

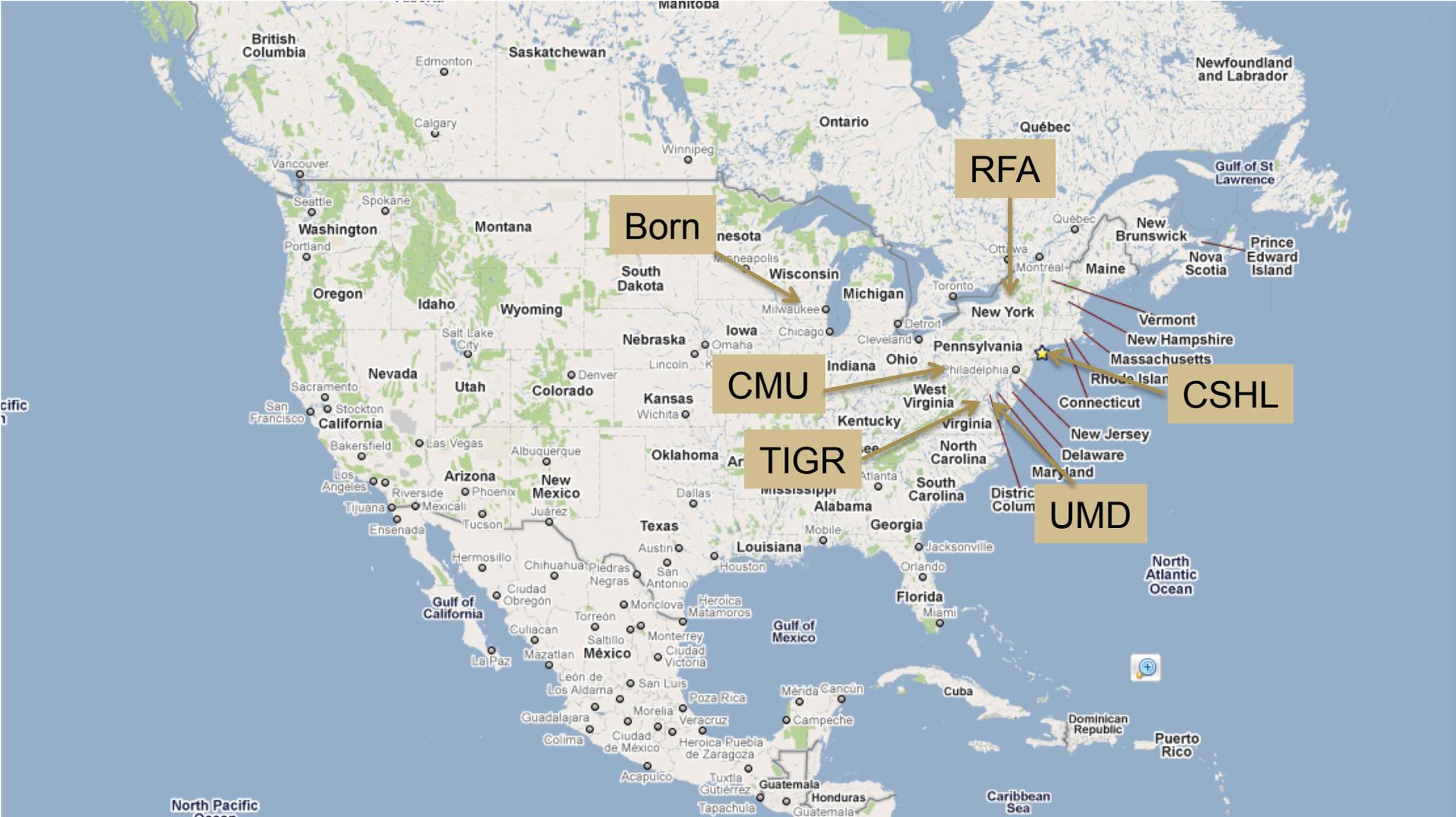
Searching for GATTACA

Michael Schatz

Bioinformatics Lecture I
Undergraduate Research Program 2013



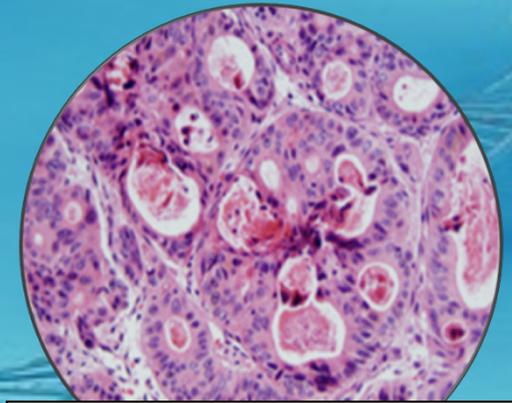
A Little About Me



Schatz Lab Overview



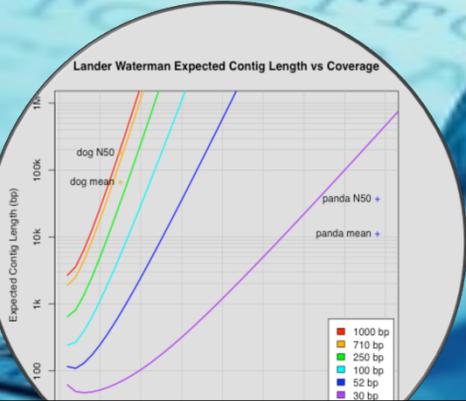
Computation



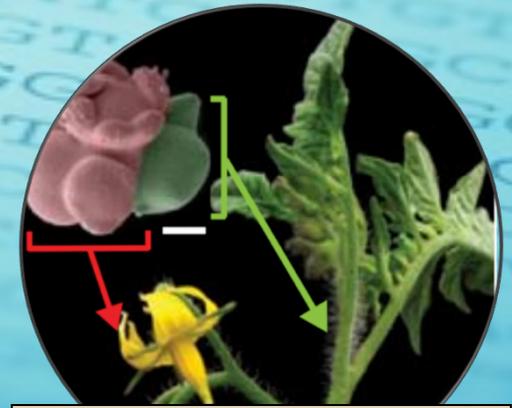
Human Genetics



Sequencing

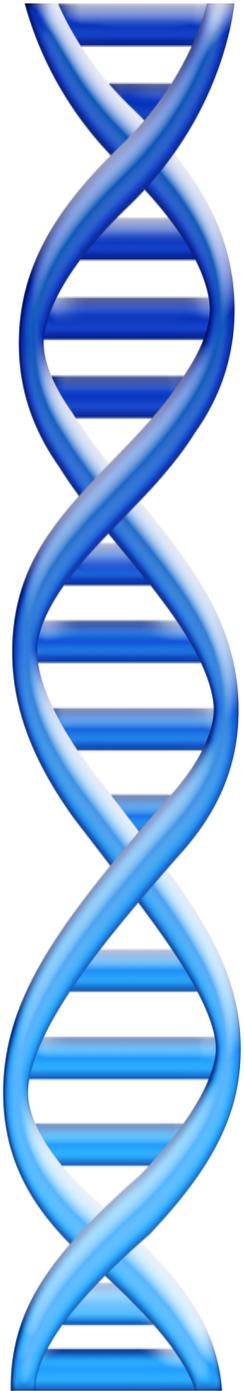


Modeling



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 Φ X174 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

A DNA sequence for the genome of bacteriophage Φ X174 of approximately 5,375 nucleotides has been determined using the rapid and simple 'plus and minus' method. The sequence identifies many of the features responsible for the production of the proteins of the nine known genes of the organism, including initiation and termination sites for the proteins and RNAs. Two pairs of genes are coded by the same region of DNA using different reading frames.

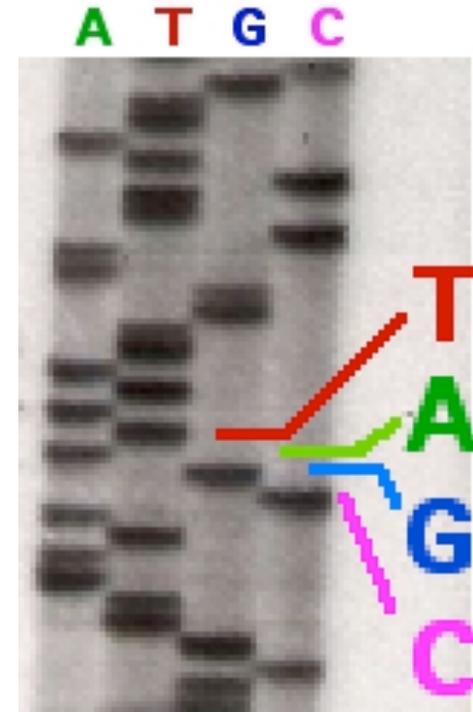
The genome of bacteriophage Φ X174 is a single-stranded, circular DNA of approximately 5,400 nucleotides coding for nine known proteins. The order of these genes, as determined by genetic techniques¹⁻⁴, is *A-B-C-D-E-F-G-H*. Genes *F*, *G* and *H* code for structural proteins of the virus capsid, and gene *J* (as defined by sequence work) codes for a small basic protein

strand DNA of Φ X has the same sequence as the mRNA and, in certain conditions, will bind ribosomes so that a protected fragment can be isolated and sequenced. Only one major site was found. By comparison with the amino acid sequence data it was found that this ribosome binding site sequence coded for the initiation of the gene *G* protein¹⁵ (positions 2,362-2,413).

At this stage sequencing techniques using primed synthesis with DNA polymerase were being developed¹⁶ and Schott¹⁷ synthesised a decanucleotide with a sequence complementary to part of the ribosome binding site. This was used to prime into the intergenic region between the *F* and *G* genes, using DNA polymerase and ³²P-labelled triphosphates¹⁸. The ribo-substitution technique¹⁶ facilitated the sequence determination of the labelled DNA produced. This decanucleotide-primed system was also used to develop the plus and minus method¹. Suitable synthetic primers are, however, difficult to prepare and as

1977

**1st Complete Organism
Bacteriophage ϕ X174
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 ϕ X174 DNA
Sanger, F. et al. (1977) *Nature*. 265: 687 - 695

Milestones in Molecular Biology



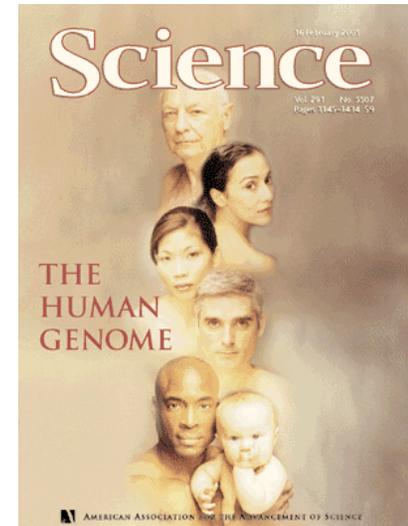
1995

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 =
1 Gbp / day



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



2011
PacBio
SMRT Sequencing
Current Specs (RS 2):
3kbp-20kbp reads
~1 Gbp / day

The DNA Data Race

Year	Genome	Technology	Cost
2001	Venter/IHGHC <i>et al.</i>	Sanger (ABI)	\$3,000,000,000
2007	Levy <i>et al.</i>	Sanger (ABI)	\$10,000,000
2008	Wheeler <i>et al.</i>	Roche (454)	\$2,000,000
2008	Ley <i>et al.</i>	Illumina	\$1,000,000
2008	Bentley <i>et al.</i>	Illumina	\$250,000
2009	Pushkarev <i>et al.</i>	Helicos	\$48,000
2009	Drmanac <i>et al.</i>	Complete Genomics	\$4,400
2013		Illumina/Complete/Ion Torrent	~\$1,000

Sequencing a single human genome uses ~100 GB of compressed sequence data in billions of short reads.

~20 DVDs / genome



Unsolved Questions in Biology

There is tremendous interest to sequence:

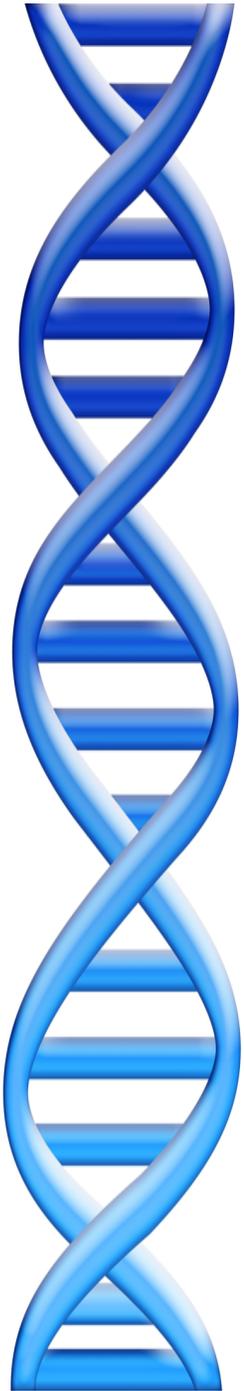
- What is your genome sequence?
- How does your genome compare to my genome?
- Where are the genes and how active are they?
- How does gene activity change during development?
- How does splicing change during development?
- How does methylation change during development?
- How does chromatin change 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 do your mutations relate to disease?

- W
- .. **Answering these questions requires specialized software & quantitative analysis**



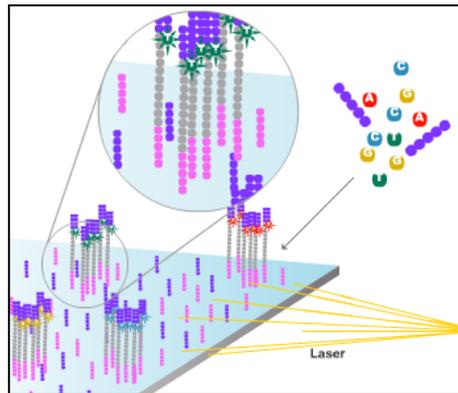
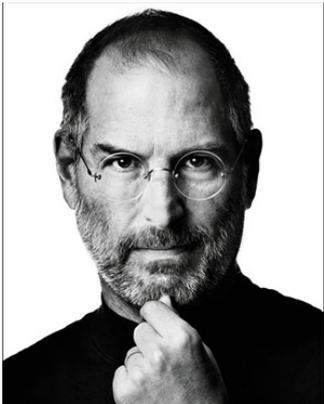
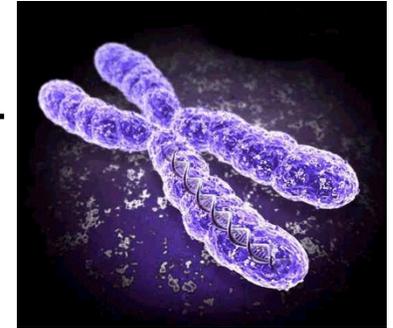
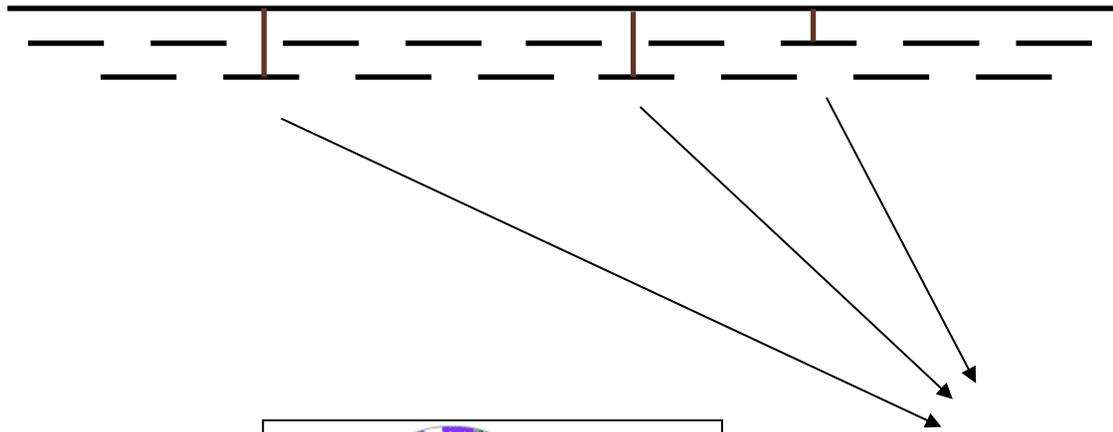
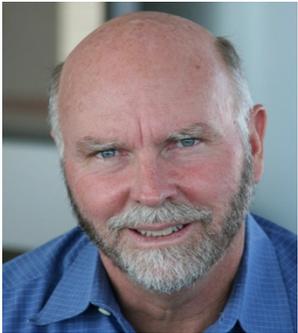
Outline



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

Personal Genomics

How does your genome compare to the reference?



Heart Disease
Cancer
Creates magical
technology

Searching for GATTACA

- Where is GATTACA in the human genome?
- Strategy I: 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 I: 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...

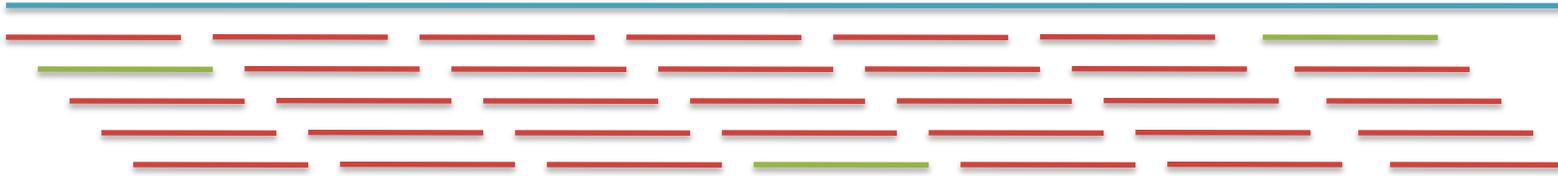
Searching for GATTACA

- Where is GATTACA in the human genome?
- Strategy I: 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?]

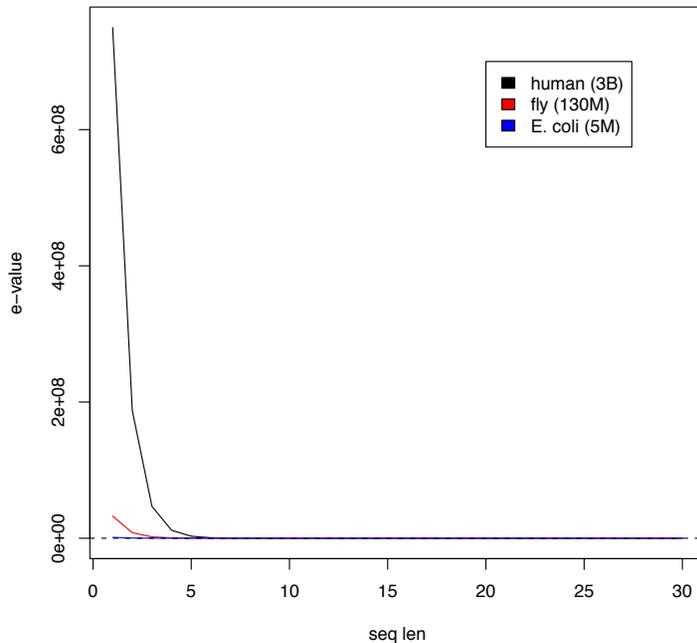
Expected Occurrences

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

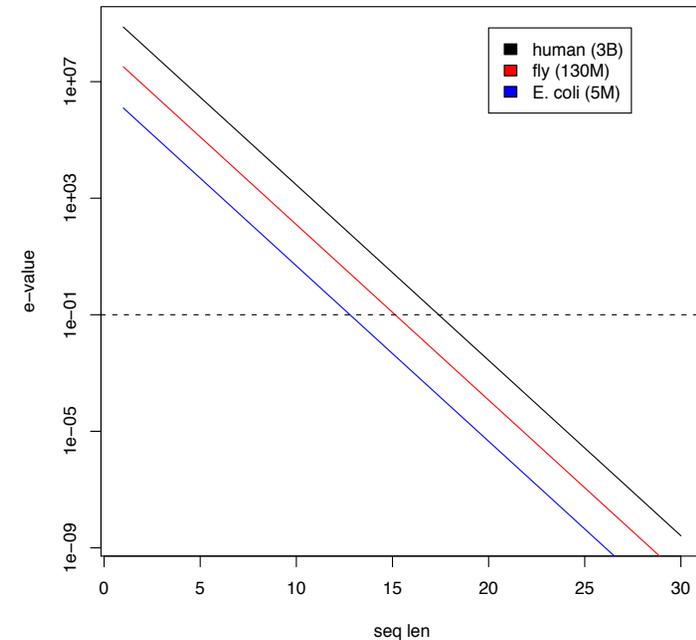
- 1 in 4 bases are G, 1 in 16 positions are GA, 1 in 64 positions are GAT, ...
- 1 in 16,384 should be GATTACA
- $E = n / (4^m)$

[183,105 expected occurrences]
[How long do the reads need to be for a significant match?]

Value and sequence length
cutoff 0.1



E-value and sequence length
cutoff 0.1



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

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
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

#	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: $Hi = Mid - 1$
 - $Lo = 9; Hi = 11;$

#	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
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
=> Lower: $Hi = Mid - 1$
 - $Lo = 9; Hi = 9;$

#	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
 - => Lower: $Hi = Mid - 1$
 - $Lo = 9; Hi = 9; Mid = (9+9)/2 = 9$
 - Middle = Suffix[9] = GATTACA...
 - => Match at position 2!

#	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
→

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: $n/(2^x) \leq 1$; $x = \lg_2(n)$

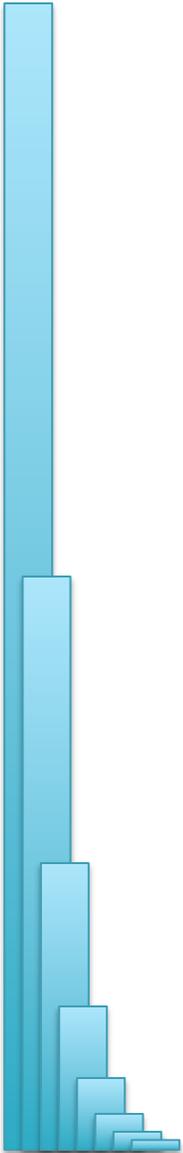
[32]

- 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?]

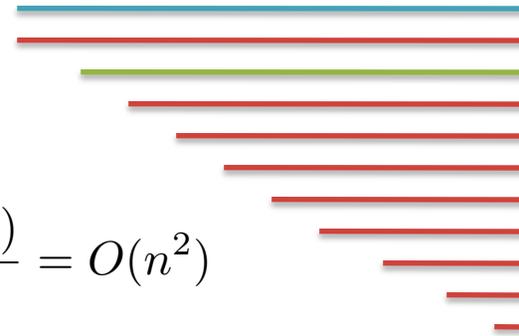
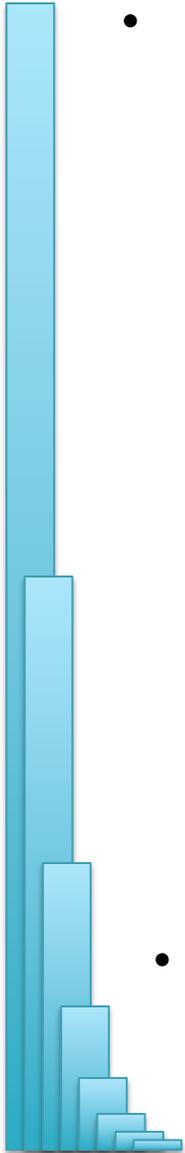


Suffix Array Construction

- How can we store the suffix array?
[How many characters are in all suffixes combined?]

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

- 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

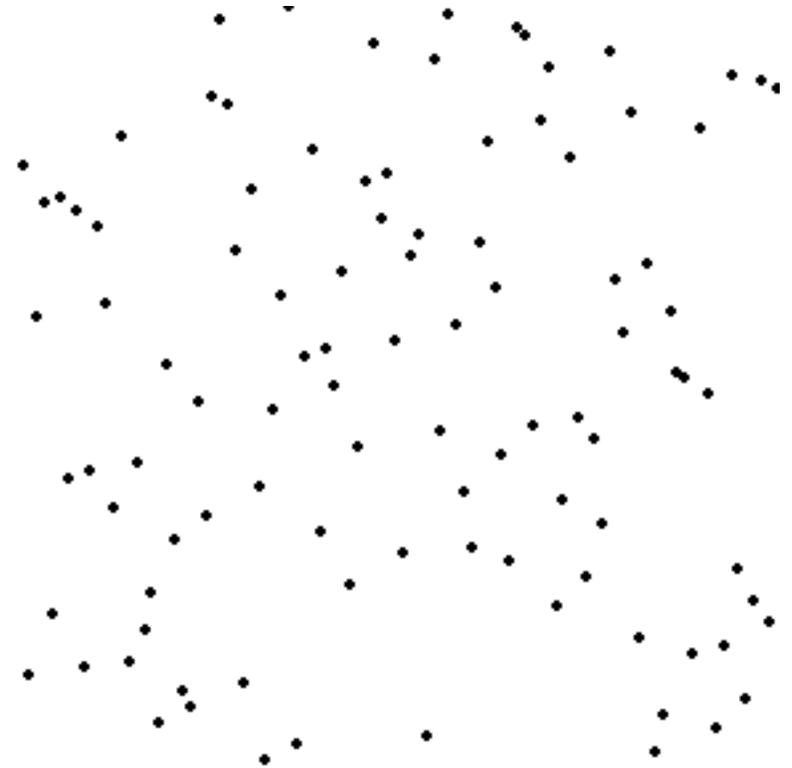
Sorting

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, 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



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

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

- Outer loop: pos = 1 to n

- Inner loop: check = pos to n

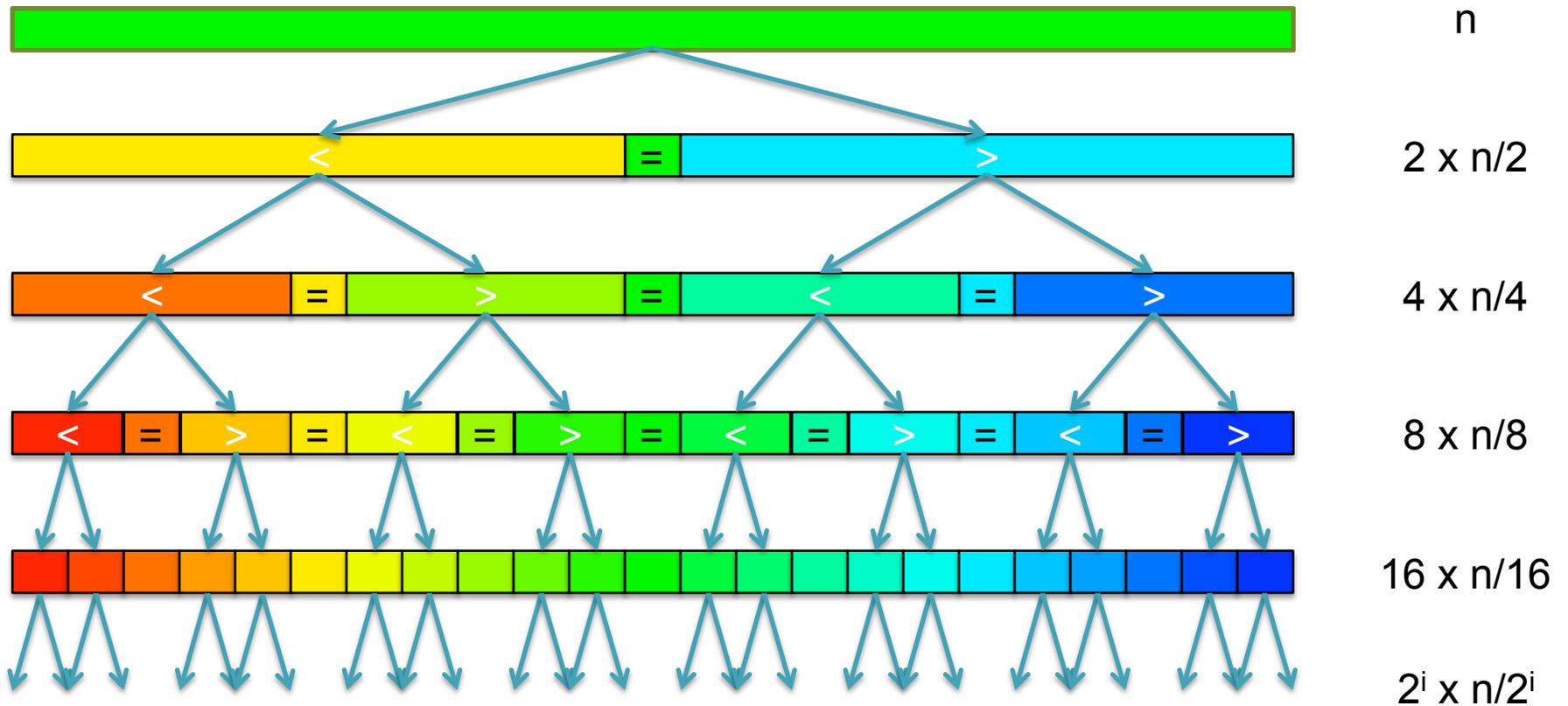
- Running time: Outer * Inner = $O(n^2)$

[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



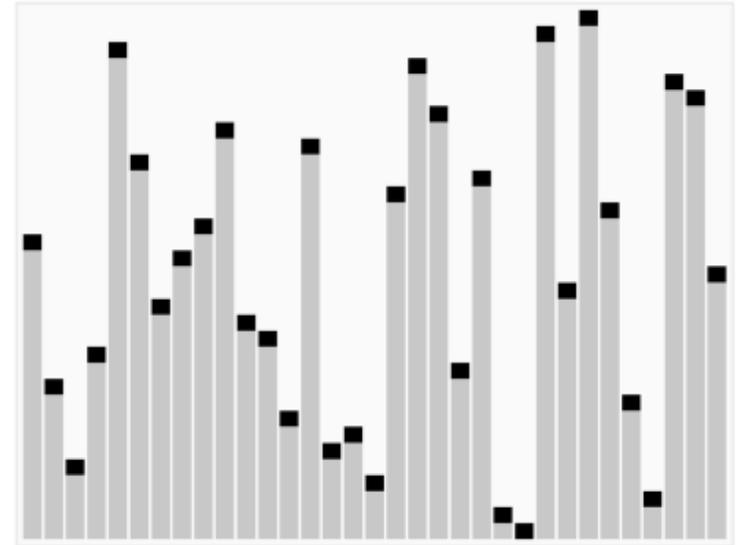
[How many times can we split a list in half?]

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 $O(n)$)

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

$$T(n) = n + 2\left(\frac{n}{2}\right) + 4\left(\frac{n}{4}\right) + \cdots + n\left(\frac{n}{n}\right) = \sum_{i=0}^{\lg(n)} \frac{2^i n}{2^i} = \sum_{i=0}^{\lg(n)} n = O(n \lg n) \quad [\sim 94B]$$

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])
```

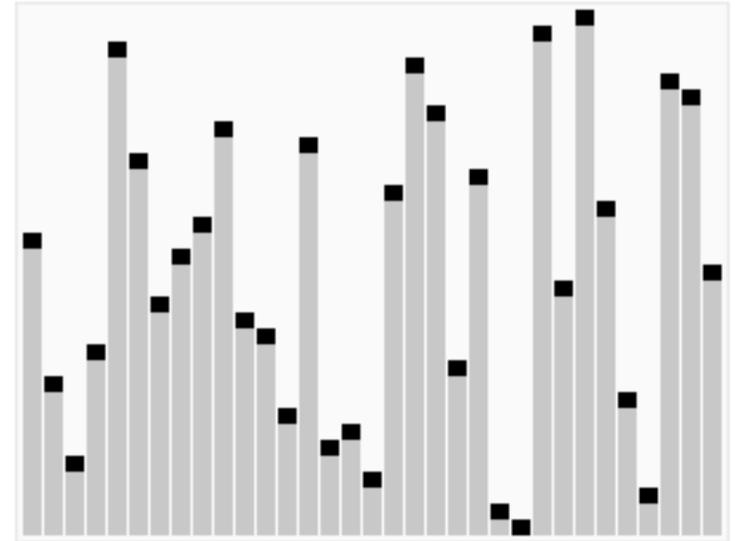
```
// recurse on sublists
```

```
return (append(QuickSort(lo), QuickSort(hi)))
```

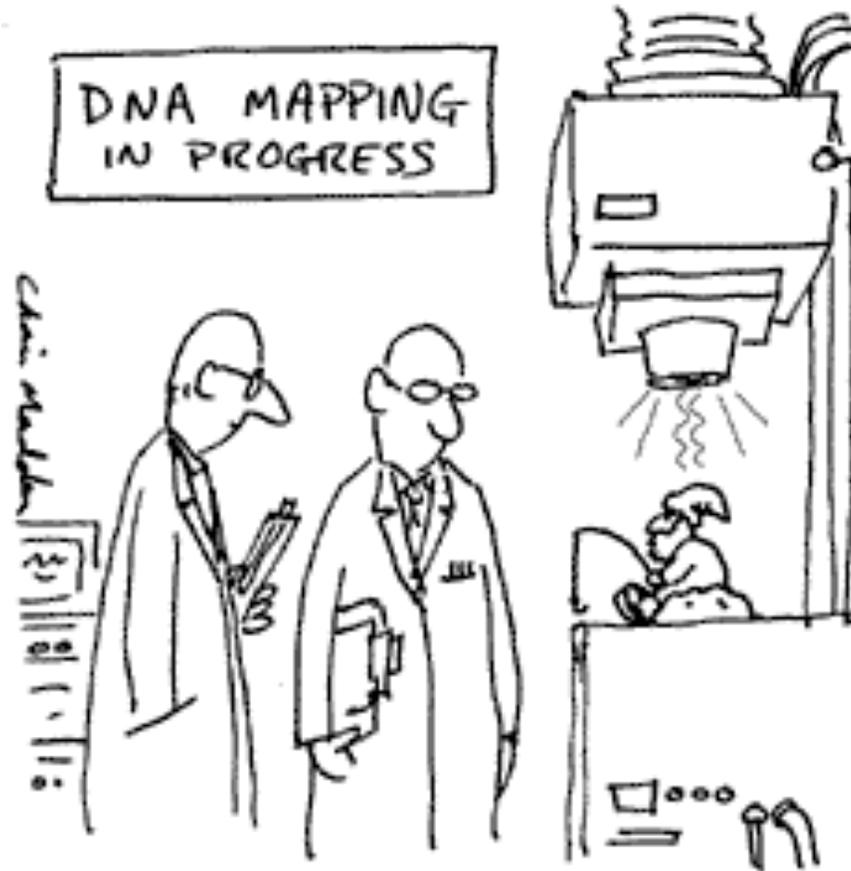
- Analysis (Assume we can find the median in $O(n)$)

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

$$T(n) = n + 2\left(\frac{n}{2}\right) + 4\left(\frac{n}{4}\right) + \dots + n\left(\frac{n}{n}\right) = \sum_{i=0}^{\lg(n)} \frac{2^i n}{2^i} = \sum_{i=0}^{\lg(n)} n = O(n \lg n) \quad [\sim 94B]$$



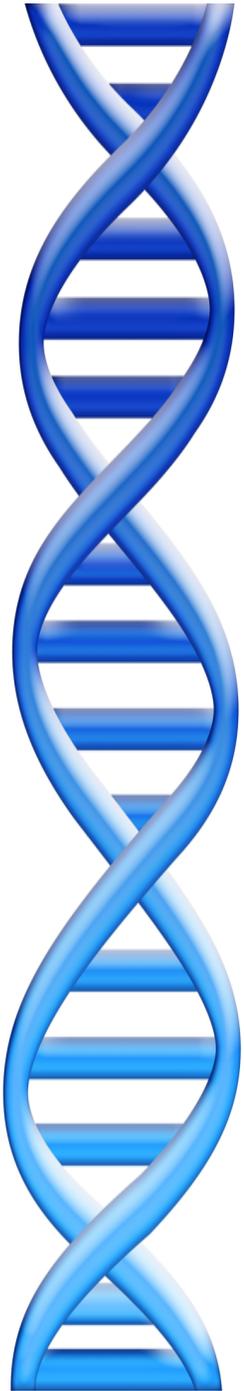
<http://en.wikipedia.org/wiki/Quicksort>



THE G-NOME PROJECT

2 minute break

Outline



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



Fast gapped-read alignment with Bowtie 2

Ben Langmead and Steven Salzberg (2012) Nature Methods. 9, 357–359

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

Similarity metrics

- Hamming distance

- Count the number of substitutions to transform one string into another

GATTACA	ATTACCC
X	XX XX X
GATCACA	GATTACA
1	5

- Edit distance

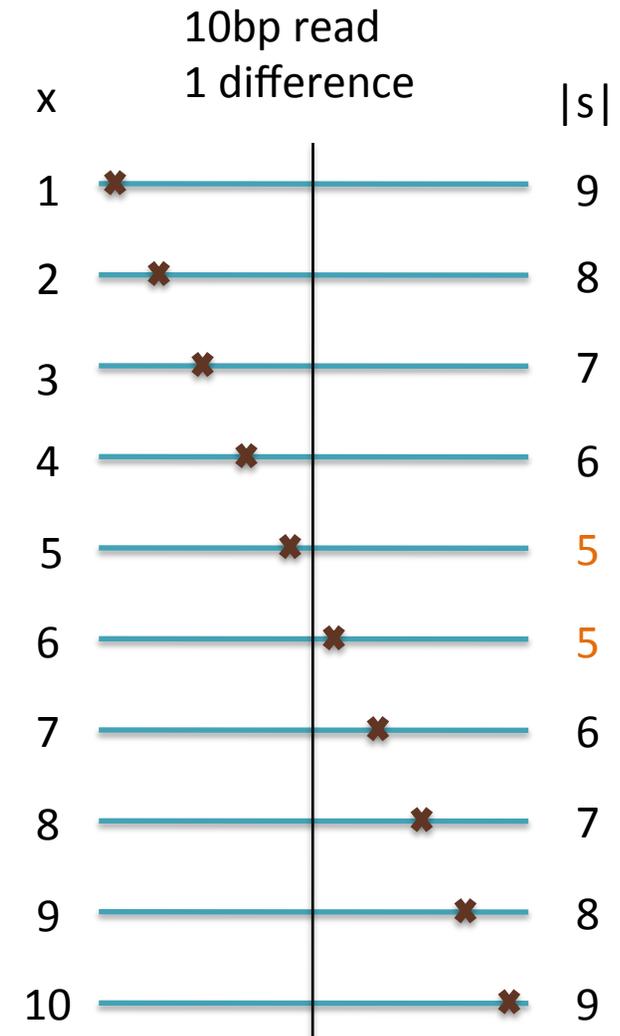
- The minimum number of substitutions, insertions, or deletions to transform one string into another

GATTACA	-ATTACCC
X	X XX
GATCACA	GATTAC-A
1	3

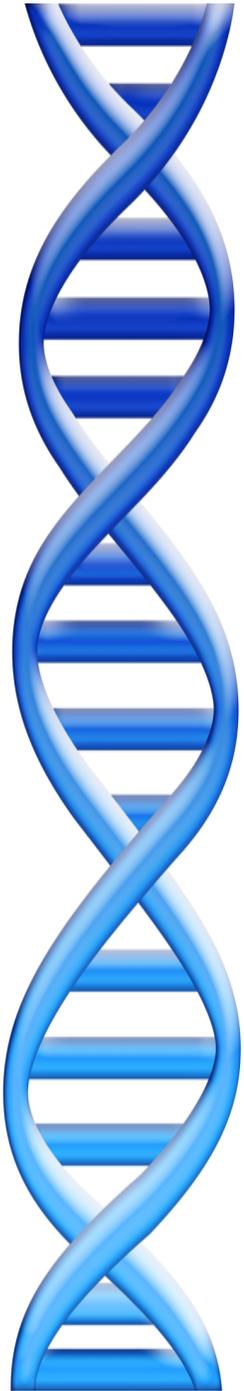
Seed-and-Extend Alignment

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



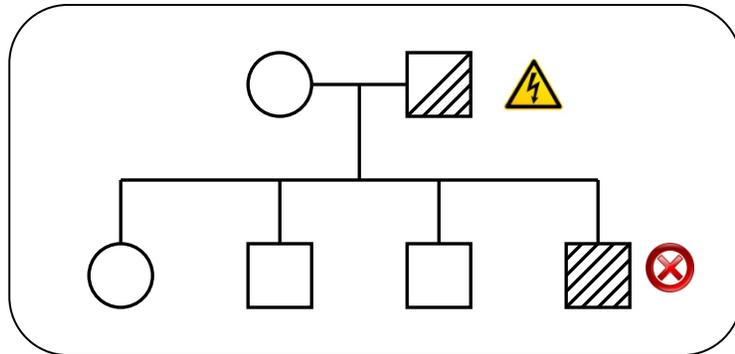
Outline



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

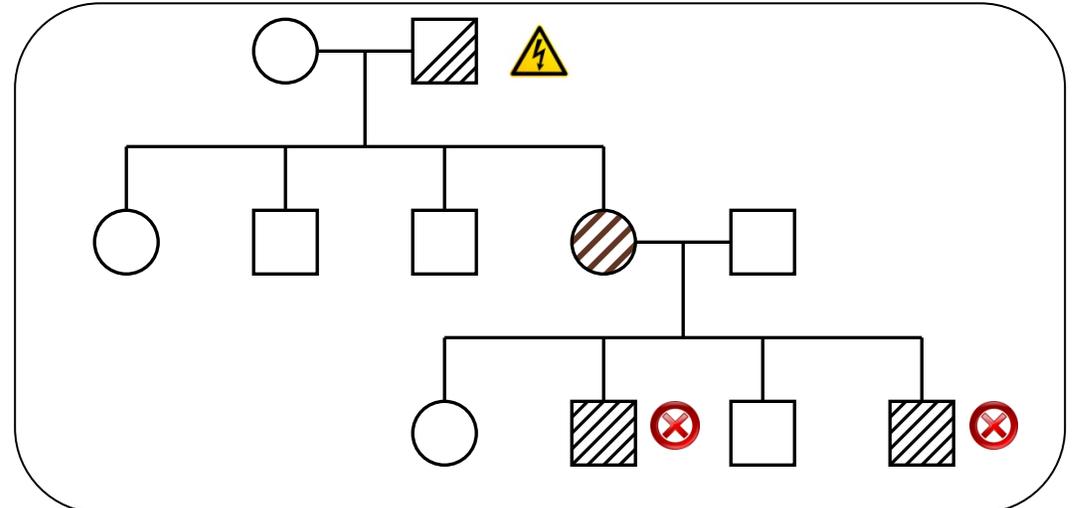
Unified Model of Autism

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

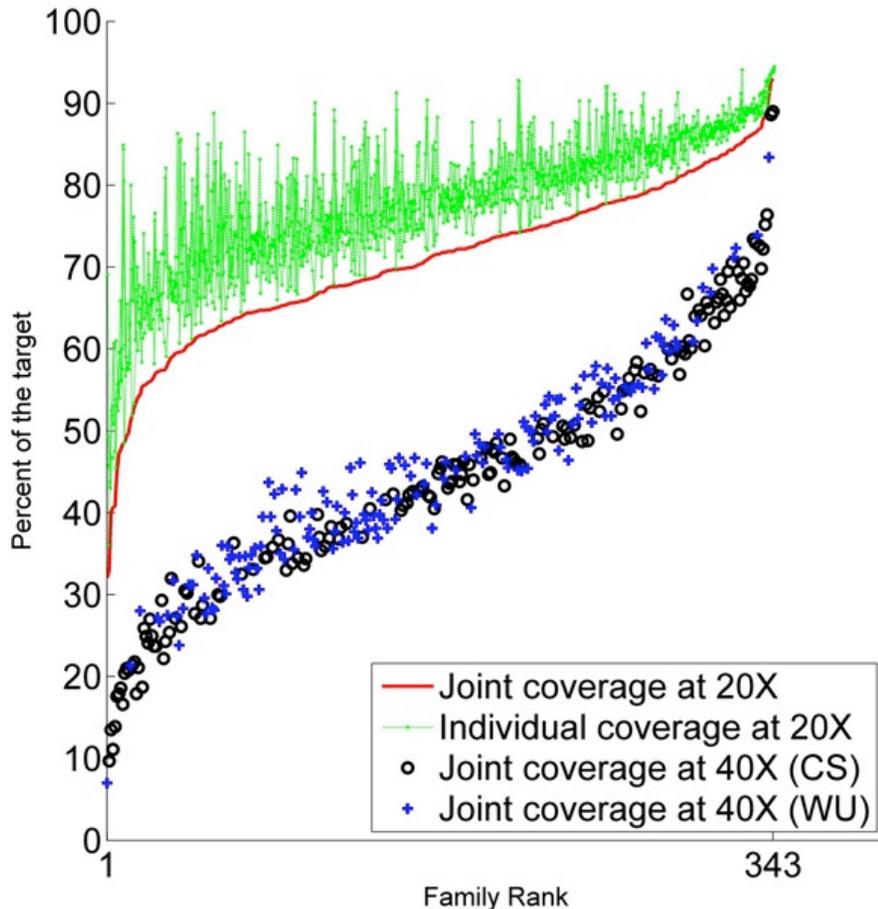


Fails to procreate

A unified genetic theory for sporadic and inherited autism

Zhao *et al.* (2007) *PNAS*. 104(31)12831-12836.

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

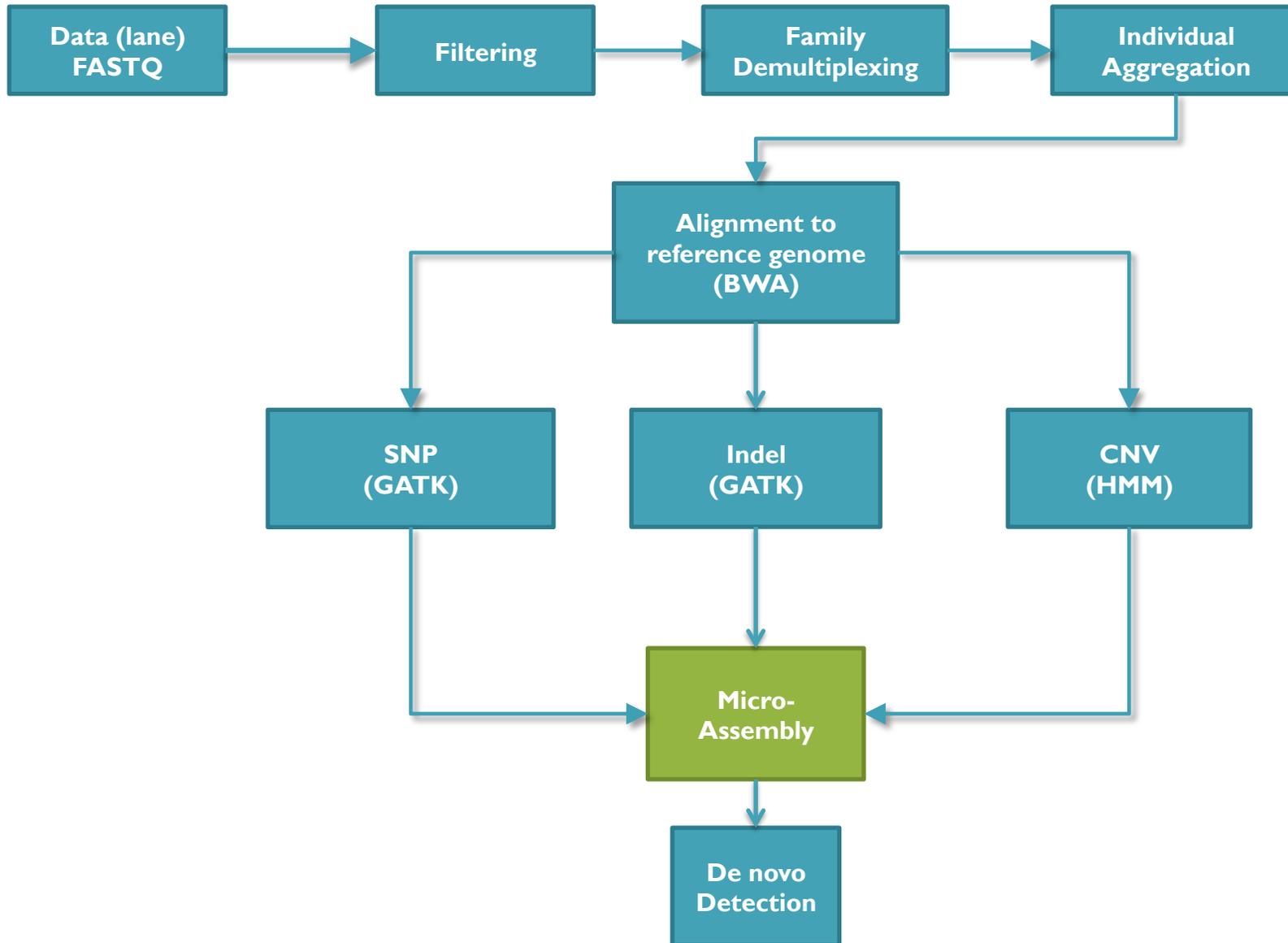
Families prepared and captured together to minimize batch effects

- 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

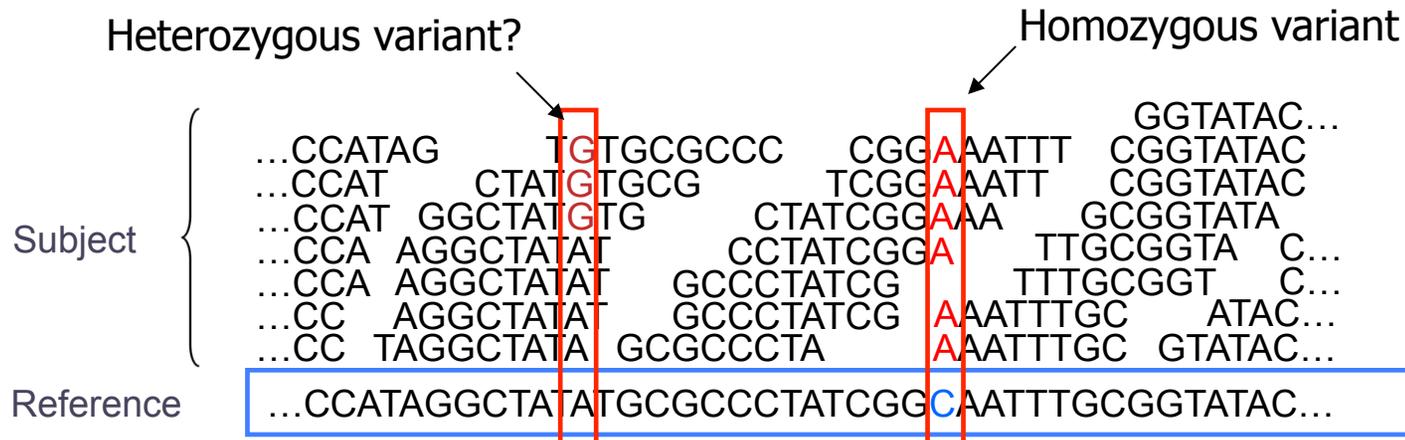
De novo gene disruptions in children on the autism spectrum

lossifov *et al.* (2012) *Neuron*. 74:2 285-299

Exome Sequencing Pipeline

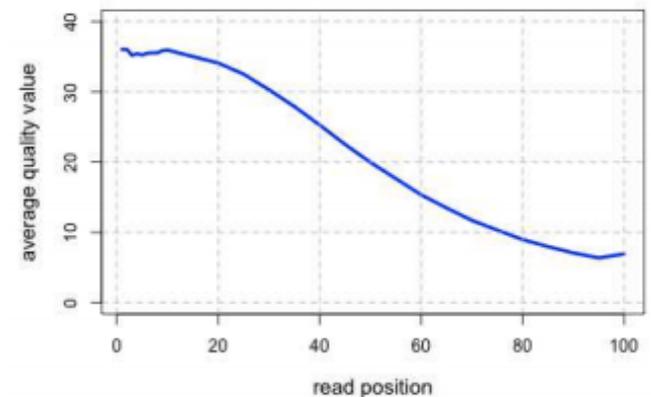


Genotyping



- 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

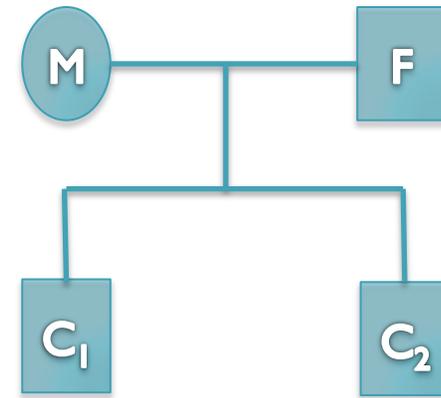
$$Q_{\text{sanger}} = -10 \log_{10} p$$



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 FMRP
 - 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

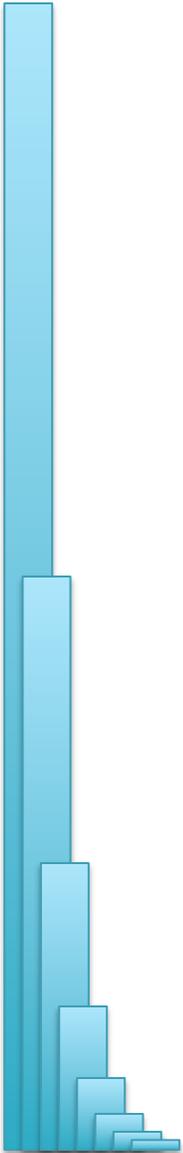
Computer Science = Science of Computation

- 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



Challenges of Modern Science



The foundations of science 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?

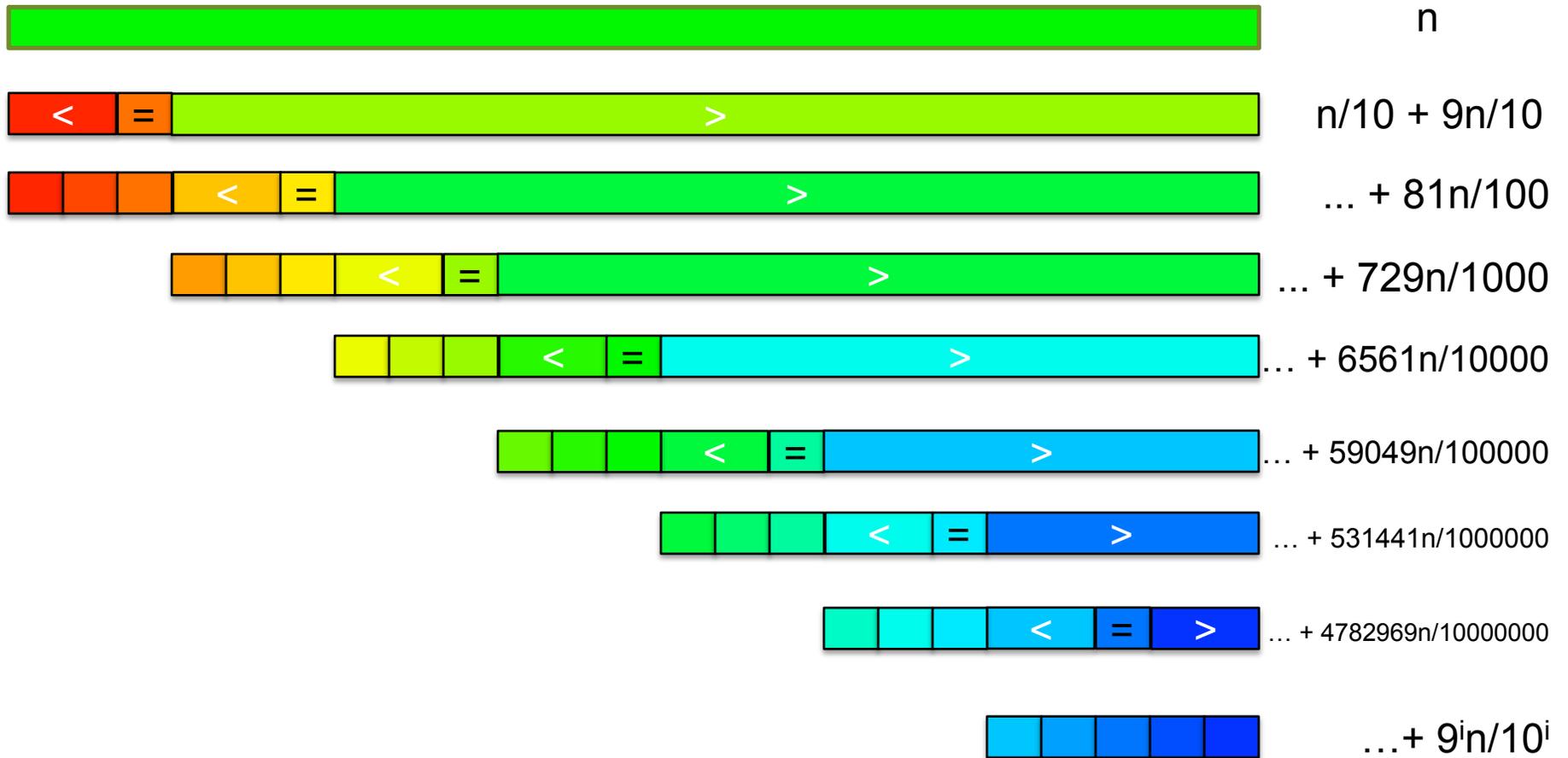
<http://schatzlab.cshl.edu>

@mike_schatz



Picking the Median

- What if we miss the median and do a 90/10 split instead?



[How many times can we cut 10% off a list?]

Randomized Quicksort

- **90/10 split runtime analysis**

Find smallest x s.t.

$$T(n) = n + T\left(\frac{n}{10}\right) + T\left(\frac{9n}{10}\right)$$

$$(9/10)^x n \leq 1$$

$$T(n) = n + \frac{n}{10} + T\left(\frac{n}{100}\right) + T\left(\frac{9n}{100}\right) + \frac{9n}{10} + T\left(\frac{9n}{100}\right) + T\left(\frac{81n}{100}\right)$$

$$(10/9)^x \geq n$$

$$T(n) = n + n + T\left(\frac{n}{100}\right) + 2T\left(\frac{9n}{100}\right) + T\left(\frac{81n}{100}\right)$$

$$x \geq \log_{10/9} n$$

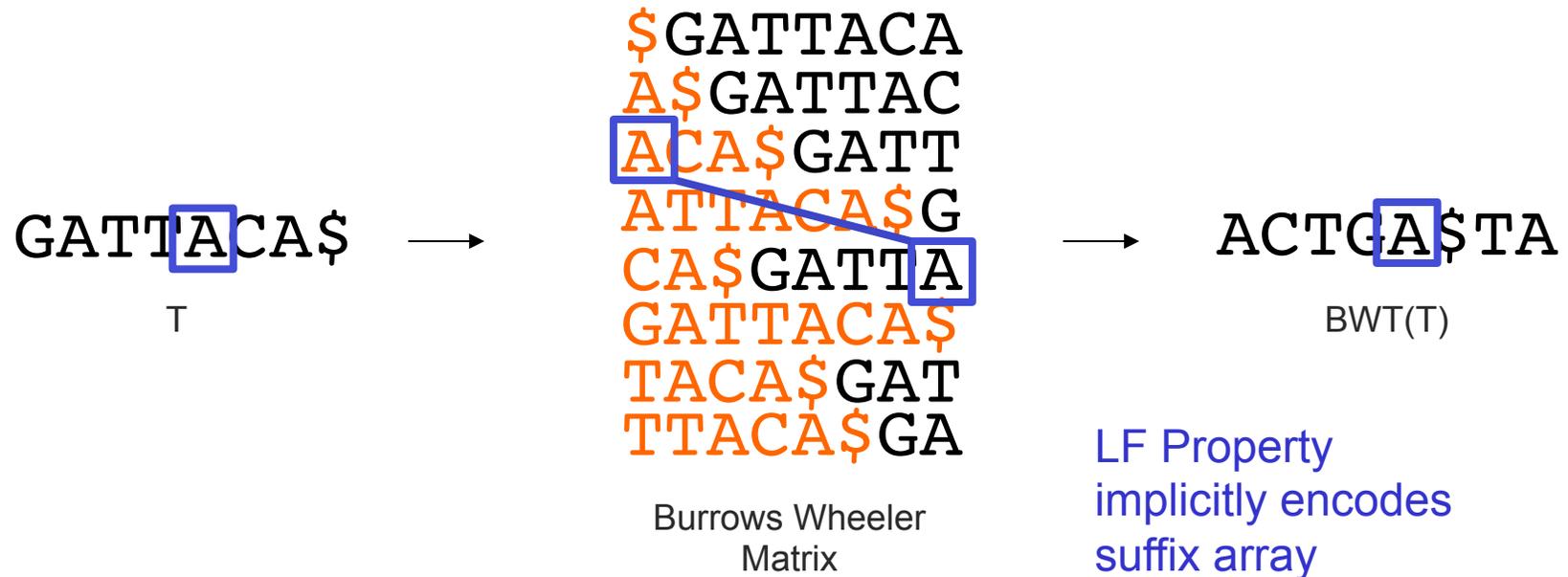
$$T(n) = \sum_{i=0}^{\log_{10/9}(n)} n = O(n \lg n)$$

- **If we randomly pick a pivot, we will get at least a 90/10 split with very high probability**

- Everything is okay as long as we always slice off a fraction of the list

[Challenge Question: What happens if we slice 1 element]

Burrows-Wheeler Transform



- 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

A block sorting lossless data compression algorithm.

Burrows M, Wheeler DJ (1994) *Digital Equipment Corporation, Palo Alto, CA 1994, Technical Report 124*