local sequence alignment
  - BLAST: Homology Searching
  - MUMmer: Whole genome alignment, short read mapping (with care)
  - Bowtie/BWA/Novoalign: short read mapping

Supplemental

Burrows-Wheeler Transform

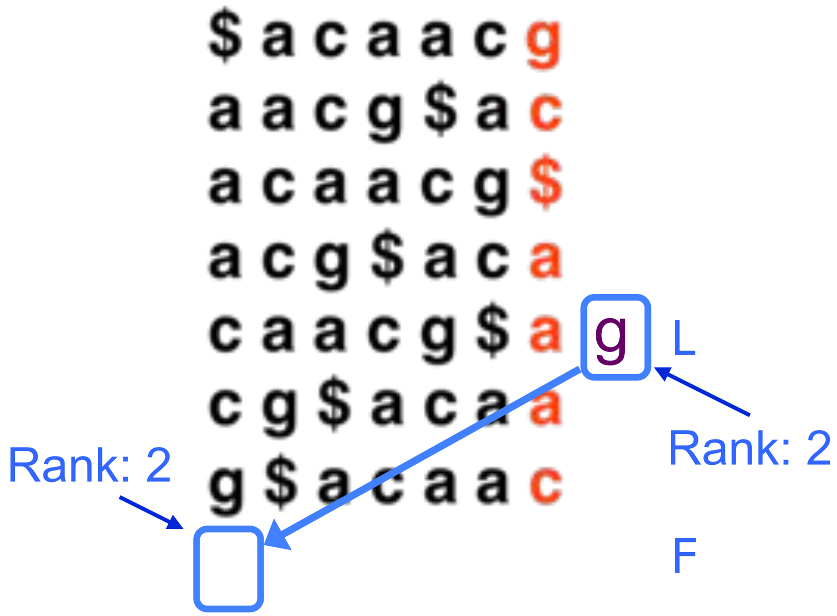
- Recreating T from BWT(T)
  - Start in the first row and apply **LF** repeatedly, accumulating predecessors along the way



BWT Exact Matching

- **LFc**(r, c) does the same thing as **LF**(r) but it ignores r' s actual final character and "pretends" it' s c:

LFc(5, g) = 8



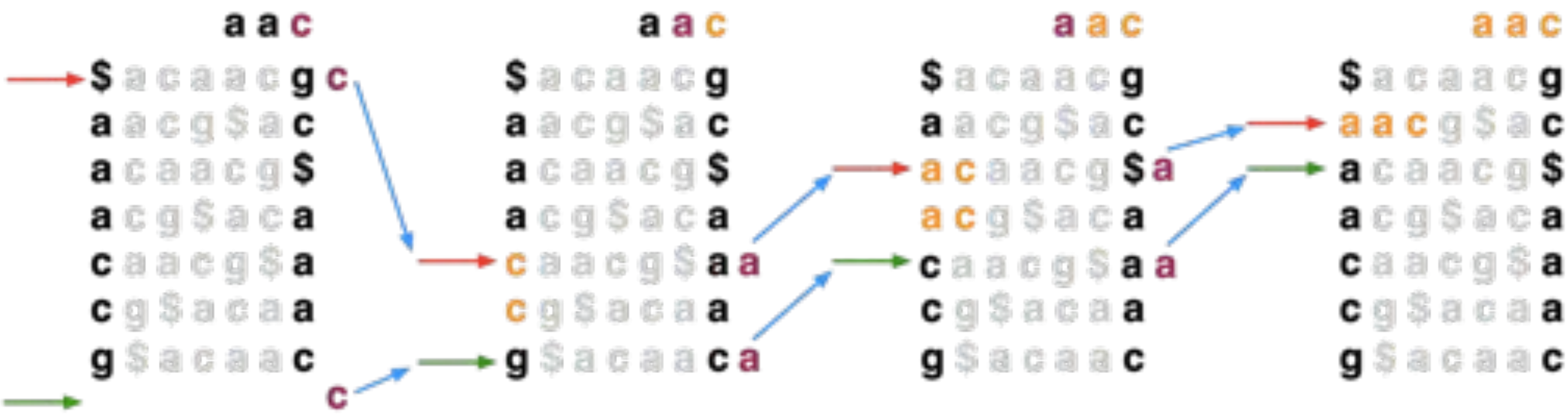
$ a c a a c g
a a c g $ a c
a c a a c g $
a c g $ a c a
c a a c g $ a
c g $ a c a a
g $ a c a a c

BWT Exact Matching

- Start with a range, (**top**, **bot**) encompassing all rows and repeatedly apply **LFc**:

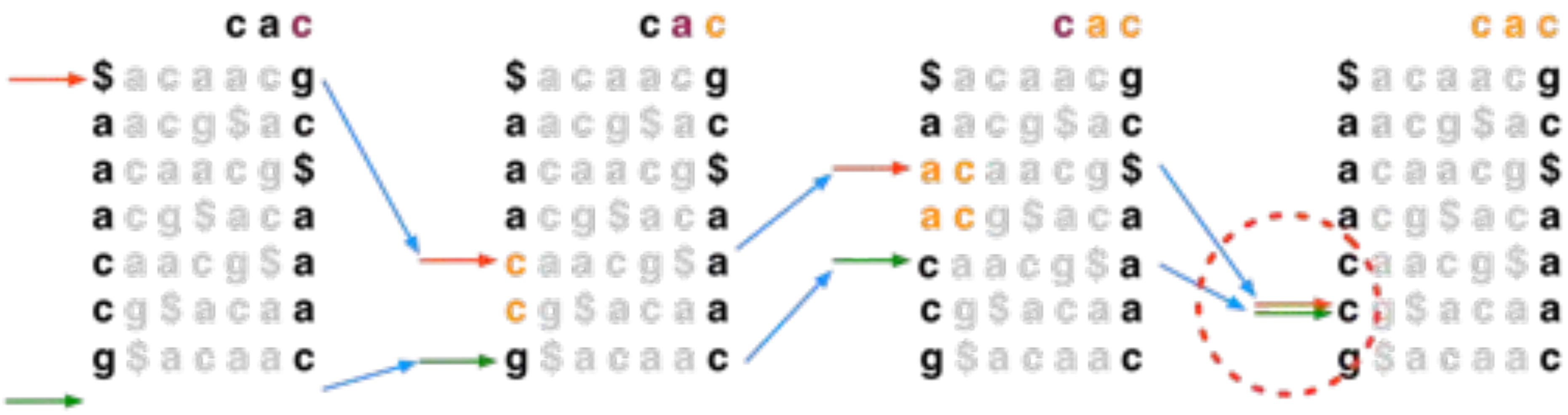
  top = **LFc**(**top**, **qc**); **bot** = **LFc**(**bot**, **qc**)

  qc = the next character to the left in the query



Ferragina P, Manzini G: Opportunistic data structures with applications. *FOCS. IEEE Computer Society; 2000.*

BWT Exact Matching



```
      c a c            c a c            c a c            c a c
$ a c a a c g    $ a c a a c g    $ a c a a c g    $ a c a a c g
a a c g $ a c    a a c g $ a c    a a c g $ a c    a a c g $ a c
a c a a c g $    a c a a c g $    a c a a c g $    a c a a c g $
a c g $ a c a    a c g $ a c a    a c g $ a c a    a c g $ a c a
c a a c g $ a    c a a c g $ a    c a a c g $ a    c a a c g $ a
c g $ a c a a    c g $ a c a a    c g $ a c a a    c g $ a c a a
g $ a c a a c    g $ a c a a c    g $ a c a a c    g $ a c a a c
```

- If range becomes empty (**top** = **bot**) the query suffix (and therefore the query as a whole) does not occur