

Low level Processing and Visualization of Genome Sequencing Data

GSND 5340Q, BMDA

W. Evan Johnson, Ph.D.
Professor, Division of Infectious Disease
Director, Center for Data Science
Rutgers University – New Jersey Medical School

2024-05-06

Section 1

Motivating Example: X-linked Disease

Rare X-linked Disease



Rare X-linked Disease



Uncle #1



Uncle #2

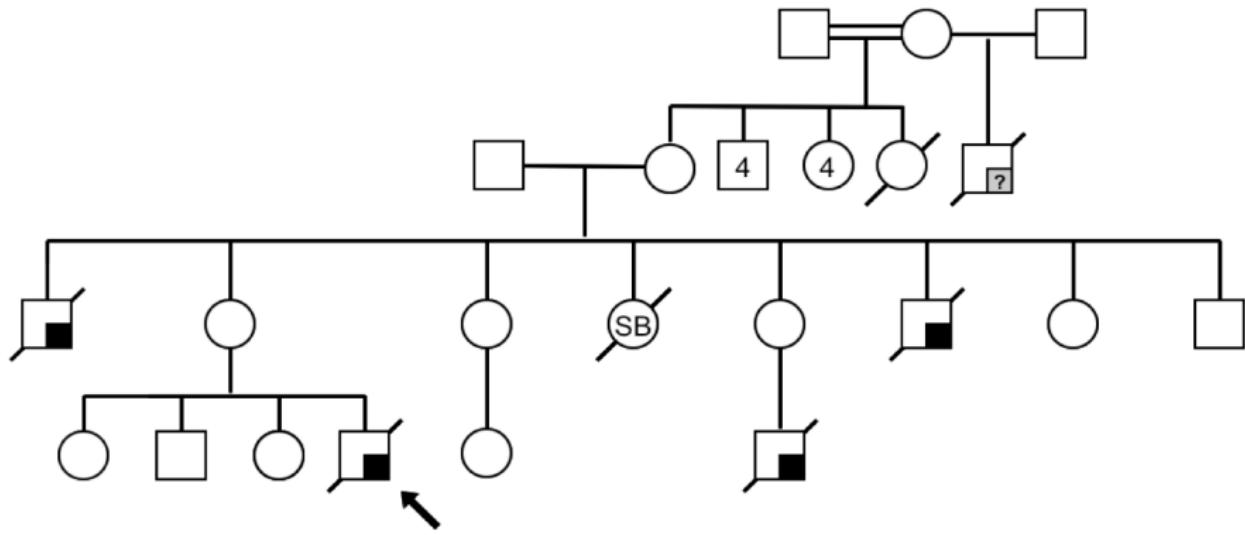


cousin

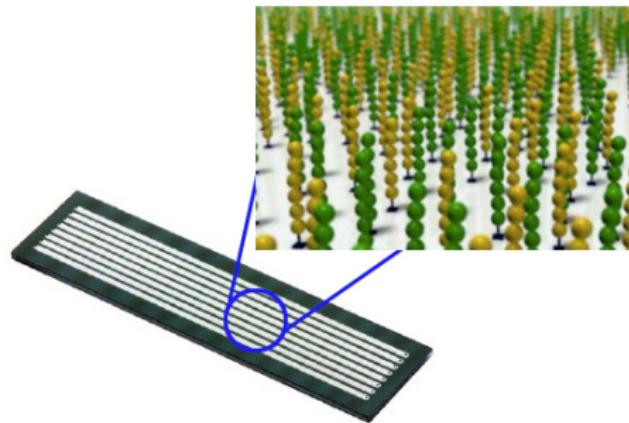


proband

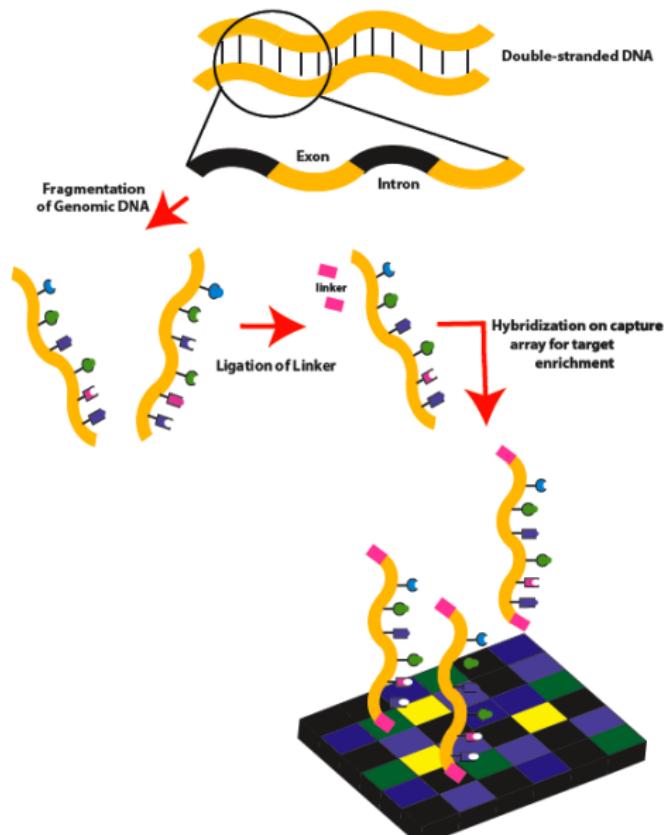
Rare X-linked Disease



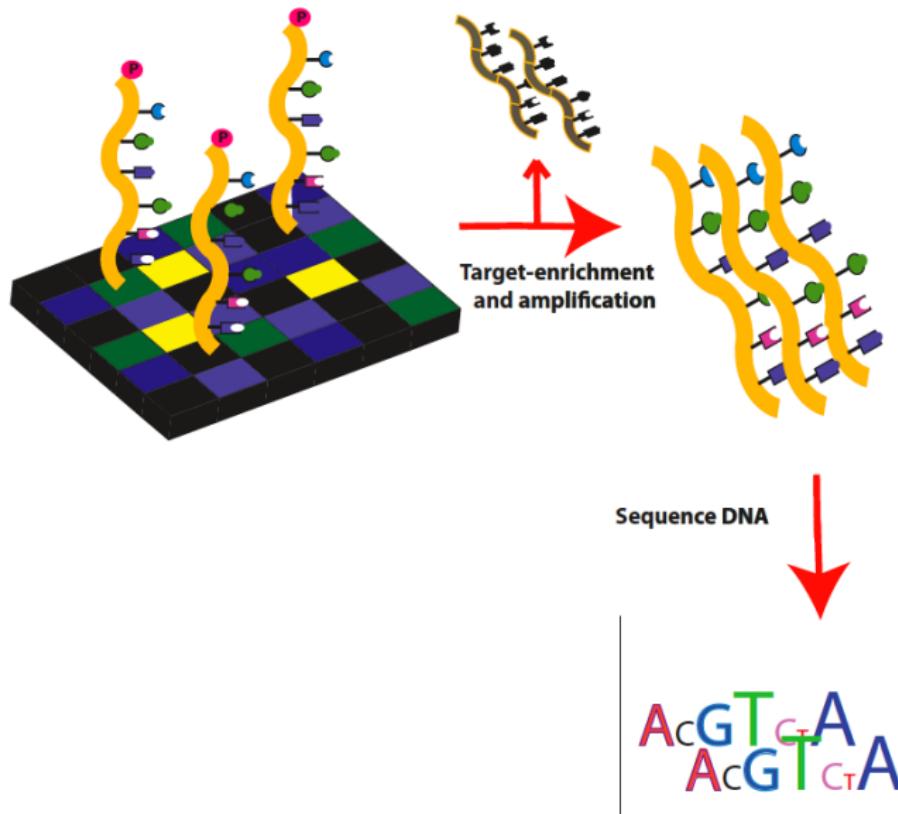
Exon Capture Sequencing



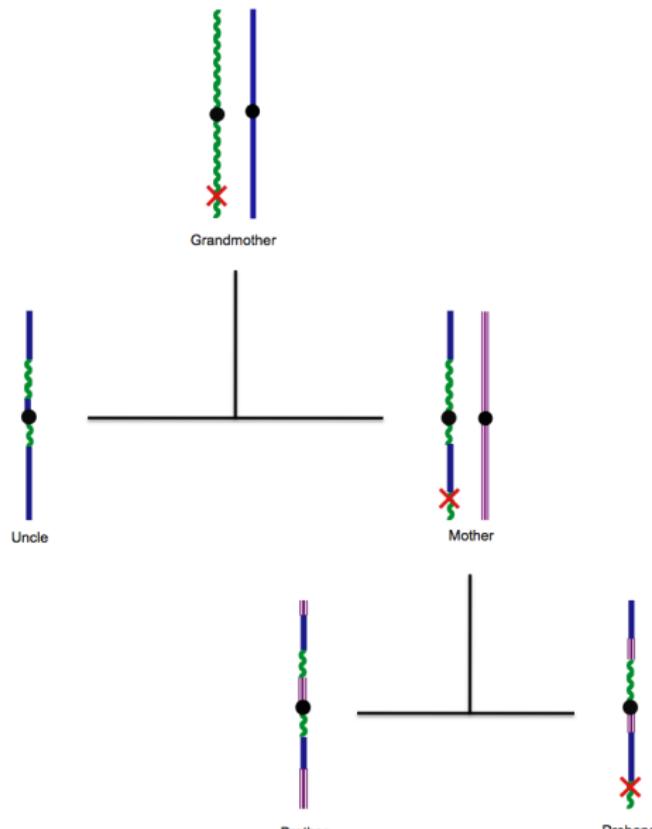
Exome Capture



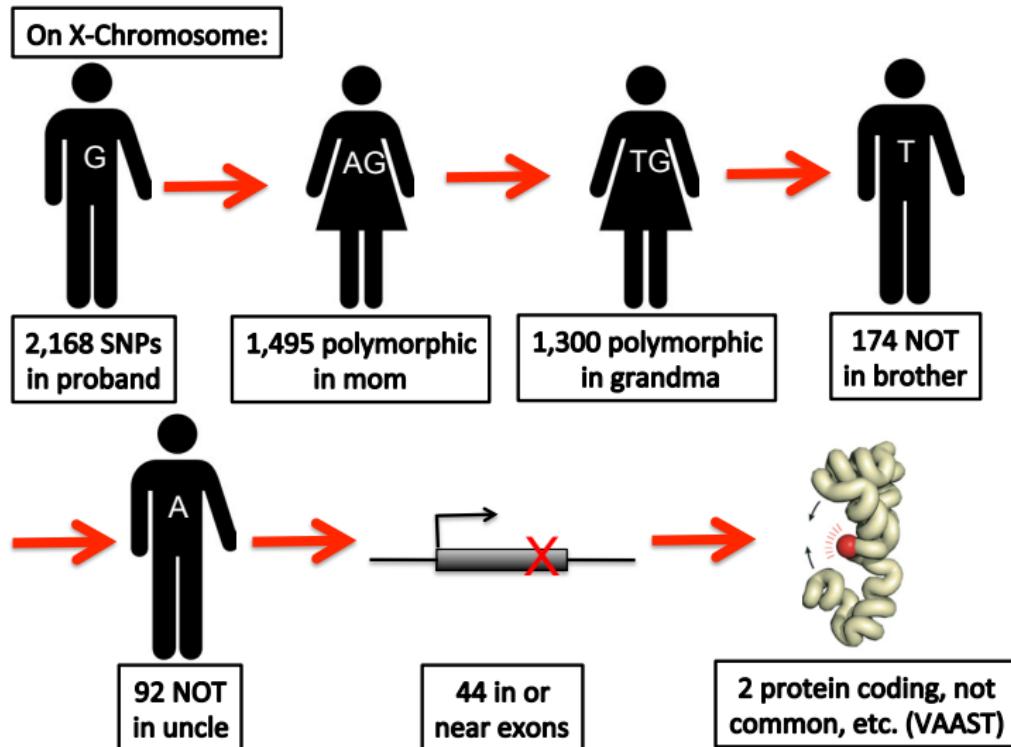
Exome Capture



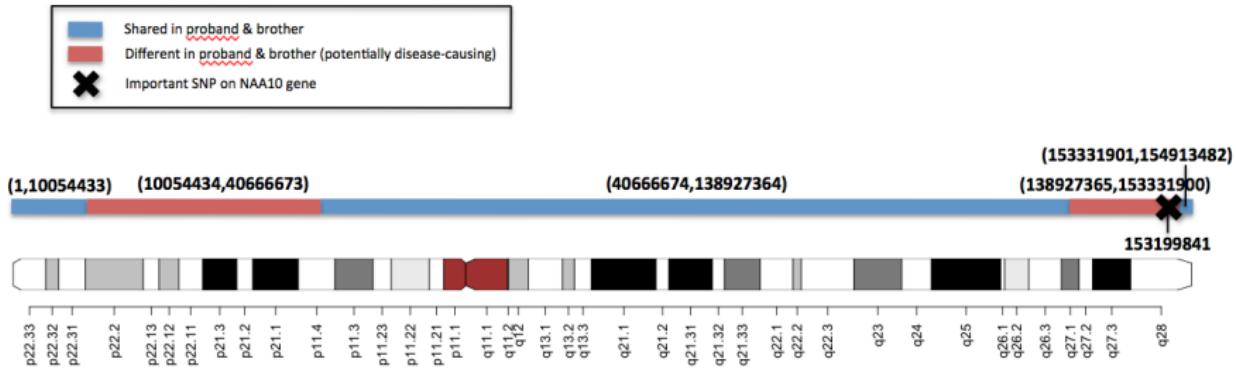
Rare X-linked Disease



Rare X-linked Disease



Rare X-linked Disease



Rare X-linked Disease

AJHG  Supports open access

Submit Log in Register Subscribe Claim

ARTICLE | VOLUME 89, ISSUE 1, P28-43, JULY 15, 2011 [Download Full Issue](#)

      PDF [603 KB] Figures Save Share Reprints Request

Using VAAST to Identify an X-Linked Disorder Resulting in Lethality in Male Infants Due to N-Terminal Acetyltransferase Deficiency

Alan F. Rope • Kai Wang ¹⁹ • Rune Enevirth • ... Mark Yandell • Thomas Arnesen • Gholson J. Lyon  20  Show all authors • Show footnotes

Open Archive • Published: June 23, 2011 • DOI: <https://doi.org/10.1016/j.ajhg.2011.05.017>

 PlumX Metrics

Introduction

We have identified two families with a previously undescribed lethal X-linked disorder of infancy; the disorder comprises a distinct combination of an aged appearance, craniofacial anomalies, hypotonia, global developmental delays, cryptorchidism, and cardiac arrhythmias. Using *X* chromosome exon sequencing and a recently developed probabilistic algorithm aimed at discovering disease-causing variants, we identified in one family a c.109T>C (*p.Ser37Pro*) variant in *NAA10*, a gene encoding the catalytic subunit of the major human N-terminal acetyltransferase (NAT). A parallel effort on a second unrelated family converged on the same variant. The absence of this variant in controls, the amino acid conservation of this region of the protein, the predicted disruptive change, and the co-occurrence in two unrelated families with the same rare

Subjects and Methods

Results

Discussion

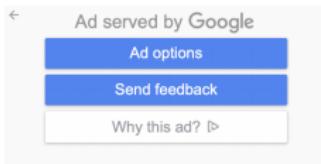
Acknowledgments

Supplemental Data

Web Resources

References

Article info



N-terminal acetyltransferase (NAA10)

N-terminal acetyltransferase (NAT):

- Common modification (~80-90% of human proteins)
- Depletion from cancer cells linked to cell cycle arrest and apoptosis
(Starheim, *BMC Proc* 2009)
- NAT genes directly implicated as cause of genetic disease
- Mutation demonstrated a significantly impaired biochemical activity *in vitro*
- NAA10 lethal if knocked out of Drosophila

Section 2

Generating Sequencing Data

Next-Generation Sequencing

- Expensive to purchase (hundreds of thousands \$USD)
- Expensive to operate (e.g. reagents, flow cells)
- You can sequence your genome at 30X depth for <\$1000 USD.

Roche 454



Ion Torrent



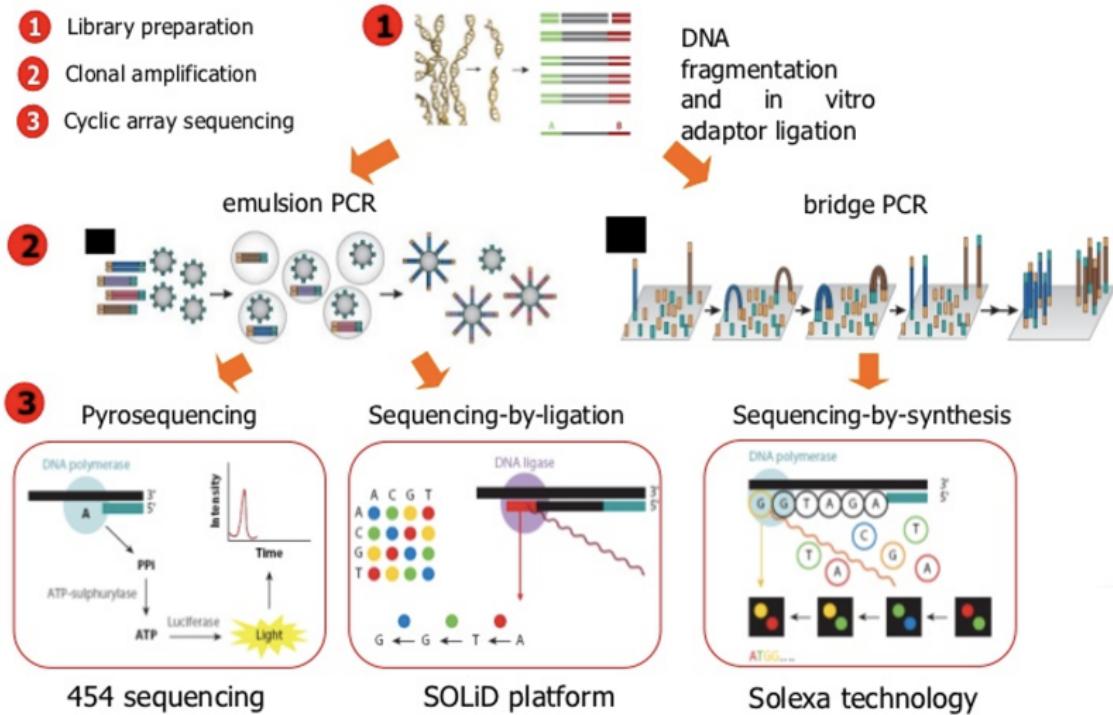
Illumina HiSeq 2500



Illumina NovaSeq 6000



Next-Generation Sequencing



Alex Sanchez, Statistics and Bioinformatics Research Group, Statistics Department, Universitat de Barcelona

Illumina Sequencing

- Most common sequencing technology today
- Sequences any DNA
- Sequencing by synthesis method
- Sequences (reads) are short (<300bp)
- 2 gigabases - 6 terabases per run
- Hours to days to complete one run

For more information, you can watch the following:

<https://www.youtube.com/watch?v=fCd6B5HRaZ8>

Illumina Sequencing

- Sequencing occurs on a **flow cell**
- Each flow cell has 1 to 8 **lanes**
- Number of reads for overall flow cell varies
- Length of reads is fixed (e.g. 250 bp)
- Read format:
 - **Single end** - one read per molecule
 - **Paired end** - two reads per molecule
- **Multiplexing:** sequence many samples at once using molecular barcodes



NovaSeq Flowcell
Courtesy of Illumina, Inc.

Sequencing Library Generation Workflow

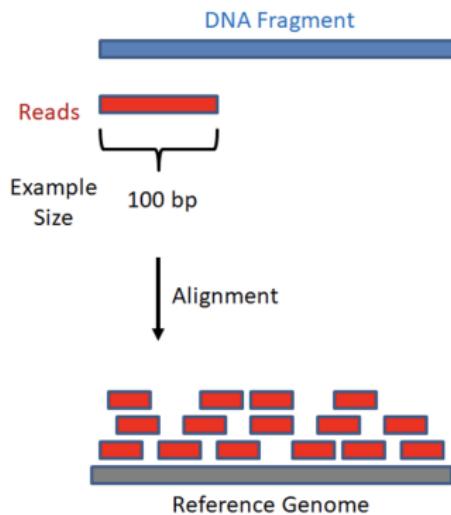
Sequencing Library: DNA prepared for sequencing

Workflow:

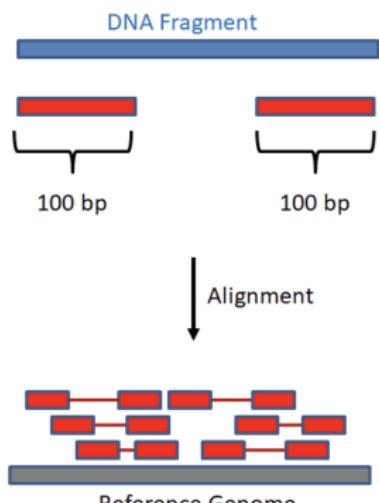
- Extract RNA/DNA from sample
 - If RNA, reverse transcribe to cDNA
- Size select using gel cut or random shearing
- PCR amplify DNA if concentration is low
- Add sequencing adapters
 - If multiplexing, use barcoded adapters
- Pool samples, load across flow lanes for sequencing
- Typically only perform 1, sequencing cores

Design Choice: Single End vs Paired End

Single End



Paired End



Design Choice: Number of Reads

Table 1: Coverage and Read Recommendations by Application

Category	Detection or Application	Recommended Coverage (x) or Reads (millions)	References
Whole genome sequencing	Homozygous SNVs	15x	Bentley et al., 2008
	Heterozygous SNVs	33x	Bentley et al., 2008
	INDELS	60x	Feng et al., 2014
	Genotype calls	35x	Ajay et al., 2011
	CNV	1-8x	Xie et al., 2009; Medvedev et al., 2010
Whole exome sequencing	Homozygous SNVs	100x (3x local depth)	Clark et al., 2011; Meynert et al., 2013
	Heterozygous SNVs	100x (13x local depth)	Clark et al., 2011; Meynert et al., 2013
	INDELS	not recommended	Feng et al., 2014
Transcriptome Sequencing	Differential expression profiling	10-25M	Liu Y. et al., 2014; ENCODE 2011 RNA-Seq
	Alternative splicing	50-100M	Liu Y. et al., 2013; ENCODE 2011 RNA-Seq
	Allele specific expression	50-100M	Liu Y. et al., 2013; ENCODE 2011 RNA-Seq
	De novo assembly	>100M	Liu Y. et al., 2013; ENCODE 2011 RNA-Seq
DNA Target-Based Sequencing	ChIP-Seq	10-14M (sharp peaks); 20-40M (broad marks)	Rozowsky et al., 2009; ENCODE 2011 Genome; Landt et al., 2012

<https://genohub.com/recommended-sequencing-coverage-by-application/>

Design Choice: Sequencing Depth

Whole Exome

- Less expensive
- Nearly complete ascertainment of variation in the coding ~1% of the genome (i.e. exons)
- Will miss functional variants outside of the coding region

Low Coverage Whole Genome

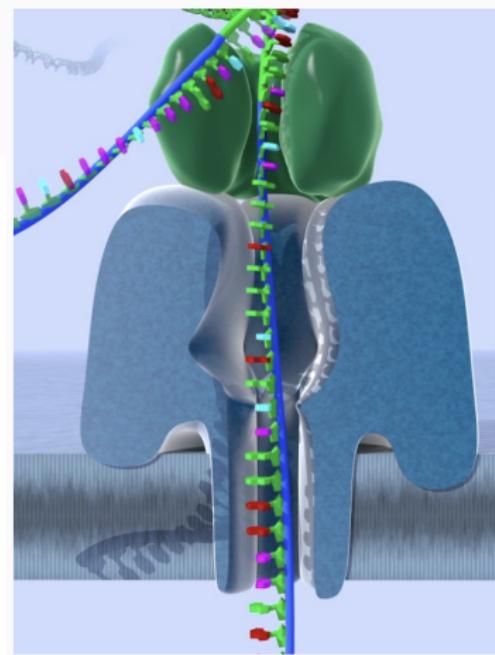
- Less expensive
- Reasonably good ascertainment of shared variation, but not unique variation

Deep Whole Genome

- More expensive
- Capture most of the genetic information
- Sequence the entire genome of each subject

3rd Gen Sequencing: Oxford Nanopore

- “Real time single molecule”



<https://www.youtube.com/watch?v=GUb1TZvMWsw>

Section 3

Sequencing Data Formats

Biological Data Formats are standardized

- Many different types of data
- Standard formats exist for many of them
- Always use/extend standard formats
- Don't save data in non standard formats when standards are available -
No .xlsx!
- For complete list of formats see:
<https://genome.ucsc.edu/FAQ/FAQformat.html>

Sequencing Data Formats

format	data	tool(s)
FASTA	sequence of nucleotides	Samtools, Biopython
FASTQ	sequenced reads	FASTQC, Biopython
SAM/BAM/CRAM	aligned reads	Samtools, Deeptools, PySam
VCF	variant calls	vcftools bedtools
BED / BED-PE	genomic regions	bedtools
GFF	general features	Biopython, Manual Parsing
GTF	gene features	Biopython, Manual Parsing

Sequencing Data Formats

format	data	tool(s)
FASTA	sequence of nucleotides	samtools faidx
FASTQ	sequenced reads	-
SAM/BAM/CRAM	aligned reads	samtools
VCF	variant calls	vcftools bedtools
BED / BED-PE		bedtools
GFF		-
GTF	gene features	-

Contains sequences
of nucleotides / AA /
bases

Sequencing Data Formats

format	data	(s)
FASTA	sequences	fastq
FASTQ	sequences	fastq
SAM/BAM/CRAM	aligned reads	samtools
VCF	variant calls	vcftools bedtools
BED / BED-PE	genomic regions	bedtools
GFF	general features	-
GTF	gene features	-

Data Formats Example

EXAMPLE

Align reads to the reference, sort and index, call SNPs, extract the SNPs on chromosome Y overlapping with all the Alu elements.



Data Formats Example

EXAMPLE

Align reads to the reference, sort and index, call SNPs on chromosome Y overlapping with all the Alu elements.

FASTQ format

Data Formats Example

EXAMPLE

Align reads to the reference, sort and index, call SNPs on chromosome Y overlapping with all the Alu elements.

FASTA format

Data Formats Example

EXAMPLE

Align reads to the reference, sort and index, call SNPs, extract the SNPs on chromosome 1 save in BAM format with all the Alu elements.

Data Formats Example

EXAMPLE

Align reads to the reference, sort and index, call SNPs, extract the SNPs on chromosome 1 in VCF format with all the Alu elements.

Data Formats Example

EXAMPLE

Align reads to the reference, sort and index, call SNPs, extract the SNPs on chromosome Y overlapping with all the Alu elements.

Read from GTF/GFF/BED

Data Formats Example

EXAMPLE

Align reads to the reference, sort and index, call SNPs, extract the SNPs on chromosome Y overlapping with all the Alu elements.

Data Formats Example

EXAMPLE

Align reads to the reference, sort and index, call SNPs, extract the SNPs on chromosome 1, and compare with all the Alu elements.

Data Formats Example

EXAMPLE

Align reads to the reference, sort and index, call SNPs, extract the SNPs on chromosome Y over Alu elements.

samtools / GATK

Data Formats Example

EXAMPLE

Align reads to the reference, sort and index, call SNPs, extract the SNPs on chromosome Y overlapping with all the Alu elements.

samtools view

Data Formats Example

EXAMPLE

Align reads to the reference, sort and index, call SNPs, extract the SNPs on chromosome Y **overlapping** with all the Alu elements.

`bedtools intersect`

Section 4

Raw Sequencing Data and QC

History and Evolution of Illumina Data Output

Illumina sequencers have given output in many different formats:

- Illumina .PRB and .INT files
 - Better access to raw data.
 - Base calling algorithms (Bravo and Irizarry, *Biometrics*, 2010)
 - Mapping algorithms (GNUMAP, NOVO)
 - Confusing formats; Large data files
- Illumina .FASTQ files
- Sanger .FASTQ files

Illumina .INT and .PRB

1	1	125	771	1651.8	2189.6	228.1	549.9	219.0	202.5	48.4	3016.8	127.8	6.1	204
1	1	478	16	1050.0	969.9	149.5	311.7	0.0	0.0	39.3	134.1	0.0	0.0	0.0
1	1	780	553	639.6	980.8	555.2	6412.8	1040.1	4408.7	750.3	638.1	946.2	4351.1	7
1	1	123	685	-116.5	341.5	-14.0	-985.9	231.5	1090.0	88.3	-102.2	240.2	513.6	
1	1	61	934	40.7	87.3	38.5	21.3	16.4	31.7	100.9	68.8	41.4	29.4	40.4
1	1	866	972	2820.5	4698.9	435.8	8502.2	4740.3	4890.2	1491.5	1241.7	2137.5	2505.6	6

40	-40	-40	-40	-40	-40	-40	-40	40	-40	-40	40	-40	-40	40	-40	-40	-40
28	-28	-40	-40	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5	-5
-17	-16	-30	13	-40	26	-26	-40	-21	17	-20	-27	21	-21	-40	-40	-40	-40
-40	40	-40	-40	-40	40	-40	-40	-1	1	-40	-40	-0	-40	-0	-14	26	
-1	-40	-0	-10	-20	-18	16	-35	-1	-8	-2	-13	12	-34	-13	-26	5	
-27	-40	-40	27	40	-40	-40	-40	-40	-40	40	-40	-40	-40	40	-40	-40	-40

.fasta or .fa

```
>chrX
ttgaactcctgacctcaggtatccgcggcctgacccaaaggcgct
cctgcctcagcctcccagtagctggactacaggtgccaccatgc
....
caggctaattttgtattttagtagagacgggtttcaccatgttagc
caggatggtctcaatctcctgacctcatgatccgcctgcctcgccccc
>chrY
tgtacacttaaatgggtgaatttatggaatgtgaattataCGTGTG
CTTGTAAAAAAAAATGATGGAGATGGAGACGTGACTCTAGCGTGAAGGGG
...
GTGGGGAGAGTAGATCTAGAGTGGAGACACCACTTTAGGAGGTATGATC
cctgccaccatgcctggctaattttgtattttagtagagacagggtt
```

.fastq or .fq (or .fq.gz)

@HWI-EAS240_0001:2:1:1142:17571#0/1

CTCTCTTTCTCCCCANGCTCCTCATGACCATATCCNTGTTGCCATTGTGTANGGNNNCTC

+HWI-EAS240 0001;2;1;1142;17571#0/1

@HWI-EAS240 0001;2;1;1142;17571#0/1

CTCTCTTTCTCCGANGTCTCTCATGACCATATCCNTGTTGCCATTGTGTANGGNNNCTT

+HWT=FAS240 0001:2:1:1142:17571#0/1

bababbb` abbbbabbhYB^ [^` ` bbbbbbbbbb` B` `` ` ^` ^bbbba^` ^` B00BBBN00

@HWT-EAS340_0001:2:1:1142:6453#0/1

CAGGACCTGGCACTATGNCATGCGATGCCAGGATGCGAAGTCTTANCAATTAGGATAACANATANNGGC

+HWT_EAS340_0001:3:1:1143:6453#0/1

CH/UT_EAC242_2001-2-1-1142-10442#2/1

@HW1-EAS240_0001:2:1:1142:19443#0/1
EAS240 EAS240 EAS240 EAS240 EAS240 EAS240

TCAGAAAACAGAAAGGTCATTTCTTACTCTTG
NNTT TACCCAG GGGCA T C A T T G G T G T G

+HW1-EAS240_0001:2:1:1142:19443#0/1

БББАББ~БББ~ББАББ]ББ]~] ББББББББYYBYХ000000ББ ББVY~НВ000ВВ0Ј0

.FASTQ Paired End

fastq_SRR1997469_1.fastq

```
@SRR1997469.1 1 length=36
CACTTCTTAGAAATATCCACTTCGGAATAAAAGA
+SRR1997469.1 1 length=36
BBBBBFFFFF<FFFFFFFFFFFFFBFFFFFFFFF
@SRR1997469.2 2 length=36
ACAGTTAACGATCCTTACAGANAGNAGNCTNGTA
+SRR1997469.2 2 length=36
<BBBBFFFBBF<BFF/<FFFFFF##########
...
```

fastq_SRR1997469_2.fastq

```
@SRR1997469.1 1 length=36
AGATAAGATGGTAATCTTGATGGAGAACATTAAGA
+SRR1997469.1 1 length=36
BBBBBFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
@SRR1997469.2 2 length=36
ACTGGGAANCCTCTGTCTAGCCTTATATGAAAAAA
+SRR1997469.2 2 length=36
BBB/BFFB#BB/FFFF/FFFFFFBBFFFFFFFBF
...
```

.FASTQ Comparison

https://en.wikipedia.org/wiki/Phred_quality_score

Sequence Quality Score (PHRED)

Phred quality scores are logarithmically linked to error probabilities

Phred Quality Score	Probability of incorrect base call	Base call accuracy
10	1 in 10	90%
20	1 in 100	99%
30	1 in 1000	99.9%
40	1 in 10,000	99.99%
50	1 in 100,000	99.999%
60	1 in 1,000,000	99.9999%

https://en.wikipedia.org/wiki/Phred_quality_score

Sequence Quality Score (PHRED)

Quality scores (Phred)

- Sanger Phred: Range=(0,40), $P = 1 - 10^{-(ASCII-33)/10}$
- Solexa: Range= (-5,40), $P = \frac{10^{(ASCII-64)/10}}{1+10^{(ASCII-64)/10}}$
- Illumina 1.3: (0,40), $P = \frac{10^{(ASCII-64)/10}}{1+10^{(ASCII-64)/10}}$
- Illumina 1.5: Range=(2,40), $P = 1 - 10^{-(ASCII-64)/10}$
- Illumina 1.8: Same as Sanger except Range=(0,41)

.FASTQ Comparison

> Sanger	> Solexa	> Illumina1.3	> Illumina1.5
ASCII Quality	ASCII Quality	ASCII Quality	ASCII Quality
! 33 0.0000	; 59 0.2403	@ 64 0.5000	B 66 0.3690
" 34 0.2057	< 60 0.2847	A 65 0.5573	C 67 0.4988
# 35 0.3690	= 61 0.3339	B 66 0.6131	D 68 0.6019
\$ 36 0.4988	> 62 0.3869	C 67 0.6661	E 69 0.6838
% 37 0.6019	? 63 0.4427	D 68 0.7153	F 70 0.7488
& 38 0.6838	@ 64 0.5000	E 69 0.7597	G 71 0.8005
' 39 0.7488	A 65 0.5573	F 70 0.7992	H 72 0.8415
(40 0.8005	B 66 0.6131	G 71 0.8337	I 73 0.8741
) 41 0.8415	C 67 0.6661	H 72 0.8632	J 74 0.9000
* 42 0.8741	D 68 0.7153	I 73 0.8882	K 75 0.9206
+ 43 0.9000	E 69 0.7597	J 74 0.9091	L 76 0.9369
,	F 70 0.7992	K 75 0.9264	M 77 0.9499
,	G 71 0.8337	L 76 0.9406	N 78 0.9602
- 45 0.9369	H 72 0.8632	M 77 0.9523	O 79 0.9684
.	I 73 0.8882	N 78 0.9617	P 80 0.9749
/ 47 0.9602	J 74 0.9091	O 79 0.9693	Q 81 0.9800
0 48 0.9684	K 75 0.9264	P 80 0.9755	R 82 0.9842
1 49 0.9749	L 76 0.9406	Q 81 0.9804	S 83 0.9874
2 50 0.9800	M 77 0.9523	R 82 0.9844	T 84 0.9900
	N 78 0.9617	S 83 0.9876	U 85 0.9921

Quality Control

Need to preprocess the reads:

- Check for quality (FASTQC)
- Trim adapter and (Cutadapt, others)
- Remove duplicate reads, trim low complexity/quality bases/reads (Prinseq)
- Complete pipelines: NCBI Toolkit, QC-Chain, PathoQC (pathoscope.sourceforge.net), others

Note: Not comprehensive or updated!

Read Quality Checks

Step 1: User inputs sequencing reads (fastq) and primer and adapter sequences

Sequencing Reads
(fastq)

Primers & Adapters
CTCGGCATTCTGT...
GATCTATTATACTCC...

Step 2: FastQC

- Detect Phred offset
- Search for sequence tags

Steps 3 & 4: Process sequencing reads

Cutadapt

- Trim sequence primers and adapters

Prinseq

- Trim low quality bases
- Remove short sequences low complexity and redundant reads

Step 5: Output quality controlled sequencing reads

Clean Sequencing Reads

Read Quality Checks (Outdated!)

Comparison of some of the more popular QC pipelines:

QC Features	QCToolkit	QC-Chain	Cutadapt	Prinseq
Parallel computation	X	X		
Phred offset detection	X	X		
Tag sequence removal	X	X	X	
Poly-A/T tail trimming			X	X
PCR duplication filtering			X	X
Low complexity filtering				X
Homopolymer removal	X			X
GC content filtering		X		X
N/X content filtering				X

FASTQC



FastQC Example

Running FastQC from command-line:

```
fastqc myfastqfile.fq --outdir=output/
```

Interactive GUI:

```
fastqc
```

Section 5

Mapping Reads

Sequencing data alignment

Next-Generation Sequencing (NGS) Data present new challenges:

- Map 'reads' to genome
- Call SNPs and variants from the reads
- Other Applications: RNA-seq, miRNAs, alternative splicing, ChIP-seq, BS-seq, RNA editing

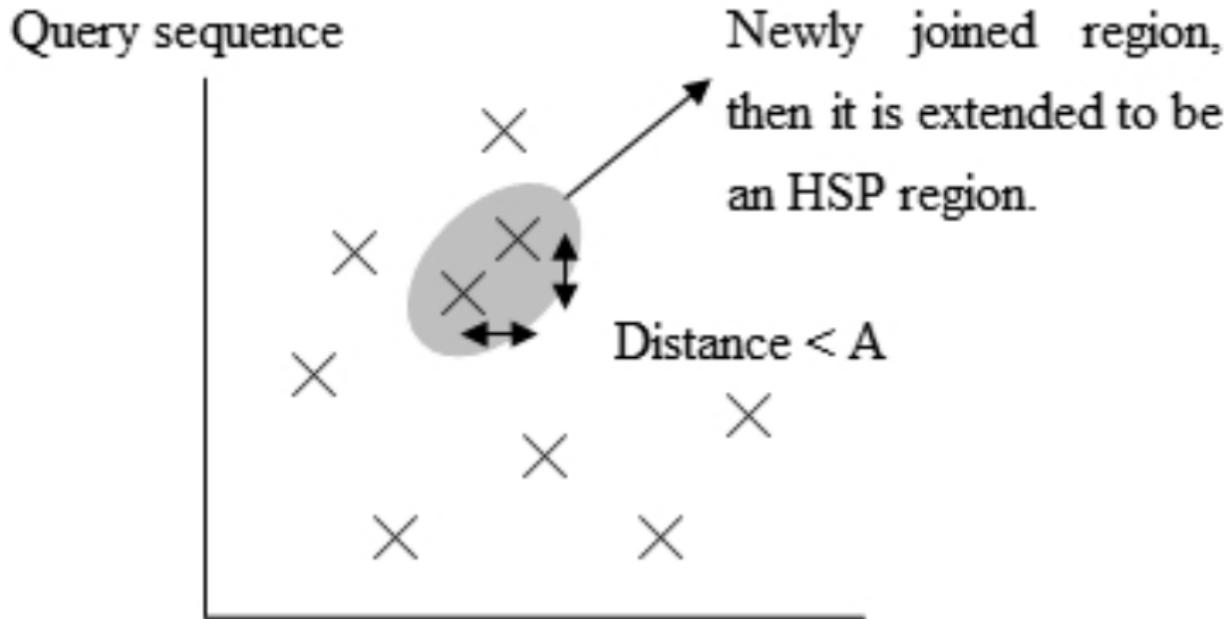
Obstacles:

- Massive data size
- Repeat regions, rare variants

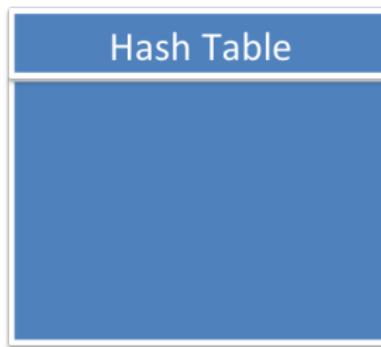
Goal: Develop an approach to put the puzzle together!

Basic Local Alignment Search Tool (BLAST)

BLAST seed extension:



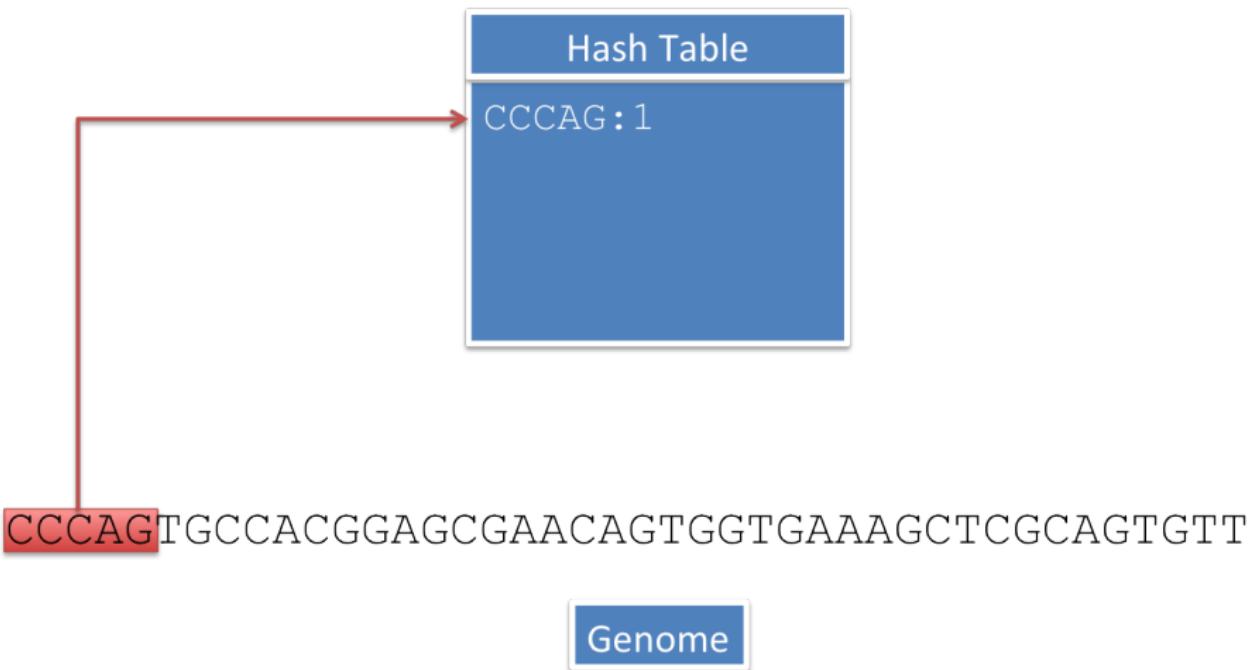
Creating a Hash Index Table



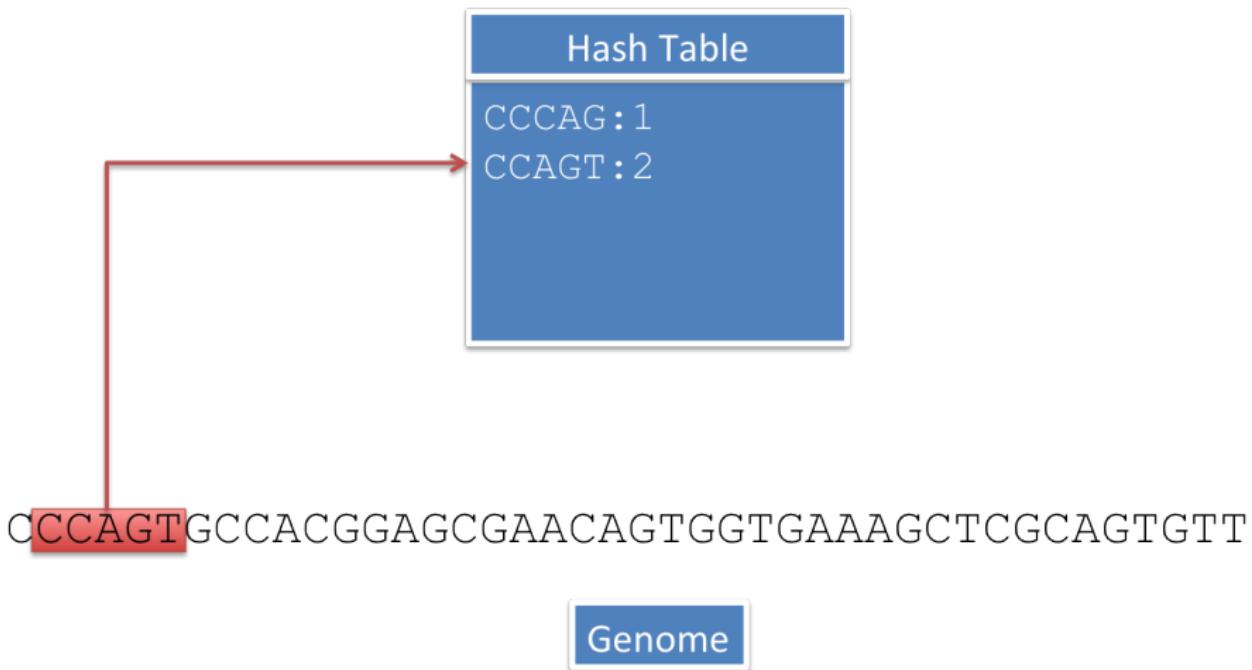
CCCAGTGCCACGGAGCGAACAGTGGTGAAAGCTCGCAGTGT

Genome

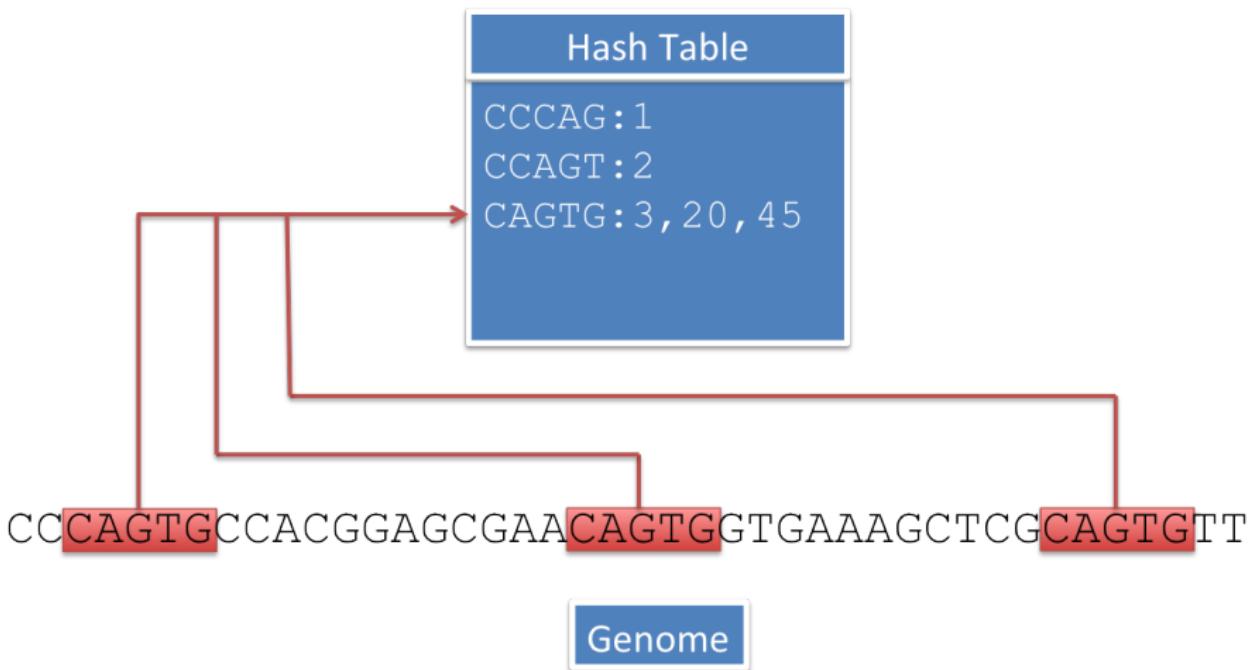
Creating a Hash Index Table



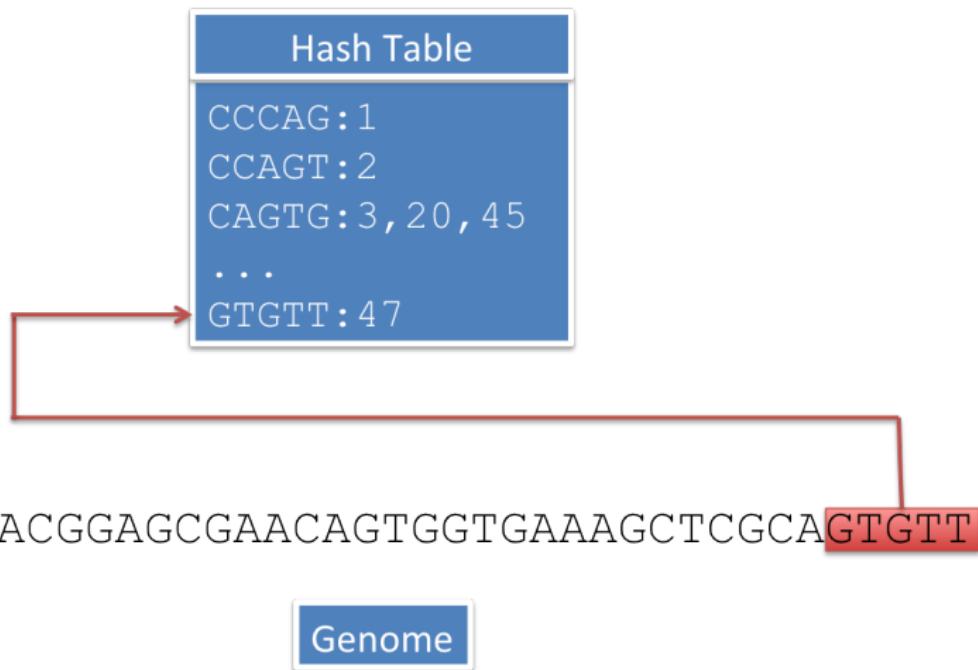
Creating a Hash Index Table



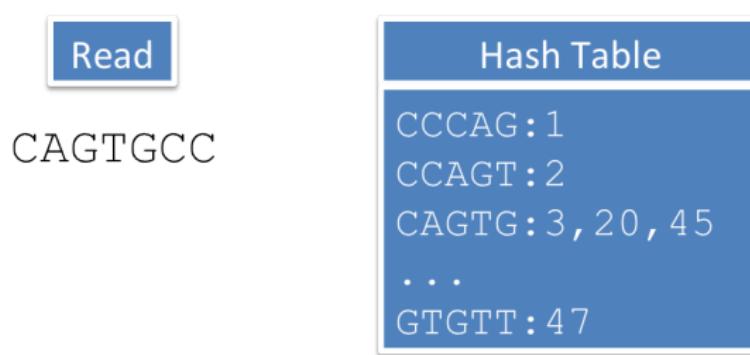
Creating a Hash Index Table



Creating a Hash Index Table



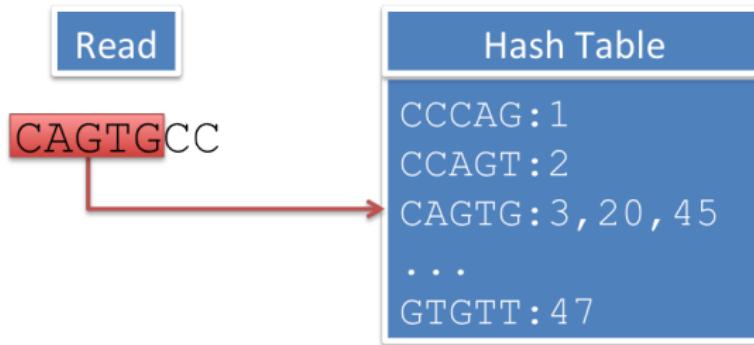
Read Lookup



CCCAGTGCCACGGAGCGAACAGTGCTGAAAGCTCGCAGTGT

Genome

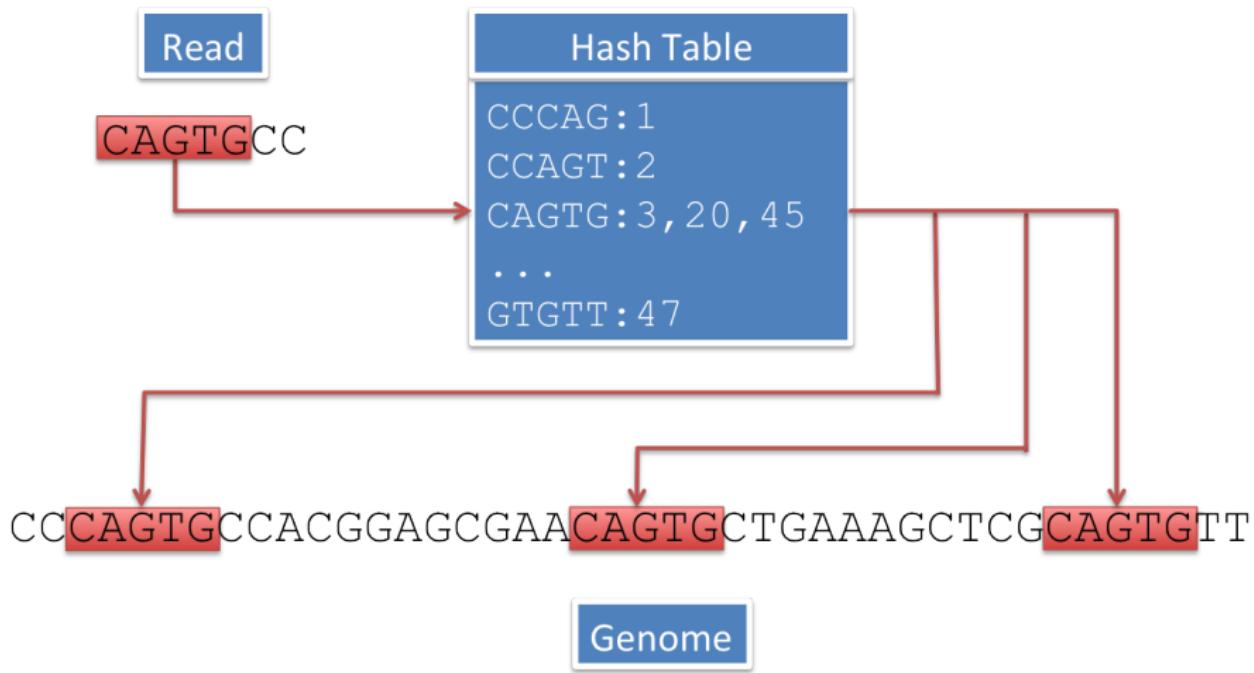
Read Lookup



CCCAGTGCCACGGAGCGAACAGTGCTGAAAGCTCGCAGTGT

Genome

Read Lookup



Read Lookup

Read

CAGTGCC

Hash Table

CCCAG:1
CCAGT:2
CAGTG:3,20,45
...
GTGTT:47

CAGTGCC

|||||||

CCCAGTGCCACGGAGCGAA**CAGTGCT**GAAAGCTCG**CAGTGTT**

Genome

Read Lookup

Read

CAGTGCC

Hash Table

CCCAG:1
CCAGT:2
CAGTG:3,20,45
...
GTGTT:47

CAGTGCC

|||||||

CCCAGTGCCACGGAGCGAA

Score=7

CAGTGCC

|||||||

CAGTGCTGAAAGCTCG

Score=5

Genome

CAGTGCC

|||||||

CAGTGTT

Score=3

Needleman-Wunsch

T A C A T C

T T C A - C

	t	a	c	a	t	c
t						
t						
c						
a						
c						

- Allows for mismatches and gaps
- Scoring function: +1 match; -1 for mismatch; -2 for gap
- Produces maximum likelihood alignment

Needleman-Wunsch

T A C A T C

T T C A - C

	t	a	c	a	t	c
t	0	-2	-4	-6	-8	-10
t	-2					
c	-4					
a	-6					
c	-8					
c	-10					

- Allows for mismatches and gaps
- Scoring function: +1 match; -1 for mismatch; -2 for gap
- Produces maximum likelihood alignment

Needleman-Wunsch

```

T A C A T C
|
T T C A - C
  
```

	t	a	c	a	t	c	
t	0	-2	-4	-6	-8	-10	-12
t	-2	1					
t	-4						
c	-6						
a	-8						
c	-10						

- Allows for mismatches and gaps
- Scoring function: +1 match; -1 for mismatch; -2 for gap
- Produces maximum likelihood alignment

Needleman-Wunsch

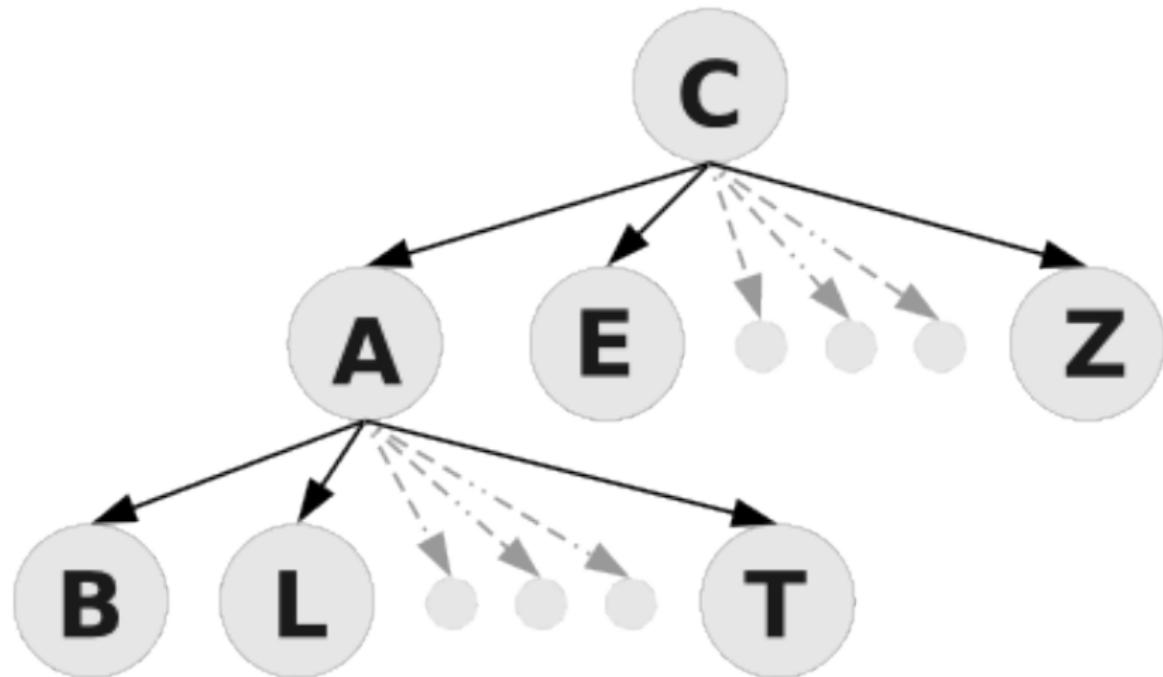
```

T A C A T C
|   |   |
T T C A - C
  
```

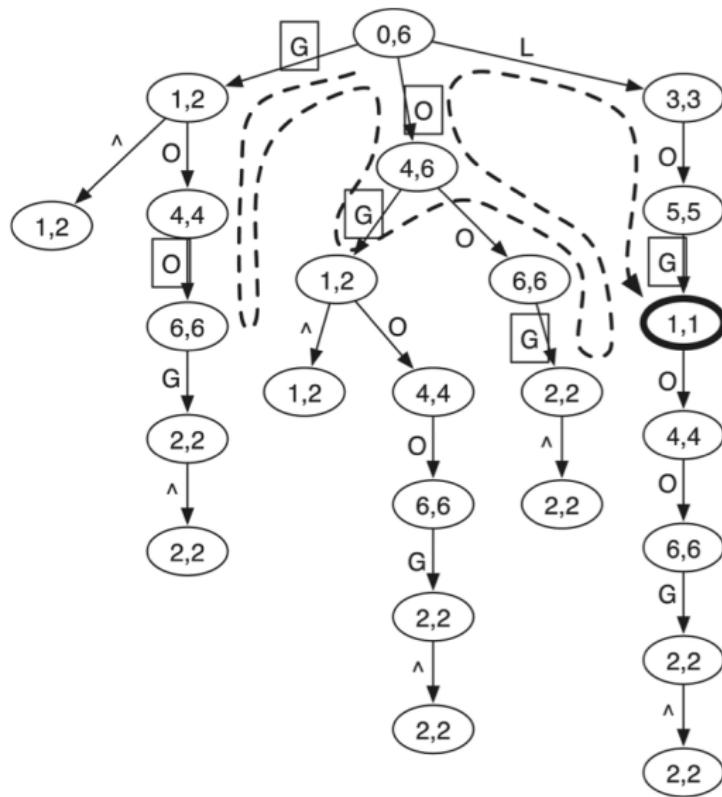
	t	a	c	a	t	c	
t	0	-2	-4	-6	-8	-10	-12
t	-2	1	-1	-3	-5	-7	-9
t	-4	-1	0	-2	-4	-4	-6
c	-6	-3	-2	1	-1	-3	-3
a	-8	-5	-2	-1	2	0	-2
c	-10	-7	-4	-1	0	1	1

- Allows for mismatches and gaps
- Scoring function: +1 match; -1 for mismatch; -2 for gap
- Produces maximum likelihood alignment

Prefix Trees



Prefix Trees



Li H , and Durbin R Bioinformatics 2009;25:1754-1760

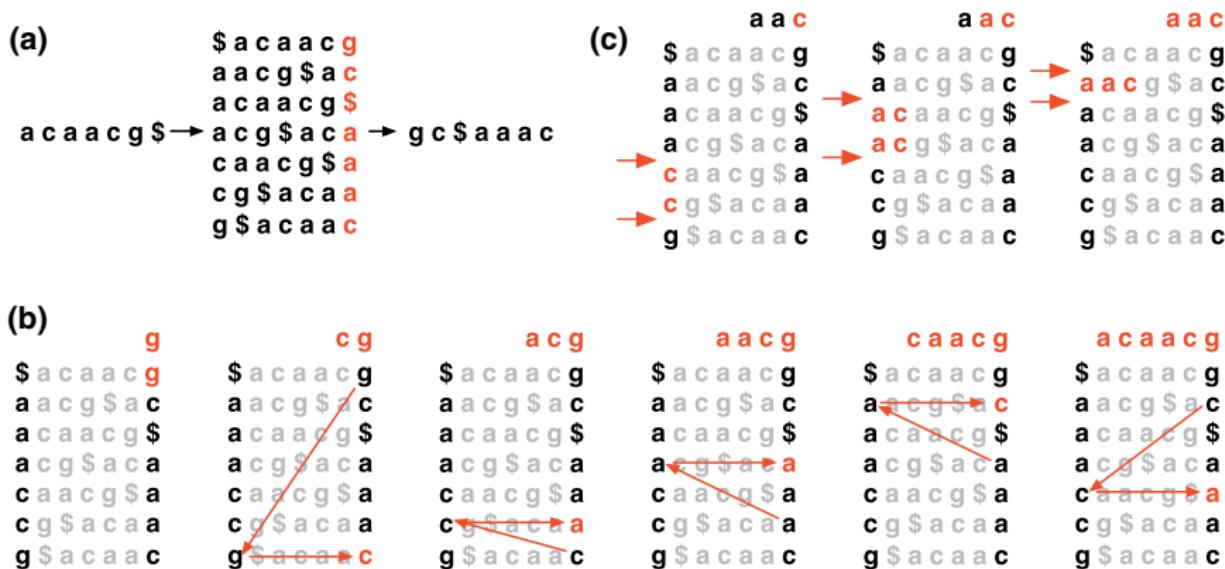
Burrows-Wheeler Transformation (BWA, Bowtie)



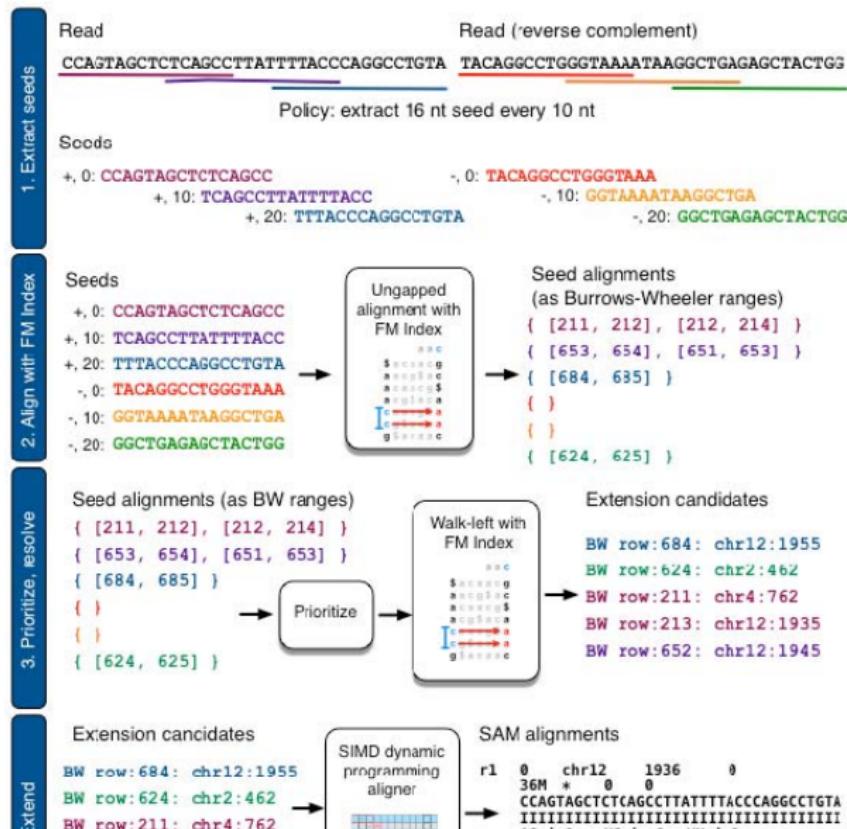
Pos

 $X = \text{googol\$}$
 i $S(i)$ $B[i]$
↓ ↓ ↓
 $(6, 3, 0, 5, 2, 4, 1)$
 $\text{lo\$oogg}$

Burrows-Wheeler Transformation (BWA, Bowtie)



Hybrid Methods (BWA-SW and MEM, Bowtie2)



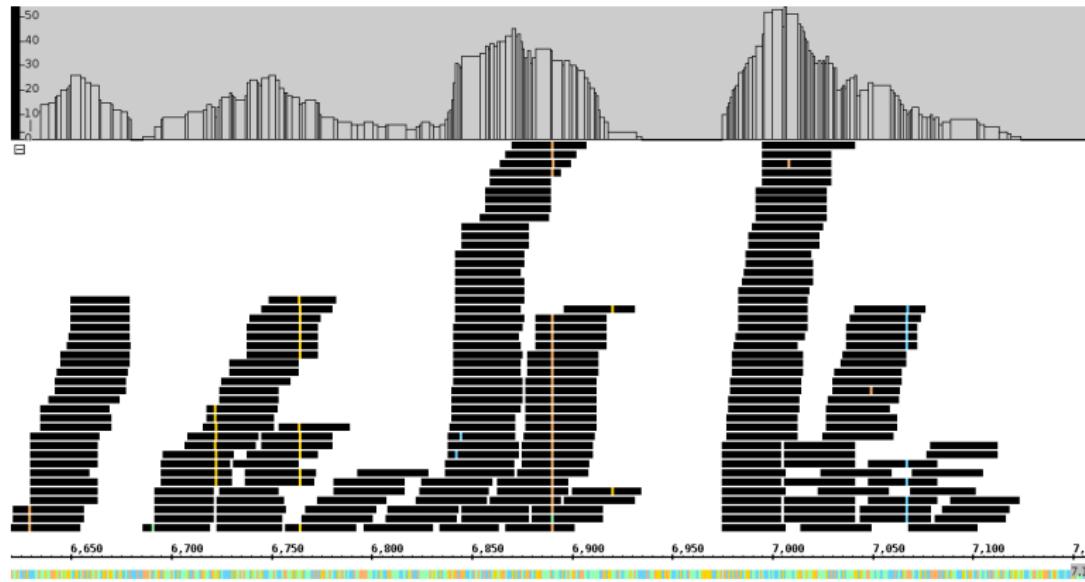
Section 6

Visualization of Aligned Sequencing Data

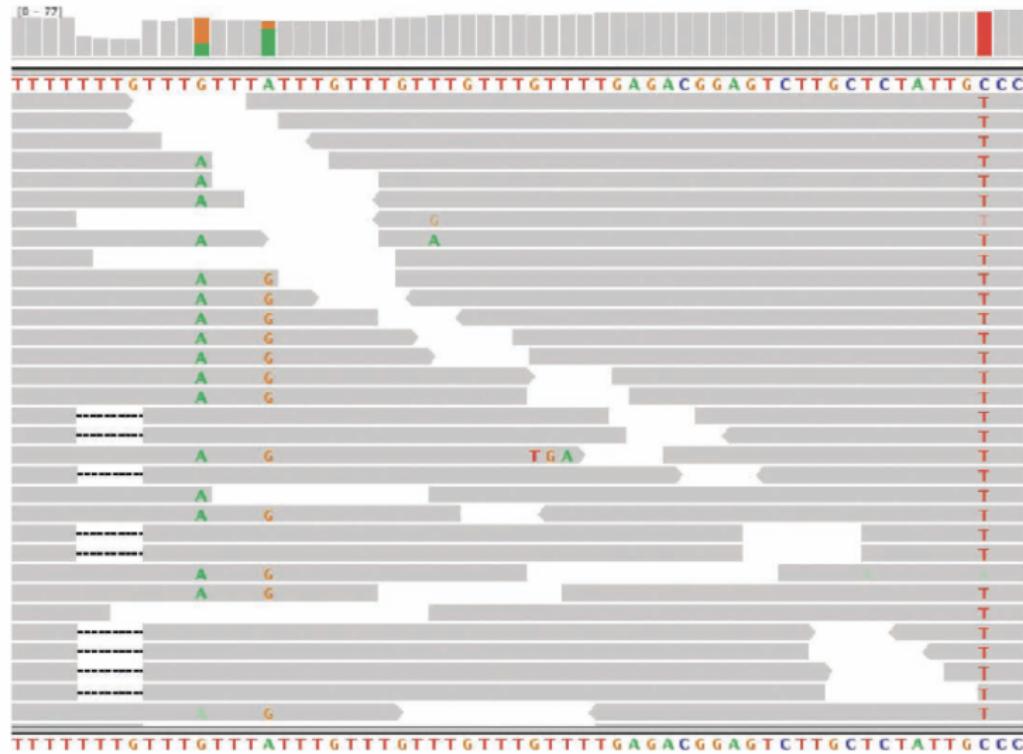
Options for data visualization

- UCSC Genome Browser
- IGB (<http://bioviz.org/igb/>)
- IGV (<http://www.broadinstitute.org/igv/>)
- SAMtools

IGB Example



IGV Example



DePristo et al. *Nature Genetics* 2011; 43: 491-8.

SAM Format

Col	Field	Type	Regexp/Range	Brief description
1	QNAME	String	[!-?A-~]{1,255}	Query template NAME
2	FLAG	Int	[0,2 ¹⁶ -1]	bitwise FLAG
3	RNAME	String	* [!-()+-<>-~] [!-~]*	Reference sequence NAME
4	POS	Int	[0,2 ²⁹ -1]	1-based leftmost mapping POSition
5	MAPQ	Int	[0,2 ⁸ -1]	MApping Quality
6	CIGAR	String	* ([0-9]+[MIDNSHPX=])+	CIGAR string
7	RNEXT	String	* = [!-()+-<>-~] [!-~]*	Ref. name of the mate/next segment
8	PNEXT	Int	[0,2 ²⁹ -1]	Position of the mate/next segment
9	TLEN	Int	[-2 ²⁹ +1,2 ²⁹ -1]	observed Template LENgth
10	SEQ	String	* [A-Za-z.=.]+	segment SEQuence
11	QUAL	String	[!-~]+	ASCII of Phred-scaled base QUALity+33

SAMtools

```

152853001 152853011 152853021 152853031 152853041 152853051 152853061
TGCCTAAATTCTAGAAGCTGCCACCTGGGCCAGGAAGGCCATGGTAGAAGTAGTATTCTGTTAGTCGGC
.....G.....
tgcc aaattcagaagctgcccacctggggccagggcatggtagaagtagtatttcatctggtagttctcgccc
tgcccaaattcagaagctgcccacc GGGGCCAGGGAAAGGCCATGGTAGAAGTAGTATTCTGTTAGTCGGC
tgcccaaattcagaagctgcccacc GGGGCCAGGGAAAGGCCATGGTAGAAGTAGTATTCTGTTAGTCGGC
tgcccaaattcagaagctgcccacct GGGCCAGGGAAAGGCCATGGTAGAAGTAGTATTCTGTTAGTCGGC
tgcccaaattcagaagctgcccacct ggcattggtagaagtagtatttcatctggtagttctcgccc
tgcccaaattcagaagctgcccacct ggggcccaggaa ggcattggtagaagtagtatttcatctggtagttctcgccc
tgcccaaattcagaagctgcccacct ggggcccaggaaagg CATGGTAGAAGTAGTATTCTGTTAGTCGGC
tgcccaaattcagaagctgcccacct ggggcccaggaaaggccatg TAGAAGTGGTATTCTGTTAGTCGGC
tgcccaaattcagaagctgcccacct ggggcccaggaaaggcatg GAAGTAGTATTCTGTTAGTCGGC
tgcccaaattcagaagctgcccacct ggggcccaggaaaggcatg gaagtagtatttcatctggtagttctcgccc
TGCCCTAAATTCTAGAAGCTGCCACCTGGGCCAGGGAAAGGCCATGGTAGAA tagtatttcatctggtagttctcgccc
TGCCCTAAATTCTAGAAGCTGCCACCTGGGCCAGGGAAAGGCCATGGTAGAA agtatttcatctggtagttctcgccc
tgcccaaattcagaagctgcccacct ggggcccaggaaaggccatggtagaagtagtattt ATCTGGTAGTCGGC
tgcccaaattcagaagctgcccacct ggggcccaggaaaggccatggtagaagtagtattt tctggtagttctcgccc
TGCCCTAAATTCTAGAAGCTGCCACCTGGGCCAGGGAAAGGCCATGGTAGAAGTAGTATTCTC TCTGGTAGTCGGC
tgcccaaattcagaagctgcccacct ggggcccaggaaaggccatggtagaagtagtattt tctggtagttctcgccc
taaaaaaattcagaagctccccacctaagggcaaggaaaggccataatagaagtagtatttcatc GGTAGTTCTCGGGC

```

Section 7

Other Data Formats

Standard format for keeping tables

field1	field2	field3	...
...

Fields (columns) separated by a character on each line:

- Comma (or Character) Separated Vector (CSV)
- Tab Separated Vector (TSV)
- Some interpreters take any space (space or tab) as a separator (such as `awk`, `cut`).
- Some have column name as first row (header), some don't

Genomic regions

-A region is defined by three required fields: - sequence name (e.g. chromosome) - start coordinate - end coordinate - Define regions of interest: introns, exons, genes, etc. - Additional information saved as fields after the first three. - Three standard tab-separated formats: BED, GFF, GTF No headers

BED Format

Mandatory fields:

- ① chrom - Name of the chromosome/scaffold/reference sequence
- ② chromStart - 0-based starting position of the feature on chrom
- ③ chromEnd - Ending position of the feature in the chromosome or scaffold.

The chromEnd base is not included in the display of the feature.

For example, the first 100 bases of a chromosome are defined as:

- chromStart=0
- chromEnd=100
- span the bases numbered 0-99

BED Format

chr1	11873	14409	uc001aaa.3	0	+	11873	11873
chr1	11873	14409	uc010nxr.1	0	+	11873	11873
chr1	11873	14409	uc010nxq.1	0	+	12189	13639
chr1	14361	16765	uc009vis.3	0	-	14361	14361
chr1	14361	19759	uc009vit.3	0	-	14361	14361
chr1	14361	19759	uc009viu.3	0	-	14361	14361
chr1	14361	19759	uc001aae.4	0	-	14361	14361
chr1	14361	29370	uc001aah.4	0	-	14361	14361
chr1	14361	29370	uc009vir.3	0	-	14361	14361
chr1	14361	29370	uc009viq.3	0	-	14361	14361
chr1	14361	29370	uc001aac.4	0	-	14361	14361
chr1	14406	29370	uc009viv.2	0	-	14406	14406
chr1	14406	29370	uc009viw.2	0	-	14406	14406
chr1	15602	29370	uc009vix.2	0	-	15602	15602
chr1	15795	18061	uc009vjd.2	0	-	15795	15795
chr1	16606	29370	uc009viy.2	0	-	16606	16606
chr1	16606	29370	uc009viz.2	0	-	16606	16606
chr1	16857	17751	uc009vjc.1	0	-	16857	16857
chr1	16857	19759	uc001aaai.1	0	-	16857	16857

BED Format

Human chr22 - UCSC Genome Browser

genome.ucsc.edu/cgi-bin/hgTracks?db=hg19&position=chr22%3A20100000-2010090...

Genomes Genome Browser Tools Mirrors Downloads My Data View Help About Us

UCSC Genome Browser on Human Feb. 2009 (GRCh37/hg19) Assembly

move <<< << < > >> zoom in 1.5x 3x 10x base zoom out 1.5x 3x 10x 100x

chr22:20,100,000-20,100,900 901 bp. enter position, gene symbol or search terms go

chr22 (q11.21) 22p13 22p12 p11.2 q11.21 q12.1 q12.2 22q12.3 q13.1 q13.2 q13.31

Scale: 200 bases hg19
chr22: | 20,100,100| 20,100,200| 20,100,300| 20,100,400| 20,100,500| 20,100,600| 20,100,700| 20,100,800|
Color by strand demonstration
Chromosome coordinates list

start

move start < 2.0 > move end < 2.0 >
track search default tracks default order hide all manage custom tracks track hubs configure reverse resize refresh
collapse all expand all

Click on a feature for details. Click or drag in the base position track to zoom in. Click side bars for track options. Drag side bars or labels up or down to reorder tracks. Drag tracks left or right to new position.

Use drop-down controls below and press refresh to alter tracks displayed. Tracks with lots of items will automatically be displayed in more compact modes.

BED Format

Optional fields:

- ④ Name
- ⑤ Score
- ⑥ Strand
- ⑦ 7-12. Display options (thick starts and end, color, blocks...) to control the view on the genome browser

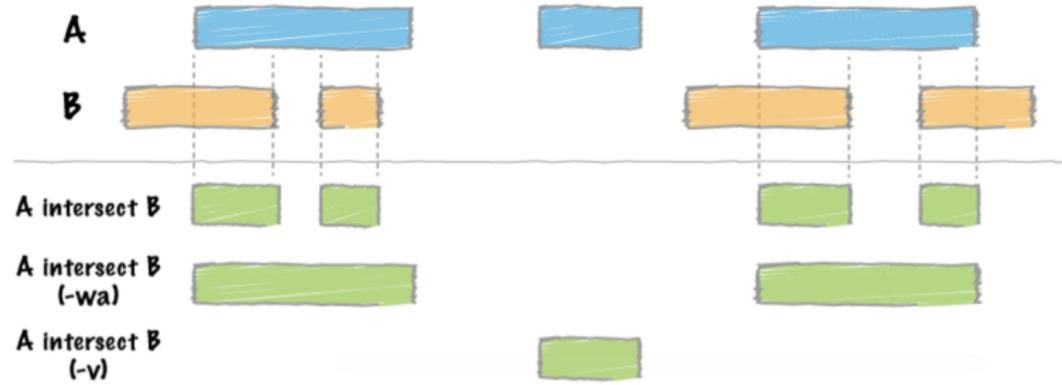
bedtools

<http://bedtools.readthedocs.io/>

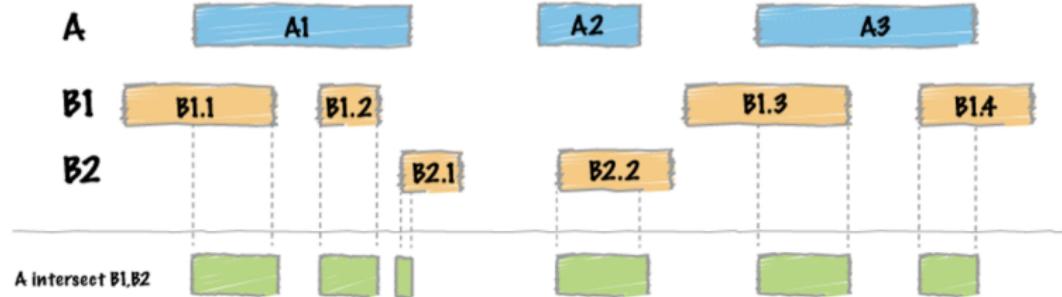
- sort (sort bed files)
- Intersect (get intersections of bed files)
- merge
- coverage
- overlap
- subtract
- ...

bedtools intersect

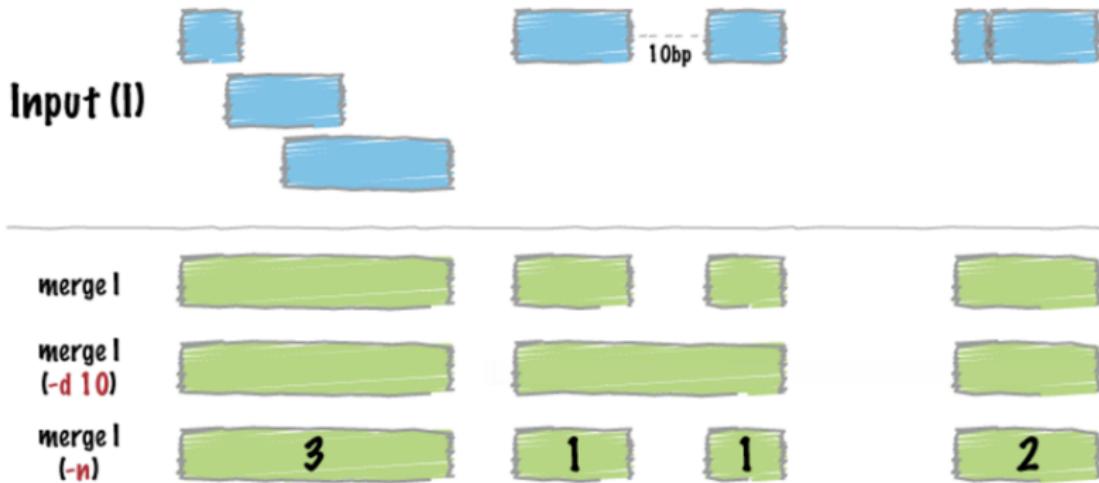
Intersect w/
1 database



Intersect w/
or more databases



bedtools merge



GFF - General features

9 mandatory fields, tab separated:

- ① seqname - The name of the sequence. Must be a chromosome or scaffold.
- ② source - The program that generated this feature.
- ③ feature - The name of this type of feature (e.g. gene, exon, etc).
- ④ start - The starting position of the feature in the sequence (1-based)
- ⑤ end - The ending position of the feature (inclusive).
- ⑥ score - A score between 0 and 1000.
- ⑦ strand - Valid entries include "+", "-", or "." (for don't know/don't care).
- ⑧ frame - If the feature is a coding exon, frame should be a number between 0-2 that represents the reading frame of the first base. If the feature is not a coding exon, the value should be ":".
- ⑨ group - All lines with the same group are linked together into a single item

Gene Information

GTF (Gene Transfer Format, GTF2.2)

- Extension to GFF2, backwards compatible
- First eight GTF fields are the same as GFF
- *feature field* is the same as GFF, has controlled vocabulary:
 - *gene, transcript, exon, CDS, 5UTR, 3UTR, inter, inter_CNS, and intron_CNS, etc*
- group field expanded into a list of attributes (i.e. key/value pairs)

The attribute list must begin with the one mandatory attribute: - gene_id value - A globally unique identifier for the genomic source of the sequence

GTF format

```
##description: evidence-based annotation of the human genome (GRCh38), version 27 (Ensembl 90)
##provider: GENCODE
##contact: gencode-help@sanger.ac.uk
##format: gtf
##date: 2017-08-01
chr1 HAVANA gene 923928 944581 . + . gene_id "ENSG00000187634.11"; gene_type
"protein_coding"; gene_name "SAMD11"; level 2; havana_gene "OTTHUMG00000040719.10";
```

- `seqname: chr1`
- `source: HAVANA`
- `feature: gene`
- `start: 923928`
- `end: 944581`
- `score: . (no score)`
- `strand: +`
- `frame: . (not coding feature)`
- `attributes:`
 - `gene_id: ENSG00000187634.11`
 - `gene_type: protein_coding`
 - `gene_name: SAMD11`
 - `level: 2`
 - `havana_gene: OTTHUMG00000040719.10`

Session info

```
sessionInfo()
```

```
## R version 4.4.0 (2024-04-24)
## Platform: aarch64-apple-darwin20
## Running under: macOS Sonoma 14.2.1
##
## Matrix products: default
## BLAS:    /Library/Frameworks/R.framework/Versions/4.4-arm64/Resources/lib/libRblas.0.dylib
## LAPACK:  /Library/Frameworks/R.framework/Versions/4.4-arm64/Resources/lib/libRlapack.dylib;  LAPACK version 3
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## time zone: America/Denver
## tzcode source: internal
##
## attached base packages:
## [1] stats      graphics   grDevices  utils      datasets   methods    base
##
## loaded via a namespace (and not attached):
## [1] compiler_4.4.0  fastmap_1.1.1   cli_3.6.2       tools_4.4.0
## [5] htmltools_0.5.8.1 rstudioapi_0.16.0 yaml_2.3.8     rmarkdown_2.26
## [9] knitr_1.46      xfun_0.43      digest_0.6.35   rlang_1.1.3
## [13] evaluate_0.23
```