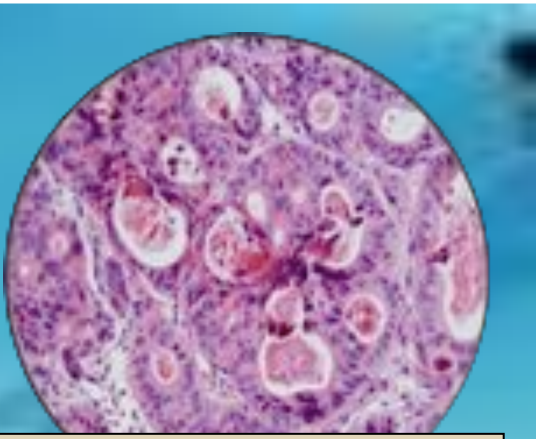
Sequence Alignment & Computational Thinking

Michael Schatz

Oct 25, 2012
SBU Graduate Genetics

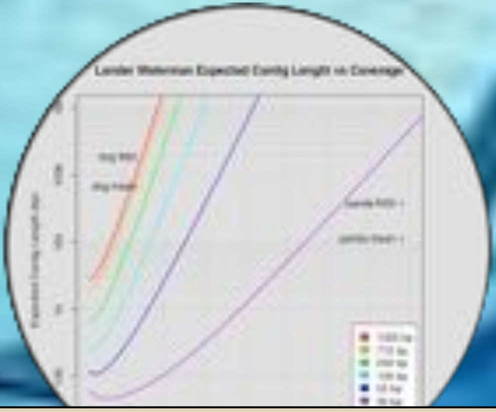
Schatz Lab Overview

Computation



Human Genetics

Sequencing



Modeling

Plant Genomics

Outline

1. Rise of DNA Sequencing
2. Sequence Alignment Basics
3. Understanding Bowtie
4. Genetics of Autism

Milestones in Molecular Biology

Nature Vol. 265 February 24 1977 687

articles

Nucleotide sequence of bacteriophage ΦX174 DNA

F. Sanger, G. M. Air, B. G. Barrell, N. L. Brown, A. R. Coulson, J. C. Fiddes, C. A. Hutchison III, P. M. Slocombe & M. Smith

MRC Laboratory of Molecular Biology, Hills Road, Cambridge CB2 2QH, UK

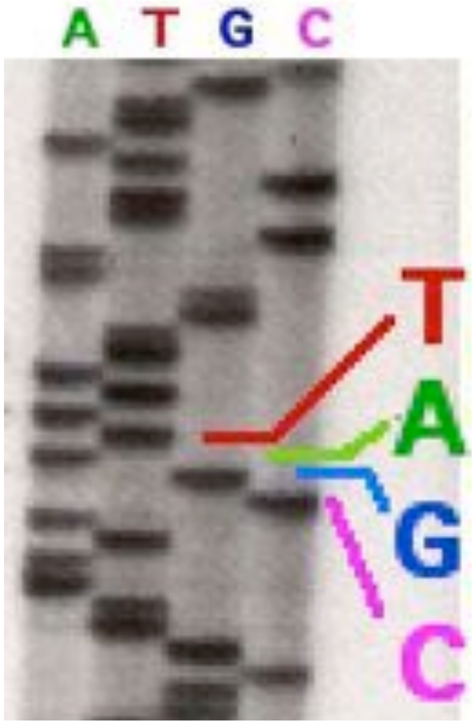
[illegible]

1977

1^{st} Complete Organism

Bacteriophage ϕX174

5375 bp



Radioactive Chain Termination

5000bp / week / person

http://en.wikipedia.org/wiki/File:Sequencing.jpg

http://www.answers.com/topic/automated-sequencer

Nucleotide sequence of bacteriophage ϕX174 DNA

Sanger, F. et al. (1977) *Nature. 265: 687 - 695*

Milestones in Molecular Biology

1995
Fleischmann *et al.*
1st Free Living Organism
TIGR Assembler. 1.8Mbp

2000
Myers *et al.*
1st Large WGS Assembly.
Celera Assembler. 116 Mbp

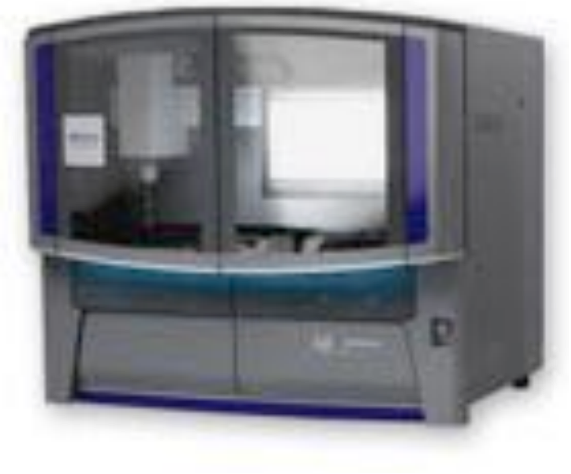
2001
Venter *et al.* / IHGSC
Human Genome
Celera Assembler. 2.9 Gbp

ABI 3700: 500 bp reads x 768 samples / day = 384,000 bp / day.
"The machine was so revolutionary that it could decode in a single day the same amount of genetic material that most DNA labs could produce in a year." J. Craig Venter

Milestones in Molecular Biology

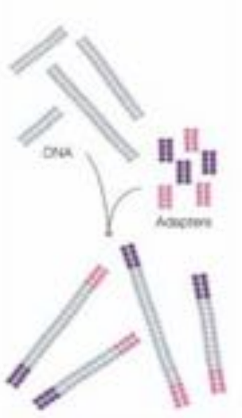
2004
454/Roche
Pyrosequencing
Current Specs (Titanium):
1M 400bp reads / run =
1Gbp / day

2007
Illumina
Sequencing by Synthesis
Current Specs (HiSeq 2000):
2.5B 100bp reads / run =
60Gbp / day

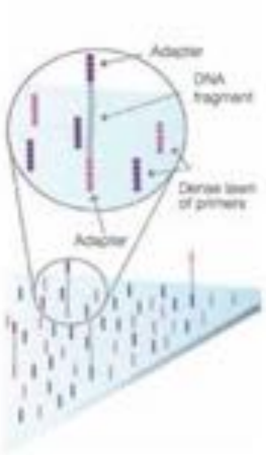


2008
ABI / Life Technologies
SOLiD Sequencing
Current Specs (5500xl):
5B 75bp reads / run =
30Gbp / day

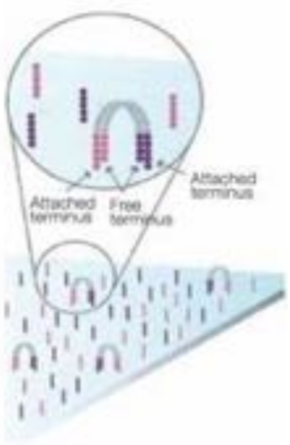
Illumina Sequencing by Synthesis



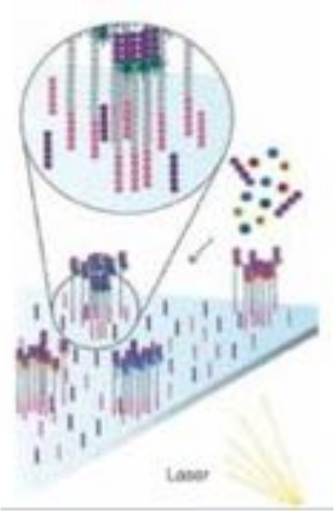
1. Prepare



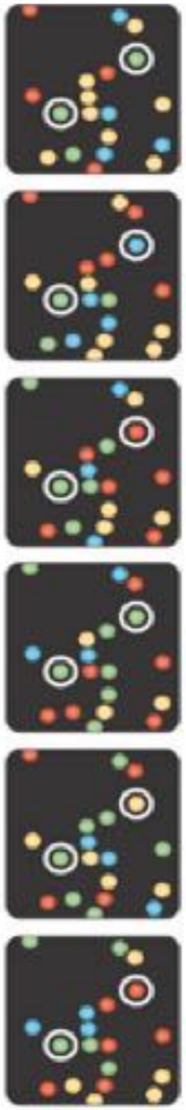
2. Attach



3. Amplify



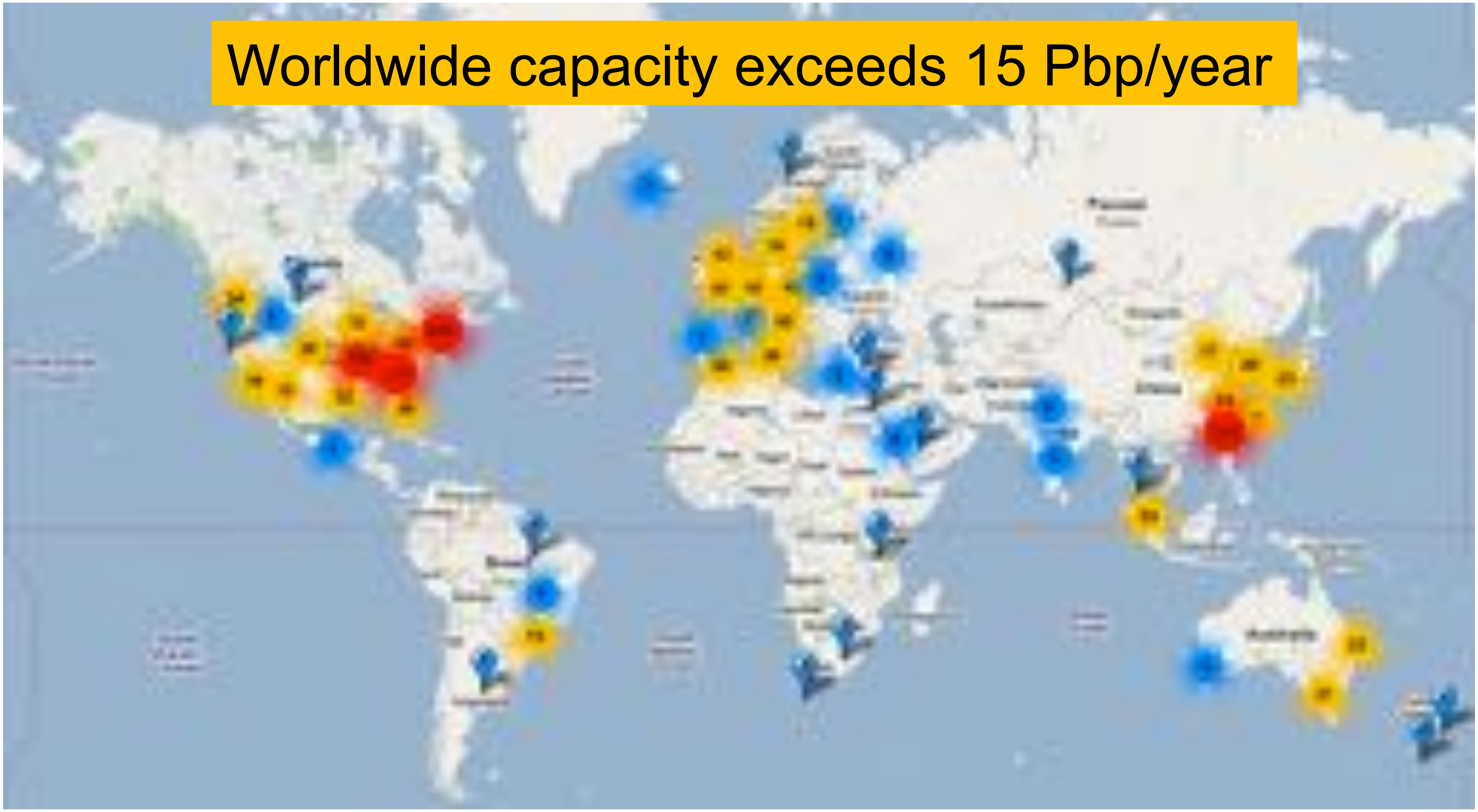
4. Image



5. Basecall

Metzker (2010) Nature Reviews Genetics 11:31-46
http://www.youtube.com/watch?v=l99aKKHcxC4

Sequencing Centers



Worldwide capacity exceeds 15 Pbp/year

Next Generation Genomics: World Map of High-throughput Sequencers
http://pathogenomics.bham.ac.uk/hts/

Milestones in Molecular Biology

There is tremendous interest to sequence:

- What is your genome sequence?
- How does your genome compare to my genome?

- Where are the genes and how active are they?
- How does gene activity change during development?
- How does splicing change during development?

- How does methylation change during development?
- How does chromatin change during development?
- How does is your genome folded in the cell?
- Where do proteins bind and regulate genes?

- What virus and microbes are living inside you?
- How has the disease mutated your genome?
- What drugs should we give you?

- …

Outline

1. Rise of DNA Sequencing

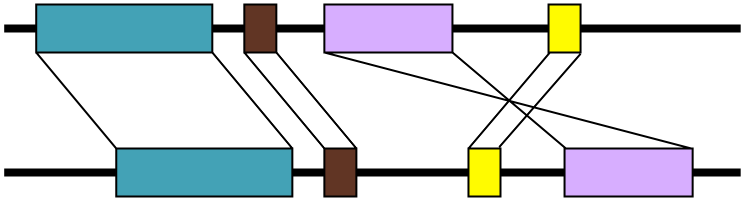
2. Sequence Alignment Basics

3. Understanding Bowtie

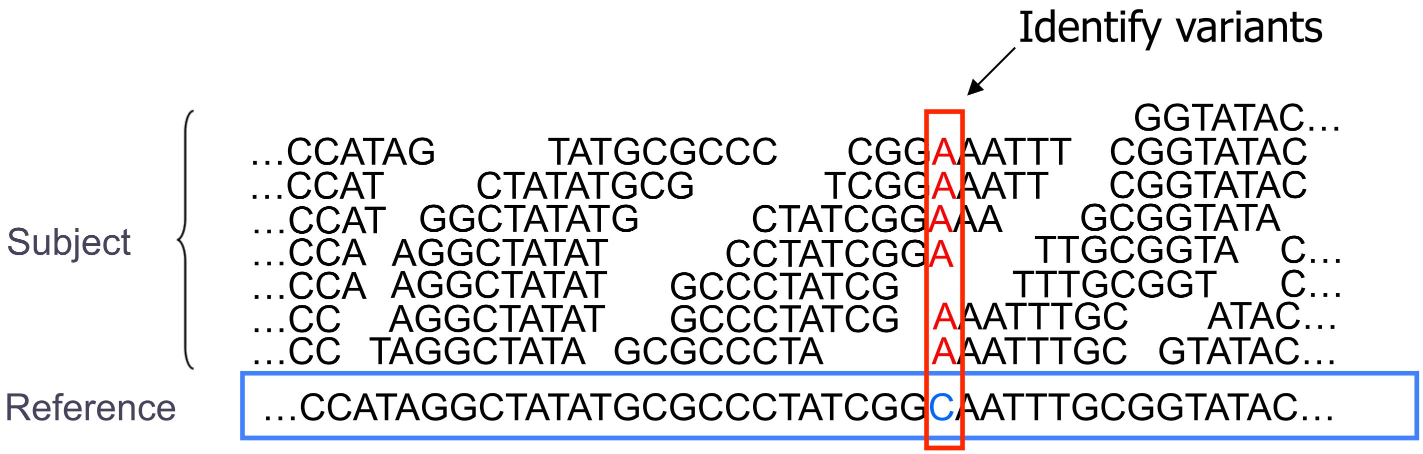
4. Genetics of Autism

Sequence Alignment

- A very common problem in computational biology is to find occurrences of one sequence in another sequence
  - Genome Assembly
  - Gene Finding
  - Comparative Genomics
  - Functional analysis of proteins
  - Motif discovery
  - SNP analysis
  - Phylogenetic analysis
  - Primer Design
  - Personal Genomics
  - …



Short Read Mapping


Identify variants
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…
Subject
Reference
…CCATAGGCTATATGCGCCCTATCGGCAATTTGCGGTATAC…

- Given a reference and many subject reads, report one or more “good” end-to-end alignments per alignable read
  - Fundamental computation to genotyping and many assays
    - RNA-seq Methyl-seq FAIRE-seq
    - ChIP-seq Dnase-seq Hi-C-seq

- Desperate need for scalable solutions
  - Single human requires >1,000 CPU hours / genome
  - **1000 hours * 1000 genomes = 1M CPU hours / project**

Searching for GATTACA

- Where is GATTACA in the human genome?

- Strategy 1: Brute Force

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									

No match at offset 1

Searching for GATTACA

- Where is GATTACA in the human genome?

- Strategy 1: Brute Force

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 at offset 2

Searching for GATTACA

- Where is GATTACA in the human genome?

- Strategy 1: Brute Force

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	…						

No match at offset 3…

Searching for GATTACA

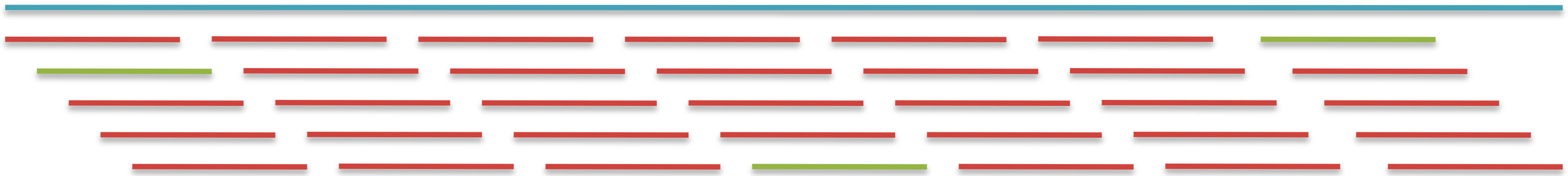
- Where is GATTACA in the human genome?

- Strategy 1: Brute Force

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	

No match at offset 9 <- Checking each possible position takes time

Brute Force Analysis



- Brute Force:
  - At every possible offset in the genome:
    - Do all of the characters of the query match?

- Analysis
  - Simple, easy to understand
  - Genome length = n [3B]
  - Query length = m [7]
  - Comparisons: (n-m+1) * m [21B]

- Overall runtime: O(nm)

[How long would it take if we double the genome size, read length?]

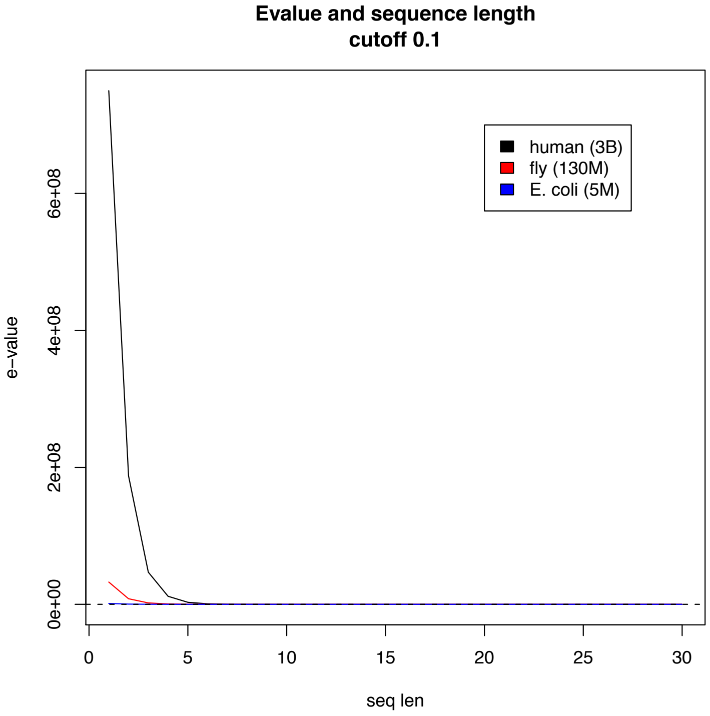
[How long would it take if we double both?]

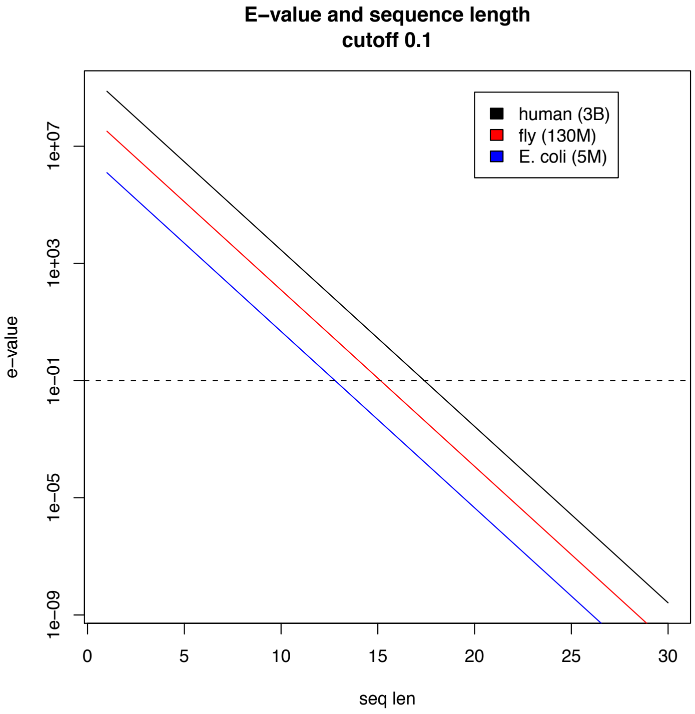
Expected Occurrences

The expected number of occurrences (e-value) of a given sequence in a genome depends on the length of the genome and inversely on the length of the sequence

- 1 in 4 bases are G, 1 in 16 positions are GA, 1 in 64 positions are GAT, …
- 1 in 16,384 should be GATTACA
- E=n/(4^m) [183,105 expected occurrences]

[How long do the reads need to be for a significant match?]





Brute Force Reflections

Why check every position?

- GATTACA can't possibly start at position 15 [WHY?]

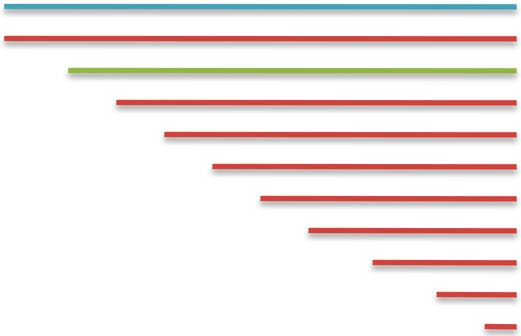
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	…
T	G	A	T	T	A	C	A	G	A	T	T	A	C	C	…
								G	A	T	T	A	C	A	

- Improve runtime to O(n + m) [3B + 7]
  - If we double both, it just takes twice as long
  - Knuth-Morris-Pratt, 1977
  - Boyer-Moyer, 1977, 1991
- For one-off scans, this is the best we can do (optimal performance)
  - We have to read every character of the genome, and every character of the query
  - For short queries, runtime is dominated by the length of the genome

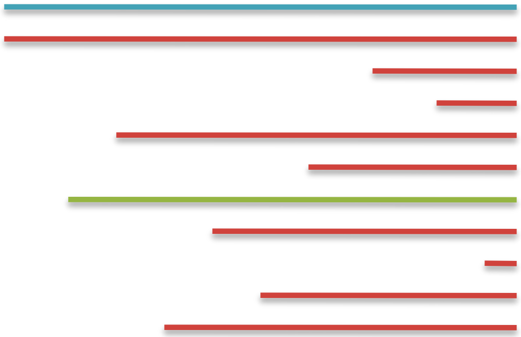
Suffix Arrays: Searching the Phone Book

- What if we need to check many queries?
  - We don't need to check every page of the phone book to find 'Schatz'
  - Sorting alphabetically lets us immediately skip 96% (25/26) of the book *without any loss in accuracy*

- Sorting the genome: Suffix Array (Manber & Myers, 1991)
  - Sort every suffix of the genome



Split into n suffixes



Sort suffixes alphabetically

[Challenge Question: How else could we split the genome?]

Searching the Index

- Strategy 2: Binary search
  - Compare to the middle, refine as higher or lower

- Searching for GATTACA
  - Lo = 1; Hi = 15;

Lo →

#	Sequence	Pos
1	ACAGATTACC…	6
2	ACC…	13
3	AGATTACC…	8
4	ATTACAGATTACC…	3
5	ATTACC…	10
6	C…	15
7	CAGATTACC…	7
8	CC…	14
9	GATTACAGATTACC…	2
10	GATTACC…	9
11	TACAGATTACC…	5
12	TACC…	12
13	TGATTACAGATTACC…	1
14	TTACAGATTACC…	4
15	TTACC…	11

Hi →

Searching the Index

- Strategy 2: Binary search
  - Compare to the middle, refine as higher or lower
- Searching for GATTACA
  - Lo = 1; Hi = 15; Mid = (1+15)/2 = 8
  - Middle = Suffix[8] = CC

Lo

#	Sequence	Pos
1	ACAGATTACC…	6
2	ACC…	13
3	AGATTACC…	8
4	ATTACAGATTACC…	3
5	ATTACC…	10
6	C…	15
7	CAGATTACC…	7
8	CC…	14
9	GATTACAGATTACC…	2
10	GATTACC…	9
11	TACAGATTACC…	5
12	TACC…	12
13	TGATTACAGATTACC…	1
14	TTACAGATTACC…	4
15	TTACC…	11

Hi

Searching the Index

- Strategy 2: Binary search
  - Compare to the middle, refine as higher or lower
- Searching for GATTACA
  - Lo = 1; Hi = 15; Mid = (1+15)/2 = 8
  - Middle = Suffix[8] = CC

    => Higher: Lo = Mid + 1

Lo

#	Sequence	Pos
1	ACAGATTACC…	6
2	ACC…	13
3	AGATTACC…	8
4	ATTACAGATTACC…	3
5	ATTACC…	10
6	C…	15
7	CAGATTACC…	7
8	CC…	14
9	GATTACAGATTACC…	2
10	GATTACC…	9
11	TACAGATTACC…	5
12	TACC…	12
13	TGATTACAGATTACC…	1
14	TTACAGATTACC…	4
15	TTACC…	11

Hi

Searching the Index

- Strategy 2: Binary search
  - Compare to the middle, refine as higher or lower

- Searching for GATTACA
  - Lo = 1; Hi = 15; Mid = (1+15)/2 = 8
  - Middle = Suffix[8] = CC
    => Higher: Lo = Mid + 1

  - Lo = 9; Hi = 15;

	#	Sequence	Pos
	1	ACAGATTACC…	6
	2	ACC…	13
	3	AGATTACC…	8
	4	ATTACAGATTACC…	3
	5	ATTACC…	10
	6	C…	15
	7	CAGATTACC…	7
	8	CC…	14
Lo →	9	GATTACAGATTACC…	2
	10	GATTACC…	9
	11	TACAGATTACC…	5
	12	TACC…	12
	13	TGATTACAGATTACC…	1
	14	TTACAGATTACC…	4
Hi →	15	TTACC…	11

Searching the Index

- Strategy 2: Binary search
  - Compare to the middle, refine as higher or lower

- Searching for GATTACA
  - Lo = 1; Hi = 15; Mid = (1+15)/2 = 8
  - Middle = Suffix[8] = CC

    => Higher: Lo = Mid + 1

  - Lo = 9; Hi = 15; Mid = (9+15)/2 = 12
  - Middle = Suffix[12] = TACC

	#	Sequence	Pos
	1	ACAGATTACC…	6
	2	ACC…	13
	3	AGATTACC…	8
	4	ATTACAGATTACC…	3
	5	ATTACC…	10
	6	C…	15
	7	CAGATTACC…	7
	8	CC…	14
Lo →	9	GATTACAGATTACC…	2
	10	GATTACC…	9
	11	TACAGATTACC…	5
	12	TACC…	12
	13	TGATTACAGATTACC…	1
	14	TTACAGATTACC…	4
Hi →	15	TTACC…	11

Searching the Index

- Strategy 2: Binary search
  - Compare to the middle, refine as higher or lower

- Searching for GATTACA
  - Lo = 1; Hi = 15; Mid = (1+15)/2 = 8
  - Middle = Suffix[8] = CC
    => Higher: Lo = Mid + 1

  - Lo = 9; Hi = 15; Mid = (9+15)/2 = 12
  - Middle = Suffix[12] = TACC
    => Lower: Hi = Mid - 1

  - Lo = 9; Hi = 11;

#	Sequence	Pos
1	ACAGATTACC…	6
2	ACC…	13
3	AGATTACC…	8
4	ATTACAGATTACC…	3
5	ATTACC…	10
6	C…	15
7	CAGATTACC…	7
8	CC…	14
9	GATTACAGATTACC…	2
10	GATTACC…	9
11	TACAGATTACC…	5
12	TACC…	12
13	TGATTACAGATTACC…	1
14	TTACAGATTACC…	4
15	TTACC…	11

Lo →

Hi →

Searching the Index

- Strategy 2: Binary search
  - Compare to the middle, refine as higher or lower

- Searching for GATTACA
  - Lo = 1; Hi = 15; Mid = (1+15)/2 = 8
  - Middle = Suffix[8] = CC

    => Higher: Lo = Mid + 1

  - Lo = 9; Hi = 15; Mid = (9+15)/2 = 12
  - Middle = Suffix[12] = TACC

    => Lower: Hi = Mid - 1

  - Lo = 9; Hi = 11; Mid = (9+11)/2 = 10
  - Middle = Suffix[10] = GATTACC

	#	Sequence	Pos
	1	ACAGATTACC…	6
	2	ACC…	13
	3	AGATTACC…	8
	4	ATTACAGATTACC…	3
	5	ATTACC…	10
	6	C…	15
	7	CAGATTACC…	7
	8	CC…	14
Lo →	9	GATTACAGATTACC…	2
	10	GATTACC…	9
Hi →	11	TACAGATTACC…	5
	12	TACC…	12
	13	TGATTACAGATTACC…	1
	14	TTACAGATTACC…	4
	15	TTACC…	11

Searching the Index

- Strategy 2: Binary search
  - Compare to the middle, refine as higher or lower

- Searching for GATTACA
  - Lo = 1; Hi = 15; Mid = (1+15)/2 = 8
  - Middle = Suffix[8] = CC

    => Higher: Lo = Mid + 1

  - Lo = 9; Hi = 15; Mid = (9+15)/2 = 12
  - Middle = Suffix[12] = TACC

    => Lower: Hi = Mid - 1

  - Lo = 9; Hi = 11; Mid = (9+11)/2 = 10
  - Middle = Suffix[10] = GATTACC

    => Lower: Hi = Mid - 1

  - Lo = 9; Hi = 9;

Lo

Hi

#	Sequence	Pos
1	ACAGATTACC…	6
2	ACC…	13
3	AGATTACC…	8
4	ATTACAGATTACC…	3
5	ATTACC…	10
6	C…	15
7	CAGATTACC…	7
8	CC…	14
9	GATTACAGATTACC…	2
10	GATTACC…	9
11	TACAGATTACC…	5
12	TACC…	12
13	TGATTACAGATTACC…	1
14	TTACAGATTACC…	4
15	TTACC…	11

Searching the Index

- Strategy 2: Binary search
  - Compare to the middle, refine as higher or lower

- Searching for GATTACA
  - Lo = 1; Hi = 15; Mid = (1+15)/2 = 8
  - Middle = Suffix[8] = CC
    => Higher: Lo = Mid + 1

  - Lo = 9; Hi = 15; Mid = (9+15)/2 = 12
  - Middle = Suffix[12] =TACC
    => Lower: Hi = Mid - 1

  - Lo = 9; Hi = 11; Mid = (9+11)/2 = 10
  - Middle = Suffix[10] = GATTACC
    => Lower: Hi = Mid - 1

  - Lo = 9; Hi = 9; Mid = (9+9)/2 = 9
  - Middle = Suffix[9] = GATTACA…
    => Match at position 2!

Lo
Hi

#	Sequence	Pos
1	ACAGATTACC…	6
2	ACC…	13
3	AGATTACC…	8
4	ATTACAGATTACC…	3
5	ATTACC…	10
6	C…	15
7	CAGATTACC…	7
8	CC…	14
9	GATTACAGATTACC…	2
10	GATTACC…	9
11	TACAGATTACC…	5
12	TACC…	12
13	TGATTACAGATTACC…	1
14	TTACAGATTACC…	4
15	TTACC…	11

Binary Search Analysis

- Binary Search

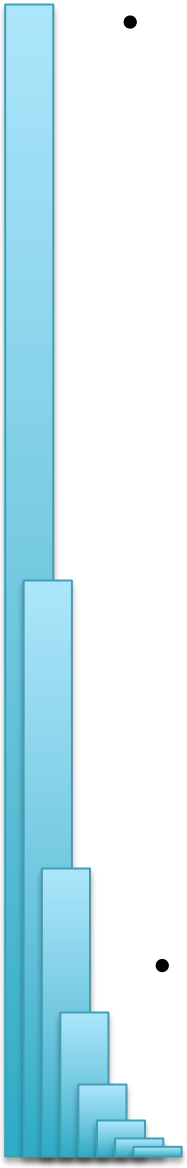
```
Initialize search range to entire list
    mid = (hi+lo)/2; middle = suffix[mid]
    if query matches middle: done
    else if query < middle: pick low range
    else if query > middle: pick hi range
Repeat until done or empty range                [WHEN?]
```

- Analysis
  - More complicated method
  - How many times do we repeat?
    - How many times can it cut the range in half?
    - Find smallest x such that: $n/(2^x) \le 1$; $x = \lg_2(n)$ [32]

- Total Runtime: $O(m \lg n)$
  - More complicated, but much faster!
  - Looking up a query loops 32 times instead of 3B

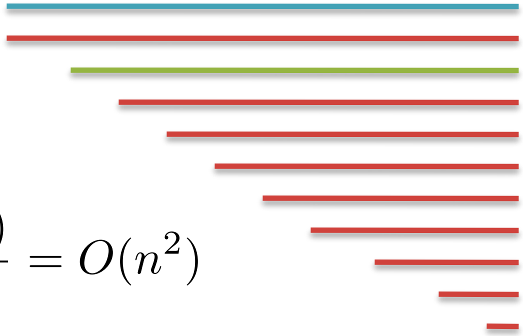
[How long does it take to search 6B or 24B nucleotides?]

Suffix Array Construction



- How can we store the suffix array?

  [How many characters are in all suffixes combined?]



$$S = 1 + 2 + 3 + \cdots + n = \sum_{i=1}^{n} i = \frac{n(n+1)}{2} = O(n^2)$$

  - Hopeless to explicitly store 4.5 billion billion characters

  - Instead use implicit representation
    - Keep 1 copy of the genome, and a list of sorted offsets
    - Storing 3 billion offsets fits on a server (12GB)

- Searching the array is very fast, but it takes time to construct
  - This time will be amortized over many, many searches
  - Run it once "overnight" and save it away for all future queries

Pos
6
13
8
3
10
15
7
14
2
9
5
12
1
4
11

TGATTACAGATTACC

Sorting

Quickly sort these numbers into ascending order:
14, 29, 6, 31, 39, 64, 78, 50, 13, 63, 61, 19

[How do you do it?]

6, 14, 29, 31, 39, 64, 78, 50, 13, 63, 61, 19

6, 13, 14, 29, 31, 39, 64, 78, 50, 63, 61, 19

6, 13, 14, 19, 29, 31, 39, 64, 78, 50, 63, 61

6, 13, 14, 19, 29, 31, 39, 64, 78, 50, 63, 61

6, 13, 14, 19, 29, 31, 39, 64, 78, 50, 63, 61

6, 13, 14, 19, 29, 31, 39, 50, 64, 78, 63, 61

6, 13, 14, 19, 29, 31, 39, 50, 61, 64, 78, 63

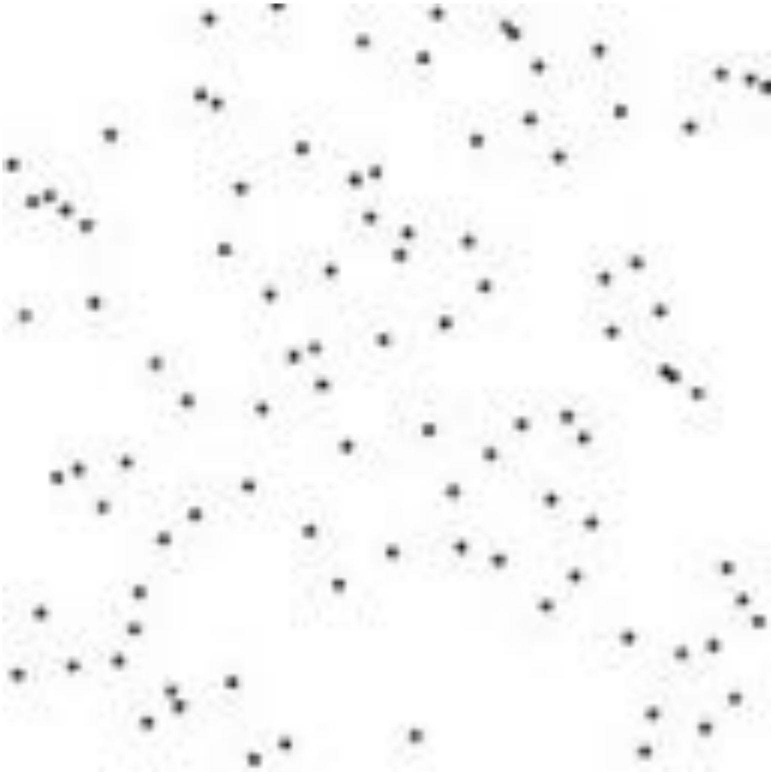
6, 13, 14, 19, 29, 31, 39, 50, 61, 63, 64, 78

6, 13, 14, 19, 29, 31, 39, 50, 61, 63, 64, 78

6, 13, 14, 19, 29, 31, 39, 50, 61, 63, 64, 78

6, 13, 14, 19, 29, 31, 39, 50, 61, 63, 64, 78

6, 13, 14, 19, 29, 31, 39, 50, 61, 63, 64, 78



http://en.wikipedia.org/wiki/Selection_sort

Selection Sort Analysis

- Selection Sort (Input: list of n numbers)

```
for pos = 1 to n
    // find the smallest element in [pos, n]
    smallest = pos
    for check = pos+1 to n
        if (list[check] < list[smallest]): smallest = check

    // move the smallest element to the front
    tmp = list[smallest]
    list[pos] = list[smallest]
    list[smallest] = tmp
```

- Analysis

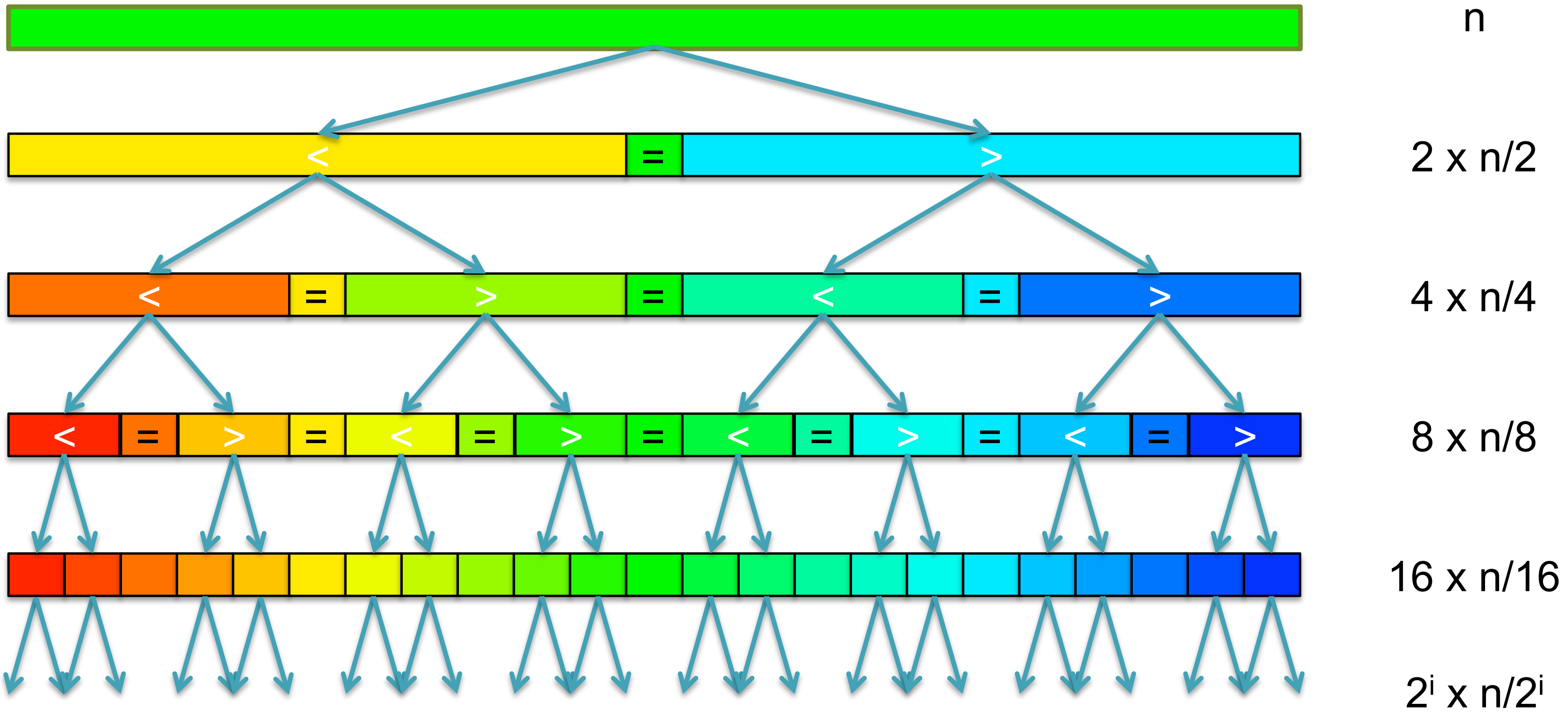
$$T = n + (n-1) + (n-2) + \cdots + 3 + 2 + 1 = \sum_{i=1}^{n} i = \frac{n(n+1)}{2} = O(n^2)$$

  - Outer loop: pos = 1 to n
  - Inner loop: check = pos to n
  - Running time: Outer * Inner = O(n^2) [4.5 Billion Billion]

[Challenge Questions: Why is this slow? / Can we sort any faster?]

Divide and Conquer

- Selection sort is slow because it rescans the entire list for each element
  - How can we split up the unsorted list into independent ranges?
  - Hint 1: Binary search splits up the problem into 2 independent ranges (hi/lo)
  - Hint 2: Assume we know the median value of a list



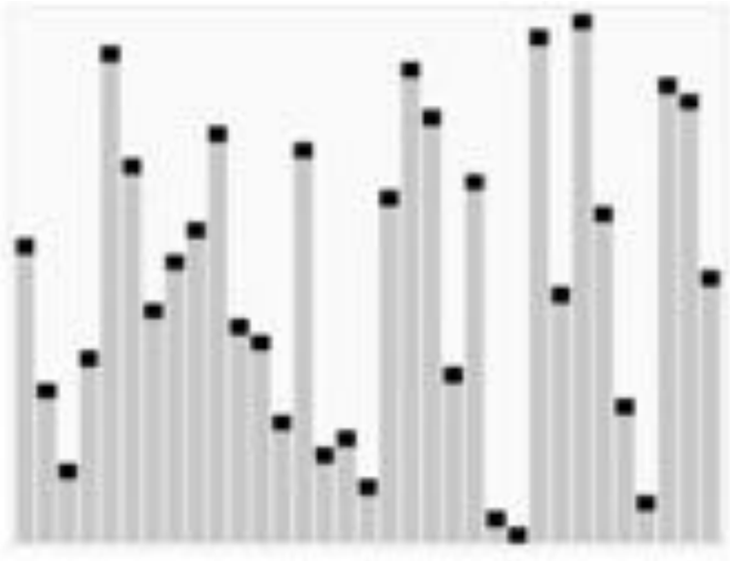
[How many times can we split a list in half?]

QuickSort Analysis

-
  ```
  QuickSort(Input: list of n numbers)
    // see if we can quit
    if (length(list)) <= 1): return list

    // split list into lo & hi
    pivot = median(list)
    lo = {}; hi = {};
    for (i = 1 to length(list))
          if (list[i] < pivot): append(lo, list[i])
          else:                 append(hi, list[i])

    // recurse on sublists
    return (append(QuickSort(lo), QuickSort(hi))
  ```



http://en.wikipedia.org/wiki/Quicksort

- Analysis (Assume we can find the median in O(n))

$$T(n) = \begin{cases} O(1) & \text{if } n \leq 1 \\ O(n) + 2T(n/2) & \text{else} \end{cases}$$

$$T(n) = n + 2(\frac{n}{2}) + 4(\frac{n}{4}) + \cdots + n(\frac{n}{n}) = \sum_{i=0}^{lg(n)} \frac{2^i n}{2^i} = \sum_{i=0}^{lg(n)} n = O(n \lg n)$$

[~94B]

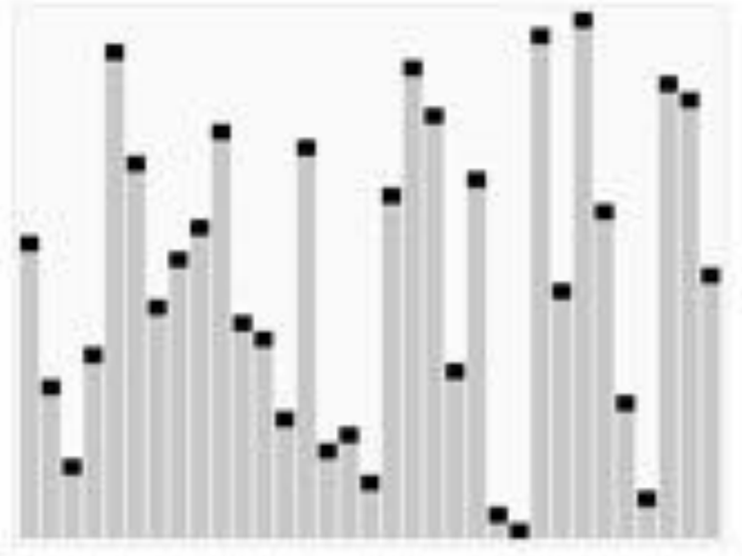
QuickSort Analysis

- QuickSort(Input: list of n numbers)

```
 // see if we can quit
 if (length(list)) <= 1): return list

 // split list into lo & hi
 pivot = median(list)
 lo = {}; hi = {};
 for (i = 1 to length(list))
         if (list[i] < pivot): append(lo, list[i])
         else:                 append(hi, list[i])

 // recurse on sublists
 return (append(QuickSort(lo), QuickSort(hi))
```



http://en.wikipedia.org/wiki/Quicksort

- Analysis (Assume we can find the median in O(n))

$$T(n) = \begin{cases} O(1) & \text{if } n \leq 1 \\ O(n) + 2T(n/2) & \text{else} \end{cases}$$

$$T(n) = n + 2(\frac{n}{2}) + 4(\frac{n}{4}) + \cdots + n(\frac{n}{n}) = \sum_{i=0}^{lg(n)} \frac{2^i n}{2^i} = \sum_{i=0}^{lg(n)} n = O(n \lg n)$$

[~94B]

Break

Outline

1. Rise of DNA Sequencing
2. Sequence Alignment Basics
3. Understanding Bowtie
4. Genetics of Autism

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-table based homology searches
    - 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

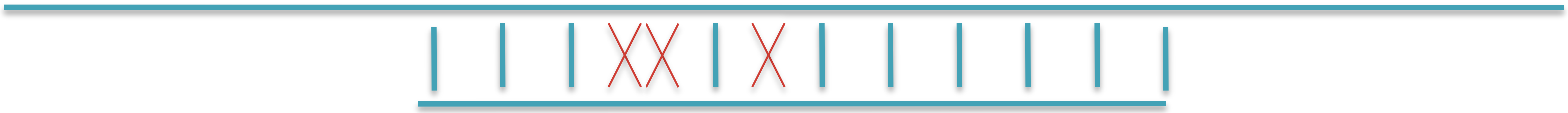
Searching for GATTACA

- 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



- 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

- Find all occurrences of GATTACA with Hamming Distance ≤ 1
- Find all occurrences with minimal Hamming Distance

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

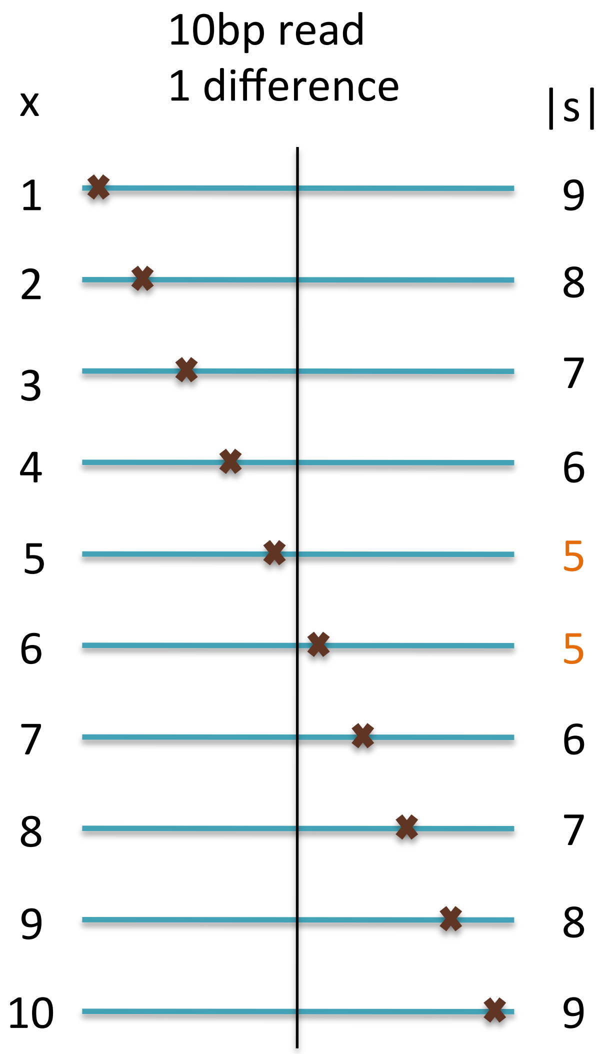
Seed-and-Extend Alignment

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: Pigeonhole principle
  - 1 pigeon can't fill 2 holes

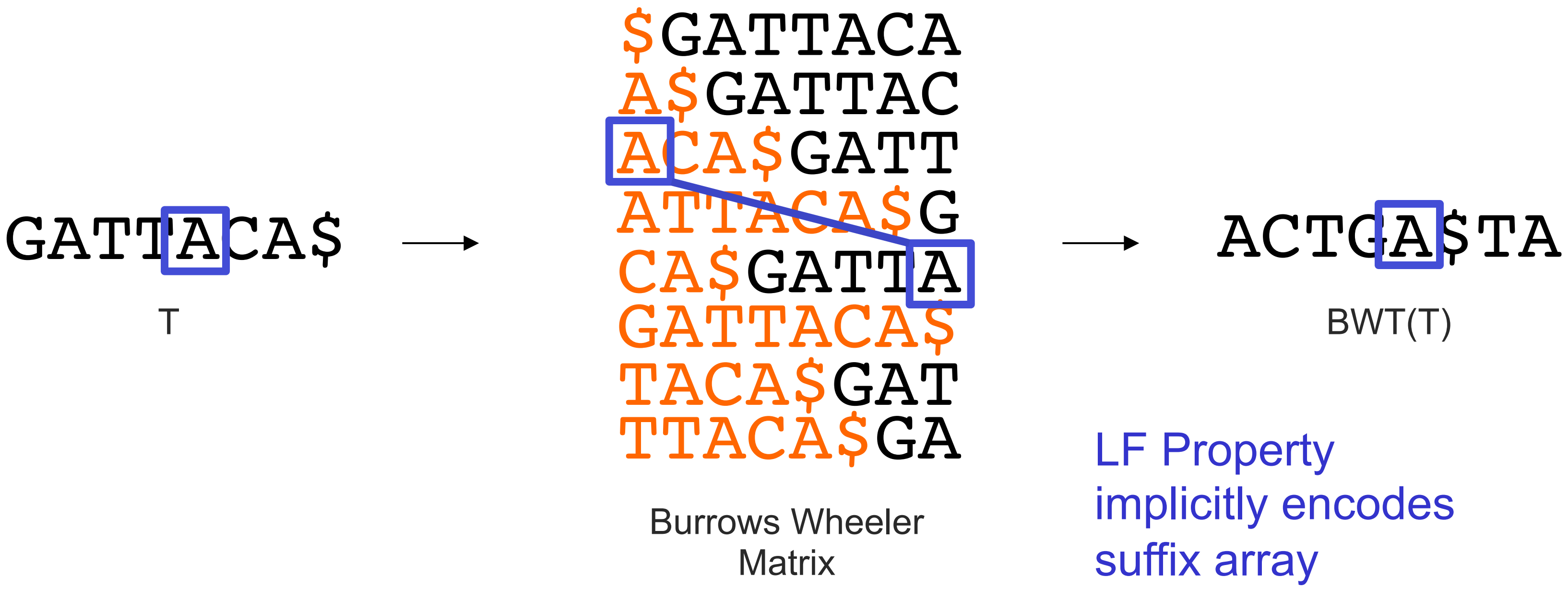
- Seed-and-extend search
  - Use an index to rapidly find short exact alignments to seed longer in-exact alignments
    - BLAST, MUMmer, Bowtie, BWA, SOAP, …
  - Specificity of the depends on seed length
    - Guaranteed sensitivity for k differences
    - Also finds some (but not all) lower quality alignments <- heuristic



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

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

Burrows-Wheeler Transform



GATTACA$ →

T

```
$GATTACA
A$GATTAC
ACA$GATT
ATTACA$G
CA$GATTA
GATTACA$
TACA$GAT
TTACA$GA
```

Burrows Wheeler Matrix

→ ACTGA$TA

BWT(T)

LF Property implicitly encodes suffix array

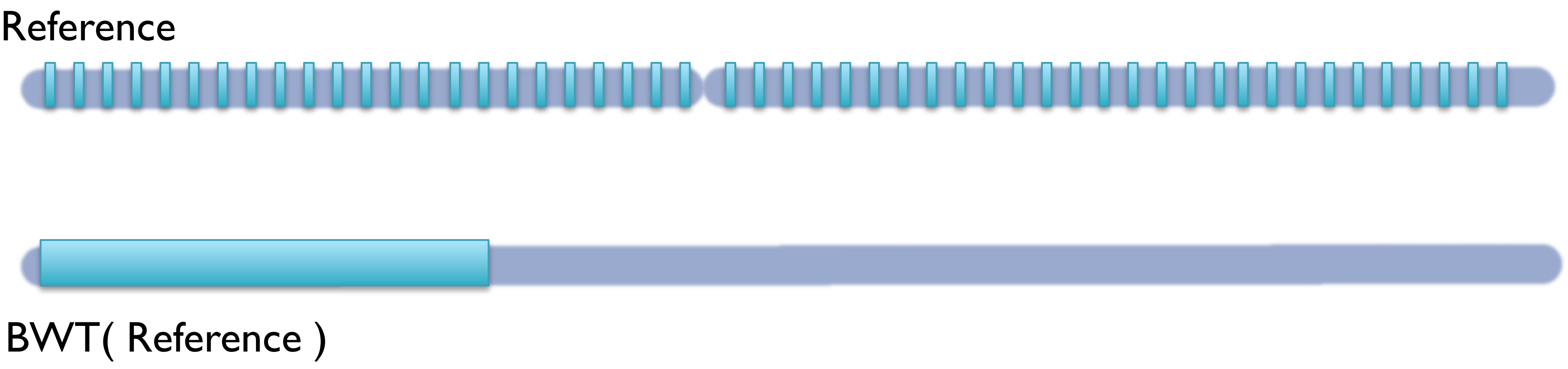
- Suffix Array is tight, but much larger than genome
  - BWT is a reversible permutation of the genome based on the suffix array
  - Core index for Bowtie (Langmead *et al.*, 2009) and most recent short read mapping applications: BWA, SOAP, BLASR, etc…

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

Bowtie algorithm

Query:
AATGATACGGCGACCACCGAGATCTA

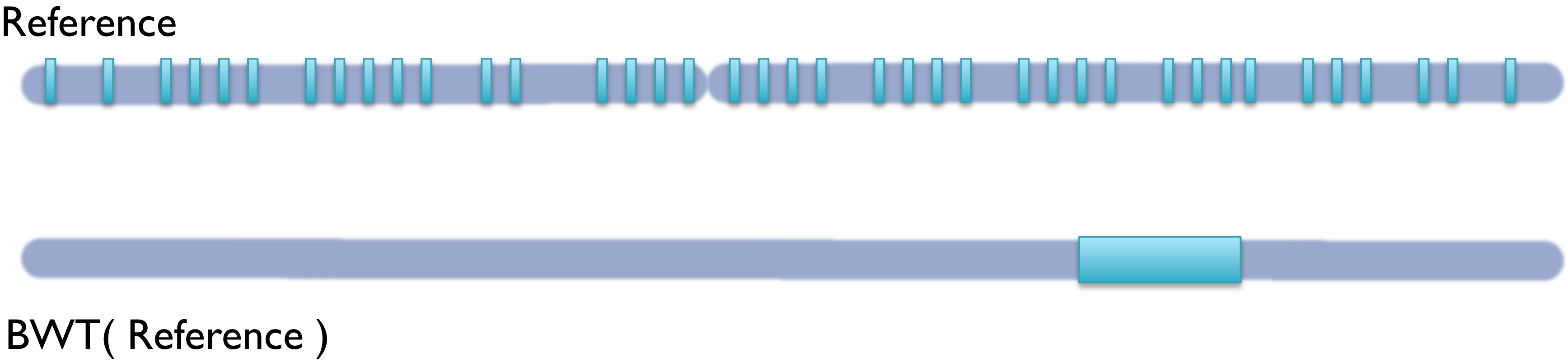
Bowtie algorithm



Query:

AATGATACGGCGACCACCGAGATCT$\boxed{A}$

Bowtie algorithm



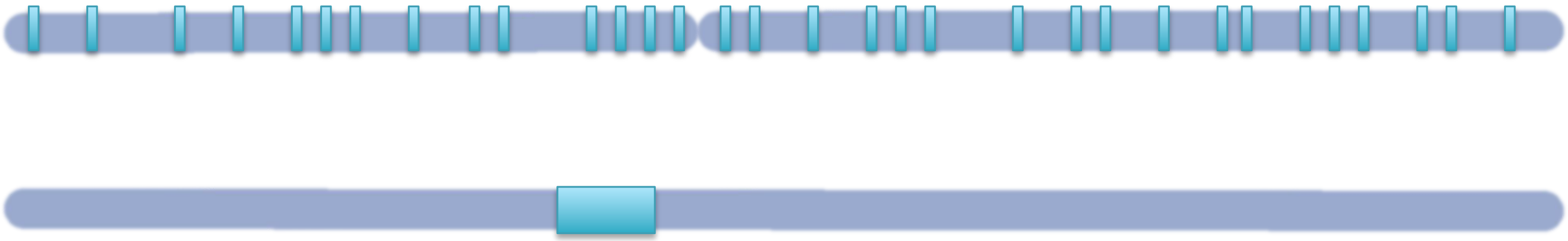
Reference

BWT(Reference)

Query:
AATGATACGGCGACCACCGAGATCTA

Bowtie algorithm

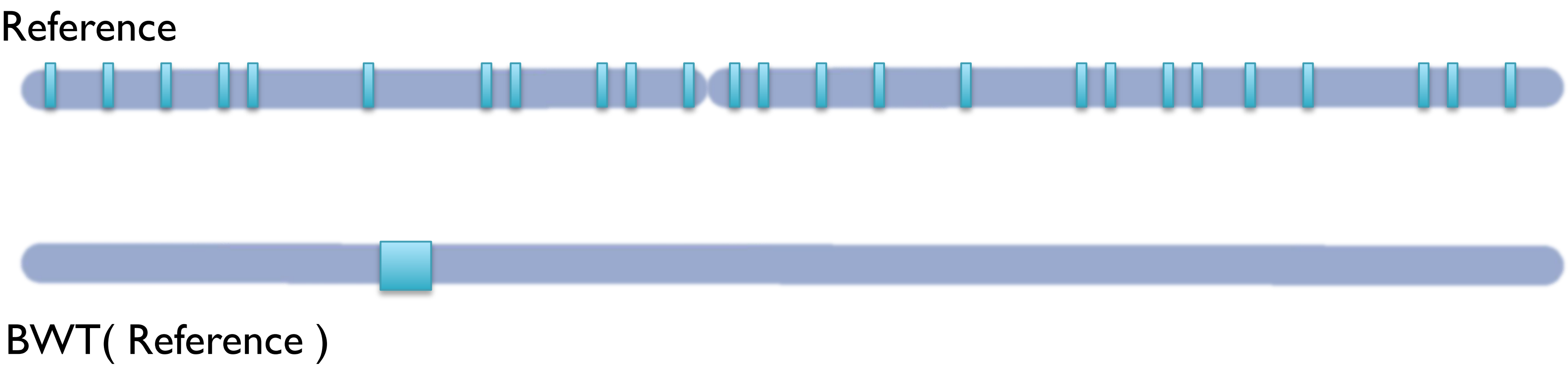
Reference



BWT(Reference)

Query:
AATGATACGGCGACCACCGAGATCTA

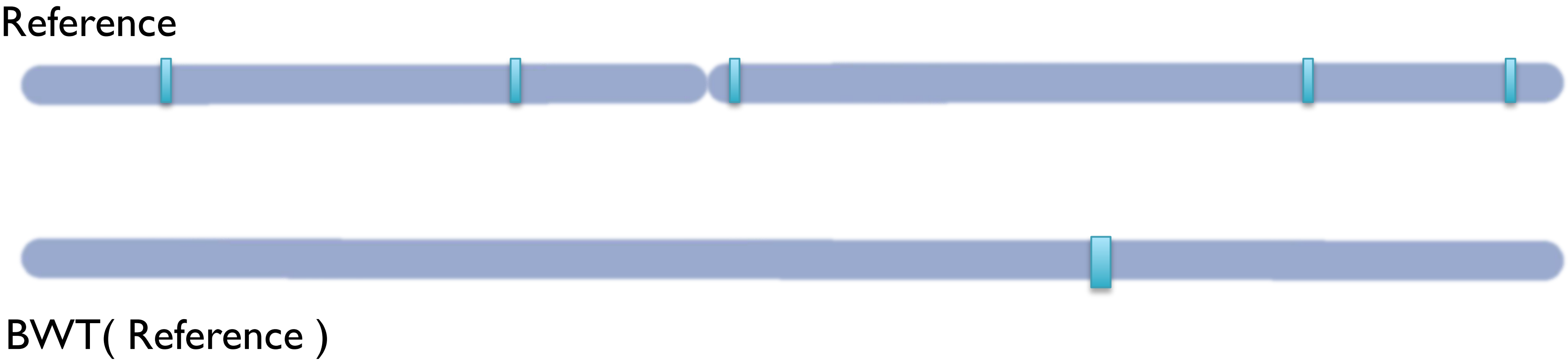
Bowtie algorithm


Reference

BWT(Reference)

Query:
AATGATACGGCGACCACCGAGATCTA

Bowtie algorithm



Query:
AATGATACGGCGACCACCGAGATCTA

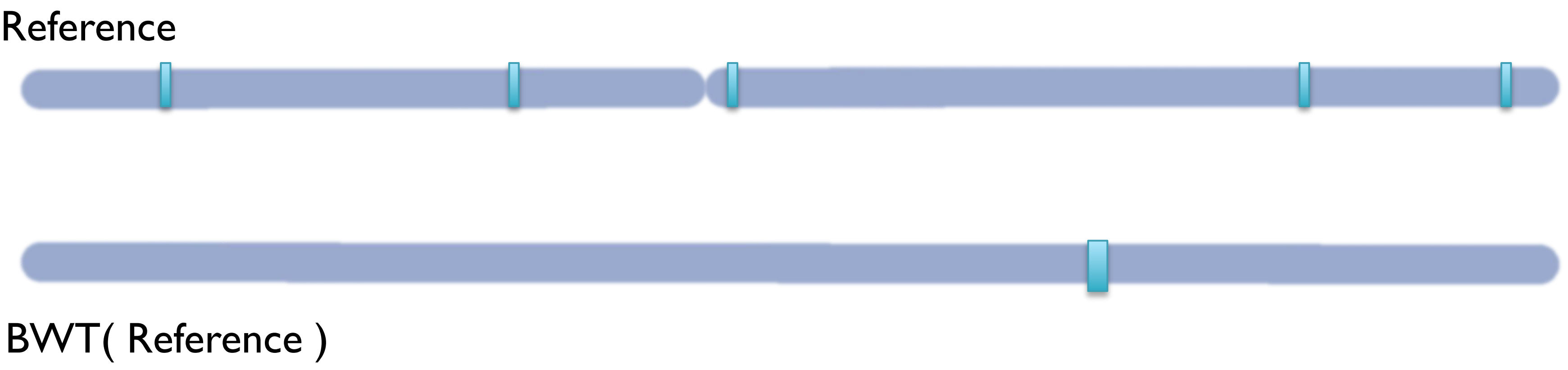
Bowtie algorithm

Reference

BWT(Reference)

Query:
AATGATACGGCGACCACCGAGATCTA

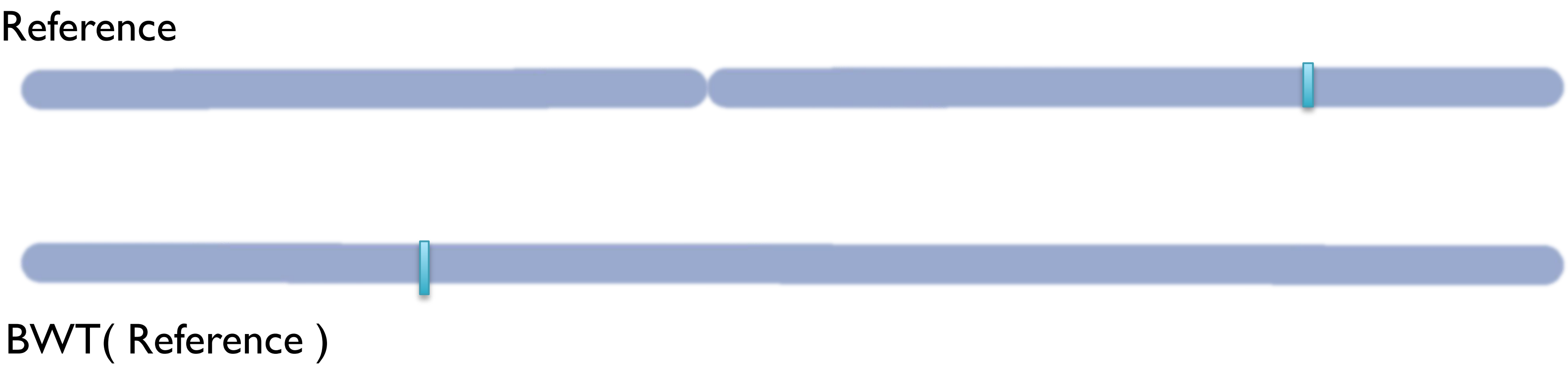
Bowtie algorithm



Query:

AATGTTACGGCGACCACCGAGATCTA

Bowtie algorithm



Query:

AATGTTACGGCGACCACCGAGATCTA

Bowtie performance

Length	Program	CPU time	Wall clock time	Peak virtual memory footprint (megabytes)	Bowtie speed-up	Reads aligned (%)
36 bp	Bowtie	7 m 11 s	7 m 20 s	[illegible]	-	[illegible]
	Maq	[illegible]	[illegible]	804	[illegible]	[illegible]
	Bowtie -v 2	[illegible]	[illegible]	[illegible]	-	[illegible]
	SOAP	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

- Seed-and-extend search of the BWT
  1. If we fail to reach the end, back-track and resume search
  2. The beginning of the read is used as high confidence seed

- Report the "best" n alignments
  1. Best = smallest hamming distance, possibly weighted by QV
  2. Some reads will have millions of equally good mapping positions

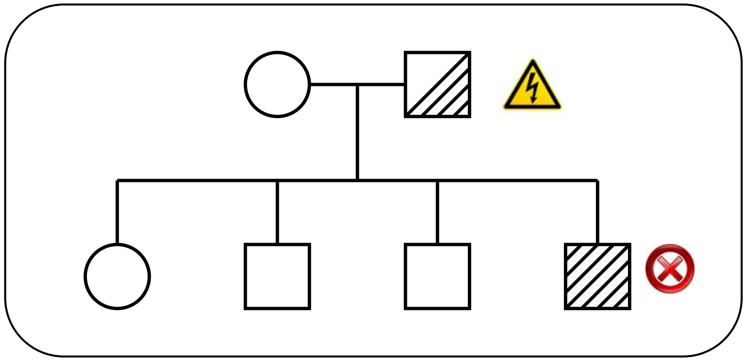
Recommendation Today: Use Bowtie2 or BWA

Outline

1. Rise of DNA Sequencing
2. Sequence Alignment Basics
3. Understanding Bowtie
4. Genetics of Autism

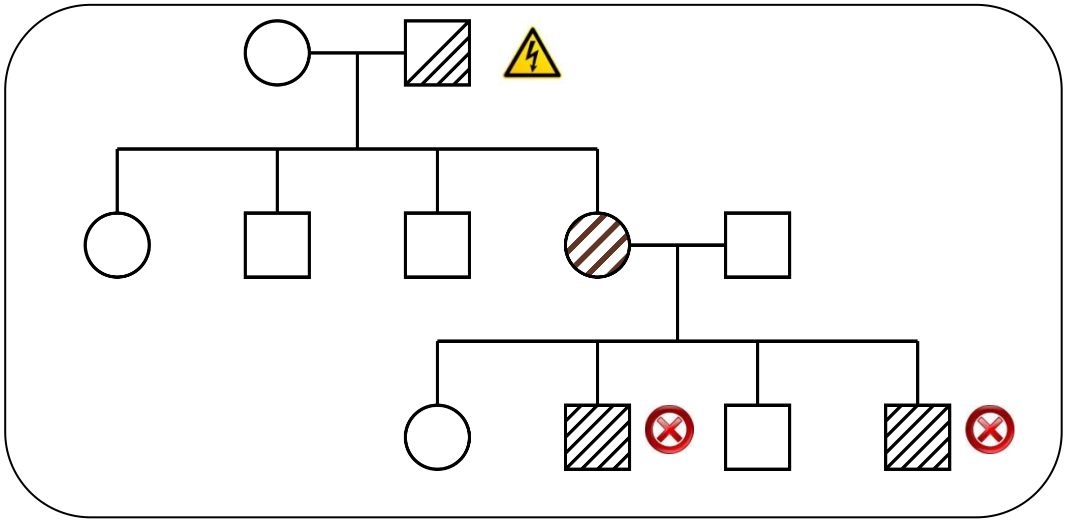
Unified Model of Autism

Sporadic Autism: 1 in 100



Prediction: De novo mutations of high penetrance contributes to autism, especially in low risk families with no history of autism.

Familial Autism: 90% concordance in twins



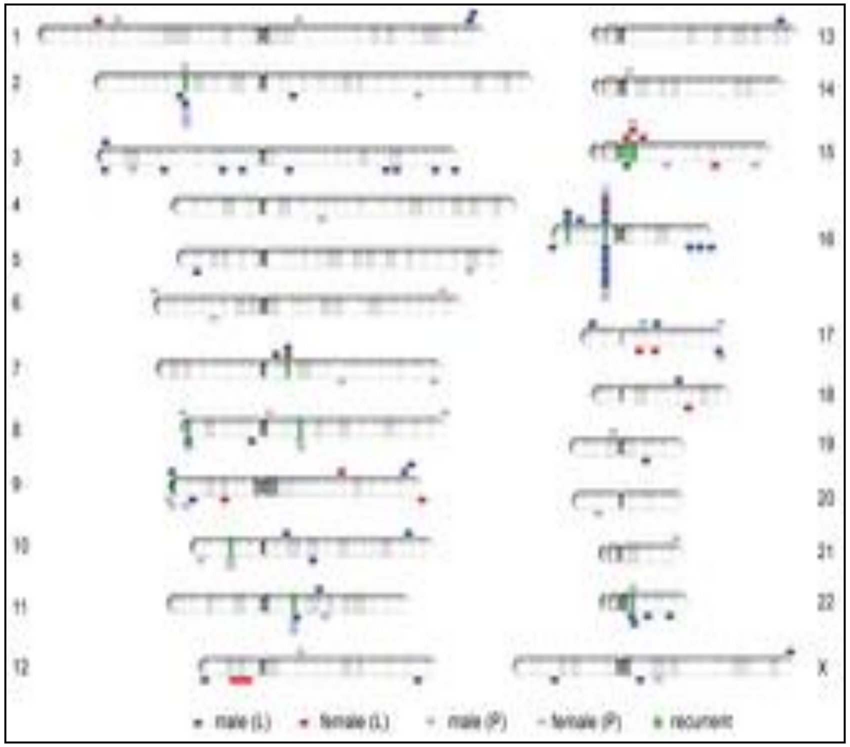
Legend

⚠ Sporadic mutation

⊗ Fails to procreate

A unified genetic theory for sporadic and inherited autism
Zhao *et al.* (2007) *PNAS. 104(31)12831-12836.*

Autism and de novo CNVs

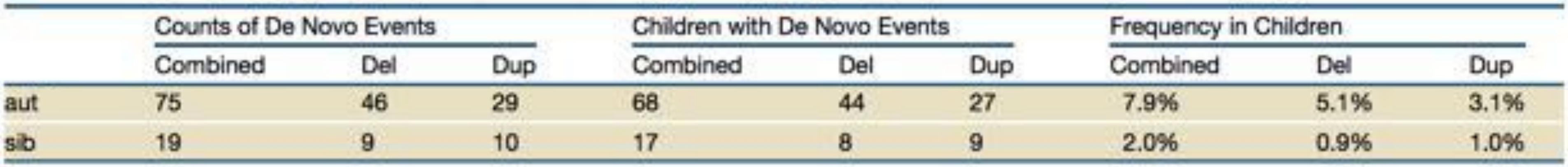


Analysis of Simons Simplex Collection

- CGH arrays of 510 family quads
- 94 total de novo CNVs discovered

De novo CNVs are more common in autistic children

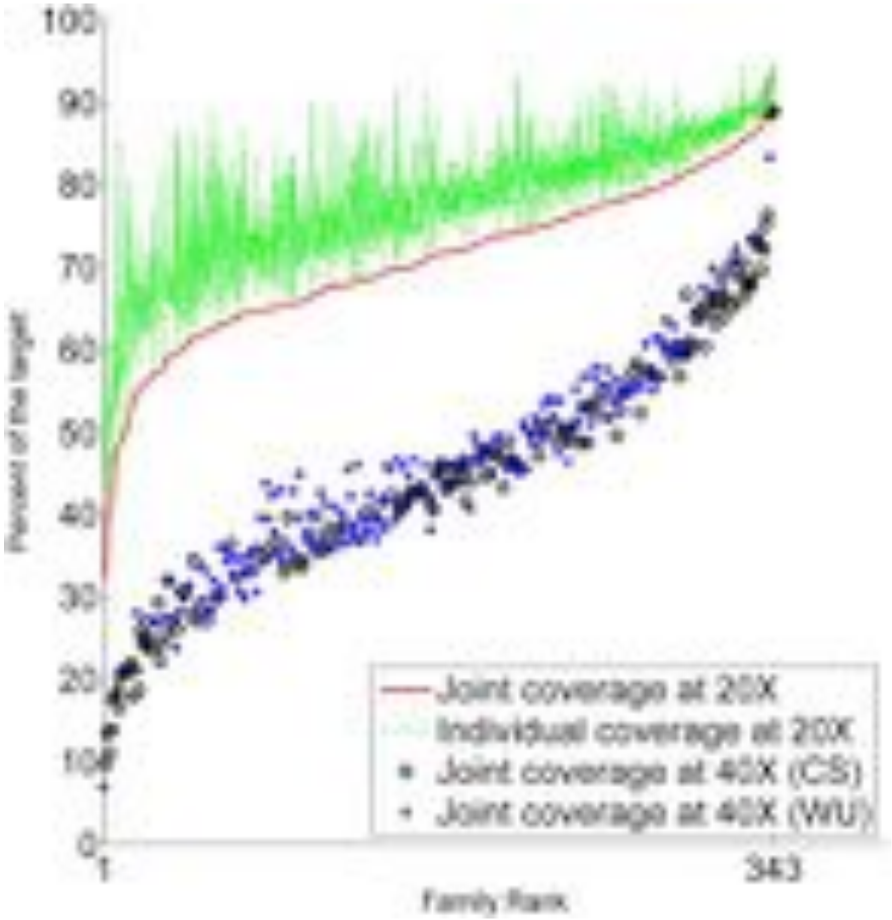
- 4:1 ratio in autistic kids relative to their non-autistic siblings
- Some recurrence at genes related to other psychiatric conditions



	Counts of De Novo Events			Children with De Novo Events			Frequency in Children		
	Combined	Del	Dup	Combined	Del	Dup	Combined	Del	Dup
aut	75	46	29	68	44	27	7.9%	5.1%	3.1%
sib	19	9	10	17	8	9	2.0%	0.9%	1.0%

Rare de novo and transmitted copy-number variation in autism spectrum disorders. Levy *et al.* (2011) *Neuron.* 70:886-897.

Exome-Capture and Sequencing



Sequencing of 343 families from the Simons Simplex Collection

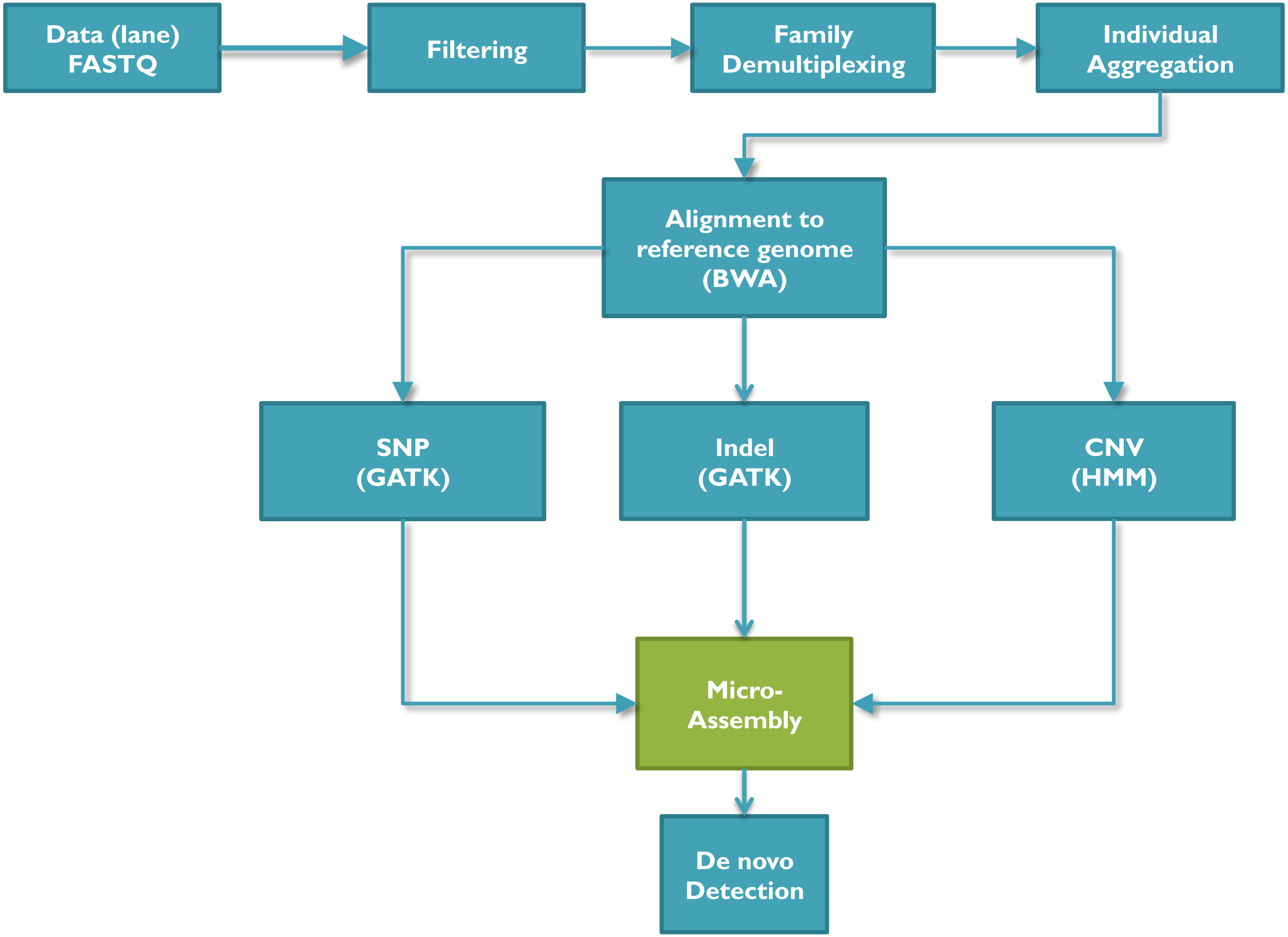
- Parents plus one child with autism and one non-autistic sibling
- Enriched for higher-functioning individuals

Families prepared and captured together to minimize batch effects

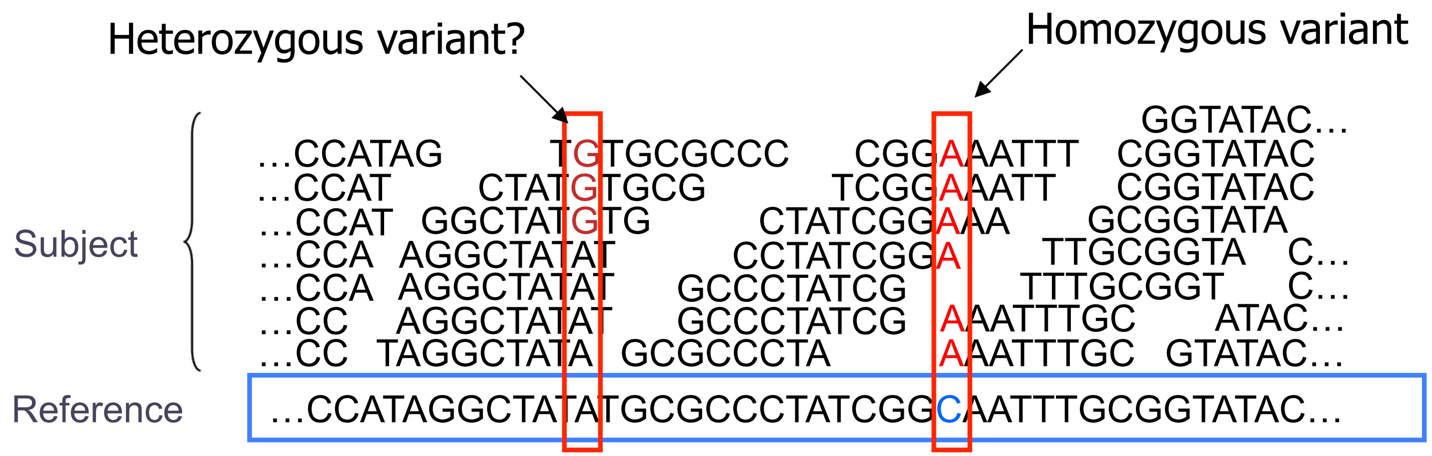
- Exome-capture performed with NimbleGen SeqCap EZ Exome v2.0 targeting 36 Mb of the genome.
- ~80% of the target at >20x coverage with ~93bp reads

De novo gene disruptions in children on the autism spectrum
Iossifov *et al.* (2012) *Neuron.* 74:2 285-299

Exome Sequencing Pipeline



Genotyping

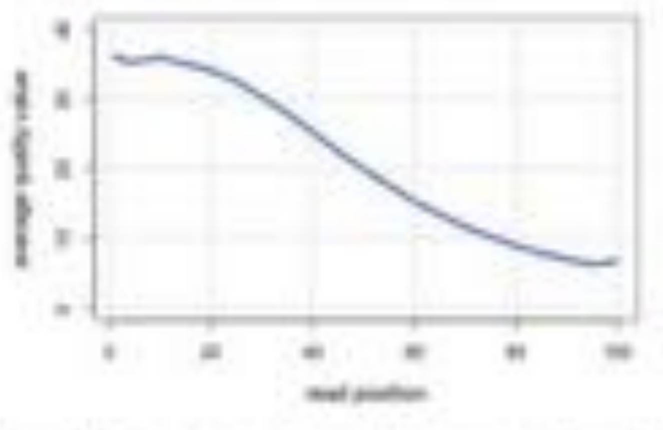


```
                                    GGTATAC...
...CCATAG    TGTGCGCCC   CGGAAATTT CGGTATAC
...CCAT   CTATGTGCG     TCGGAAATT  CGGTATAC
...CCAT GGCTATGTG    CTATCGGAAA   GCGGTATA
...CCA AGGCTATAT    CCTATCGGA   TTGCGGTA  C...
...CCA AGGCTATAT  GCCCTATCG    TTTGCGGT   C...
...CC  AGGCTATAT  GCCCTATCG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

- Sequencing instruments make mistakes
  - Quality of read decreases over the read length

- A single read differing from the reference is probably just an error, but it becomes more likely to be real as we see it multiple times
  - Often framed as a Bayesian problem of more likely to be a real variant or chance occurrence of N errors
  - Accuracy improves with deeper coverage

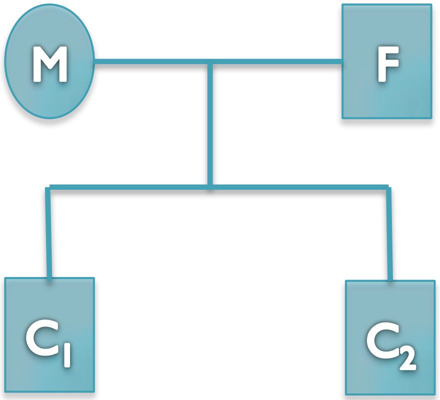
$$Q_{\text{sanger}} = -10 \log_{10} p$$



De novo mutation discovery and validation

Concept: Identify mutations not present in parents.

Challenge: Sequencing errors in the child or low coverage in parents lead to false positive de novos



```
Ref:     ...TCAGAACAGCTGGATGAGATCTTAGCCAACTACCAGGAGATTGTCTTTGCCCGGA...

Father:  ...TCAGAACAGCTGGATGAGATCTTAGCCAACTACCAGGAGATTGTCTTTGCCCGGA...
Mother:  ...TCAGAACAGCTGGATGAGATCTTAGCCAACTACCAGGAGATTGTCTTTGCCCGGA...
Sib:     ...TCAGAACAGCTGGATGAGATCTTAGCCAACTACCAGGAGATTGTCTTTGCCCGGA...
Aut(1):  ...TCAGAACAGCTGGATGAGATCTTAGCCAACTACCAGGAGATTGTCTTTGCCCGGA...
Aut(2):  ...TCAGAACAGCTGGATGAGATCTTACC------CCGGGAGATTGTCTTTGCCCGGA...
```

6bp heterozygous deletion at chr13:25280526 ATP12A

De novo Genetics of Autism

- In 343 family quads so far, we see significant enrichment in de novo ***likely gene killers*** in the autistic kids
  - Overall rate basically 1:1 (432:396)
  - 2:1 enrichment in nonsense mutations
  - 2:1 enrichment in frameshift indels
  - 4:1 enrichment in splice-site mutations
  - Most de novo originate in the paternal line in an age-dependent manner (56:18 of the mutations that we could determine)

- Observe strong overlap with the 842 genes known to be associated with fragile X protein FMPR
  - Related to neuron development and synaptic plasticity

De novo gene disruptions in children on the autism spectrum
Iossifov *et al.* (2012) *Neuron.* 74:2 285-299

Computational Biology

"Computer science is no more about computers than astronomy is about telescopes."
Edsger Dijkstra

- Computer Science = Science of Computation
  - Solving problems, designing & building systems
  - Computers are very, very dumb, but we can instruct them
    - Build complex systems out of simple components
    - They will perfectly execute instructions forever

- CompBio = Thinking Computationally about Biology
  - Processing: Make more powerful instruments, analyze results
  - Designing & Understanding: protocols, procedures, systems

"Think Harder & Compute Less"
Dan Gusfield

Modern Biology Challenges

The foundations of biology will continue to be *observation*, *experimentation*, and *interpretation*

- Technology will continue to push the frontier
- Measurements will be made *digitally* over large populations, at extremely high resolution, and for diverse applications

Rise in Quantitative and Computational Demands

1. *Experimental design*: selection, collection & metadata
2. *Observation*: measurement, storage, transfer, computation
3. *Integration*: multiple samples, assays, analyses
4. *Discovery*: visualizing, interpreting, modeling

Ultimately limited by the human capacity to execute extremely complex experiments and interpret results

Questions?

http://schatzlab.cshl.edu