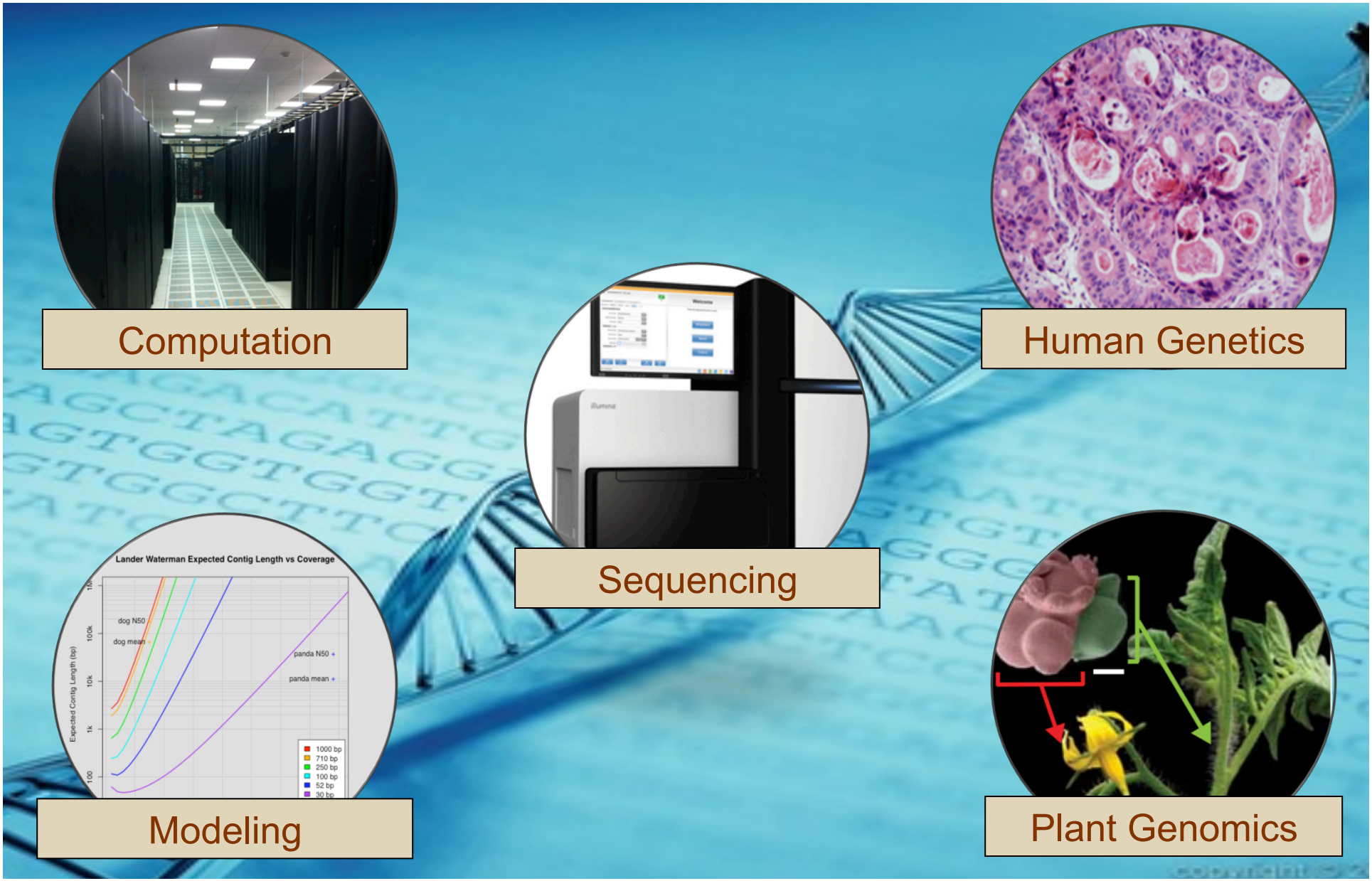
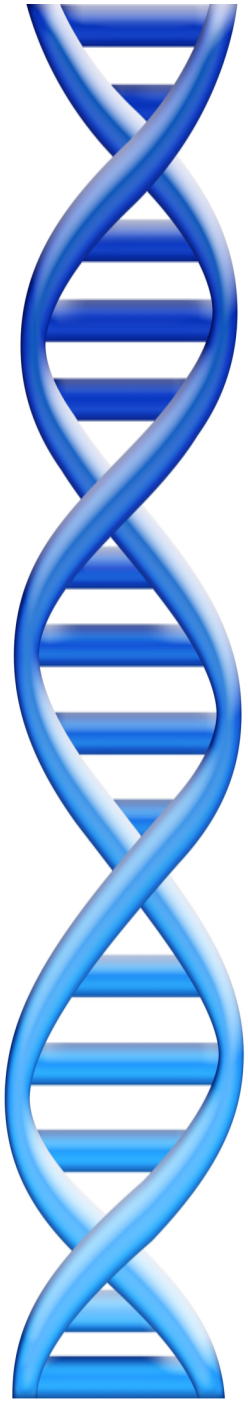
Sequence Alignment & Computational Thinking

Michael Schatz

Oct 29, 2013
SBU Graduate Genetics

Schatz Lab Overview

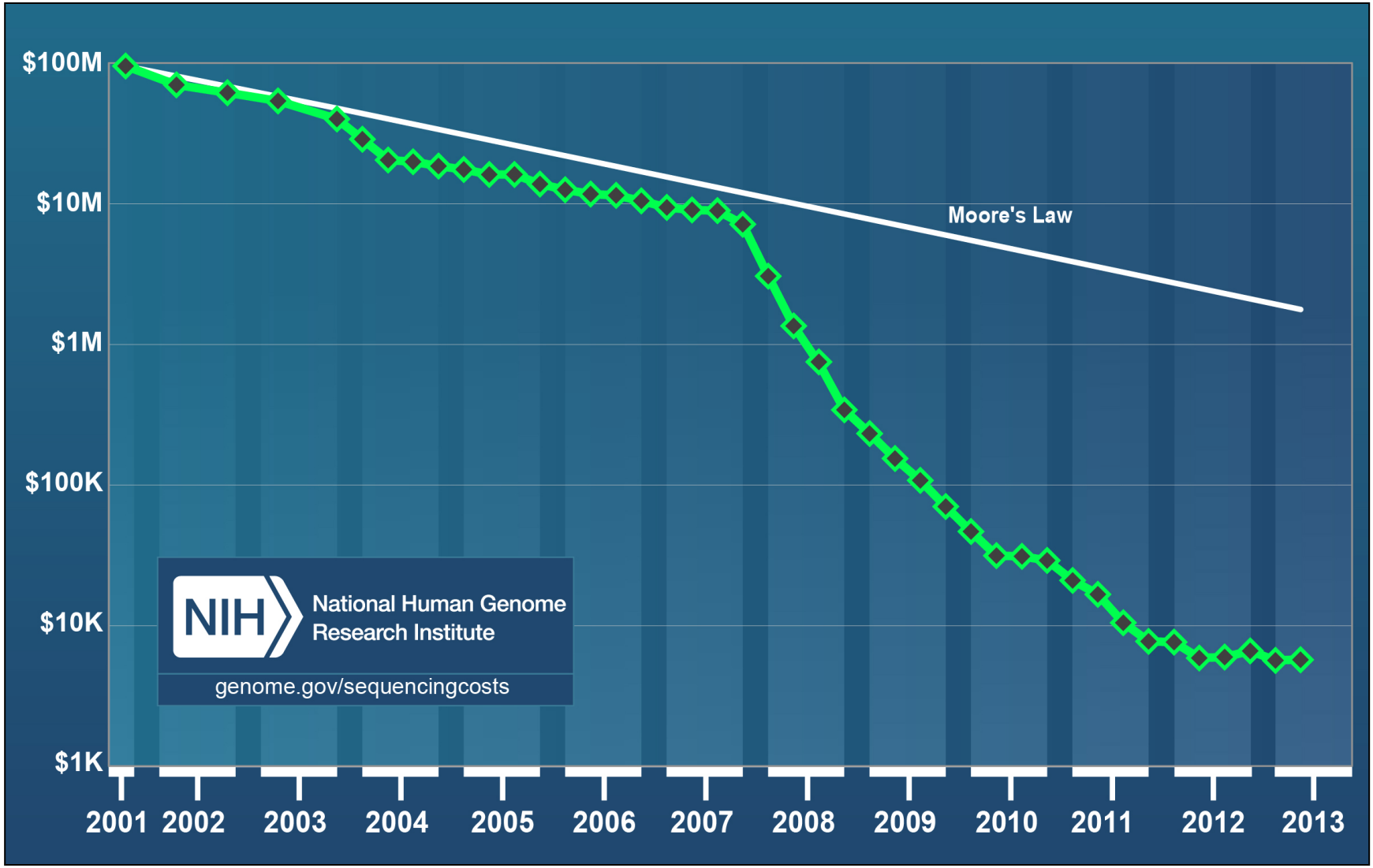




Outline

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

Cost per Genome

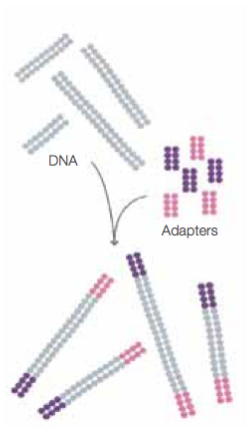


http://www.genome.gov/sequencingcosts/

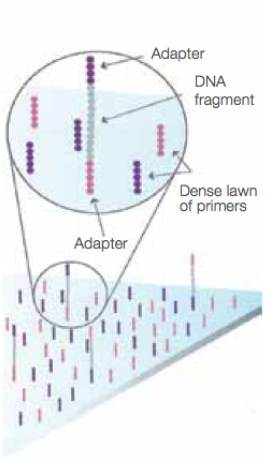
Inside the NY Genome Center

Sequencing Capacity: 16 HiSeq 2500 @ 600 Gbp / 11 day = 872 Gbp / day

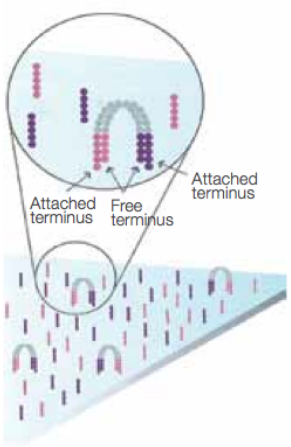
Illumina Sequencing by Synthesis



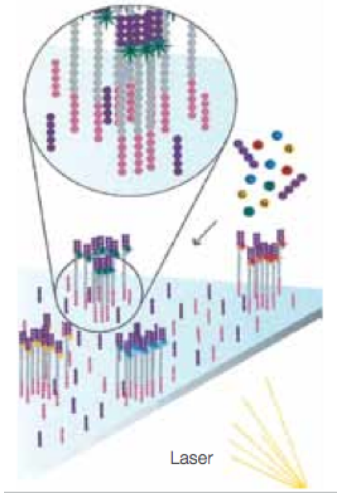
1. Prepare



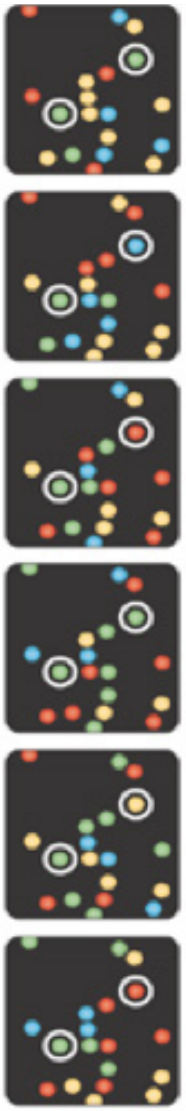
2. Attach



3. Amplify



4. Image

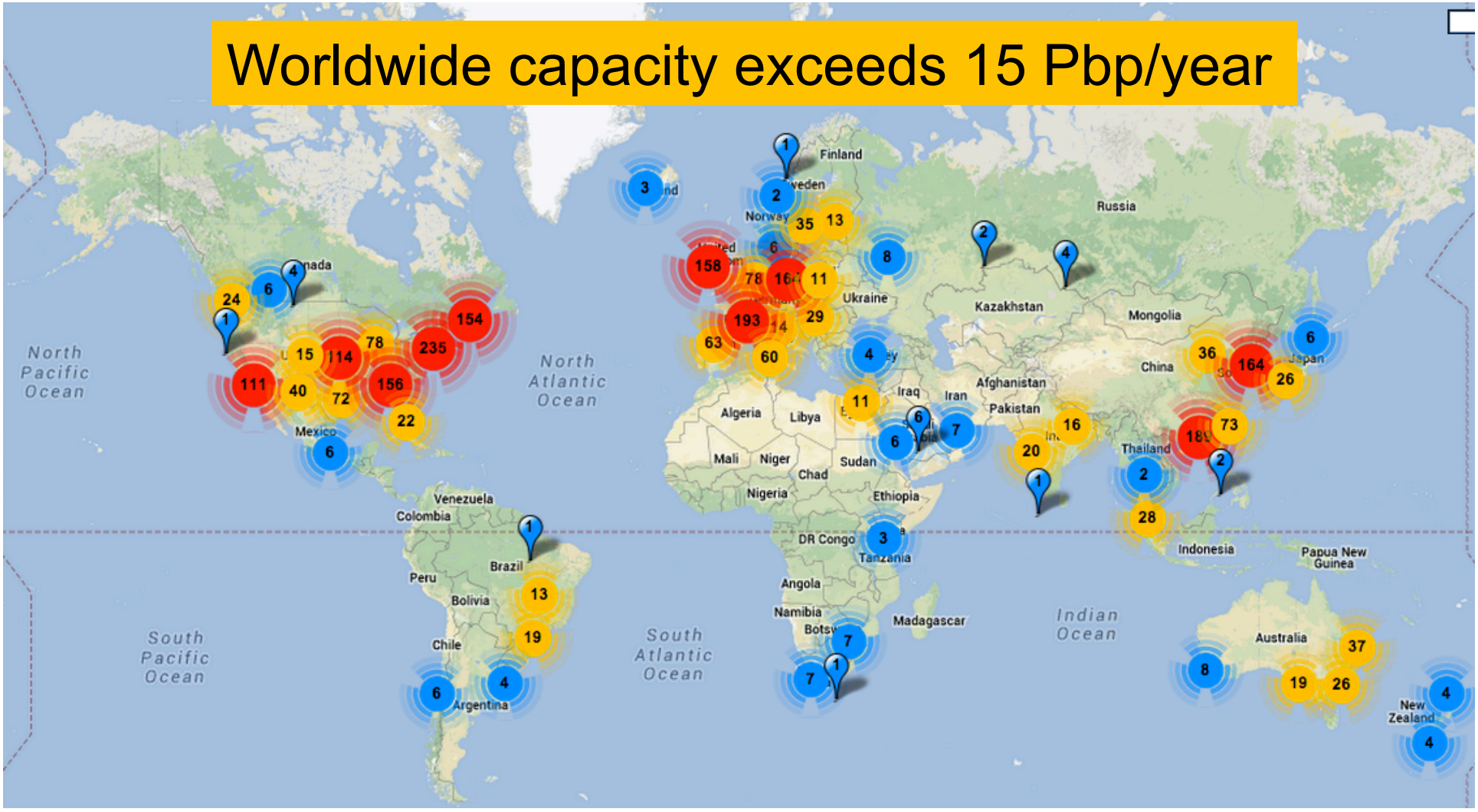


5. Basecall

Metzker (2010) Nature Reviews Genetics 11:31-46

http://www.youtube.com/watch?v=l99aKKHcxC4

Sequencing Centers



Next Generation Genomics: World Map of High-throughput Sequencers
http://omicsmaps.com

Milestones in Molecular Biology

There is tremendous interest to sequence:

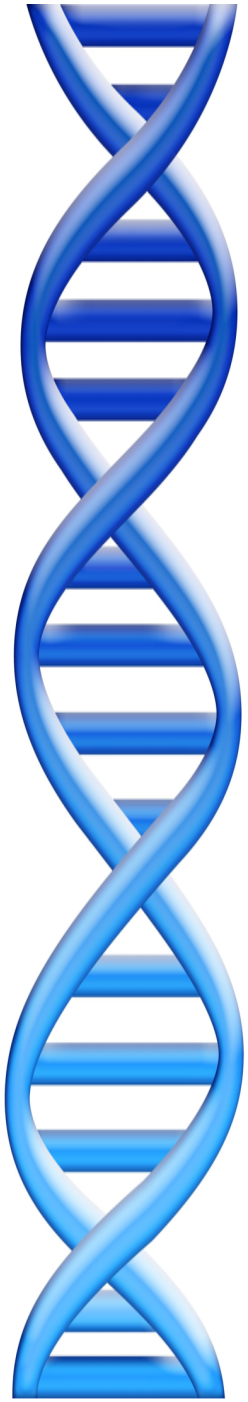
- What is your genome sequence?
- How does your genome compare to my genome?

- 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?

- …

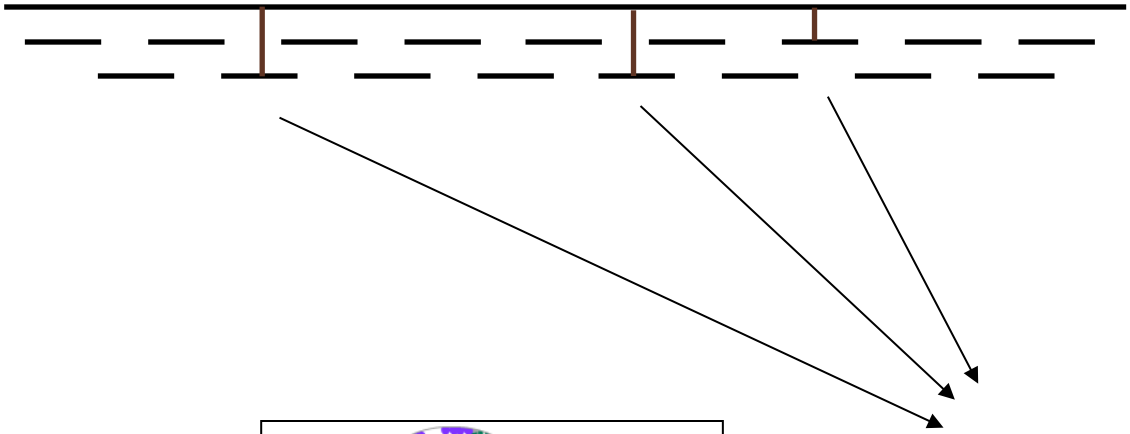


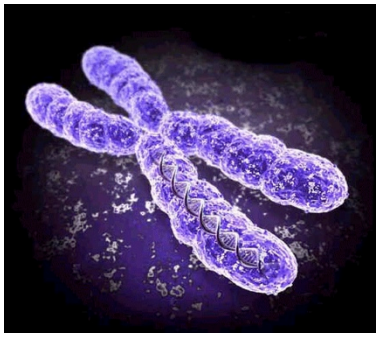
Outline

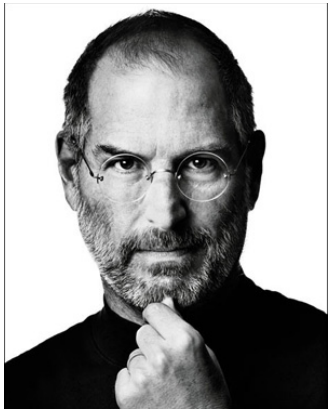
1. Rise of DNA Sequencing
2. Sequence Alignment Basics
3. Understanding Bowtie
4. Genetics of Autism

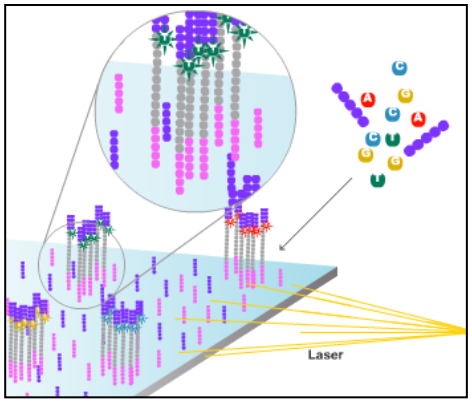
Personal Genomics

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 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…

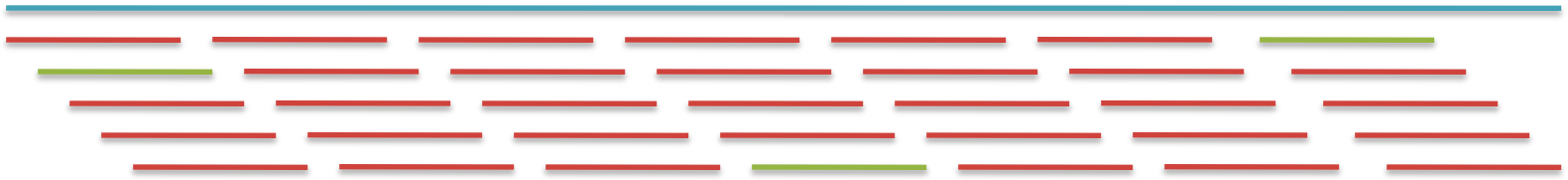
Searching for GATTACA

- 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?]

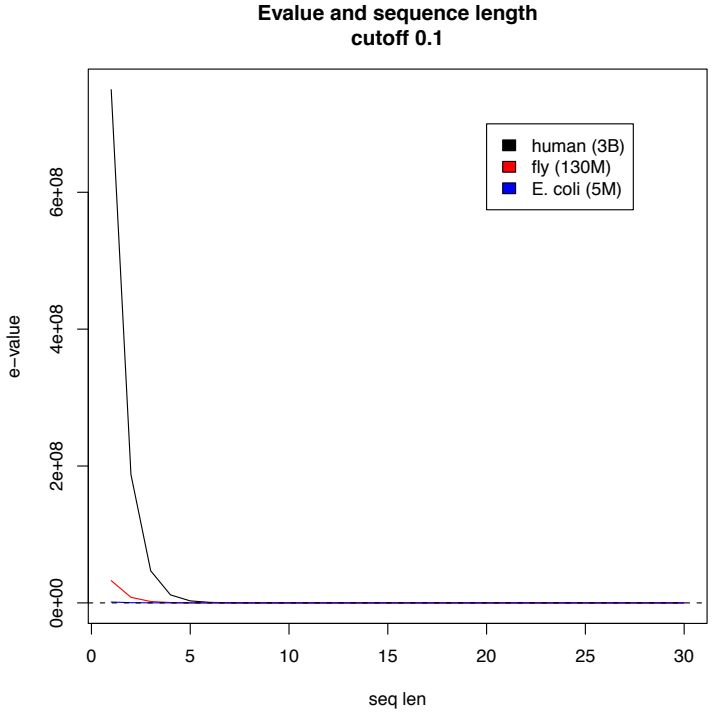
[How long would it take if we double both?]

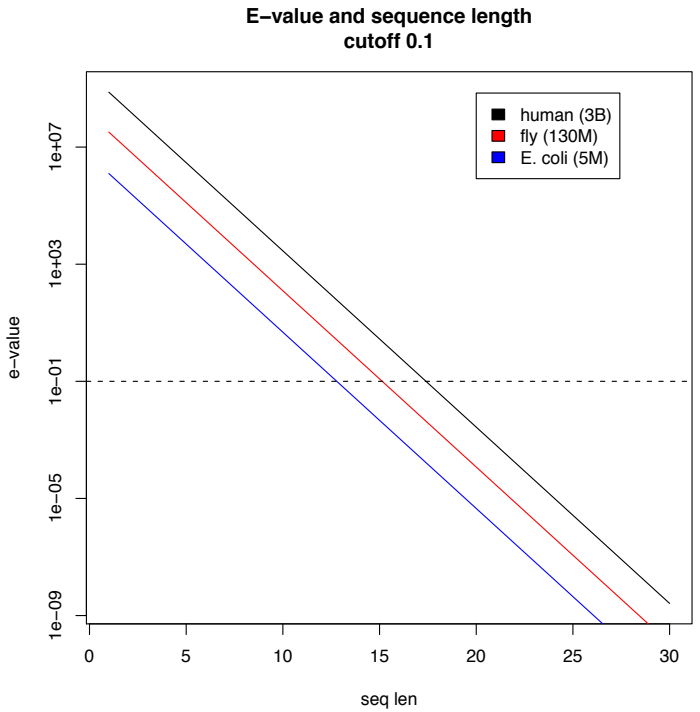
Expected Occurrences

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	

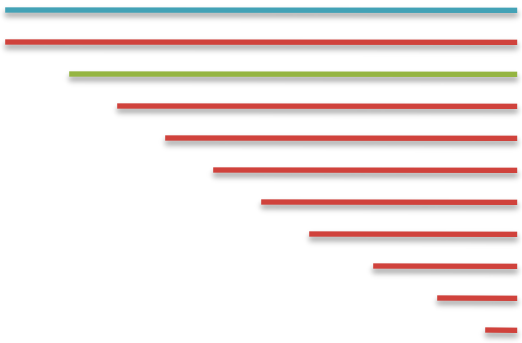
– Improve runtime to O(n + m) [3B + 7]
  - If we double both, it just takes twice as long
  - Knuth-Morris-Pratt, 1977
  - Boyer-Moyer, 1977, 1991

– 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

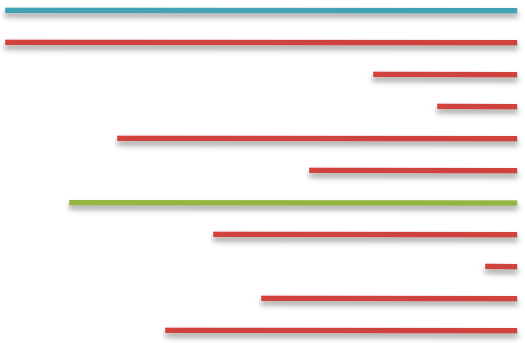
Suffix Arrays: Searching the Phone Book

- 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
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

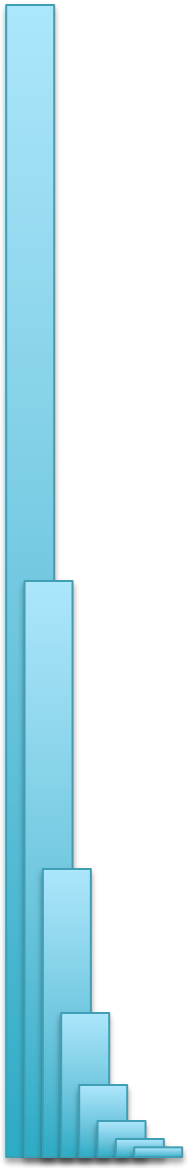
  - Lo = 9; Hi = 15; Mid = (9+15)/2 = 12
  - Middle = Suffix[12] = TACC
    => Lower: Hi = Mid - 1

  - 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

Binary Search Analysis

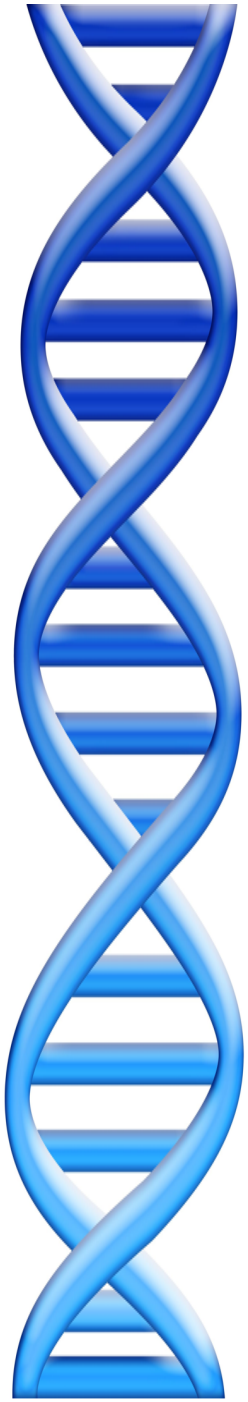


- Binary Search
  - Initialize search range to entire list
    - mid = (hi+lo)/2; middle = suffix[mid]
    - if query matches middle: done
    - else if query < middle: pick low range
    - else if query > middle: pick hi range
  - Repeat until done or empty range [WHEN?]

- 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?]



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

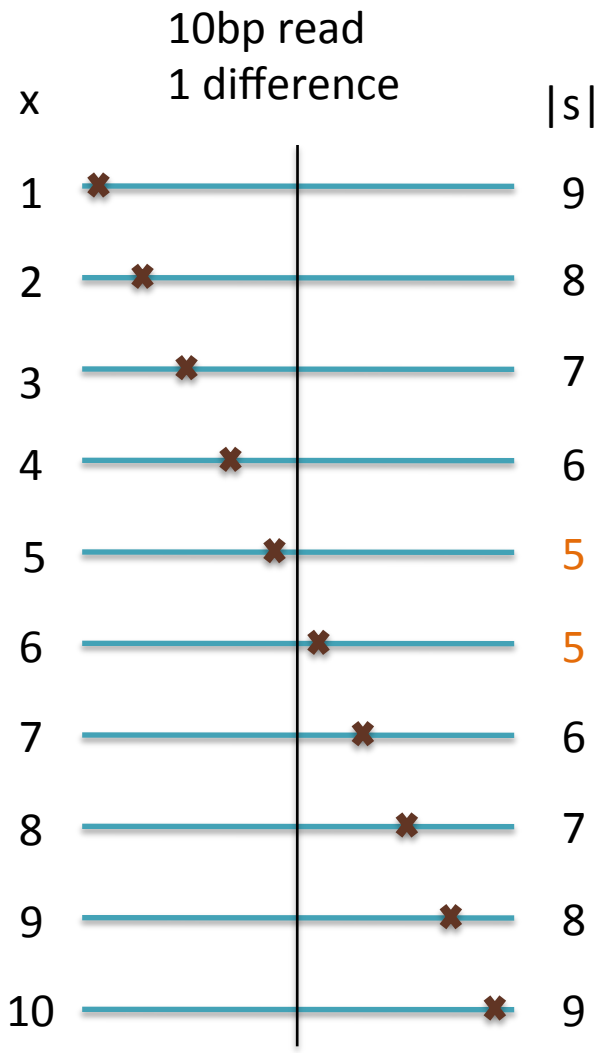
```
GATTACA          GATTTTTACA
|||x|||          ||||xxxxxx
GATCACA          GATTACA
   1                6
```

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

```
GATTACA          GATTTTTACA
|||x|||          ||||xxx|||
GATCACA          GATT---ACA
   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

Algorithm Overview

1. Split read into segments

Read

CCAGTAGCTCTCAGCCTTATTTTACCCAGGCCTGTA

Read (reverse complement)

TACAGGCCTGGGTAAAATAAGGCTGAGAGCTACTGG

Policy: extract 16 nt seed every 10 nt

Seeds

+, 0: CCAGTAGCTCTCAGCC

+, 10: TCAGCCTTATTTTACC

+, 20: TTTACCCAGGCCTGTA

-, 0: TACAGGCCTGGGTAAA

-, 10: GGTAAAATAAGGCTGA

-, 20: GGCTGAGAGCTACTGG

2. Lookup each segment and prioritize

Seeds

+, 0: CCAGTAGCTCTCAGCC

+, 10: TCAGCCTTATTTTACC

+, 20: TTTACCCAGGCCTGTA

-, 0: TACAGGCCTGGGTAAA

-, 10: GGTAAAATAAGGCTGA

-, 20: GGCTGAGAGCTACTGG

→ Ungapped alignment with FM Index

→ Seed alignments (as B ranges)

{ [211, 212], [212, 214] }

{ [653, 654], [651, 653] }

{ [684, 685] }

{ }

{ }

{ [624, 625] }

3. Evaluate end-to-end match

Extension candidates

SA:684, chr12:1955

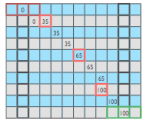
SA:624, chr2:462

SA:211: chr4:762

SA:213: chr12:1935

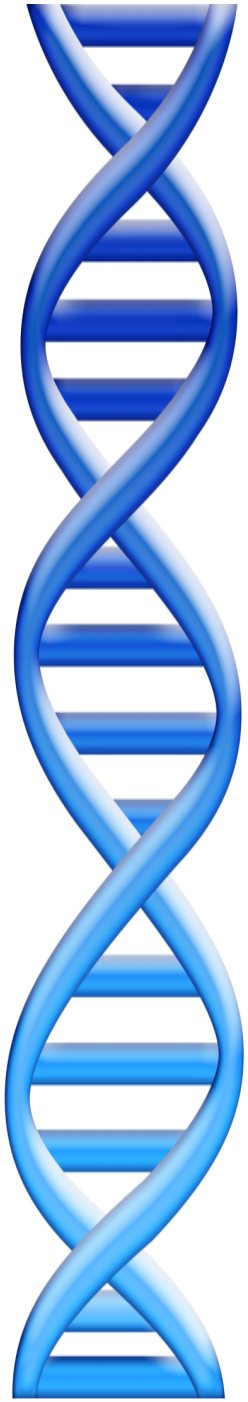
SA:652: chr12:1945

→ SIMD dynamic programming aligner



→ SAM alignments

```
r1   0    chr12    1936     0
     36M  *    0    0
     CCAGTAGCTCTCAGCCTTATTTTACCCAGGCCTGTA
     IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
     AS:i:0    XS:i:-2   XN:i:0
     XM:i:0    XO:i:0    XG:i:0
     NM:i:0    MD:Z:36   YT:Z:UU
     YM:i:0
...
```

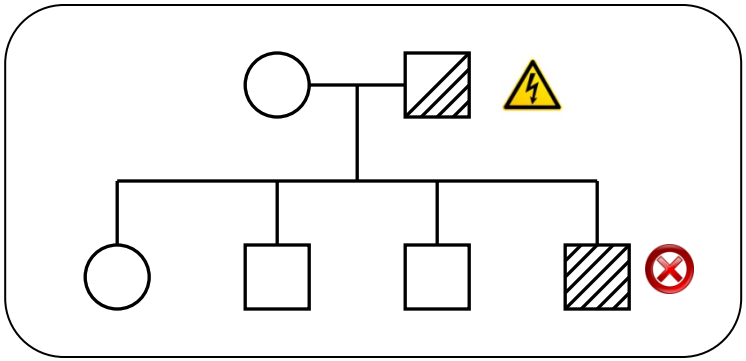


Outline

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

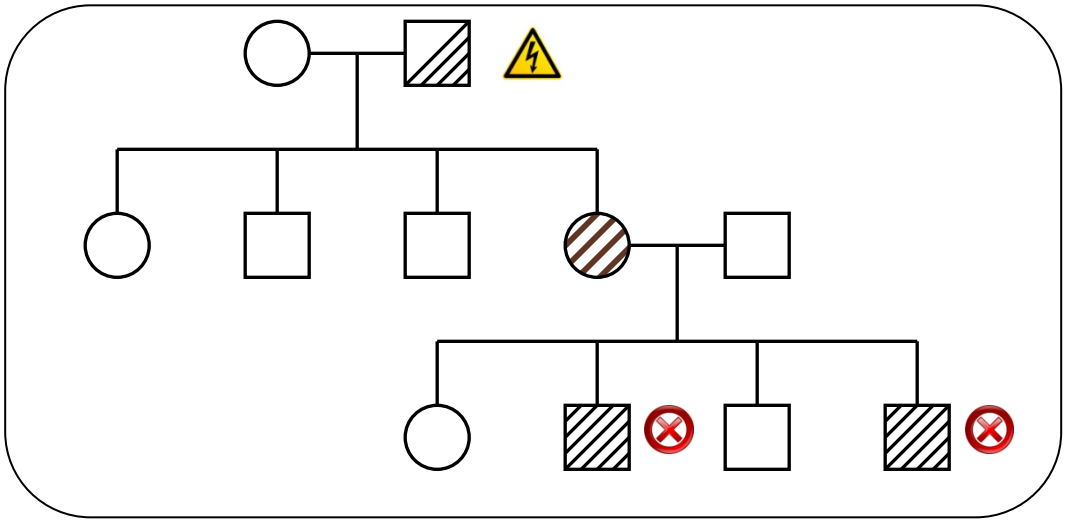
Unified Model of Autism

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



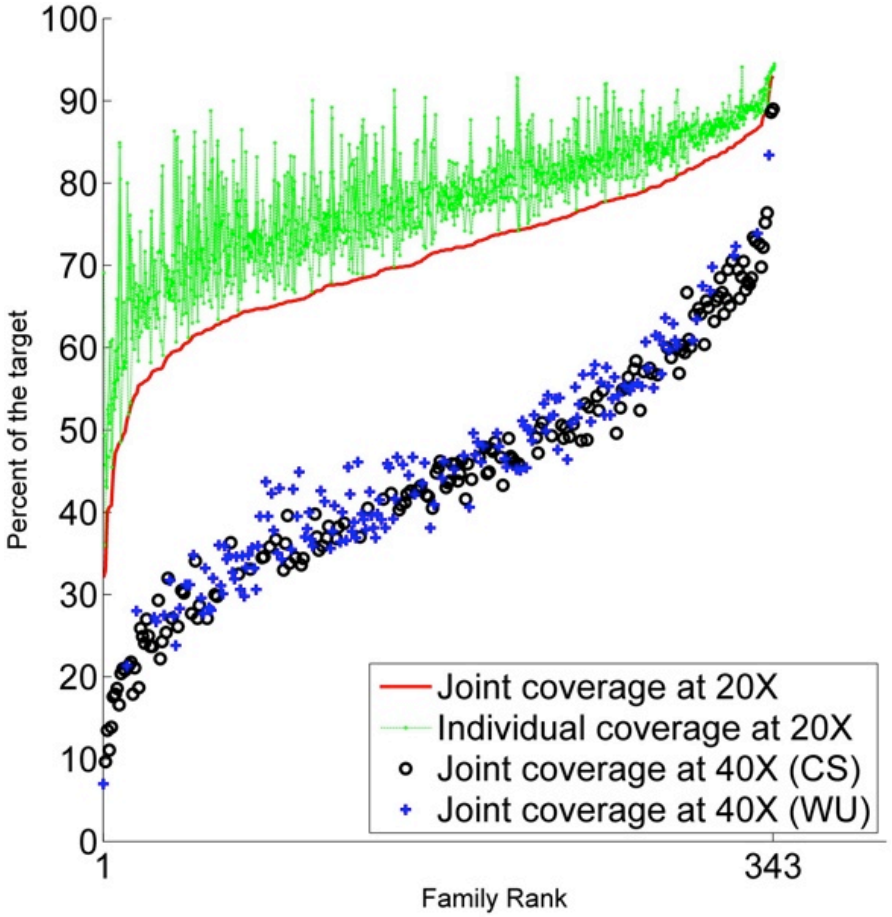
Legend

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

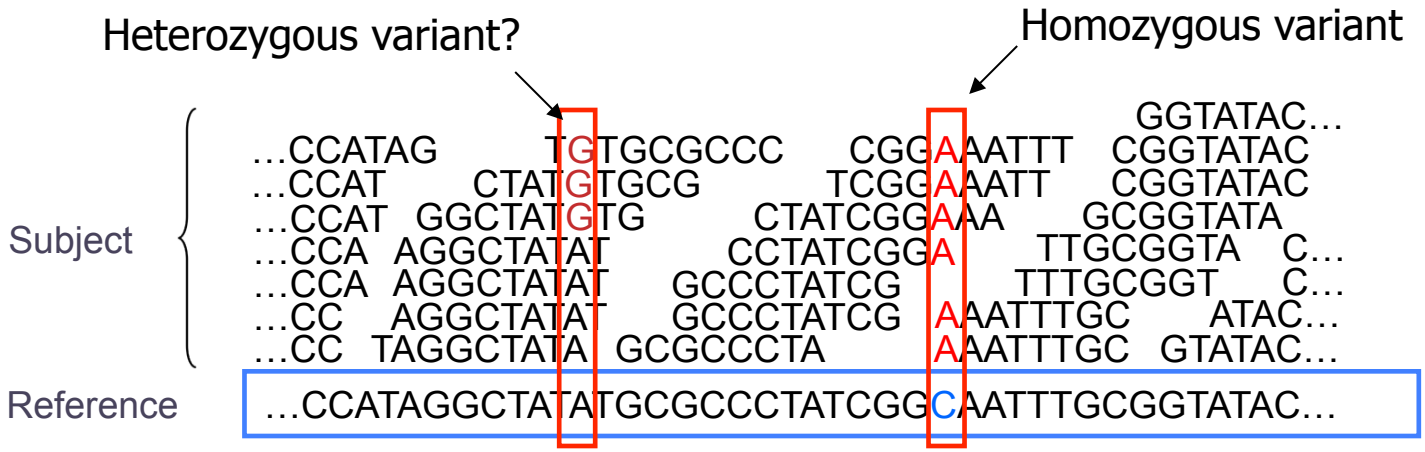
- Parents plus one child with autism and one non-autistic sibling
- Enriched for higher-functioning individuals

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

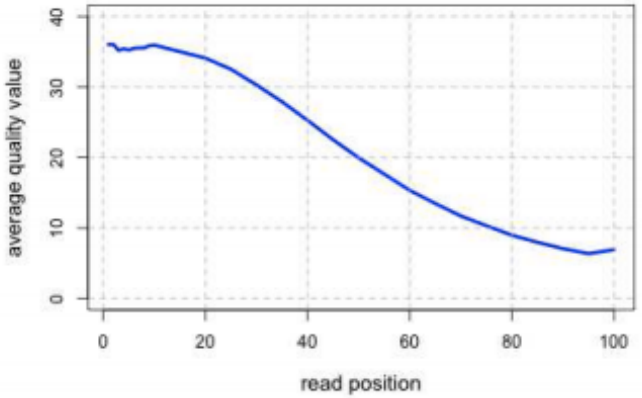
De novo gene disruptions in children on the autism spectrum
Iossifov *et al.* (2012) *Neuron*. 74:2 285-299

Genotyping



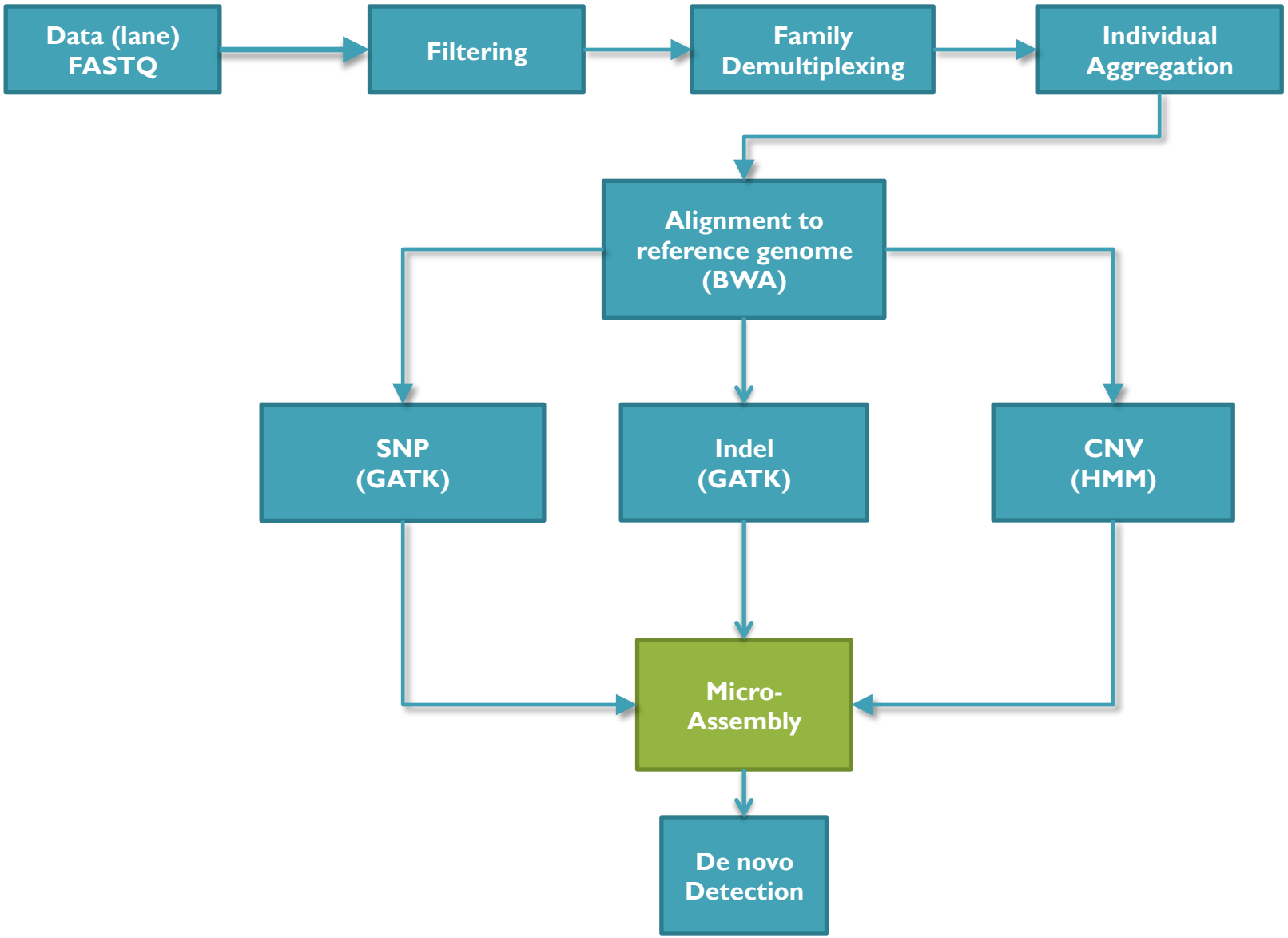
- Sequencing instruments make mistakes
  - Quality of read decreases over the read length

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



- 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

Exome Sequencing Pipeline



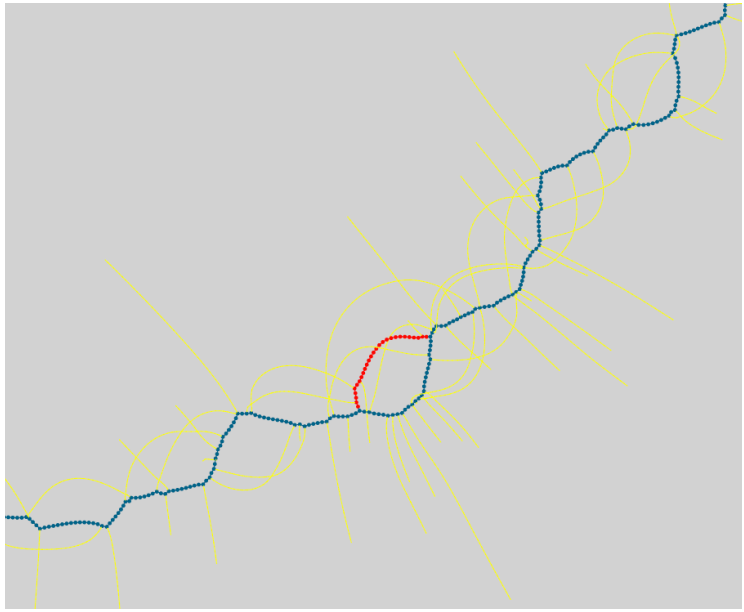
Scalpel: Haplotype Microassembly

G. Narzisi, J. O'Rawe, I. Iossifov, Y. Lee, Z. Wang, G. Lyon, M. Wigler, and M. C. Schatz

DNA sequence **micro-assembly** pipeline for accurate detection and validation of *de novo* mutations (SNPs, indels) within exome-capture data.

Features

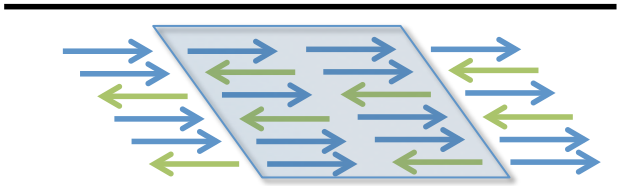
1. Combine mapping and assembly
2. Exhaustive search of haplotypes
3. De novo mutations



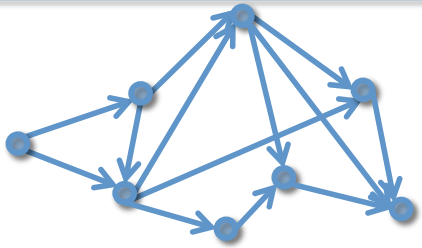
NRXN1 *de novo* SNP
(auSSC12501 chr2:50724605)

Scalpel Pipeline

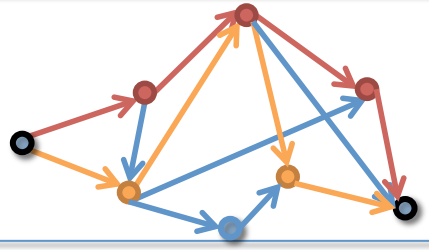
Extract reads mapping within the exon including (1) well-mapped reads, (2) soft-clipped reads, and (3) anchored pairs



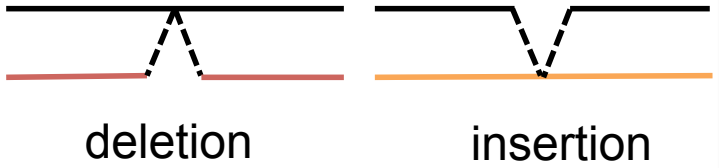
Decompose reads into overlapping *k*-mers and construct de Bruijn graph from the reads



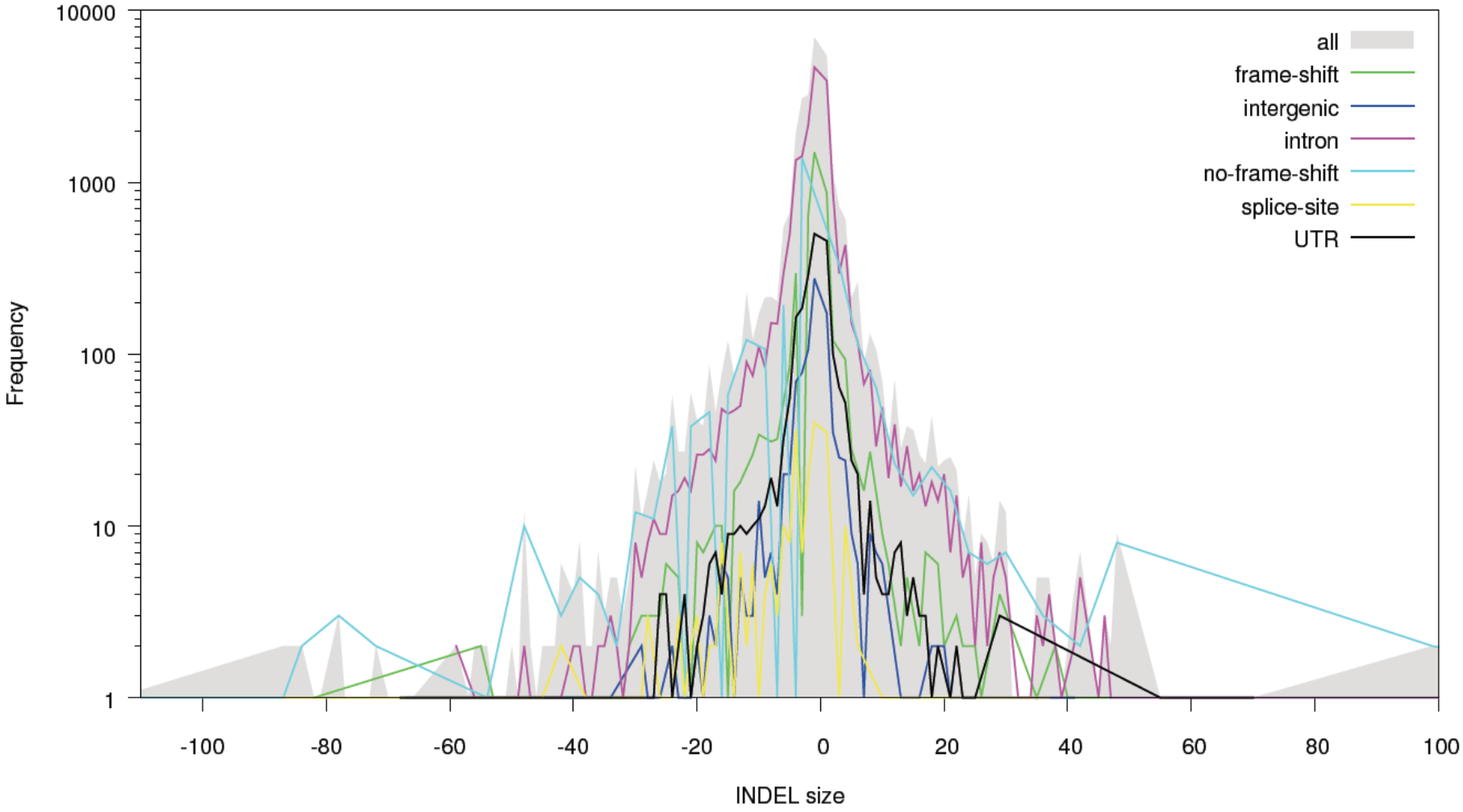
Find end-to-end haplotype paths spanning the region



Align assembled sequences to reference to detect mutations



Revised Analysis of the SSC

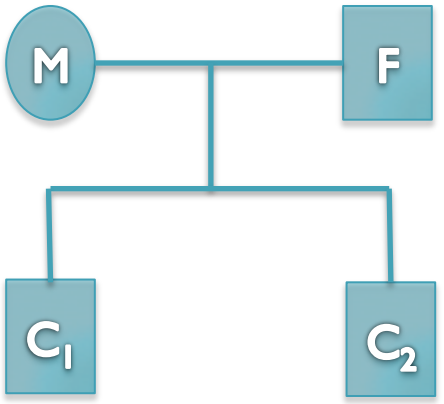


Constructed database of >1M transmitted and de novo indels
Many new gene candidates identified, population analysis underway

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------CCGGAGATTGTCTTTGCCCGGA...
```

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

Recommended: CSE 549 - Introduction to Computational Biology

Acknowledgements

Schatz Lab
Giuseppe Narzisi
Shoshana Marcus
James Gurtowski
Srividya Ramakrishnan
Hayan Lee
Rob Aboukhalil
Mitch Bekritsky
Charles Underwood
Tyler Gavin
Alejandro Wences
Greg Vurture
Eric Biggers
Aspyn Palatnick

CSHL
Hannon Lab
Gingeras Lab
Iossifov Lab
Levy Lab
Lippman Lab
Lyon Lab
Martienssen Lab
McCombie Lab
Ware Lab
Wigler Lab

IT Department

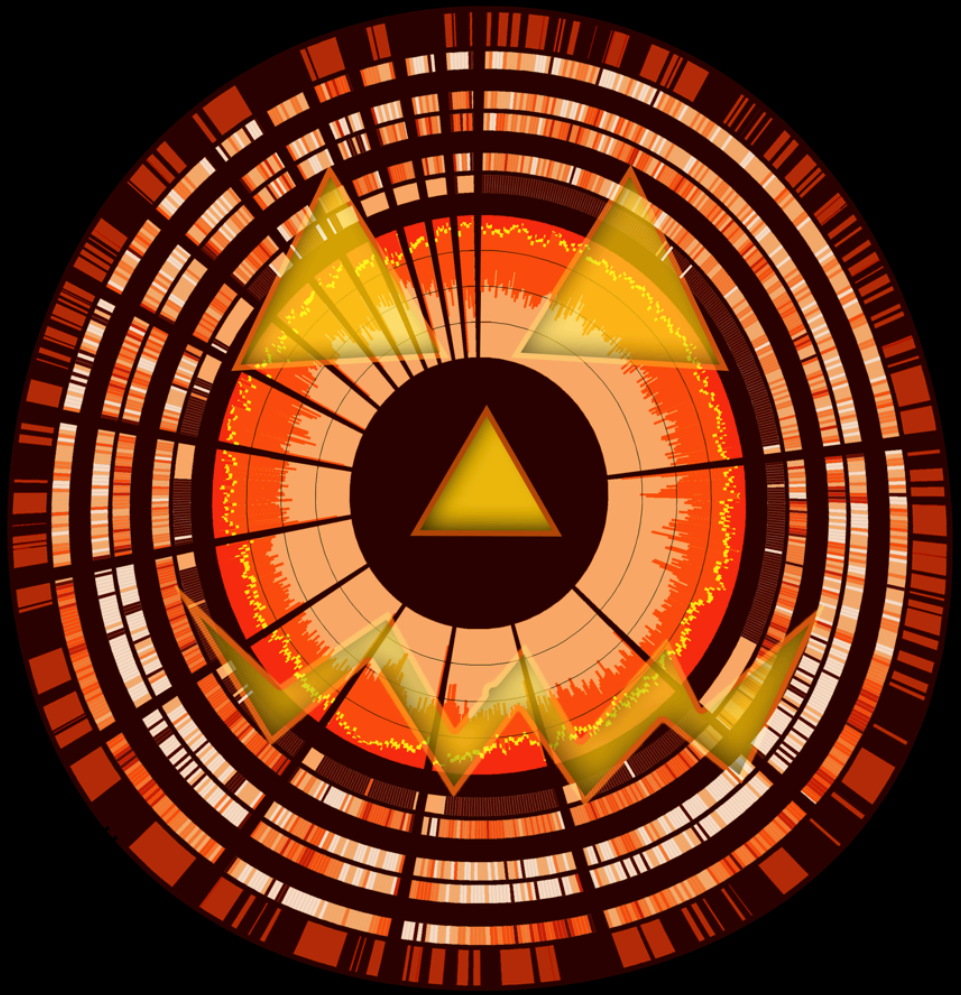
NBACC
Adam Phillippy
Sergey Koren

SFARI
SIMONS FOUNDATION AUTISM RESEARCH INITIATIVE

National Human Genome Research Institute

U.S. DEPARTMENT OF ENERGY

NSF



See you at

Genome Informatics

Oct 30 – Nov 2

http://schatzlab.cshl.edu
@mike_schatz