Mining the genome.

Aaron Quinlan

quinlanlab.org

University of Virginia, Charlottesville VA

Center for Public Health Genomics

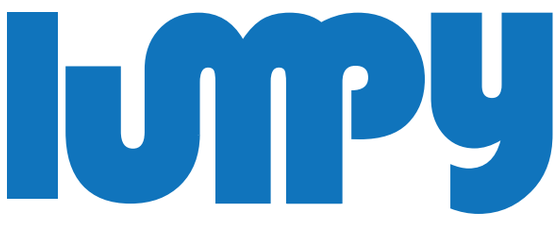
Biochemistry and Molecular Genetics

Our research

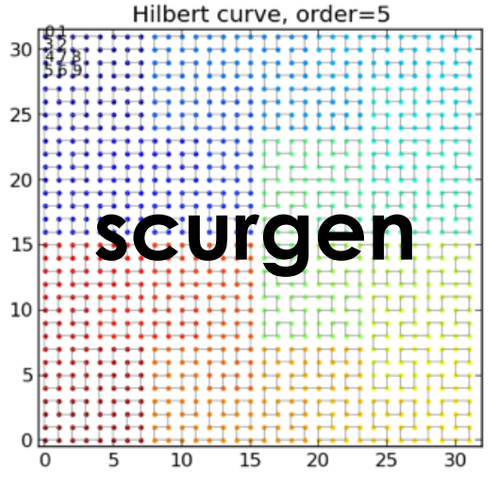
Software for genome research

gemini

hydra



cyvcf
pybedtools
grabix
bioawk



scurgen

Our research

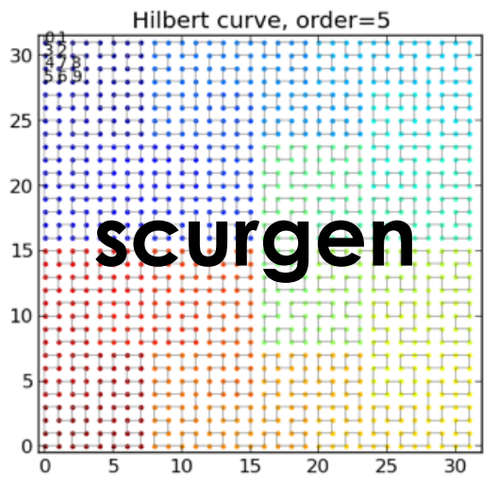
Software for genome research

gemini

hydra

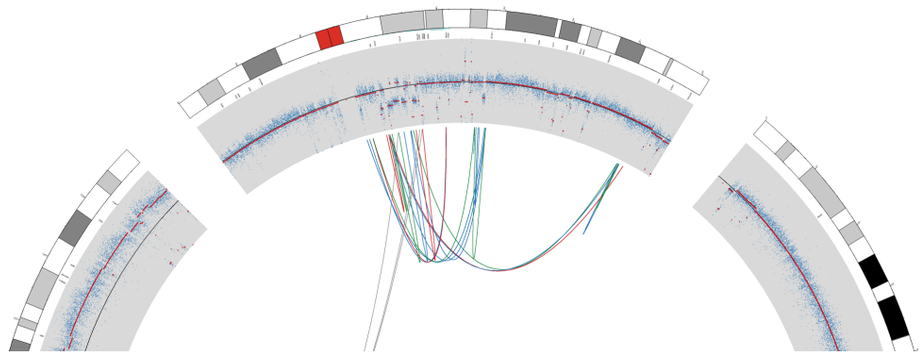
cyvcf
pybedtools
grabix
bioawk

lumpy



scurgen

Cancer genomics



1. Cellular and molecular origins of GBM
2. Initiating mutations underlying OV?
3. Tumor evolution. **Ira Hall's talk on Thurs.**

Human evolution

Understanding "unexplained" conservation

Disease genetics

Type 1 diabetes
SLE (Lupus)
Unexplained development disorders
Radiation hypersensitivity

Key contributors

Uma Paila, Ph.D.

Postdoctoral Research Associate

udp3f @ virginia.edu

Research Projects and Interests: Investigation of the genetic basis of extreme sensitivity to ionizing radiation; development of new analytical tools for exploring genetic variation identified through next-generation sequencing projects.

John Kubinksi
Undergraduate wunderkind

Neil Kindlon, M.S.

Staff Scientist and Software Engineer

nek3d @ virginia.edu

Research Projects and Interests: Software development for genomic analysis. Structural variation discovery and interpretation using DNA sequencing technologies.

Ryan Layer

Graduate student

rl6sf @ virginia.edu

Research Interests: Scalable algorithm development for high-throughput genomic analysis; genome data mining and analysis; structural variation discovery and interpretation.

Outline

1. **Exploring genomic variation**
    - Challenges
    - Approaches
    - Exploring the genetic basis of radiation sensitivity
2. **Break**
3. **Mining genomic datasets ➤ genome biology**
    - Algorithms and Tools
    - Visualization
    - Future

Gemini:

a flexible framework for mining genome variation

Uma Paila, Ph.D.

Postdoctoral Research Associate

udp3f @ virginia.edu

https://github.com/arq5x/gemini

Gemini overview

- **Goal**: unified framework for exploring genetic variation for disease and population genetics.
- Samples ➤ Genomes ➤ Variants ➤ VCF ➤ Gemini ➤ Test **H**
- Annotate variants with wealth of genome annotations
- Structured querying interface
- Extensible for new tool development
- Powerful, yet easy to use

Other options

PLINK/SEQ

A library for the analysis of genetic variation data

ANNOVAR

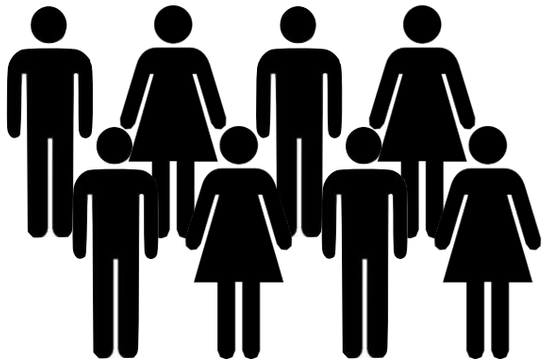
gSearch

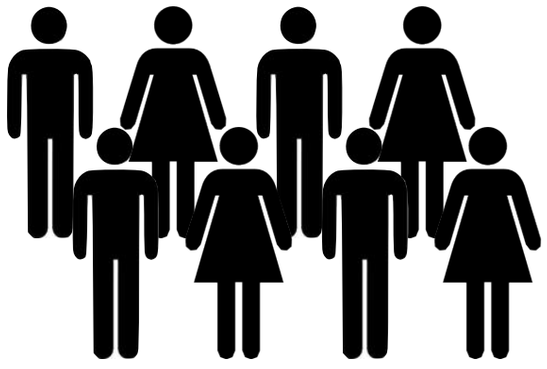
AnnTools

VAAST

A typical study

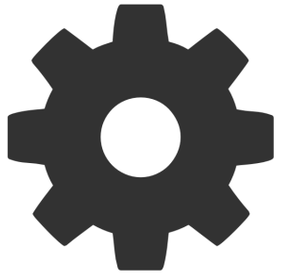
with phenotype





without phenotype

Genome sequencing



Align sequences
Scrub data
Call variants

Interpretation

Variants for all samples are stored in a VCF file*

1bp

accgtca
acctca

Single-nucleotide polymorphisms (SNPs)

3 million / genome

1–100 bp

accgggtca
acc---tca

Insertion/deletions (INDELs)

400K / genome

101bp – 10mb

Structural Variants (SVs)

3,000 - 10,000 / genome

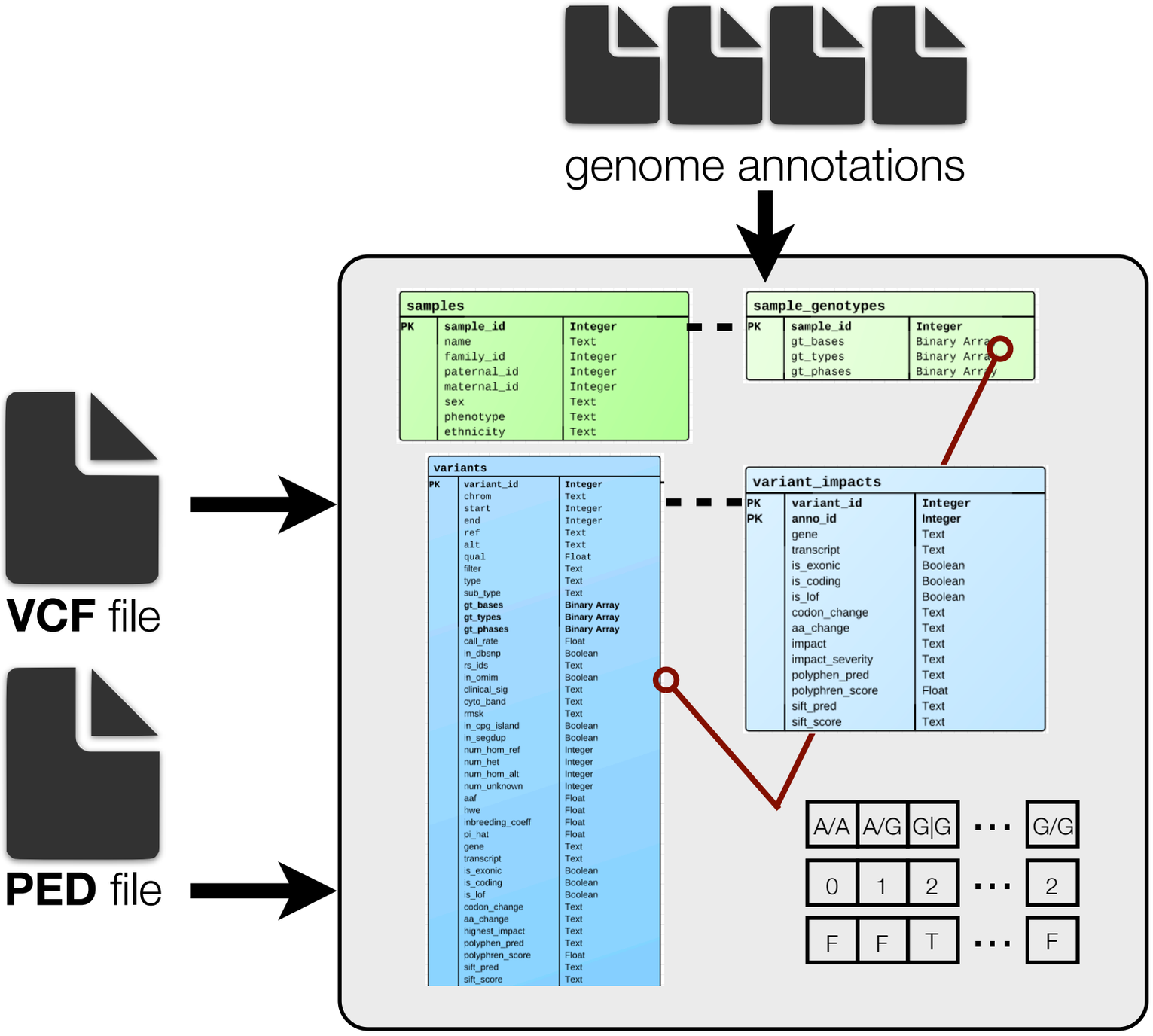
Genotypes

Variants

VCF file

*much more on this later

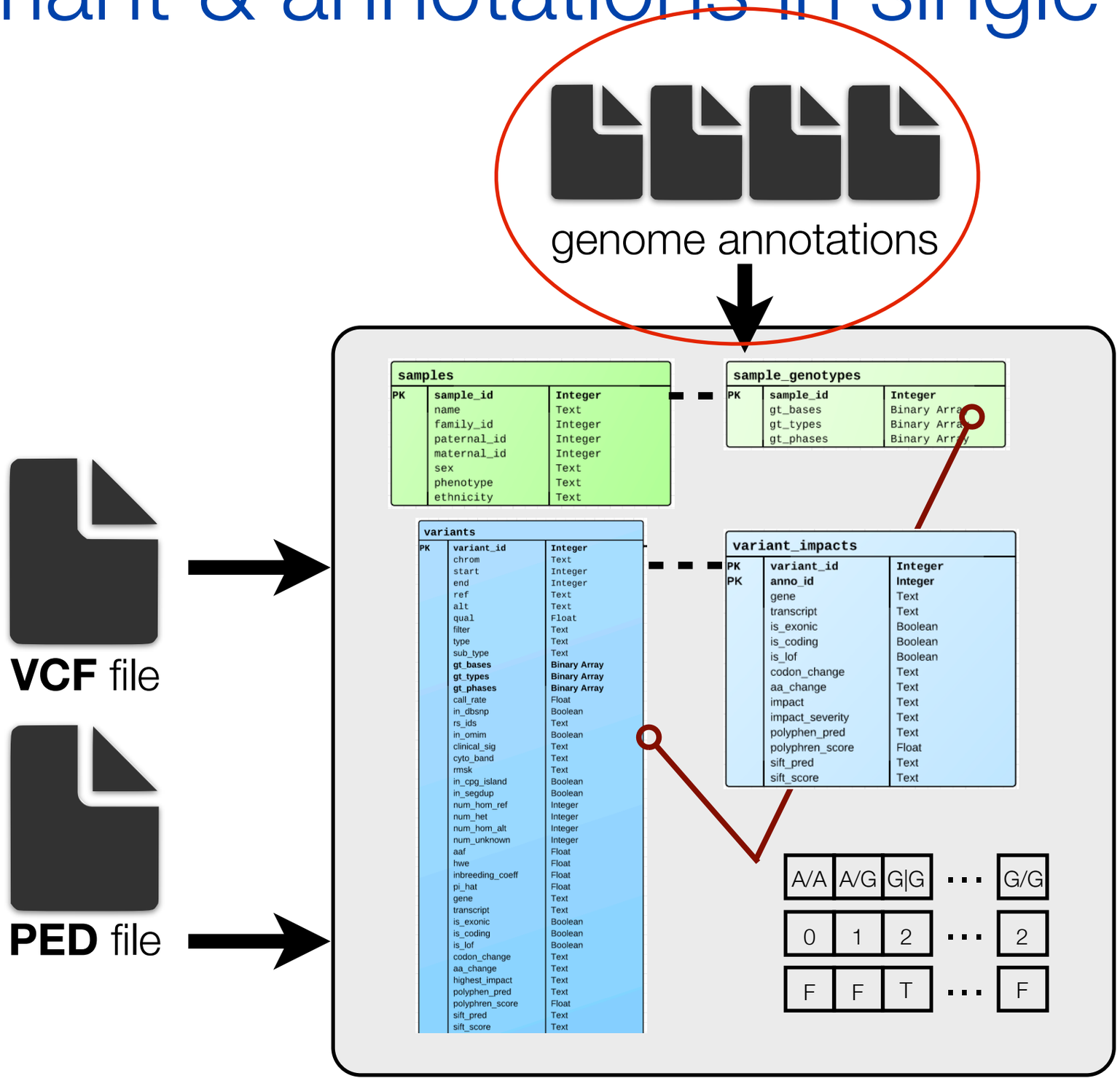
Variant & annotations in single framework



Compressed arrays of genotype information accommodates 1000s of genotypes per variant.

```
$ gemini load -v my.vcf -t VEP my.db
```

Variant & annotations in single framework



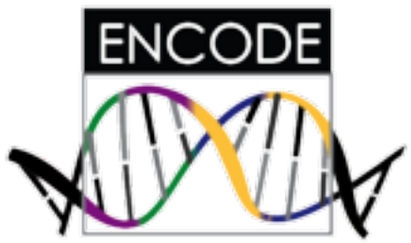
Compressed arrays of genotype information accommodates 1000s of genotypes per variant.

```
$ gemini load -v my.vcf -t VEP my.db
```

Genome annotations place variants in context

OMIM

CpG
Repeats
SegDups
Conservation
(more)



DNase
FAIRE
TF Binding
chromatin states

Recombination

COSMIC

dbSNP

ESP

pathways

protein interactions

Variants, genotypes, & sample info all in a single database

samples		
PK	**sample_id**	**Integer**
	name	Text
	family_id	Integer
	paternal_id	Integer
	maternal_id	Integer
	sex	Text
	phenotype	Text
	ethnicity	Text

sample_genotypes		
PK	**sample_id**	**Integer**
	gt_bases	Binary Array
	gt_types	Binary Array
	gt_phases	Binary Array

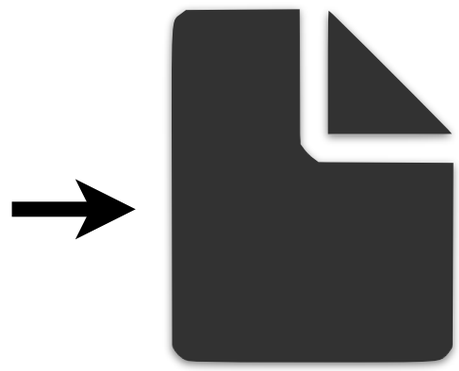
variants		
PK	**variant_id**	**Integer**
	chrom	Text
	start	Integer
	end	Integer
	ref	Text
	alt	Text
	qual	Float
	filter	Text
	type	Text
	sub_type	Text
	gt_bases	**Binary Array**
	gt_types	**Binary Array**
	gt_phases	**Binary Array**
	call_rate	Float
	in_dbsnp	Boolean
	rs_ids	Text
	in_omim	Boolean
	clinical_sig	Text
	cyto_band	Text
	rmsk	Text
	in_cpg_island	Boolean
	in_segdup	Boolean

variant_impacts		
PK	**variant_id**	**Integer**
PK	**anno_id**	**Integer**
	gene	Text
	transcript	Text
	is_exonic	Boolean
	is_coding	Boolean
	is_lof	Boolean
	codon_change	Text
	aa_change	Text
	impact	Text
	impact_severity	Text
	polyphen_pred	Text
	polyphren_score	Float
	sift_pred	Text
	sift_score	Text

Ad hoc data exploration

Enhanced SQL engine allows **selection** and **filtering** based on individual genotypes. **Scales to 1000s of samples.**

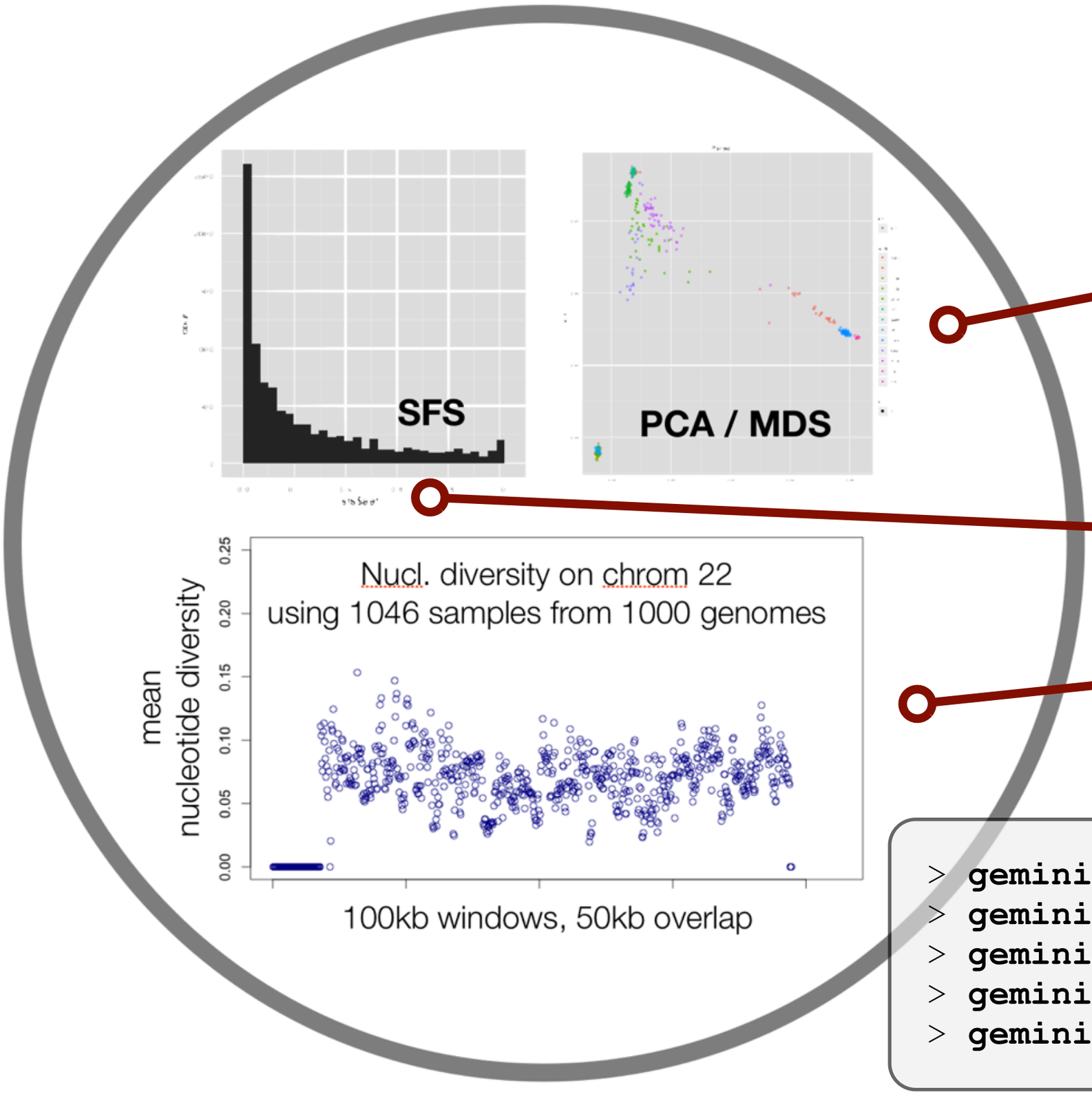
```
SELECT chrom,
start, gene,
impact, hwe,
in_dbsnp, in_omim,
gts.NA12878
FROM variants
WHERE is_lof = 1
AND aaf <= 0.01
AND call_rate \
    >= 0.95
```



Output files in standard formats (BED, etc.)

```
> gemini query -q [QUERY] my.db
```

Population genetics



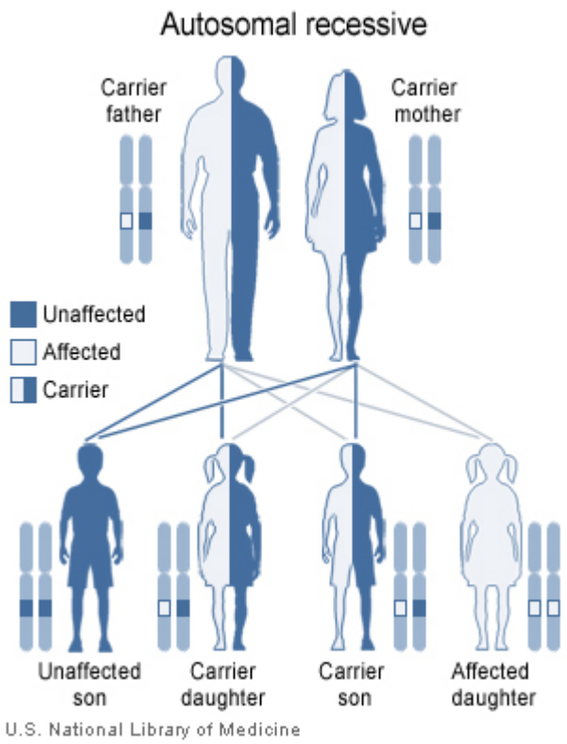
Multidimensional scaling,
Kinship coefficients
PCA

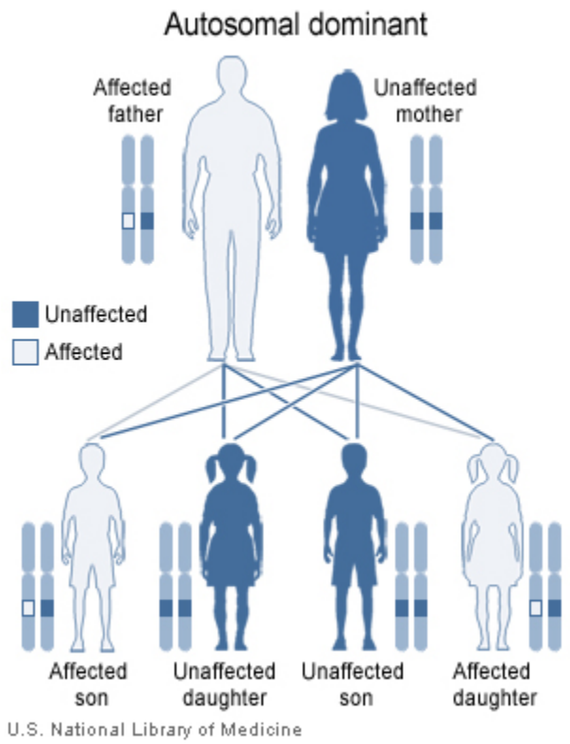
Variant QC and profiling

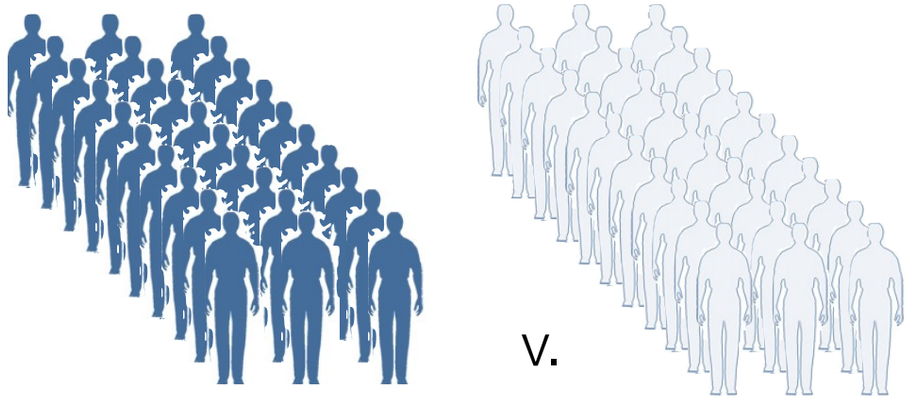
“windowing” tools

```
> gemini stats --sfs my.db
> gemini stats --mds my.db
> gemini stats --tstv my.db
> gemini stats --vars-by-sample my.db
> gemini stats --gts-by-sample my.db
```

Disease genetics







Cases
(have disease)

v.

Controls
(no disease)

Complex diseases
(multiple genes contribute to risk)

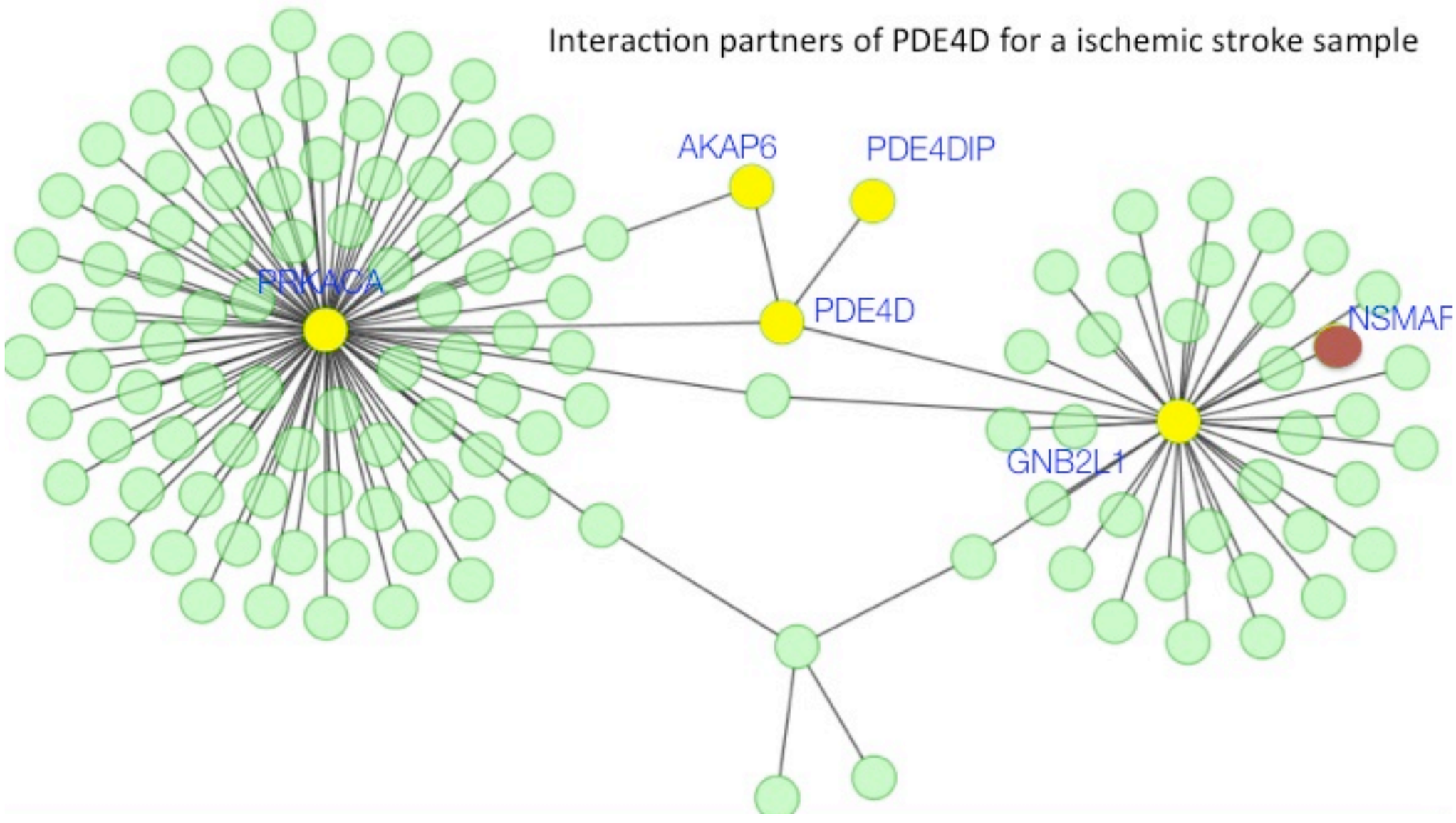
Tumor / Normal comparisons

Trio sequencing
(pilot study at UVA for undiagnosed dev. disorders)

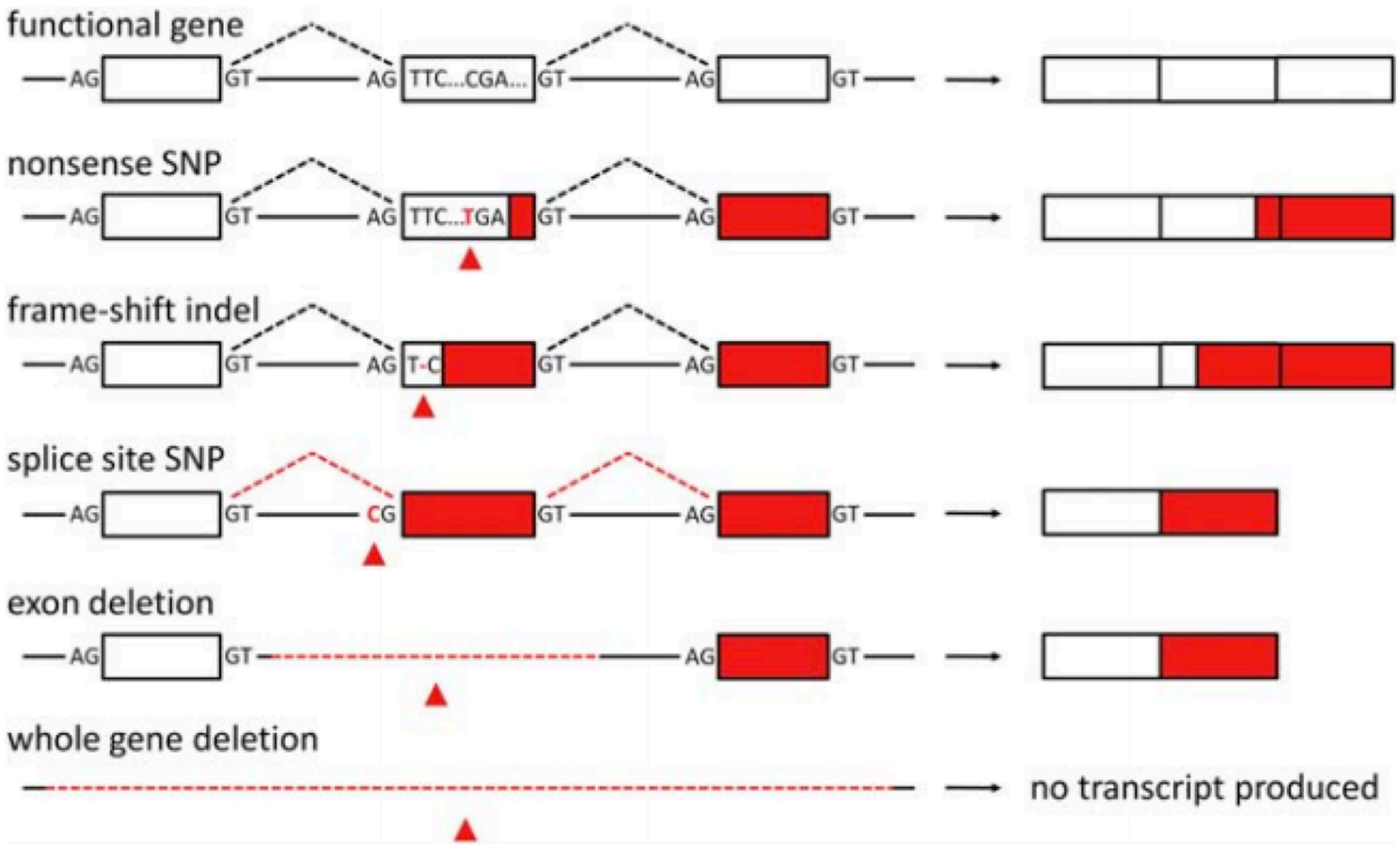
Pathway and interaction analysis

pathways

protein interactions



Loss-of-function (LoF) variants



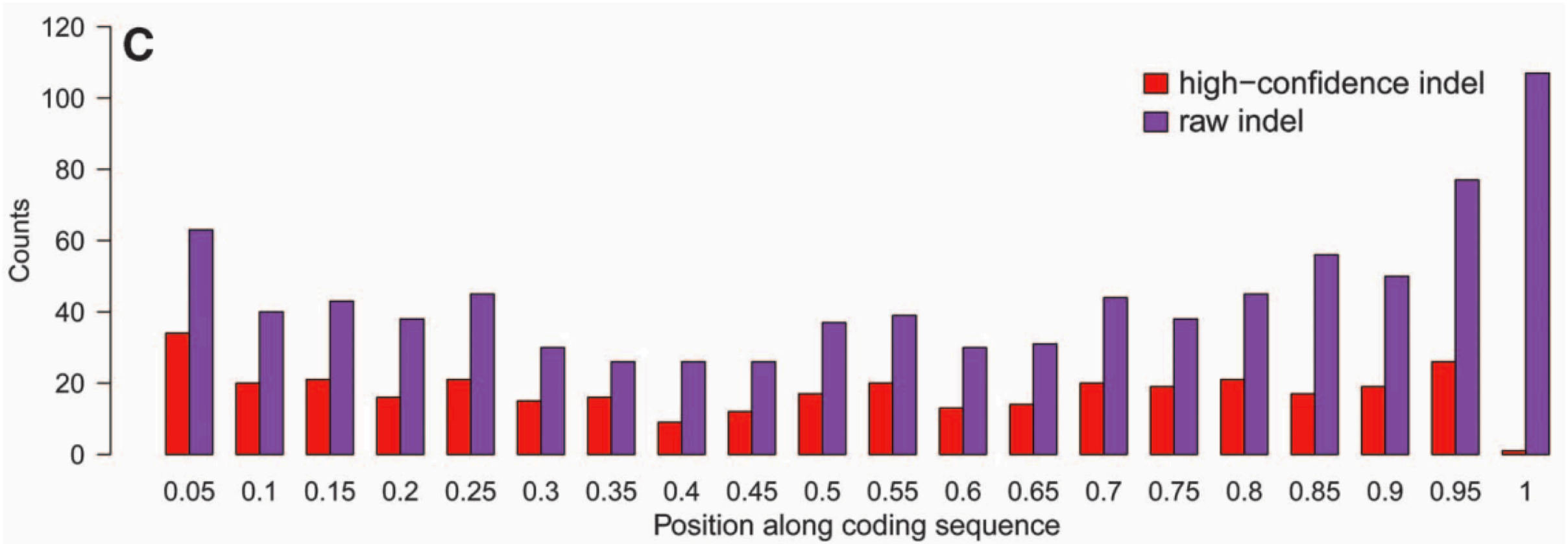
Loss-of-function variants in the genomes of healthy humans

Daniel G. MacArthur* and Chris Tyler-Smith

Human Molecular Genetics, 2010, Vol. 19, Review Issue 2 **R125–R130**
doi:10.1093/hmg/ddq365
Advance Access published on August 30, 2010

Loss-of-function (LoF) variants

Not all LoF variants are created equal.



1. Consider the position of the LoF variant in the polypeptide.
2. Test whether a second frameshift restores reading frame.

A Systematic Survey of Loss-of-Function Variants in Human Protein-Coding Genes
Daniel G. MacArthur *et al.*
Science **335**, 823 (2012);
DOI: 10.1126/science.1215040

Burden tests for rare variants

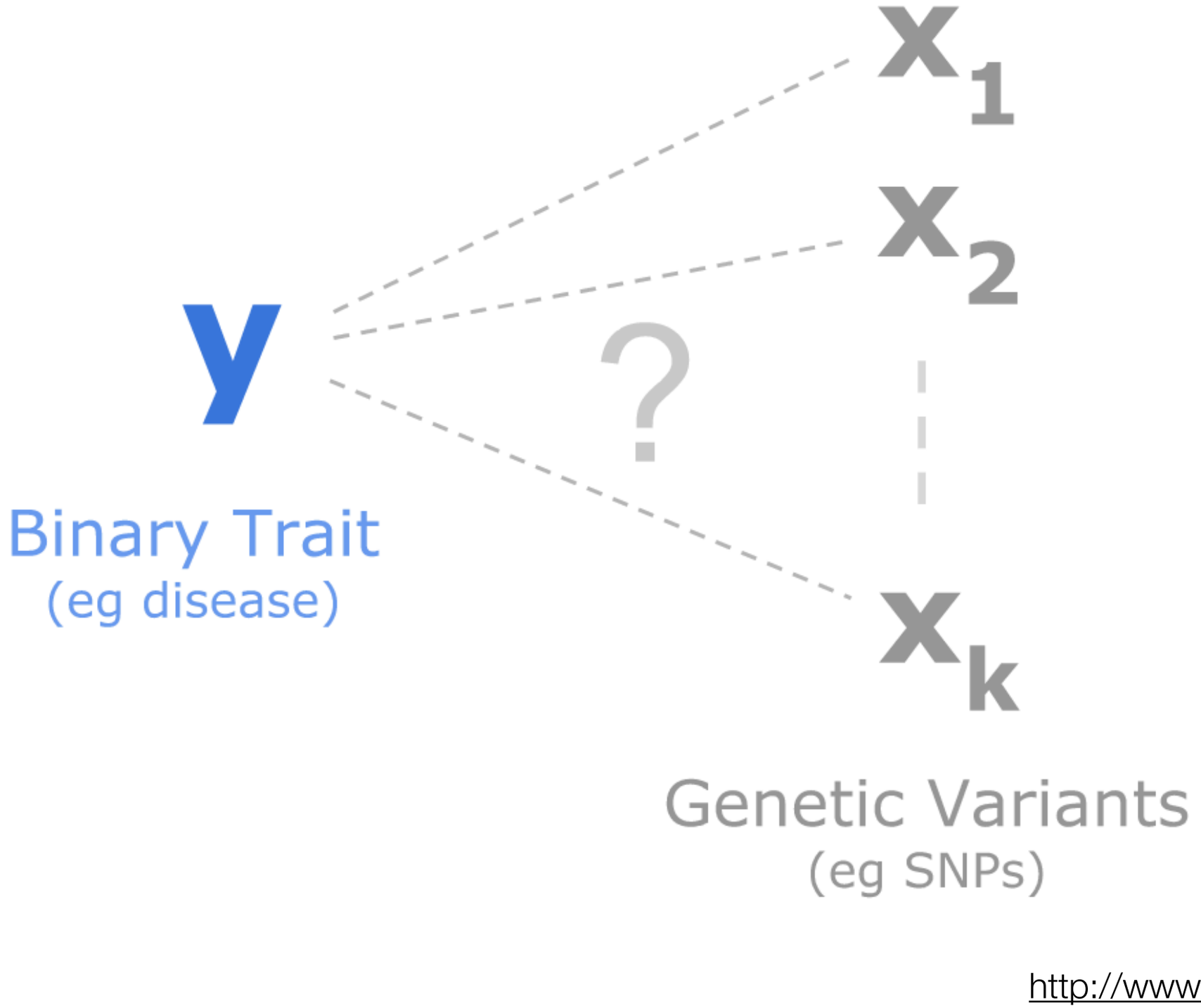
- GWAS based on the notion that common variants underly common disease.
- There are many, many more rare variants than common
- Standard, GWAS-like single-locus tests will be underpowered unless there is a huge effect size.
- Burden tests and Collapsing methods.
  - C-alpha, SKAT, KBAT, Morris-Zeggini, Hotelling's T, ...
  - Active area of research. Gemini is a common framework.

Burden tests for rare variants

	binary trait	genetic variants				
cases	1	0	1	0		0
	1	1	1	2		1
	1	1	1	0	----	0
	1	1	1	0		0
	1	0	1	0		0
controls	0	1	0	1		0
	0	1	1	0		2
	0	1	1	0	----	0
	0	2	1	0		0
	0	0	1	0		0
	y	x_1	x_2	x_3		x_k

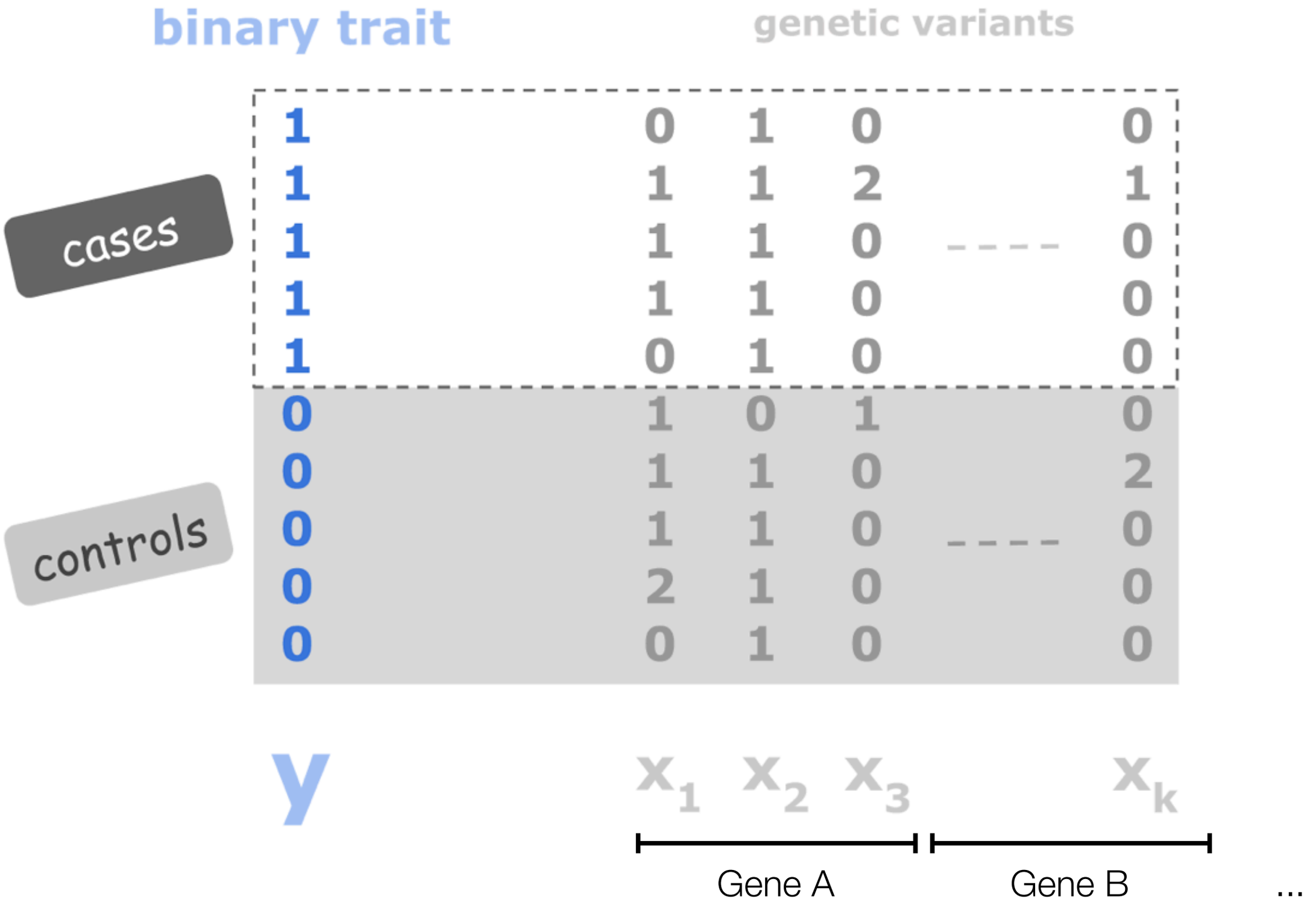
http://www.gastonsanchez.com

Burden tests for rare variants



http://www.gastonsanchez.com

Burden tests for rare variants



~Simple and extensible.

Framework for new tool development

```python
# gemini imports
import gemini_utils as util

def my_tool(c, args):
    """
    Execute a query against the gemini database and
    conduct a custom analysis on the results
    """
    # build and execute the relevant query against the da
    query = "SELECT * FROM variants \
             WHERE is_coding = 1"
    c.execute(query)

    # loop through the results.
    for row in c:
        gt_types  = np.array(unpack(row['gt_types'])))
        gt_phases = np.array(unpack(row['gt_phases'])))
        gt_bases  = np.array(unpack(row['gts'])))

        # MAGIC HAPPENS HERE

def run(parser, args):
    if os.path.exists(args.db):
        conn = sqlite3.connect(args.db)
        conn.isolation_level = None
        conn.row_factory = sqlite3.Row
        c = conn.cursor()
```

Development in Python with C and C++ code for heavy lifting.

1. Issue a query against DB
2. Iterate through results
3. Work with genotypes
4. Your genius.

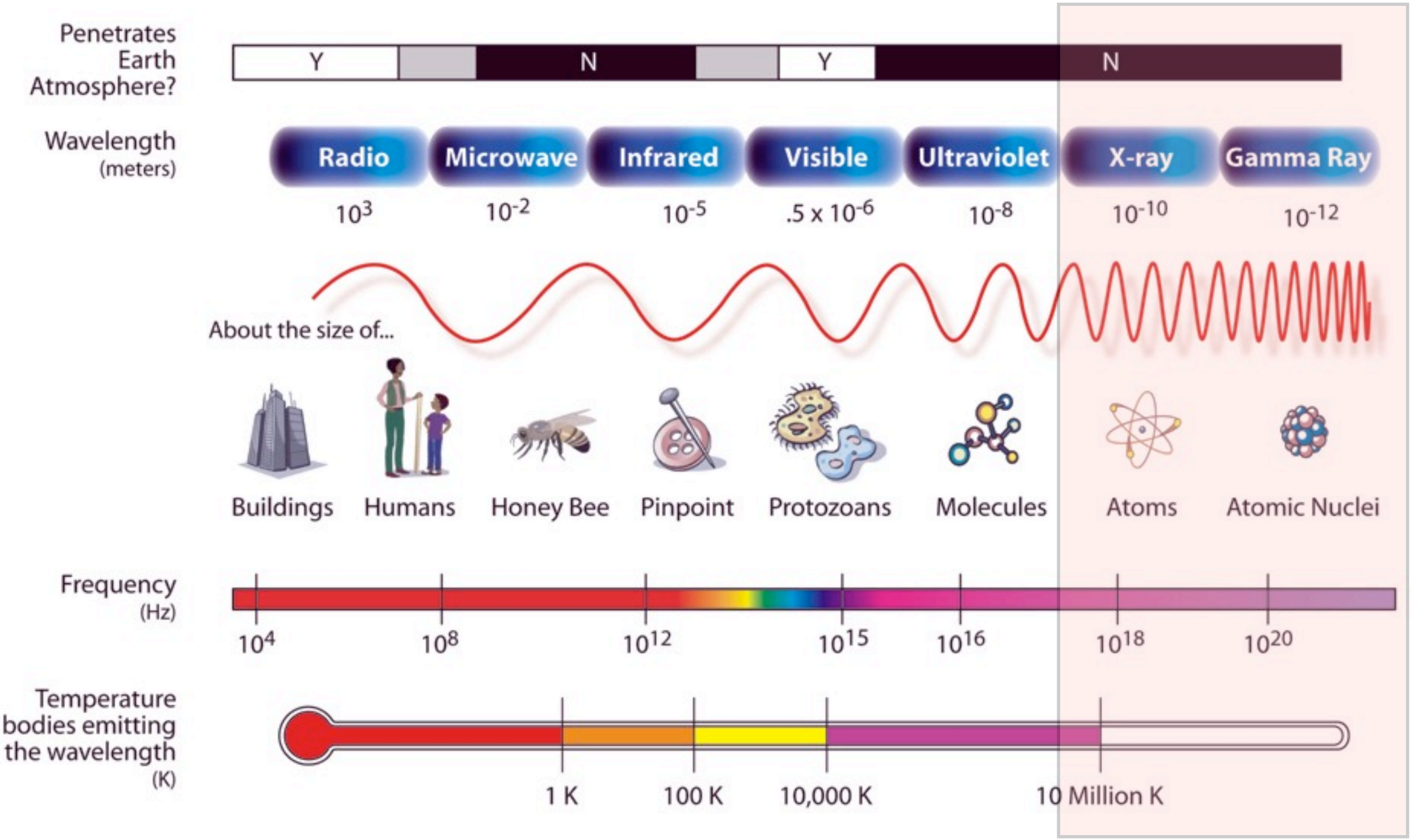
```
> gemini my_tool -arg1 -arg2 my.db
```

What is the genetic basis of extreme, unexplained sensitivity to ionizing radiation?

A collaboration with the wonderful Pat Concannon, UVA

The electromagnetic spectrum

Ionizing radiation (IR)



Beneficial applications of IR

1. Cancer therapy
2. Medical imaging
3. Food sterilization
4. Smoke detectors
5. QC for industrial production processes

Cellular consequences of IR

Direct effects

1. Causes direct damage to DNA and proteins in cell
2. More likely when the beam of charged particles consists of alpha particles, protons and electrons

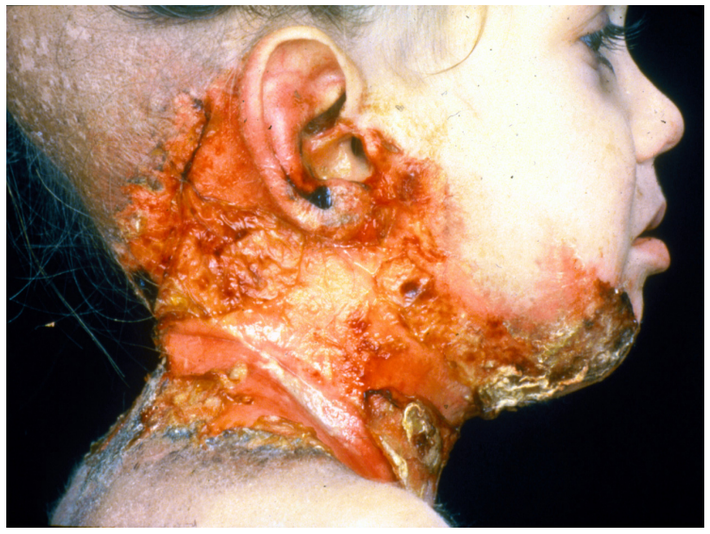
Indirect effects

1. Causes damage by interacting with the cellular medium producing free radicals which, in turn, can damage DNA
2. Typical effect of X-rays or gamma rays

Damage to DNA can include base loss or modification, single strand gaps or double-strand breaks (DSBs)

1. Often, damage is complex with multiple lesions occurring
2. Of these types of damage, the most dangerous is DSBs

Impact of standard radiation therapy in an undiagnosed ataxia-telangiectasia (A-T) patient



Background

Several rare Mendelian disorders have been described sharing the phenotype of radiation hypersensitivity with a constellation of other phenotypes including increased cancer incidence, immunodeficiency, neurologic defects, and developmental delay.

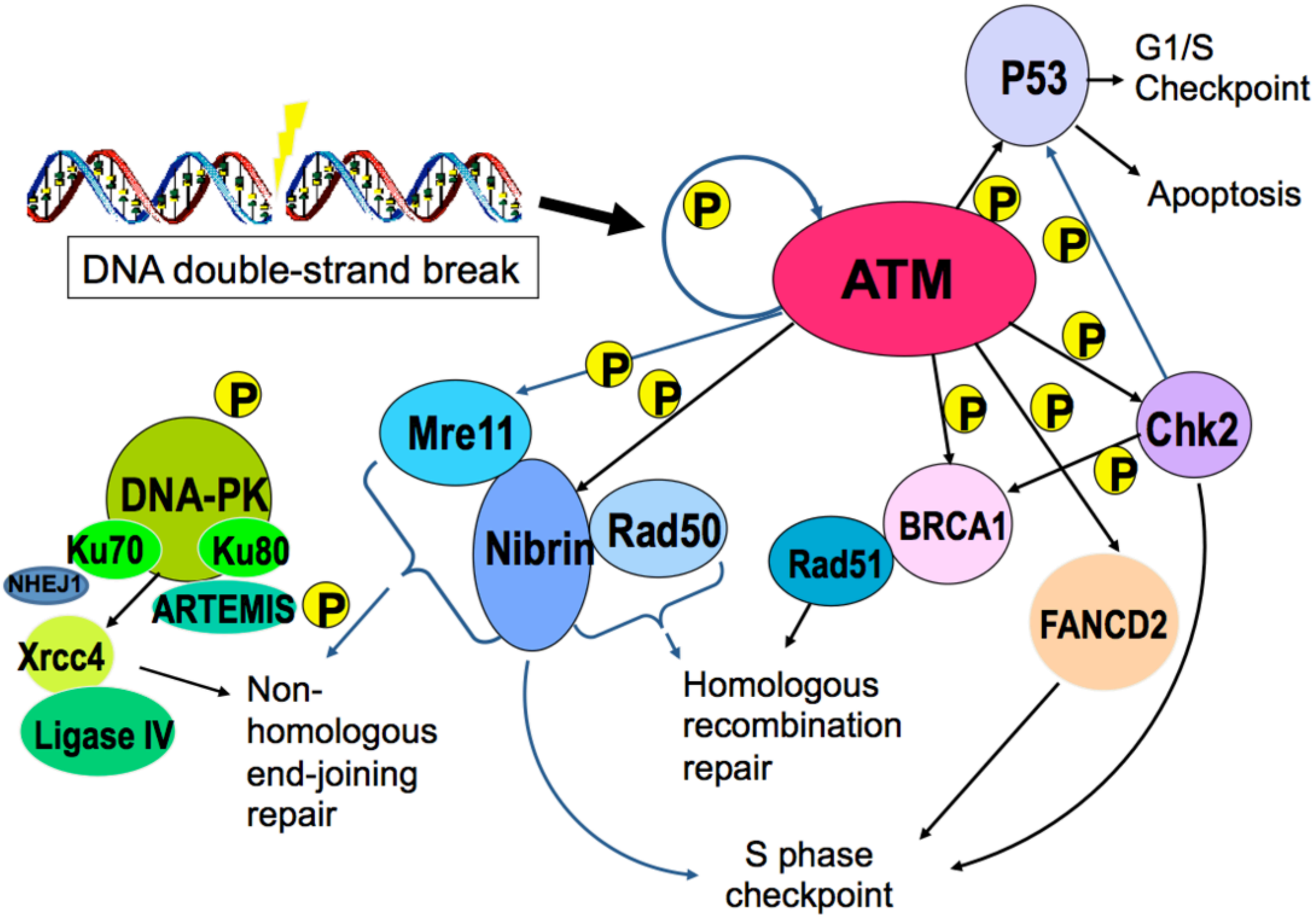
Studies of these rare, autosomal recessive genetic disorders has **identified key molecules in the human DNA damage response pathways**.

- Studies of Ataxia-Telangiectasia (A-T) identified ATM
- Studies of Nijmegen Breakage Syndrome (NBS), identified NBN as part of the MRE11-RAD50-NBN complex

Known radiosensitivity disorders

	A-T	NBS	ATLD	LIG4S
Incidence	$1/10^5$	>100 cases	40 cases	18 cases
Ataxia	+		+	
Telangiectasia	+			
Radiation sensitivity	+	+	+	+
Immunodeficiency	+	+	+	pancytopenia
Chromosomal instability	+ (7,14)	+ (7,14)	+ (7,14)	+
Cell cycle defects	+	+	+	
Lymphoid cancers	+	+	ND	+
Microcephaly		+		+
Growth retardation		+		+
Gene mutated:	ATM	NBN	MRE11	LIG4

ATM regulates the mammalian cellular response to DNA DSBs



RNF168 ubiquitin ligase: precedent for finding new ***rads*** genes among ***rads*** cell lines

- 2009: Discovery of radiosensitivity disorder RIDDLE syndrome caused by mutations in ubiquitin ligase RNF168
  - *Stewart et al. Cell 136:420*
- RS66 found to have mutations in RNF168, leading to lack of the protein.
  - *Devgan et al. 2011*

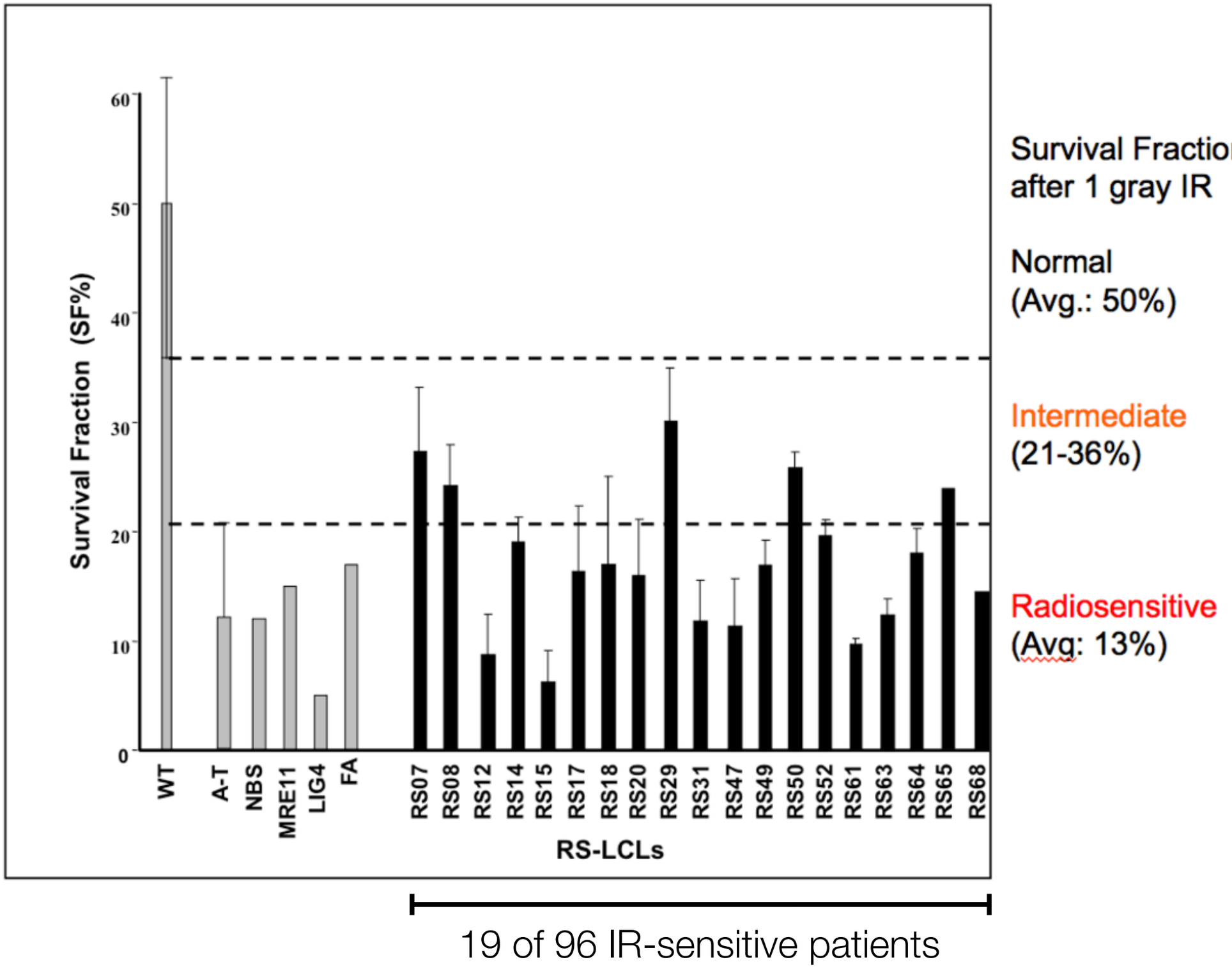
96 patients with ***unexplained*** radiosensitivity

- Collaboration w Pat Concannon & Richard Gatti, UCLA
- All patients have been previously screened for dysfunction in known radiosensitivity genes (e.g., ATM and NBN).
- Thus, opportunity to **discover new genes underlying response to DNA damage.**
- We hypothesize that each patient has a single gene disorder - yet the phenotype is only observed when they receive IR.
- We have cell lines - functional assays to confirm variants via complementation assays

Rationale for exome sequencing

- The RS cell lines derive from patients who display multiple clinical phenotypes known to be associated with A-T and NBS, autosomal recessive, monogenic, DNA damage response disorders
- Prior candidate gene approaches yielded biallelic mutations in single genes LIG4 or RNF168
- >84% of disease-causing mutations in ATM are in the coding regions
- The majority of disease-causing mutations in ATM and NBS were truncating mutations, readily recognizable (though not totally solved)

Survival after Ionizing Radiation

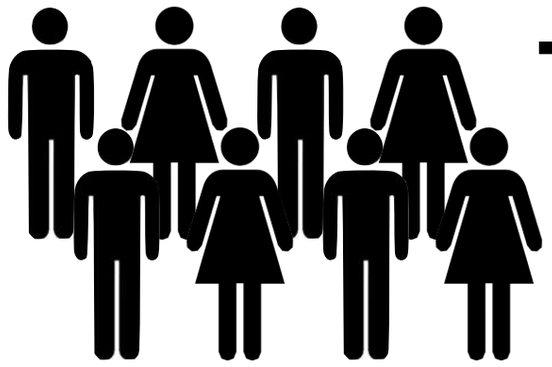


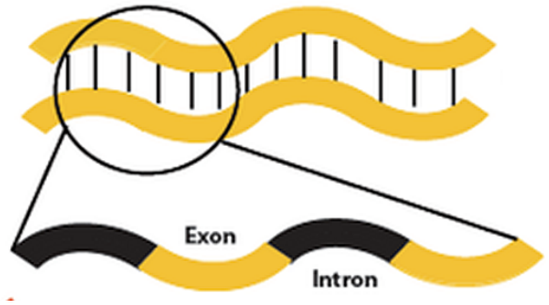
The pipeline

Project Title: *Identification of radiation sensitivity alleles by whole exome sequencing.*
PI: Pat Concannon
Co-investigator: Aaron Quinlan
Source: NIH/NIEHS (R21 ES020521-01)

@HudsonAlpha

96 patients w/ unexplained radiosensitivity



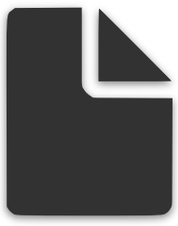


Nimblegen exome capture

Paired-end exome sequencing on Illumina HiSeq2

Raw Sequence files FASTQ; N = 96

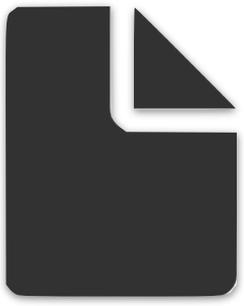
Align sequences to human genome with BWA



Aligned sequence files in BAM format N = 96

Duplicate removal
Alignment refinement
Recalibration
Picard, bedtools, samtools

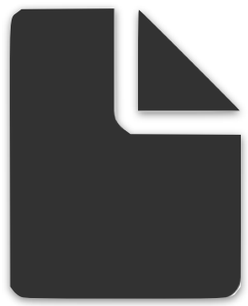
Variant detection among all 96 samples GATK



VCF file including all variants among all samples

The pipeline

Project Title: *Identification of radiation sensitivity alleles by whole exome sequencing.*
PI: Pat Concannon
Co-investigator: Aaron Quinlan
Source: NIH/NIEHS (R21 ES020521-01)



VCF file
including
all variants
among all
96 samples

Predict functional
impact of variants
on coding genes

Annotate with **gemini**.
**Explore variants
and screen for
candidates underlying
radiosensitivity.**

Preliminary variant filters

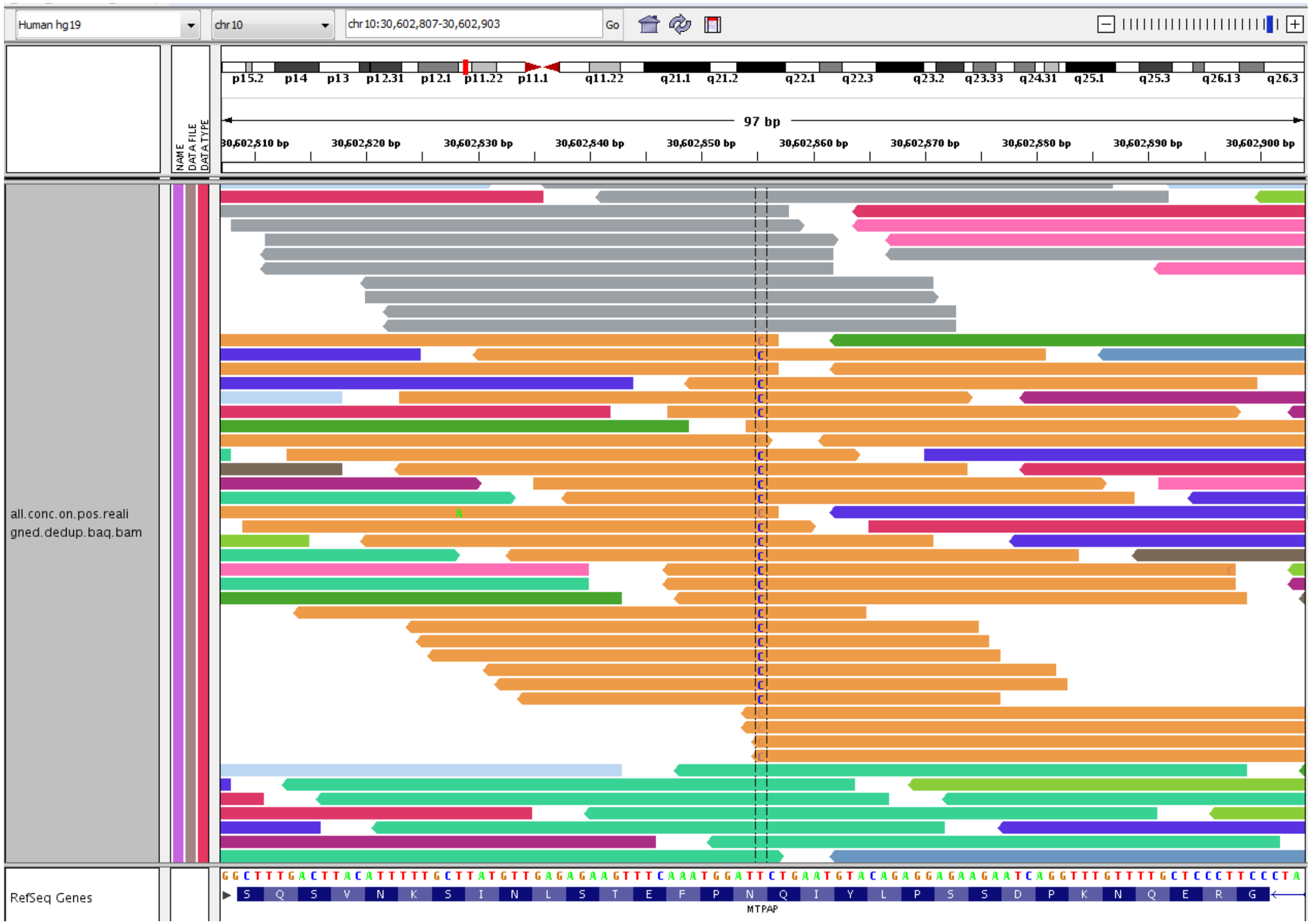
- We are looking for rare, highly penetrant alleles.

- **Exclude if:**
  - appears in dbSNP, except if has clin. sig.
  - in 1000 Genomes with MAF > 1%
  - in Exome Sequencing Project w/ MAF > 1%

- **Looking for:**
  - LoF variants fitting recessive model
  - LoF compound heterozygotes
  - Genes with 2 het LoF mutations
  - Genes known to be phosphorylation targets of ATM/ATR
  - Genes with functions consistent with a role in DNA damage response.

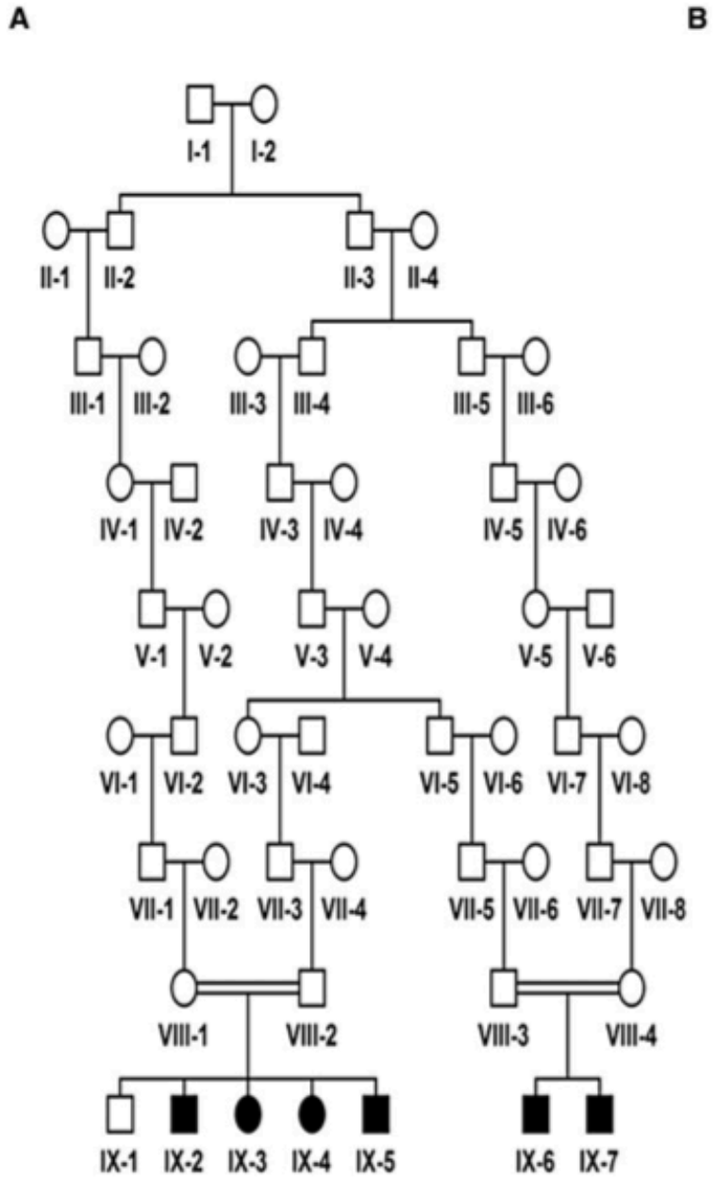
A quick, unlikely success

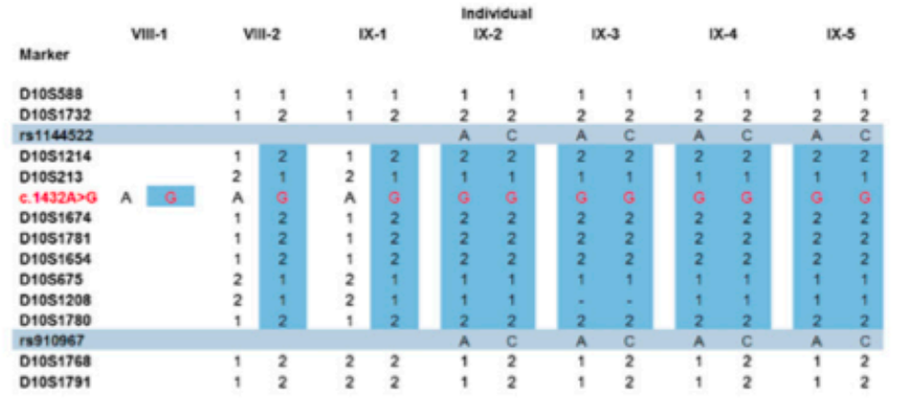
Sample A21: chr10, MTPAP, exon9, N478D homozygote

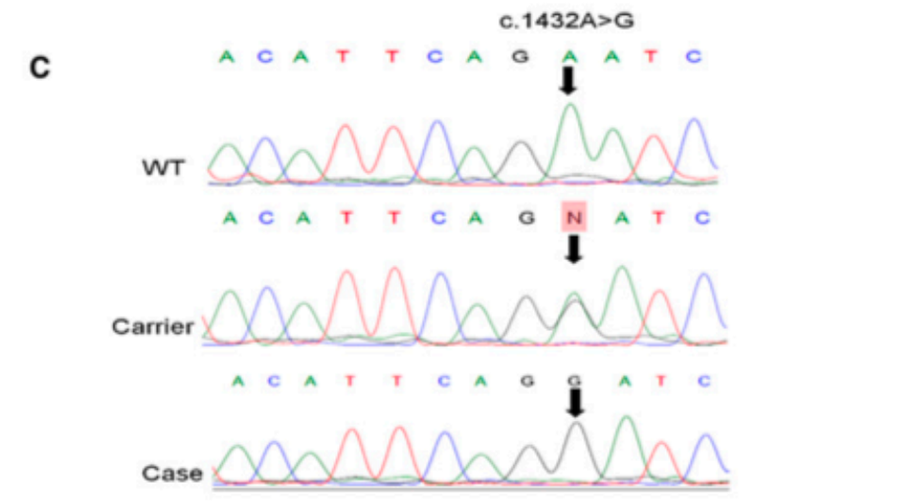


Defective Mitochondrial mRNA Maturation Is Associated with Spastic Ataxia

Andrew H. Crosby,[1,7,*] Heema Patel,[1,7] Barry A. Chioza,[1] Christos Proukakis,[1,2] Kay Gurtz,[3] Michael A. Patton,[1] Reza Sharifi,[1] Gaurav Harlalka,[1] Michael A. Simpson,[1] Katherine Dick,[1] Johanna A. Reed,[1] Ali Al-Memar,[4] Zofia M.A. Chrzanowska-Lightowlers,[5] Harold E. Cross,[6] and Robert N. Lightowlers[5]







Role for MTPAP in stability of miRNA

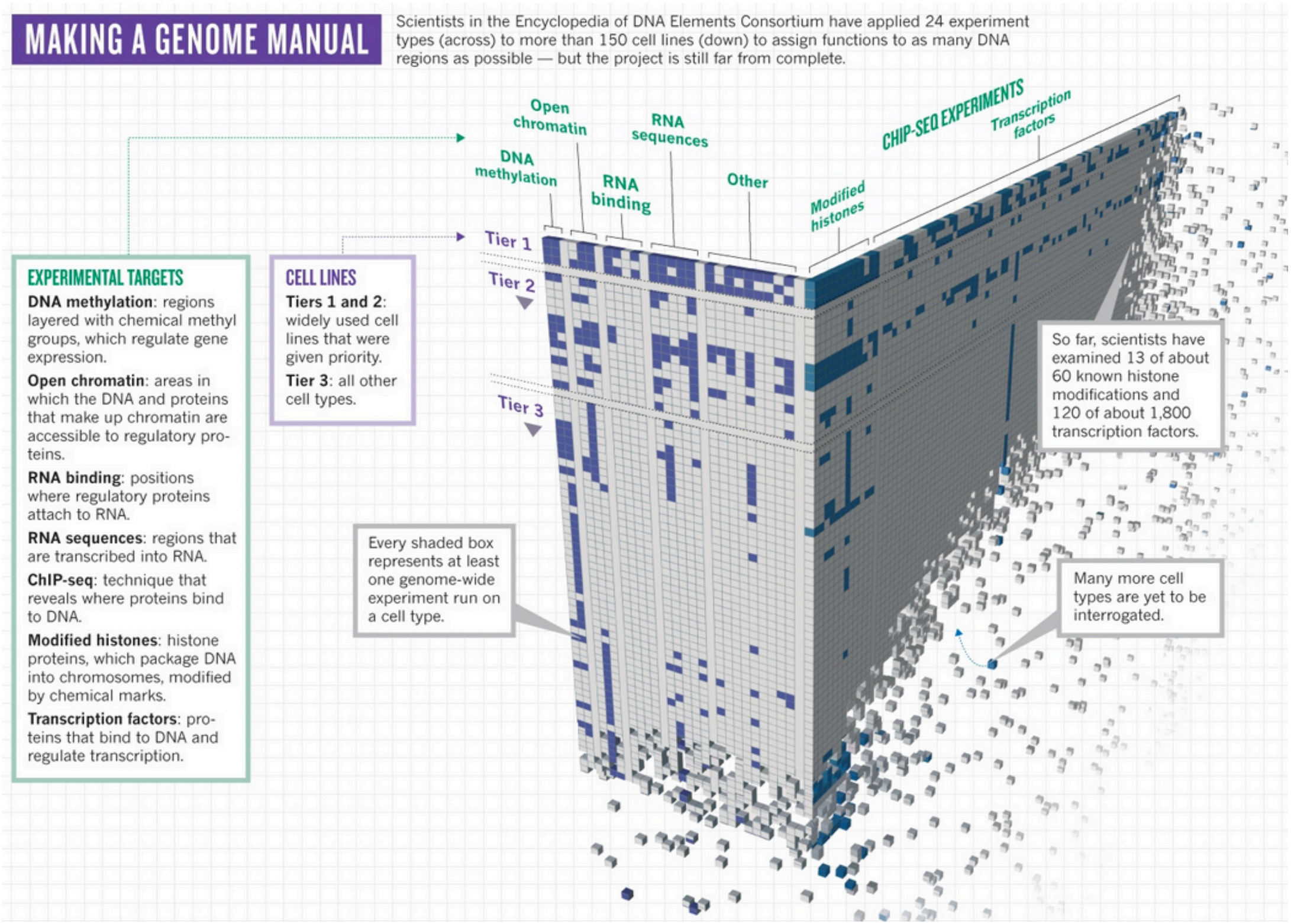
RS cell lines tested with miRNA expression array: MTPAP mutant cell line had global decrease in miRNA compared to others—Hailang Hu, Gatti lab

Further evidence: MTPAP adds nucleotides to the 3' end of miRNA, increasing their stability
Wyman et al. Genome Res. 2011 21: 1450-1461

Current: complementation assays in patient's cell lines

coffee.

Making sense of complex datasets



http://www.nature.com/news/encode-the-human-encyclopaedia-1.11312

This is a hard and important problem.

- Understand the function (or lack) of every base pair in different cell types and contexts.
- **Challenges** (among many):
  - Basic exploratory data analysis: slicing and dicing.
  - Testing for significance
  - Visualization: unbiased exploration; let the data tell its story.
- Necessary for new discovery and understanding of genome biology

bedtools: *genome arithmetic*

Neil Kindlon, M.S.

Staff Scientist and Software Engineer

nek3d @ virginia.edu

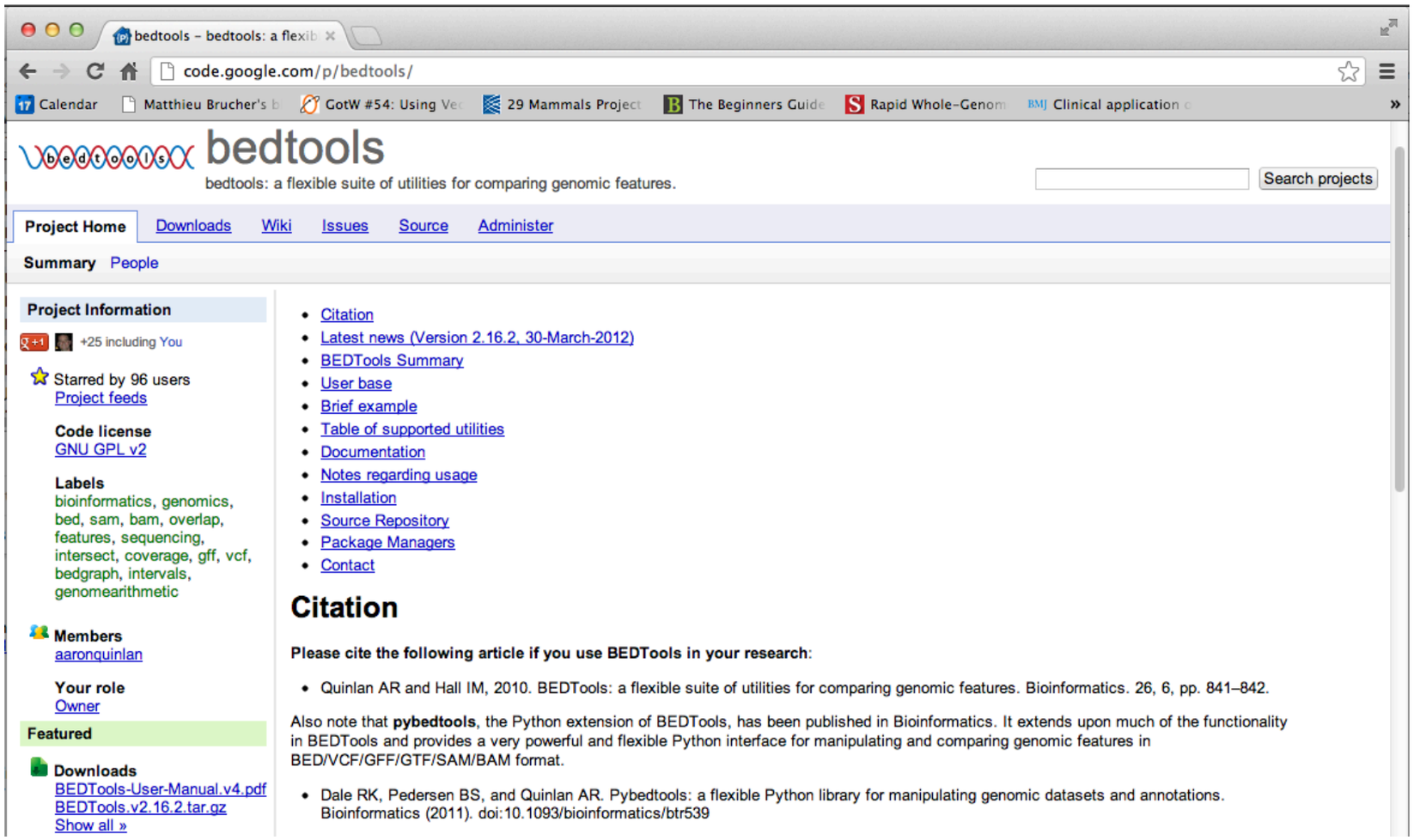
Research Projects and Interests: Soft
Structural variation discovery and interpret

Ryan Layer

Graduate student

rl6sf @ virginia.edu

Research Interests: Scalable algorithm analysis; genome data mining and analysi interpretation.



Software http://code.google.com/p/bedtools/

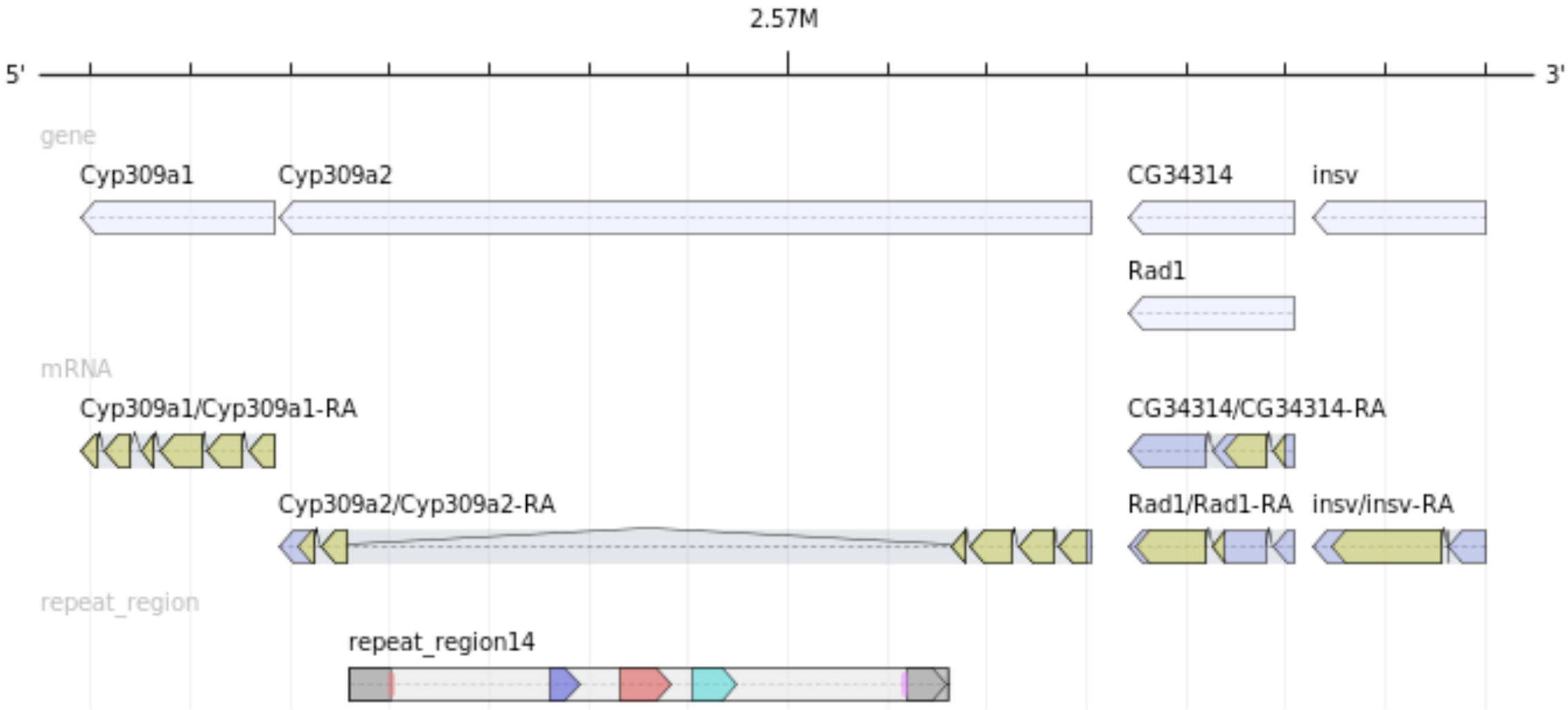
Docs https://bedtools.googlecode.com/files/BEDTools-User-Manual.v4.pdf

Tools for comparing and exploring genome “intervals”

What is a genome “interval”?

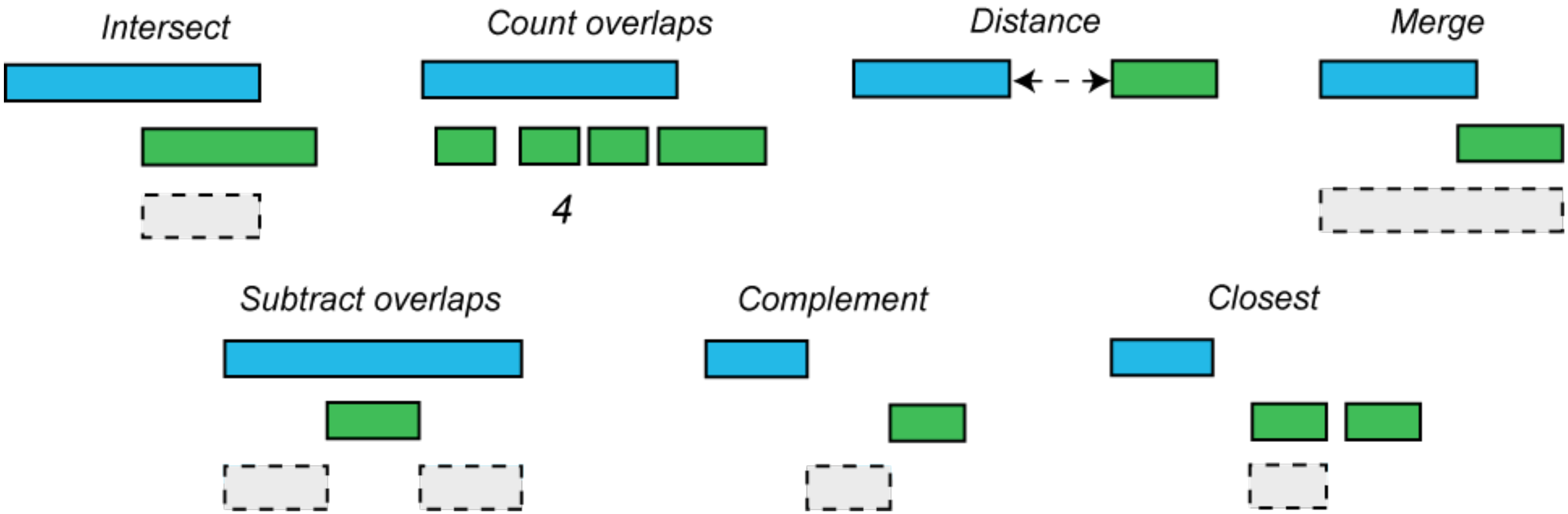
- Genes: exons, introns, UTRs, promoters
- Conservation
- Genetic variation
- Transposons
- Origins of replication
- TF binding sites
- CpG islands
- Segmental duplications
- Sequence alignments
- Chromatin annotations
- Gene expression data
- …

Genome intervals



What is genome arithmetic?

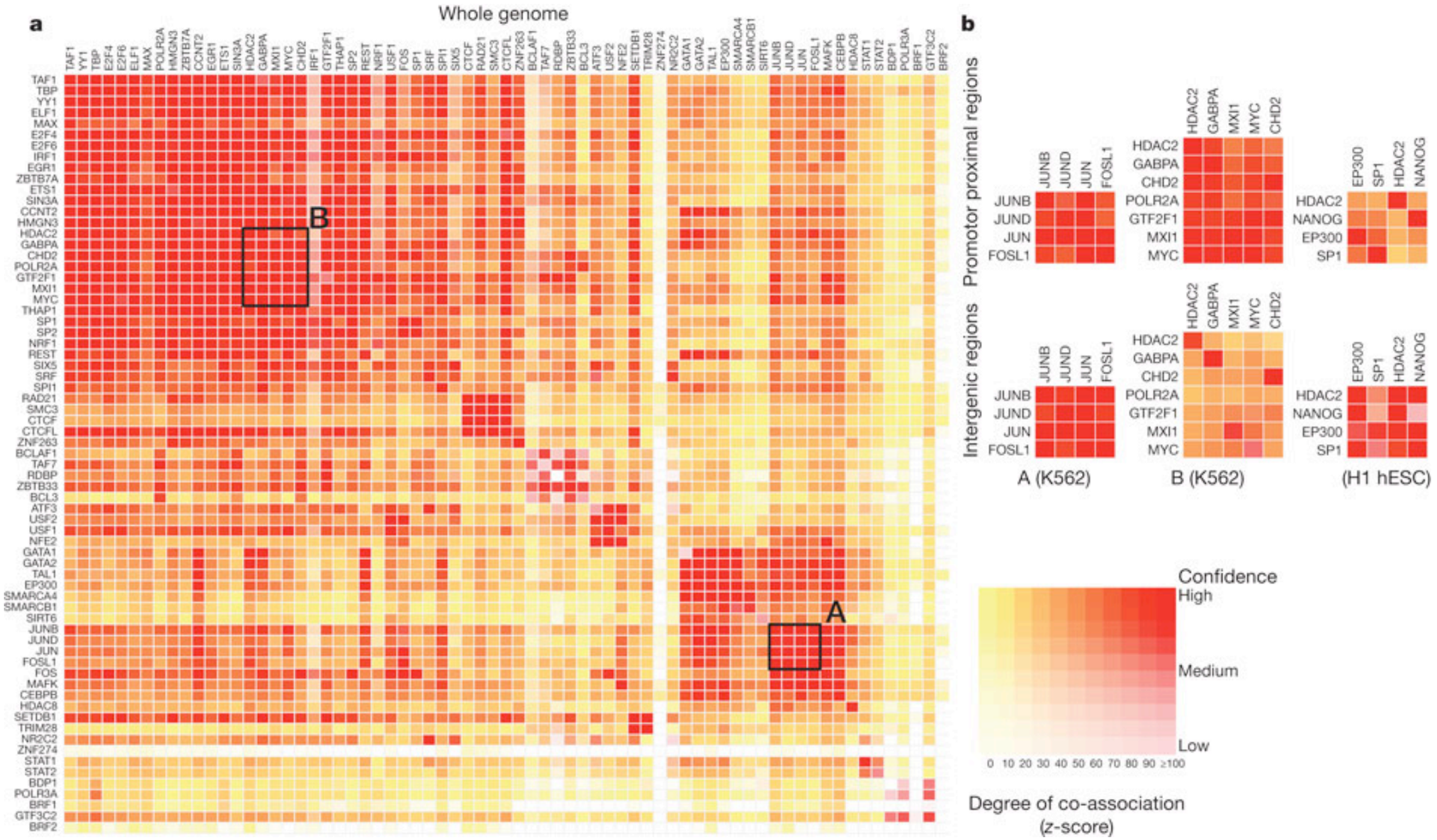
set theory on genome intervals



Answerable questions

- Closest gene to a ChIP-seq peak.
- Is my latest discovery novel w.r.t. other datasets?
- Is there strand bias in my data?
- How many genes does this mutation affect?
- Where did I fail to collect sequence coverage?
- Is my favorite feature significantly correlated with some other feature?

How do we measure significance?



Co-association between transcription factors.

Nature **489**, 57–74 (06 September 2012) doi:10.1038/nature11247

Monte-Carlo simulation

Are the observed feature overlaps more common than what you would expect by chance?

1. **Observed.** Count the number of overlaps between the two sets of features (e.g. ChIP peaks for TF 1 v. TF 2)
2. **Expected.**
    2.1. Randomly reassign the features in each set to new genomic locations.
    2.2. Count the number of overlaps.
    2.3. Repeat 1000 or more times. **SLOW!**
3. **Return either:**
    3.1. **P-value:** how many times were the shuffled overlaps > observed?
        3.1.1. e.g., if answer is 3 and there were 1000 simulations, P = 3e-3
    3.2. **Enrichment score:** log2(observed / median expected)

How do we speed this up?

We invent a new algorithm, of course!

PREPRINT

Vol. 00 no. 00 2012
Pages 1–8

Binary Interval Search (BITS):
A Scalable Algorithm for Counting Interval Intersections

Ryan M. Layer[1], Kevin Skadron[1], Gabriel Robins[1], Ira M. Hall[2], and Aaron R. Quinlan[3*]

[1]Department of Computer Science, University of Virginia, Charlottesville, VA
[2]Department of Biochemistry and Molecular Genetics, University of Virginia, Charlottesville, VA
[3]Department of Public Health Sciences and Center for Public Health Genomics, University of Virginia, Charlottesville, VA

Ryan Layer

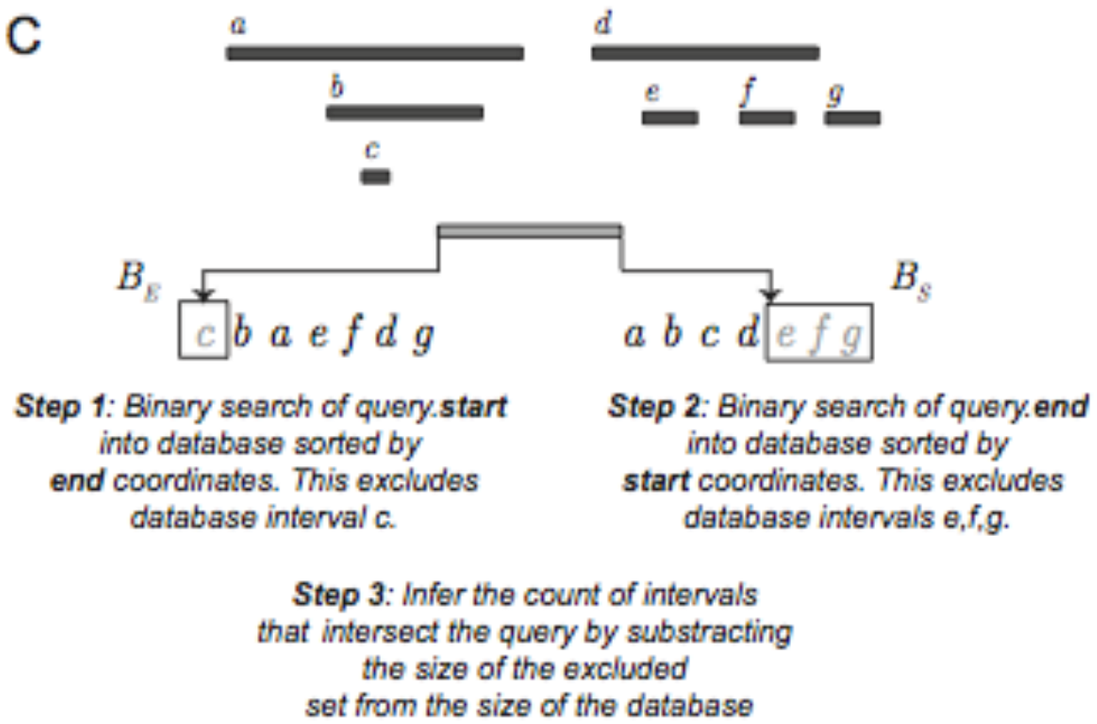
Graduate student

rl6sf @ virginia.edu

Research Interests: Scalable algorithm analysis; genome data mining and analysi interpretation.

https://github.com/arq5x/bits

Binary InTerval Search BITS



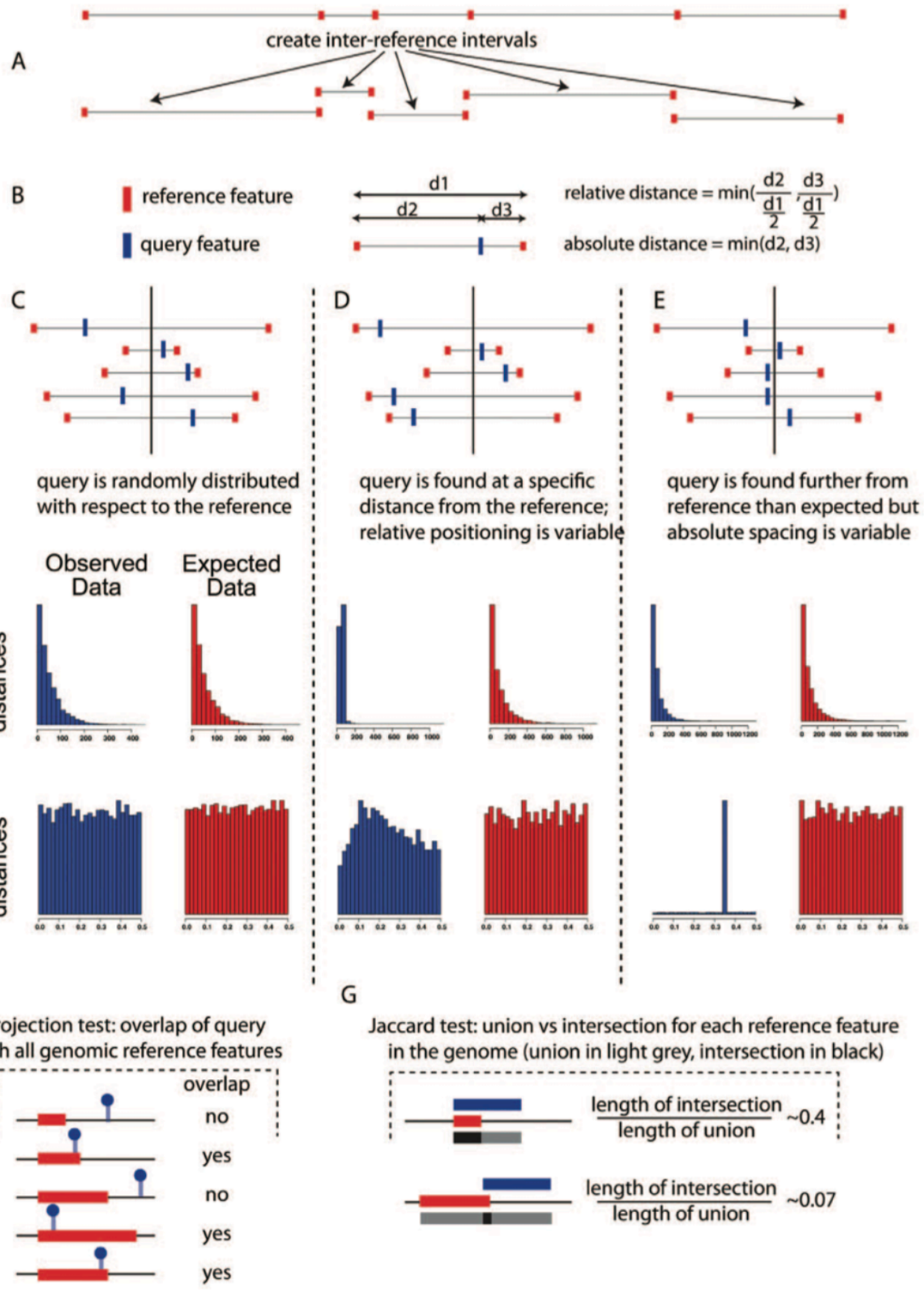
Algorithm 1: Single interval intersection counter

Input: Sorted interval starts and ends B_S and B_E, query interval a
Output: Number of intervals c intersecting a

Function ICOUNT(B_S, B_E, a) **begin**
    $first \leftarrow$ BINARYSEARCH($B_S, a.end$)
    $last \leftarrow$ BINARYSEARCH($B_E, a.start$)
    $c \leftarrow first - last$ /* $= |B| - (last + (|B| - first))$ */
    return c

- Novel algorithm for detecting genomic interval intersections.
- Clever aspect: unlike any other algorithm, it can deduce the **count** of overlaps without having to **enumerate** each individual intersection.
- Uses two binary searches. Very, very fast.
- If you haven't heard, speed is good.

Other approaches



Exploring Massive, Genome Scale Datasets with the GenometriCorr Package

Alexander Favorov[1,2,3]*, Loris Mularoni[1]¤a, Leslie M. Cope[1], Yulia Medvedeva[2,3]¤b, Andrey A. Mironov[4,5], Vsevolod J. Makeev[2,3], Sarah J. Wheelan[1]*

1 Department of Oncology, Division of Biostatistics and Bioinformatics, Johns Hopkins University School of Medicine, Baltimore, Maryland, United States of America, **2** Vavilov Institute of General Genetics, Russian Academy of Sciences, Moscow, Russia, **3** Research Institute of Genetics and Selection of Industrial Microorganisms, Moscow, Russia, **4** Department of Bioengineering and Bioinformatics, Moscow State University, Moscow, Russia, **5** Institute for Information Transmission Problems, Russian Academy of Sciences, Moscow, Russia

NON PARAMETRIC METHODS FOR GENOMIC INFERENCE

BY PETER J. BICKEL*,†, NATHAN BOLEY*,†, JAMES B. BROWN*,†, HAIYAN HUANG*,†, NANCY R. ZHANG*,‡

Statistics, University of California at Berkeley † *and Statistics, Stanford University*‡

Large-scale statistical analysis of data sets associated with genome sequences plays an important role in modern biology. A key component of such statistical analyses is the computation of p-values and confidence bounds for statistics defined on the genome. Currently such computation is commonly achieved through ad hoc simulation measures. The method of randomization, which is at the heart of these simulation procedures, can significantly affect the resulting statistical conclusions. Most simulation schemes introduce a variety of hidden assumptions regarding the nature of the randomness in the data, resulting in a failure to capture biologically meaningful relationships. To address the need for a method of assessing the significance of observations within large scale genomic studies, where there often exists a complex dependency structure between observations, we propose a unified solution built upon a data subsampling approach. We propose a piecewise stationary model for genome sequences and show that the subsampling approach gives correct answers under this model. We illustrate the method on three simulation studies and two real data examples.

Used by ENCODE project

So, what can we do?

Large, unbiased screens for novel genome biology

Scale
chr21:
200 kb
hg19
32,700,000
32,800,000
32,900,000
33,000,000
33,100,000
33,200,000
33,300,000
33,400,000
33,500,000
Uniqueness of 35bp Windows from ENCODE/OpenChrom(Duke)
UCSC Genes (RefSeq, UniProt, CCDS, Rfam, tRNAs & Comparative Genomics)
TIAM1
TIAM1
TIAM1
TIAM1
BC014150
BC041449
SOD1
HUNK
LINC00159
SCAF4
SCAF4
SCAF4
SCAF4
SCAF4
RefSeq Genes
RefSeq Genes
Human mRNAs from GenBank
Human mRNAs
Human ESTs That Have Been Spliced
Spliced ESTs
DNaseI/FAIRE/ChIP Synthesis from ENCODE/OpenChrom(Duke/UNC/UTA)
GM12878 Syn Pk
H1-hESC Syn Pk
K562 Syn Pk
100
H3K27Ac Mark (Often Found Near Active Regulatory Elements) on 7 cell lines from ENCODE
Layered H3K27Ac
0
Digital DNaseI Hypersensitivity Clusters from ENCODE
DNase Clusters
Transcription Factor ChIP-seq from ENCODE
Txn Factor ChIP
Open Chromatin by DNaseI HS from ENCODE/OpenChrom(Duke University)
GM12878 Pk
GM12878 DS
GM12878 OS
H1hESC Pk
H1hESC DS
H1hESC OS
K562 Pk
K562 DS
K562 OS
Open Chromatin by FAIRE from ENCODE/OpenChrom(UNC Chapel Hill)
GM12878 FAIRE Pk
GM12878 FAIRE DS
GM12878 FAIRE OS
H1-hESC FAIRE Pk
H1-hESC FAIRE DS
H1-hESC FAIRE OS
K562 FAIRE Pk
K562 FAIRE DS
K562 FAIRE OS
DNaseI Hypersensitivity by Digital DNaseI from ENCODE/University of Washington
GM12878 Ht 1
GM12878 Pk 1
GM12878 Sg 1
GM12878 Ht 2
GM12878 Pk 2
GM12878 Sg 2
H1hESC Ht 1
H1hESC Pk 1
H1hESC Sg 1
K562 Ht 1
K562 Pk 1
K562 Sg 1
K562 Ht 2
K562 Pk 2
K562 Sg 2
DNaseI Digital Genomic Footprinting from ENCODE/University of Washington
K562 Sig
HepG2 Sig
HUVEC Sig

HeLa-S3 G2 1

HeLa-S3 Repli-seq Wavelet-smoothed Signal Rep 1 from ENCODE/UW

Replication Timing by Repli-seq from ENCODE/University of Washington

HepG2 G1b 1
HepG2 S2 1
HepG2 S3 1
HepG2 S4 1
HepG2 G2 1

HepG2 Repli-seq Wavelet-smoothed Signal Rep 1 from ENCODE/UW

Replication Timing by Repli-seq from ENCODE/University of Washington

HUVEC G1b 1
HUVEC S2 1
HUVEC S3 1
HUVEC S4 1
HUVEC G2 1

HUVEC Repli-seq Wavelet-smoothed Signal Rep 1 from ENCODE/UW

Placental Mammal Basewise Conservation by PhyloP

Mammal Cons

4 _
0 -
-4 _

Multiz Alignments of 46 Vertebrates

Rhesus
Mouse
Dog
Elephant
Opossum
Chicken
X_tropicalis
Zebrafish

GERP scores for mammalian alignments

6.17 _
0 -
-12.1 _

GERP

Simple Nucleotide Polymorphisms (dbSNP 135) Found in >= 1% of Samples

Common SNPs(135)

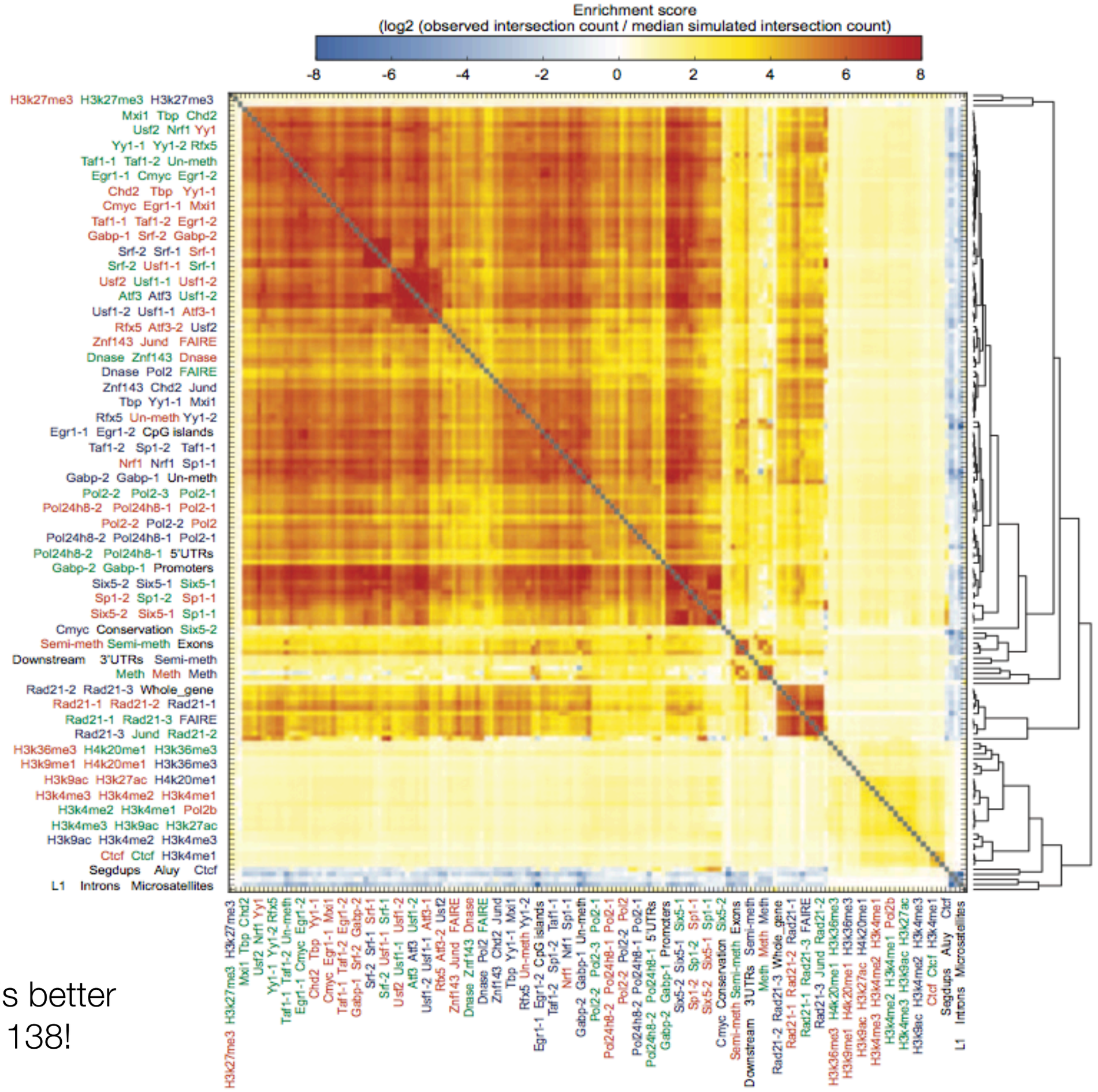
Simple Nucleotide Polymorphisms (dbSNP build 131)

Duplications of >1000 Bases of Non-RepeatMasked Sequence

Repeating Elements by RepeatMasker

RepeatMasker

Simple Tandem Repeats by TRF



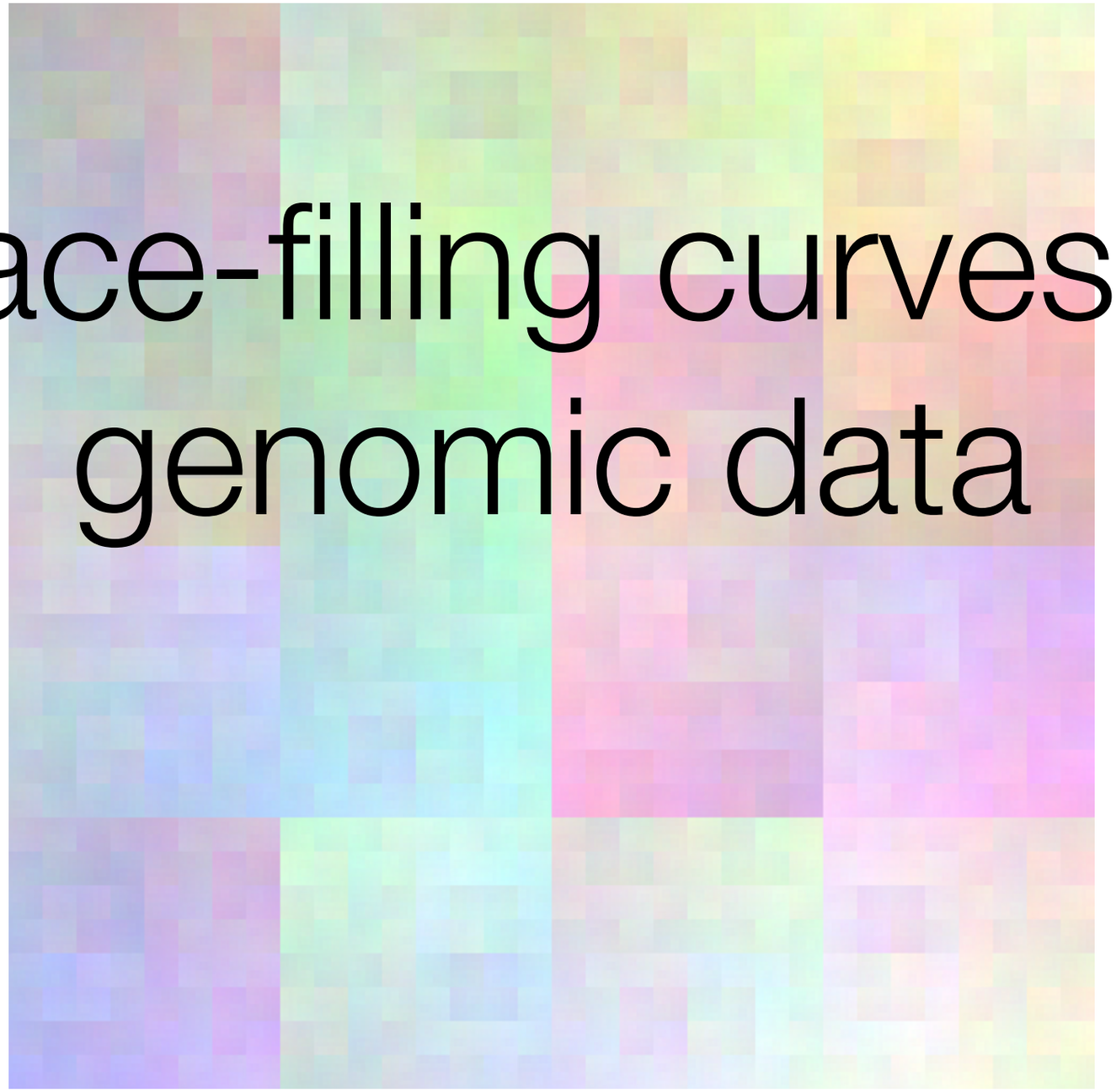
6 days is better than 138!

How do we understand the details of these signals?

Problem: *The genome is big. Patterns are hard to detect and quantify.*

We need better tools for data visualization.

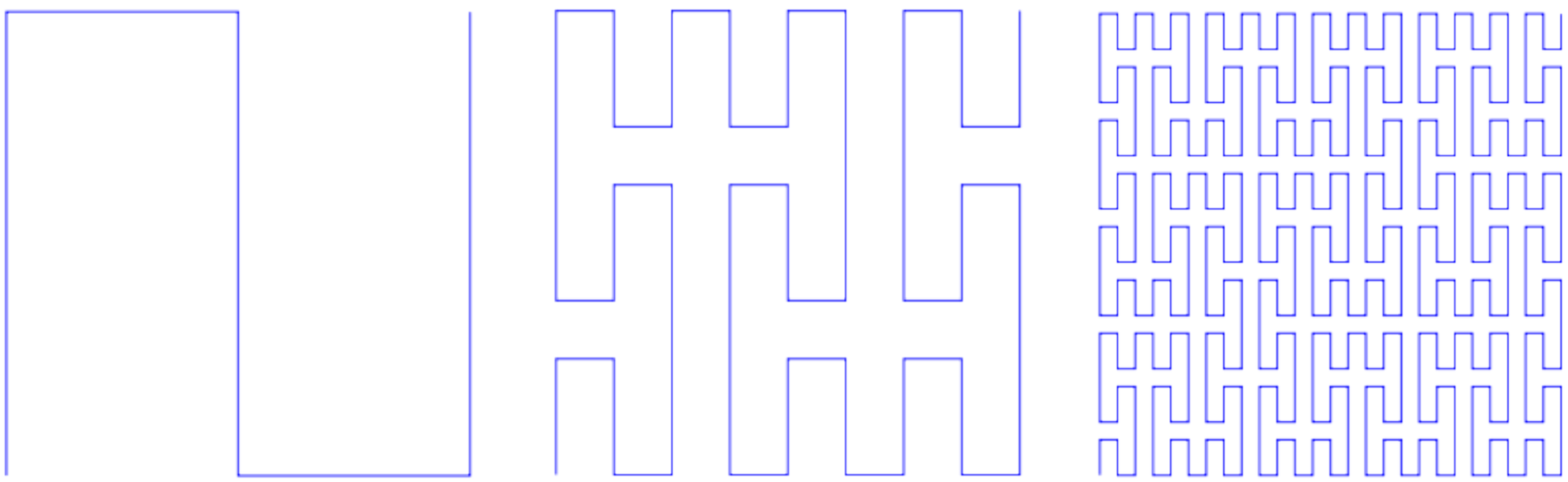
Space-filling curves for genomic data



What is a space-filling curve?

Fractal algorithm to convert a 1D line to a ***curve*** that fills a 2 (or more) dimension space.

1D



2D

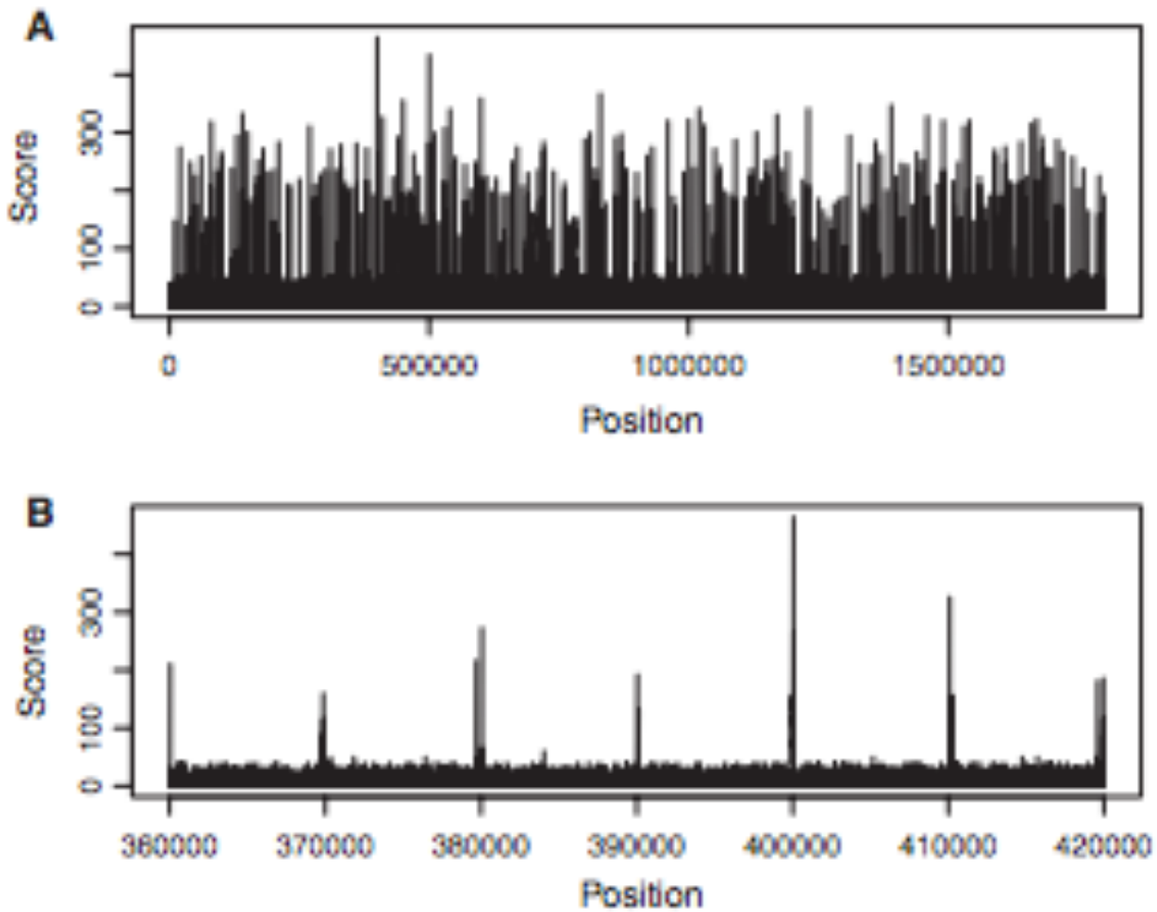
first iteration

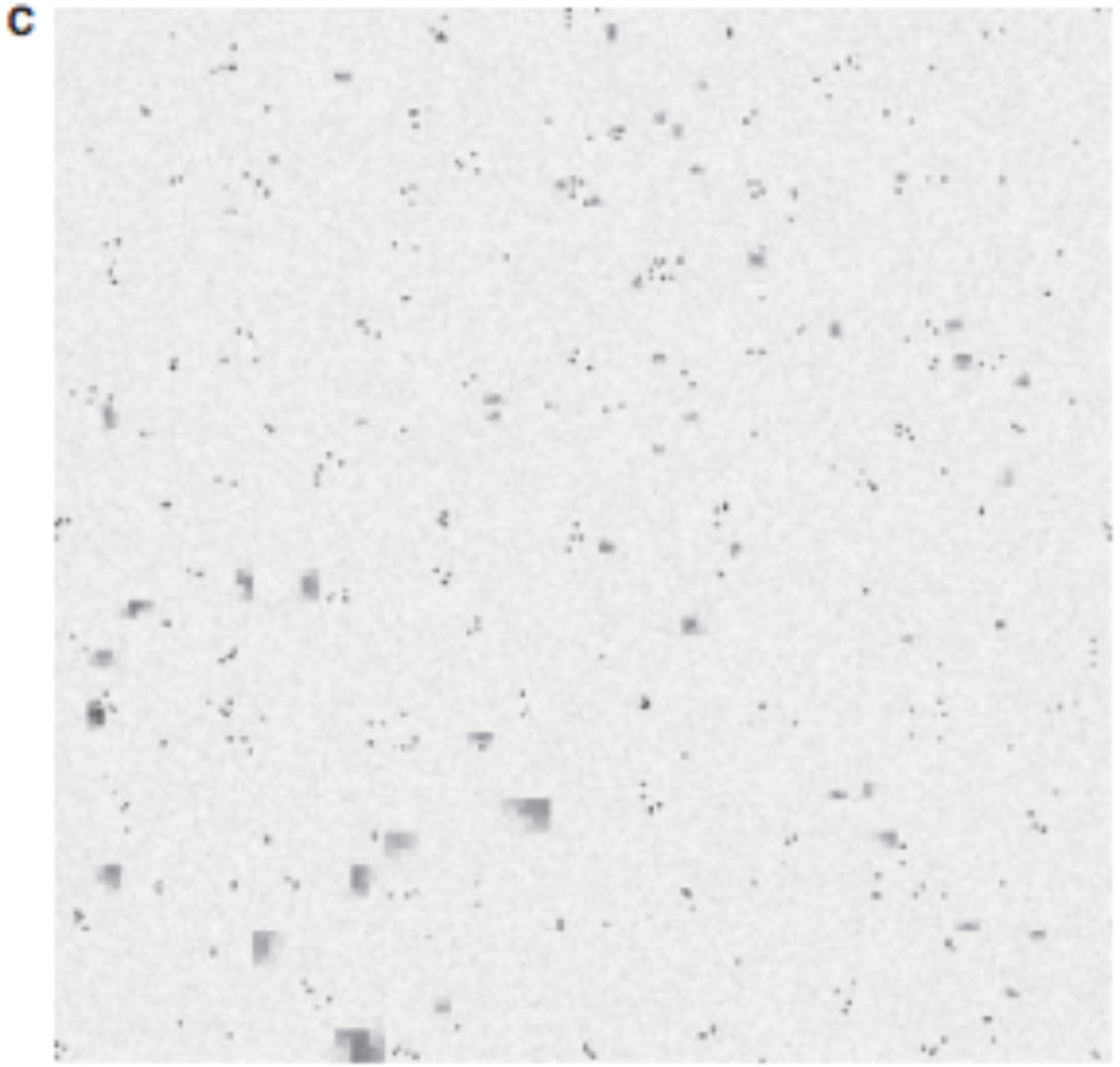
second iteration

third iteration

Hilbert curves use 2D.

Patterns revealed.





BIOINFORMATICS **ORIGINAL PAPER** *Vol. 25 no. 10 2009, pages 1231–1235*
doi:10.1093/bioinformatics/btp152

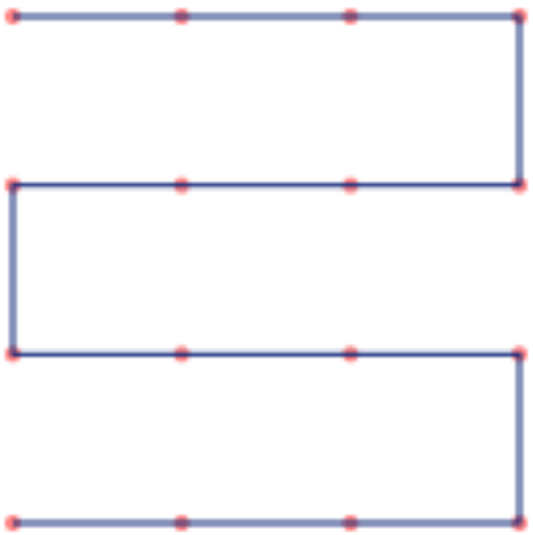
Genome analysis

Visualization of genomic data with the Hilbert curve

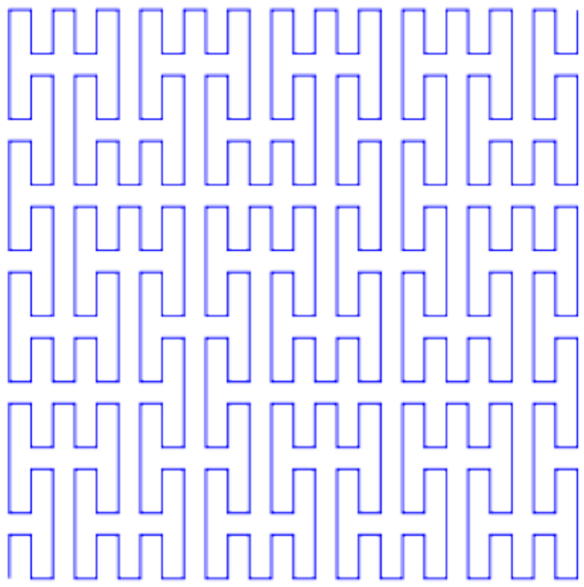
Simon Anders
European Bioinformatics Institute (EMBL-EBI), Hinxton, CB10 1SD, UK

Received on August 27, 2008; revised on March 11, 2009; accepted on March 14, 2009
Advance Access publication March 17, 2009
Associate Editor: John Quackenbush

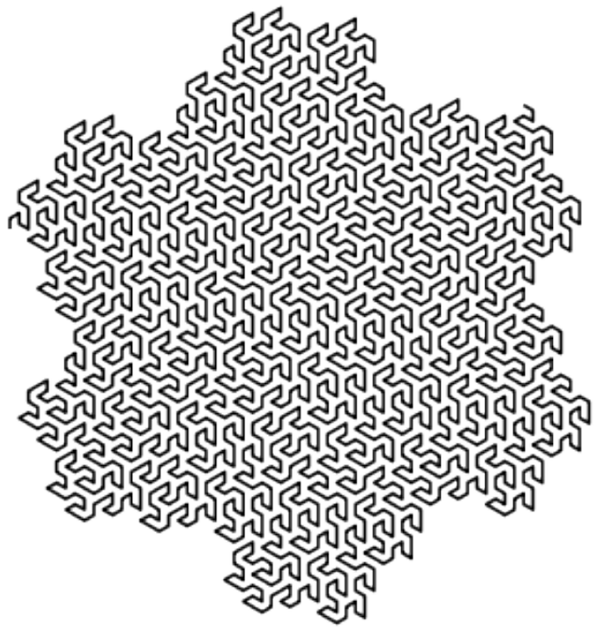
Types of SFCs



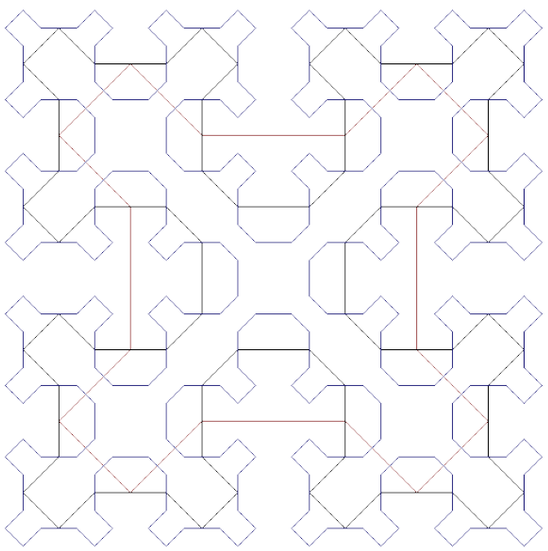
Zig-zag



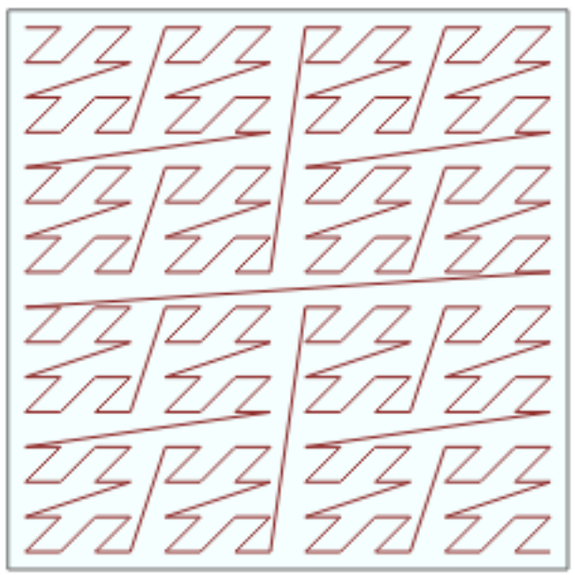
Peano



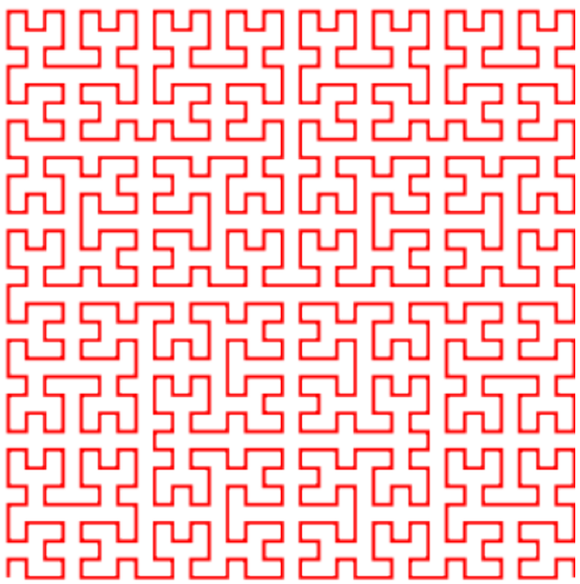
Gosper



Sierpinski



Z-order

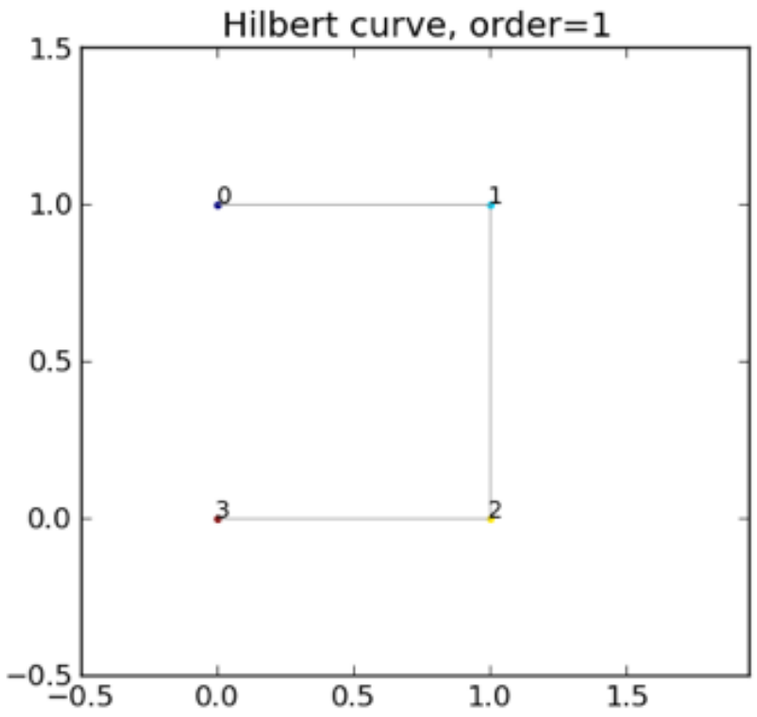


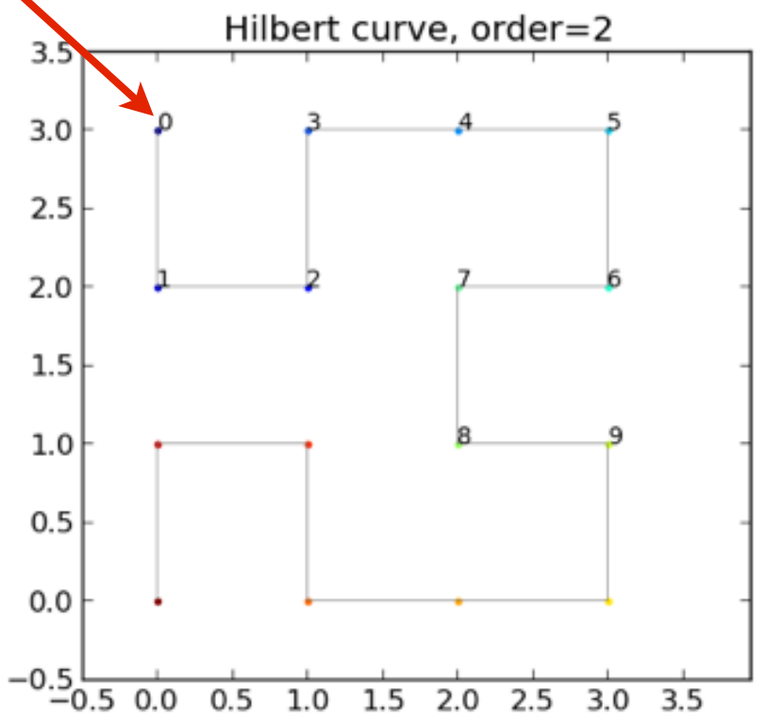
Hilbert

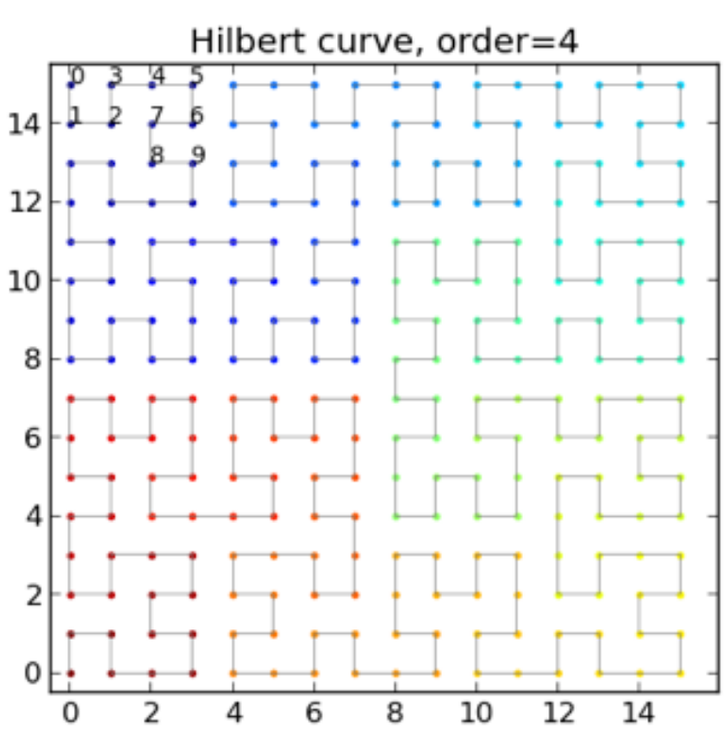
Why Hilbert curves for genomics?

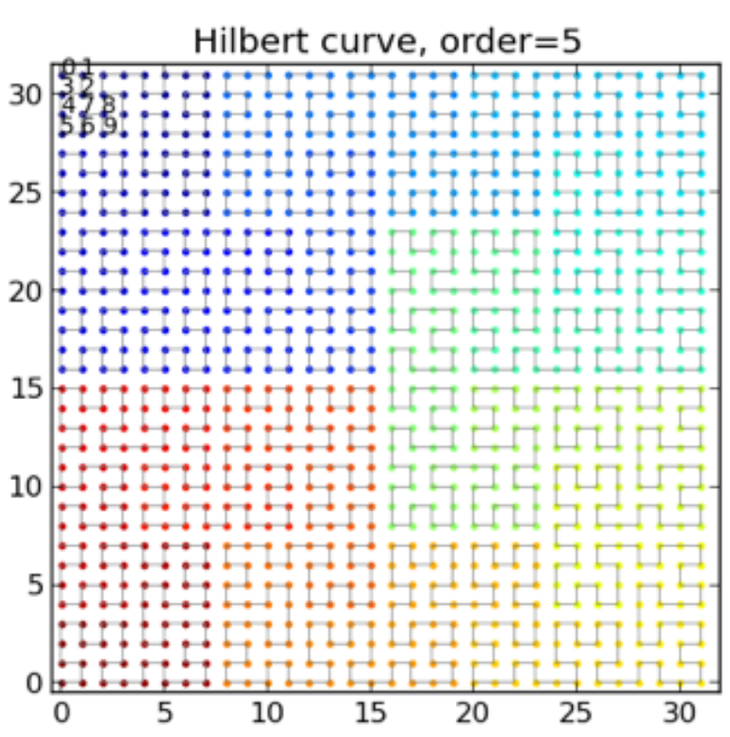
- The human genome is enormous. 3 billion base pairs.
- Difficult to see patterns in 1D plots when the coordinate space is so gargantuan.

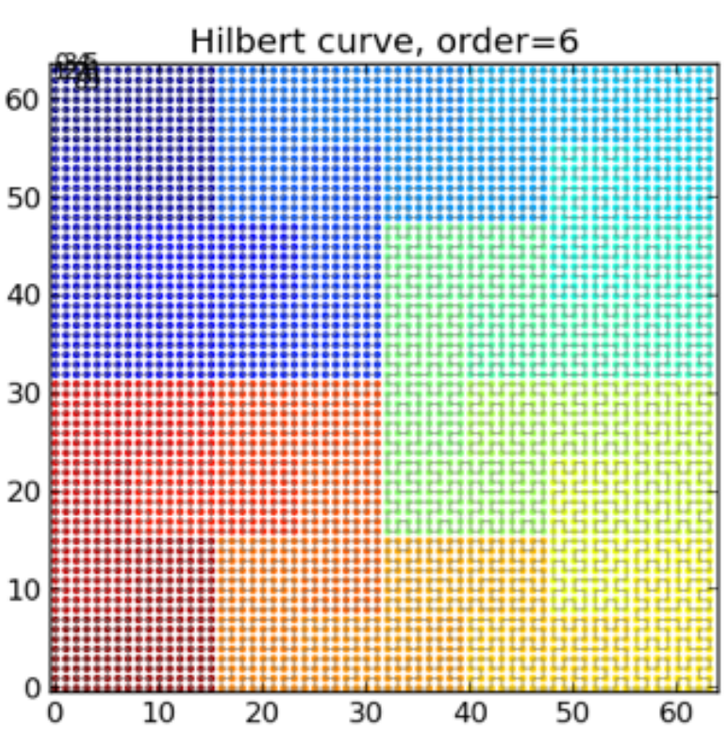
Origin (start of chrom)



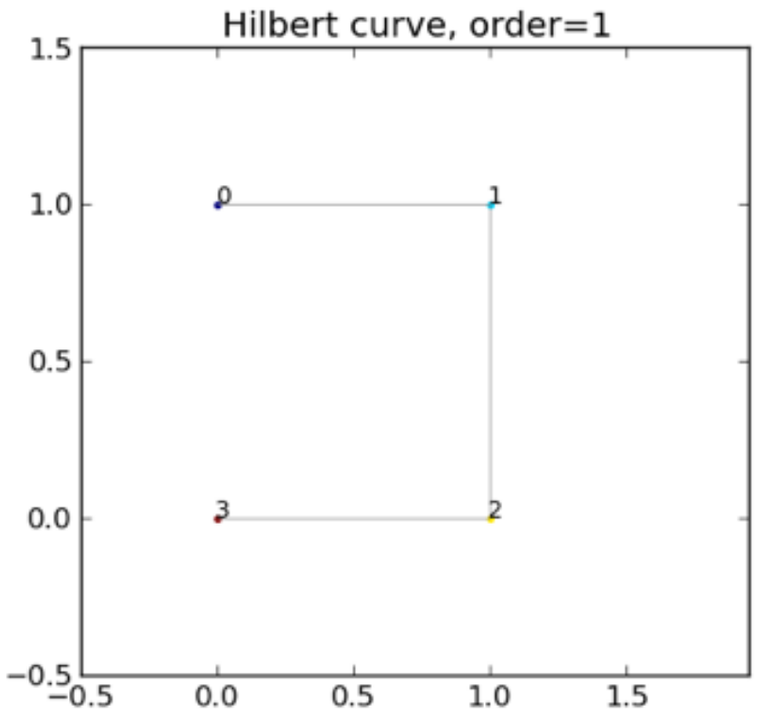


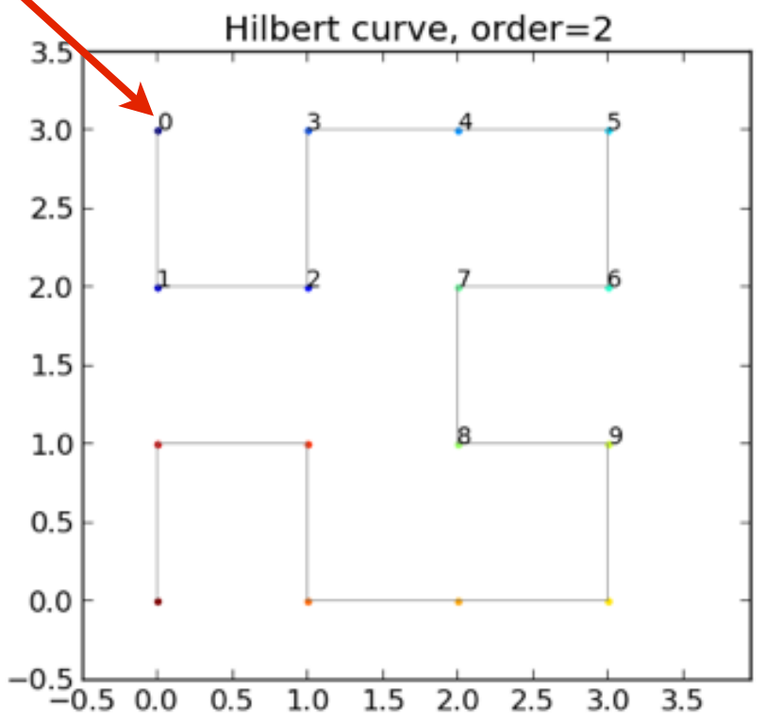


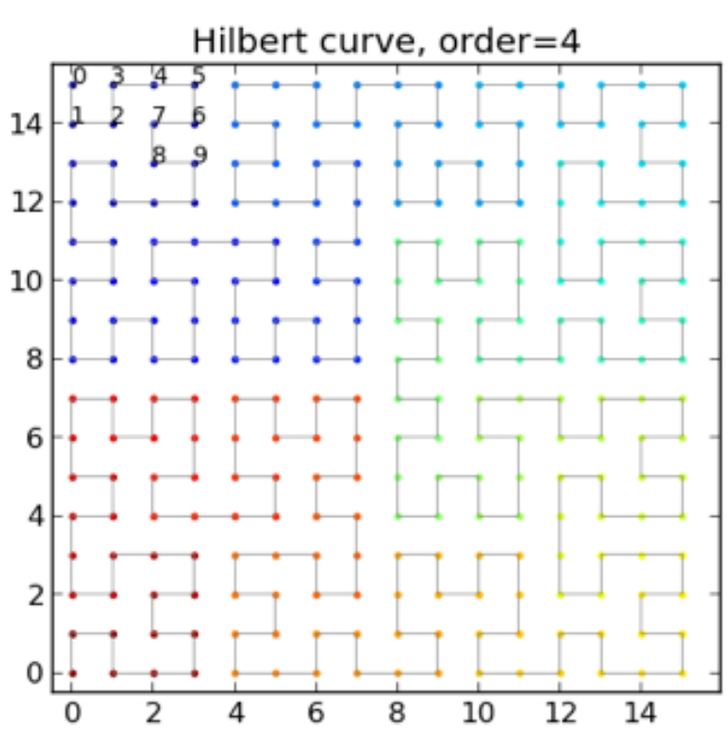


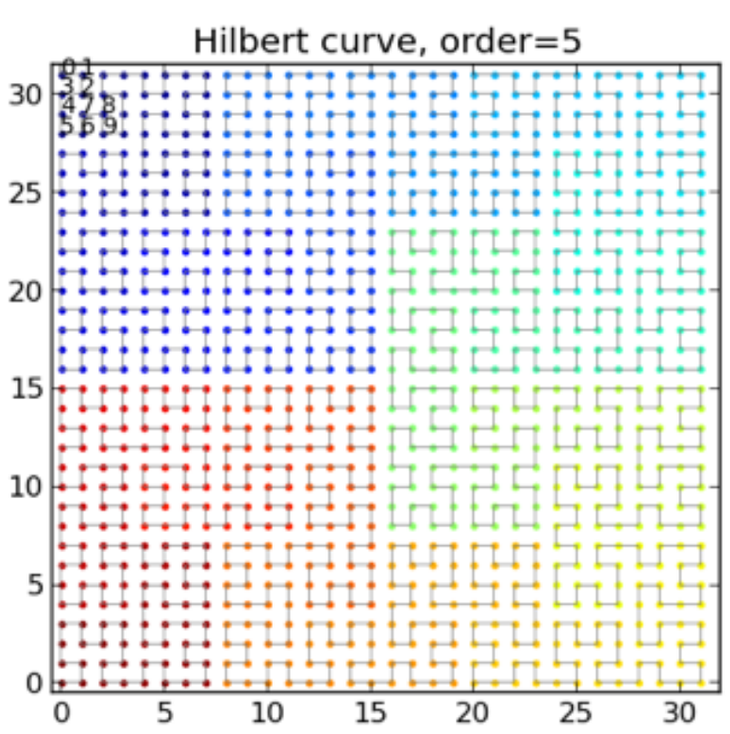


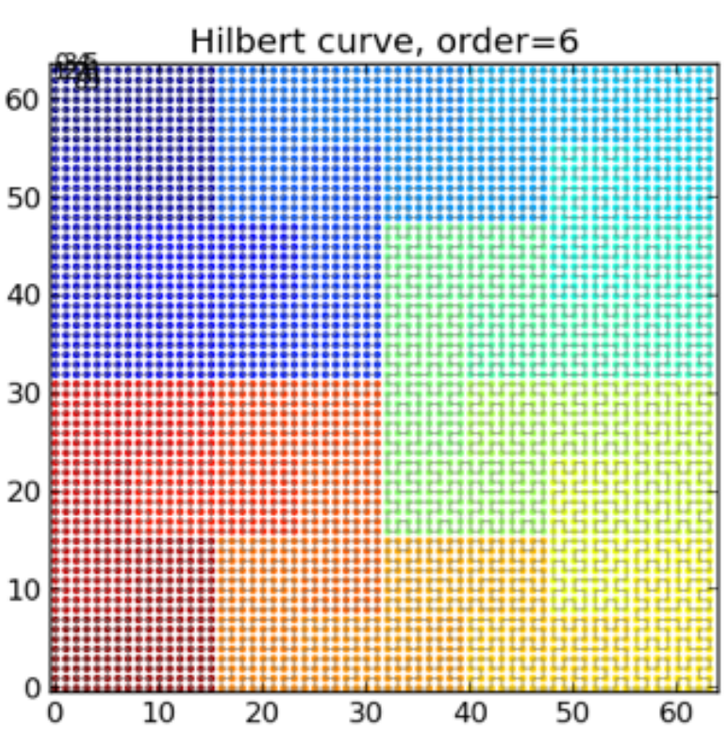
Origin (start of chrom)







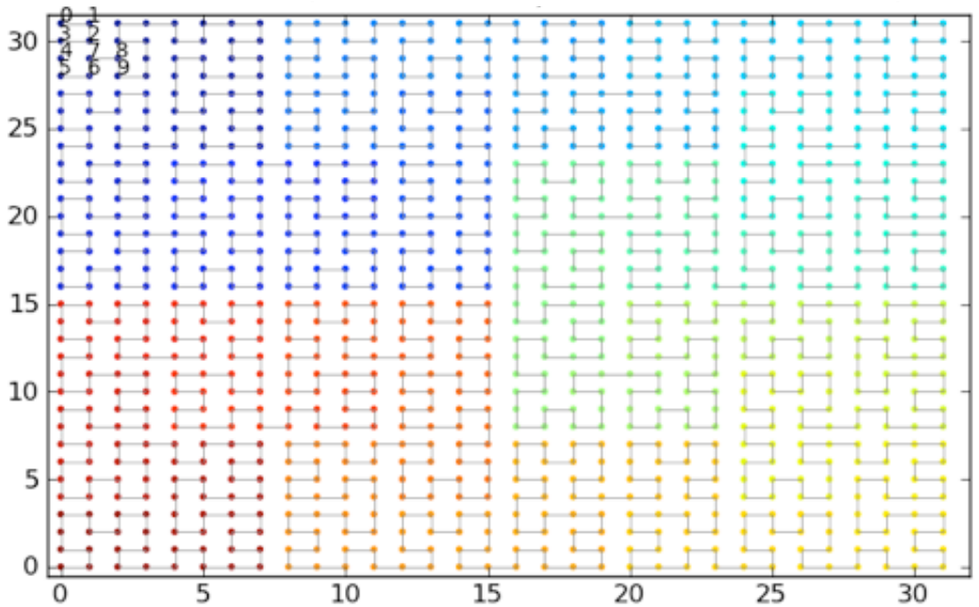




The higher the "order" the higher the resolution *- that is, each cell represents less genome space*

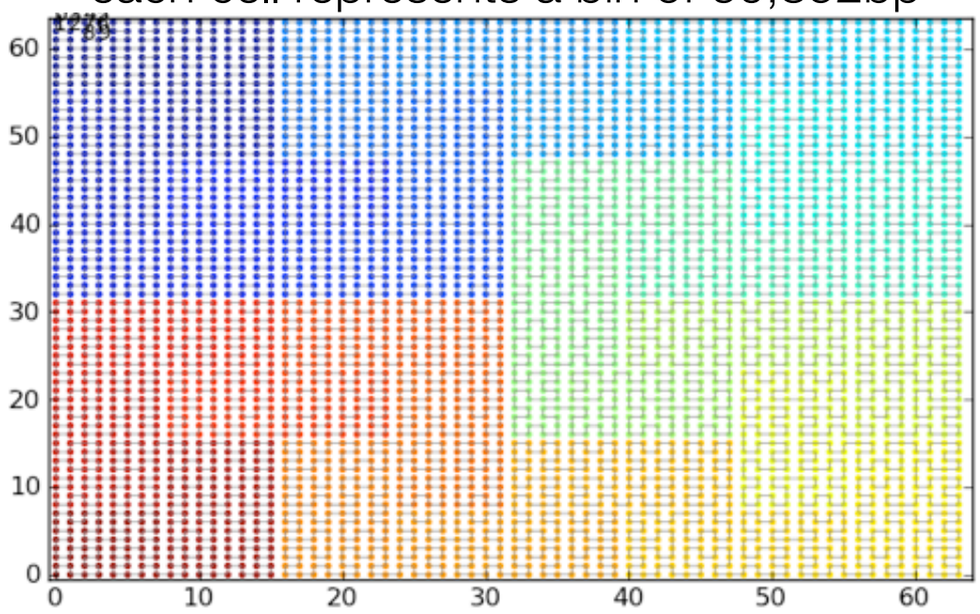
Human chromosome 1: 249,250,621 nucleotides

order=5;
2^5 (32*32 = 1024) cells
each cell represents a bin of 243,408bp

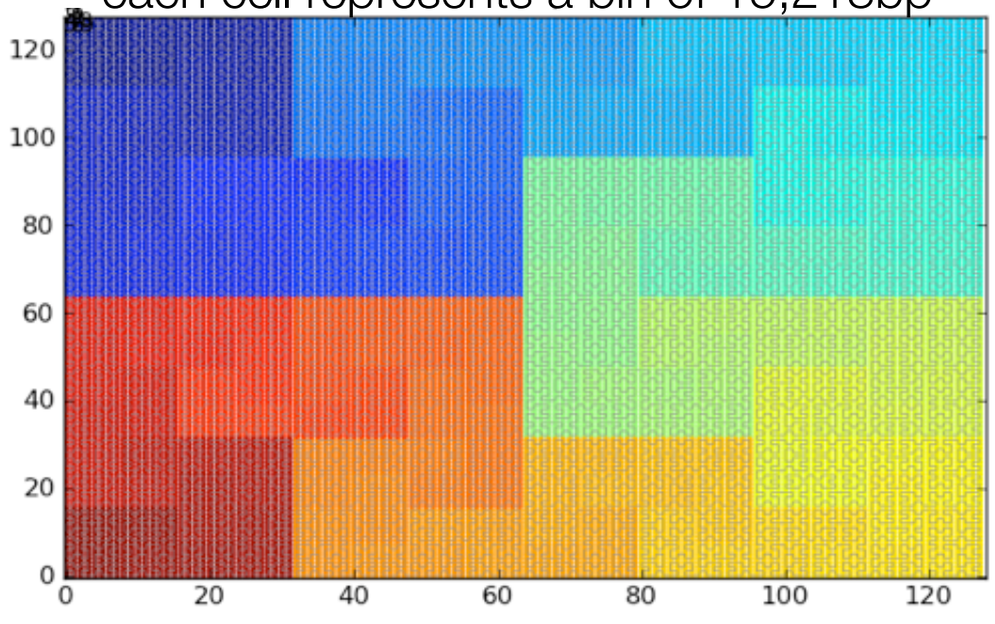


4x higher res.

order=6;
2^6 (64*64 = 4096) cells
each cell represents a bin of 60,852bp

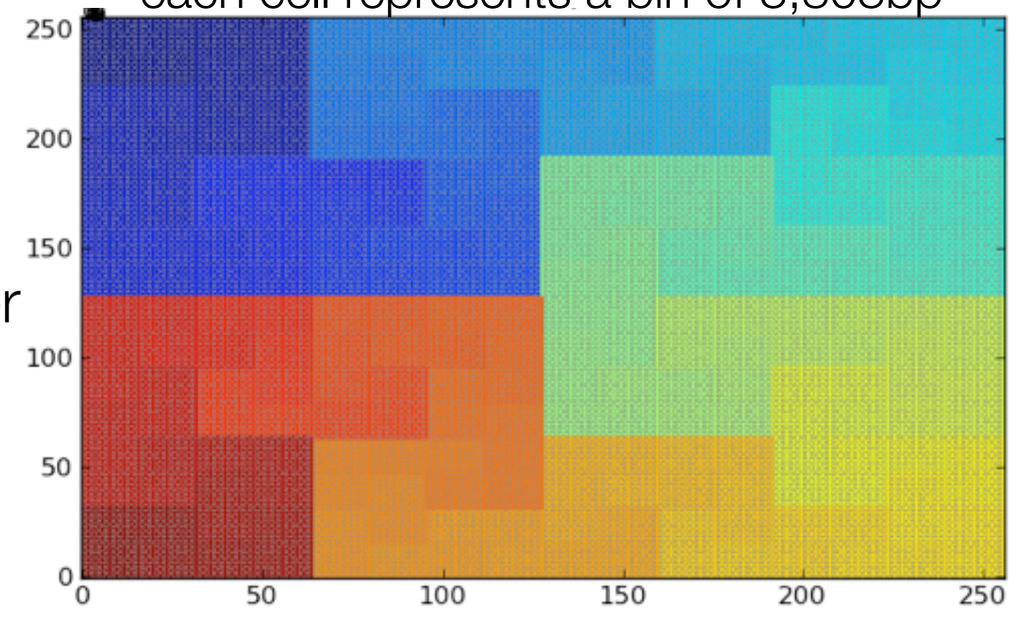


order=7;
2^7 (128*128 = 16,384) cells
each cell represents a bin of 15,213bp



4x higher res.

order=8;
2^8 (256*256 = 65,536) cells
each cell represents a bin of 3,803bp



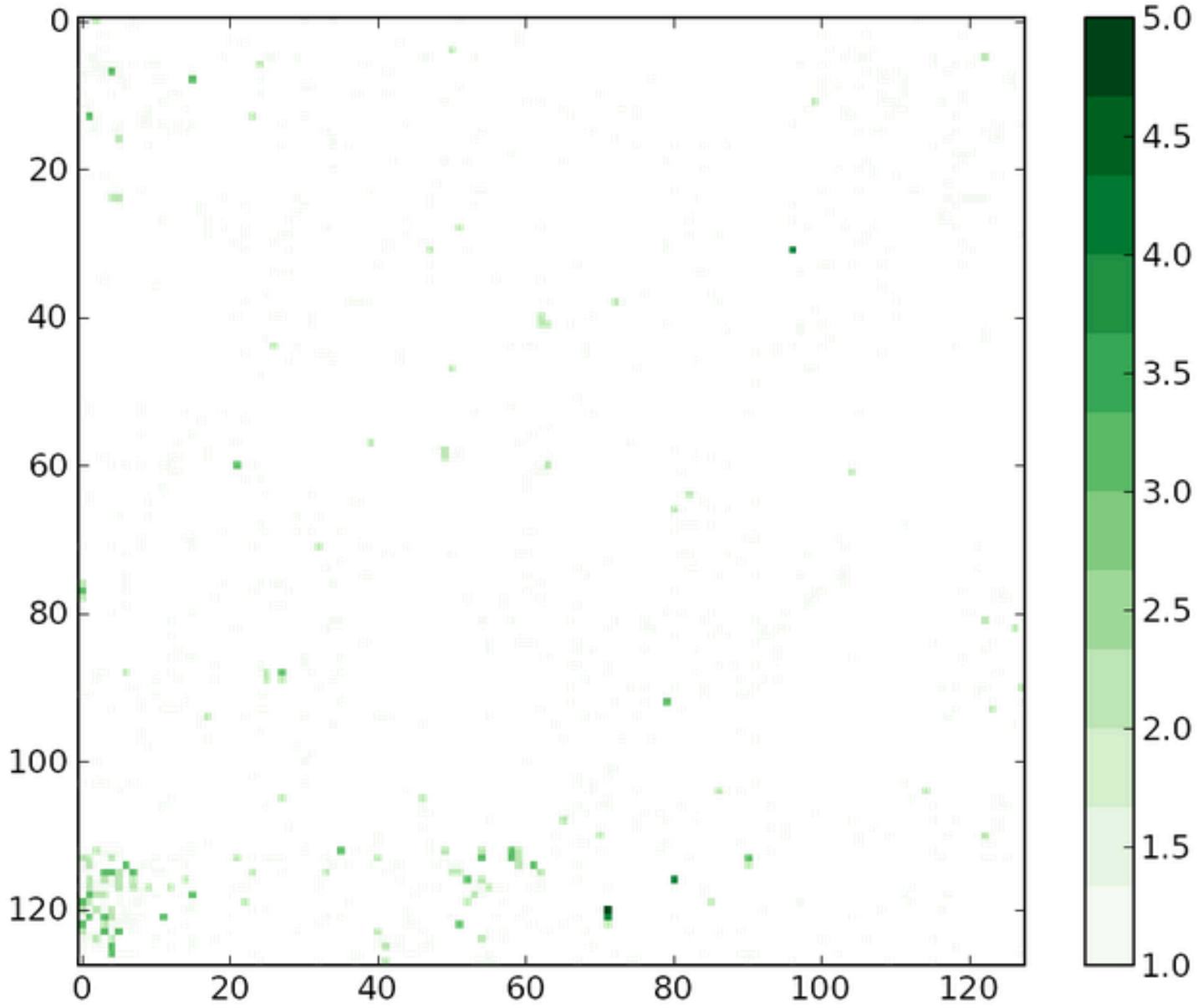
modENCODE browser

Very powerful, but an artistic toy.
Not reusable. Our motivation.

http://compbio.med.harvard.edu/flychromatin/

scurgen: Hilbert visualization toolkit for genomics

```
scurgen plot --chrom chr10 \
                          --cmap Greens \
                          --format png \
                          --dim 128 \
                          data/cpg-islands.hg19.chr10.bed
```



Ryan Layer

Graduate student

rl6sf @ virginia.edu

Research Interests: Sc
analysis; genome data mi
interpretation.

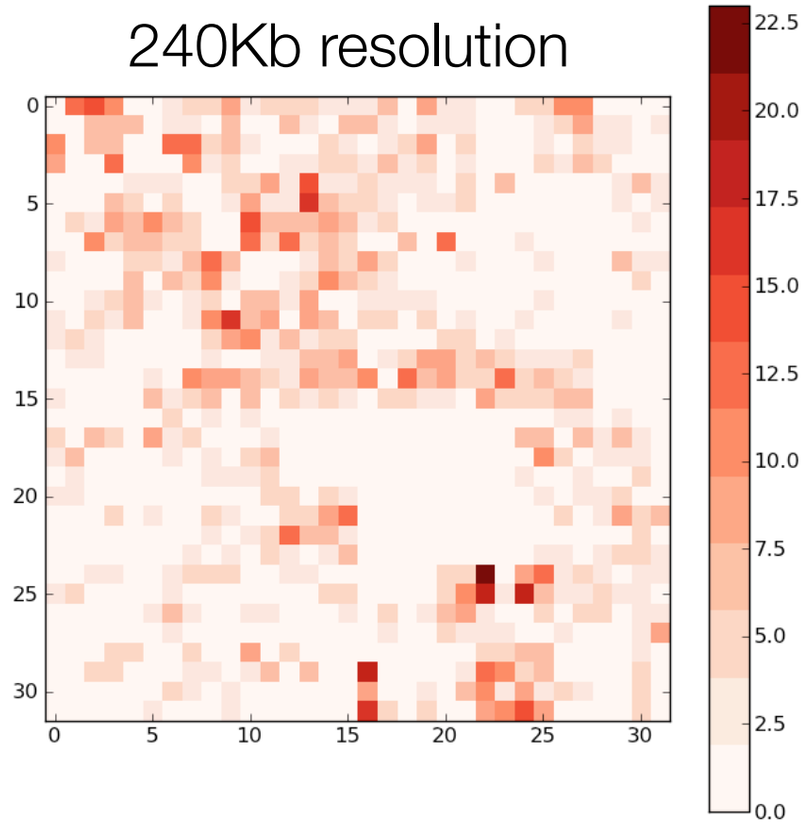
Ryan Dale

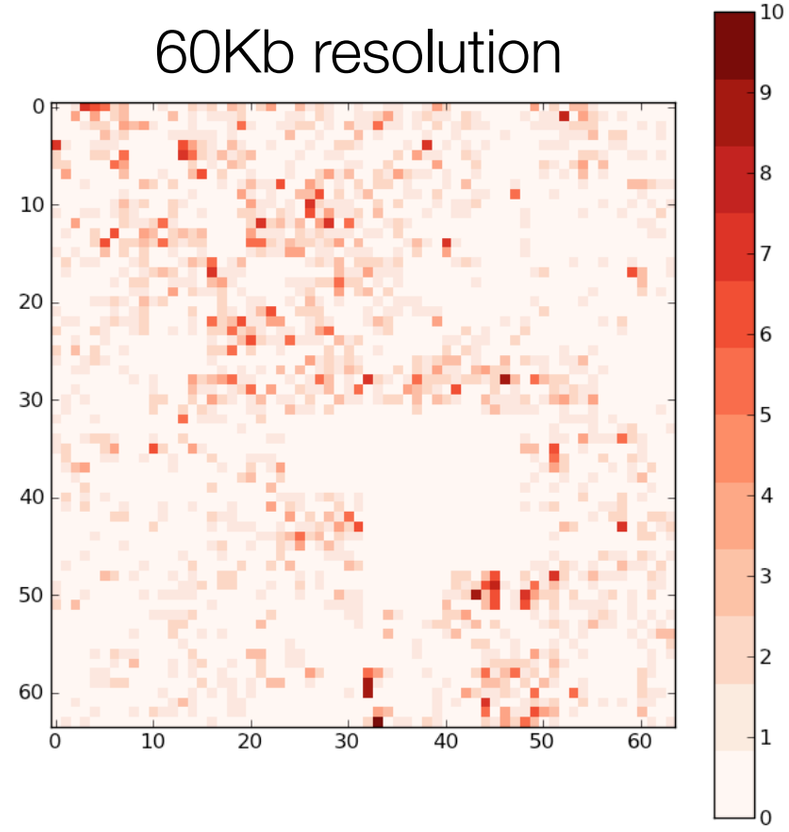
https://github.com/arq5x/scurgen

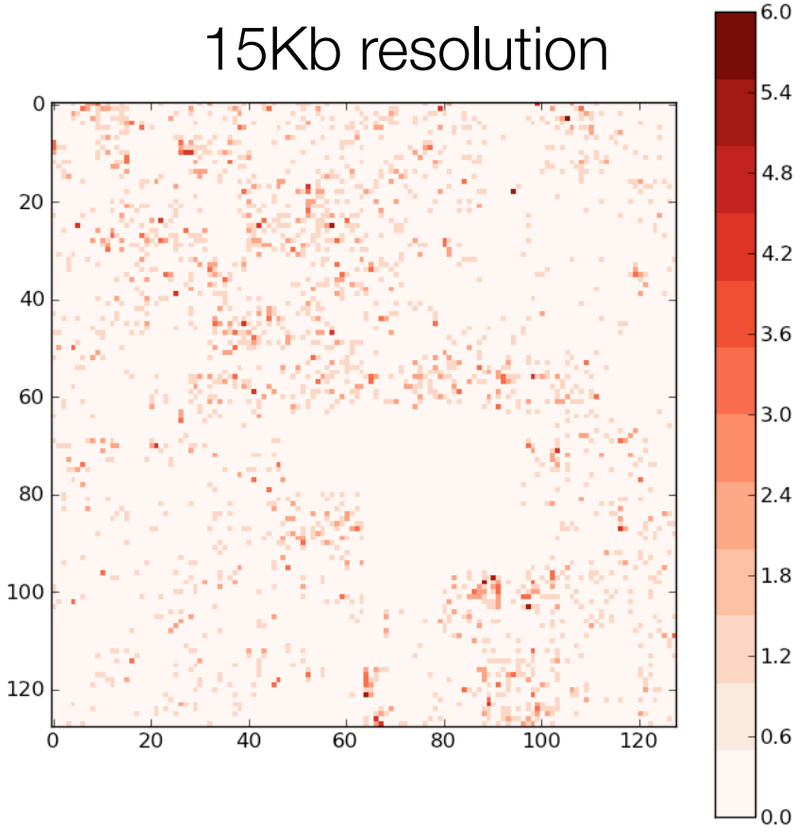
Algorithm for plotting genomic data

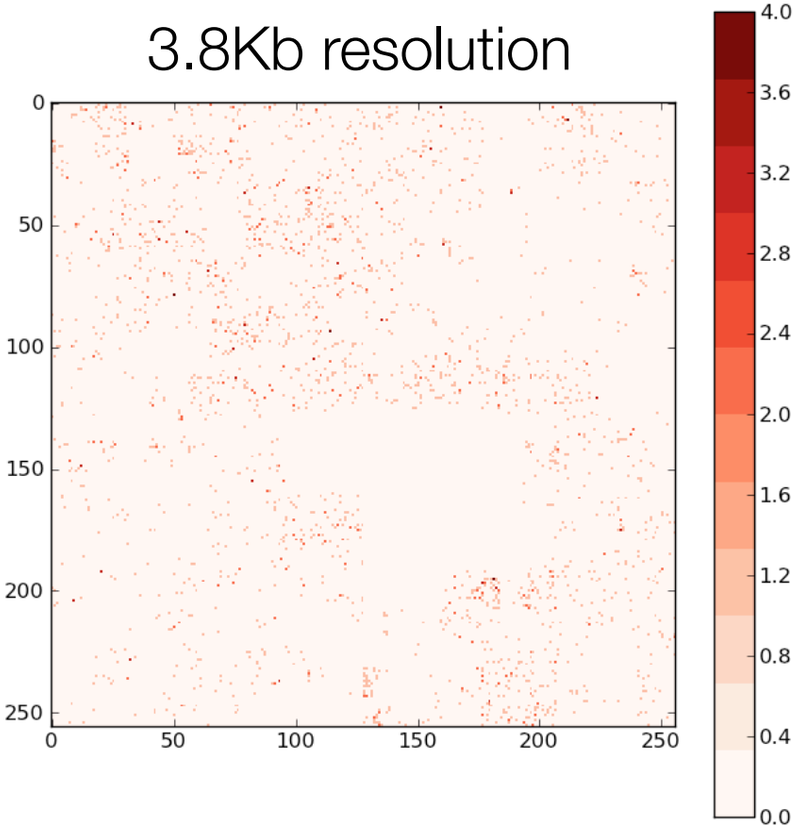
- Determine the size of the curve.
  - e.g. if 32x32 matrix, 1024 cells. curve length = 1024
- Determine the genomic size of each cell.
  - if 1024 cells and chrom length is 1048576 bp, then each cell represents 1024 bp.
- For each chrom interval in input file (BED, BAM, VCF, etc.)
  - figure out how at which point(s) along the curve (which cell) the interval belongs. **existing algorithms for this.**
  - update the cell. if BED, ++. if BEDGRAPH, += score.
- Plot matrix

Human chromosome 1: ENCODE ChIPseq of Nrf1



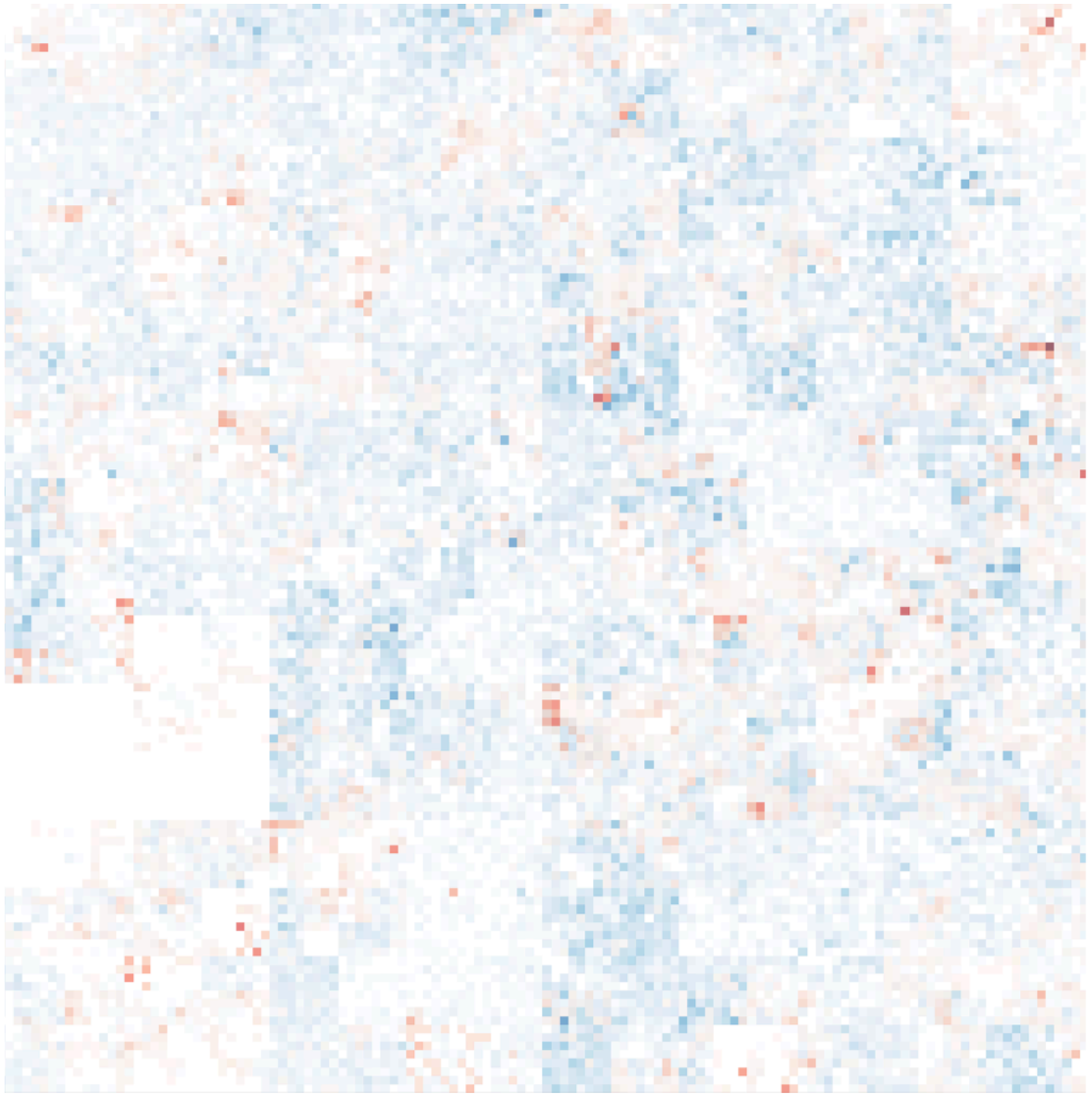






One dataset is okay. Ideal for *comparisons*

chr10
conserved elements: blue
coding exons: red



each cell is 8kb

centromere ⟶

What does this tell us (that we already know)?

Digging deeper into genomic “relationships”

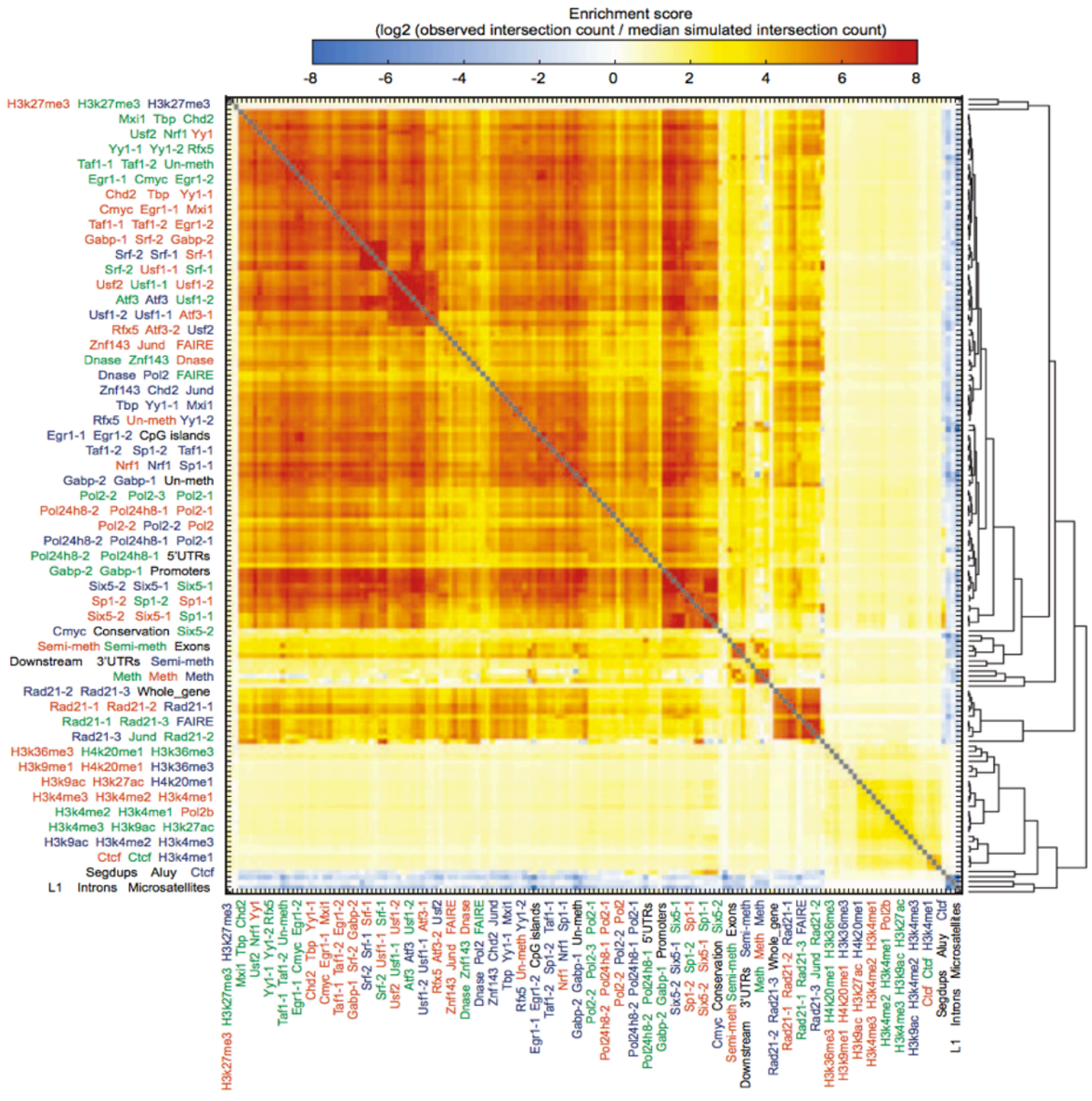
PREPRINT

Vol. 00 no. 00 2012
Pages 1–8

Binary Interval Search (BITS):
A Scalable Algorithm for Counting Interval Intersections

Ryan M. Layer[1], Kevin Skadron [1], Gabriel Robins[1], Ira M. Hall[2], and Aaron R. Quinlan[3]*

[1]Department of Computer Science, University of Virginia, Charlottesville, VA
[2]Department of Biochemistry and Molecular Genetics, University of Virginia, Charlottesville, VA
[3]Department of Public Health Sciences and Center for Public Health Genomics, University of Virginia, Charlottesville, VA

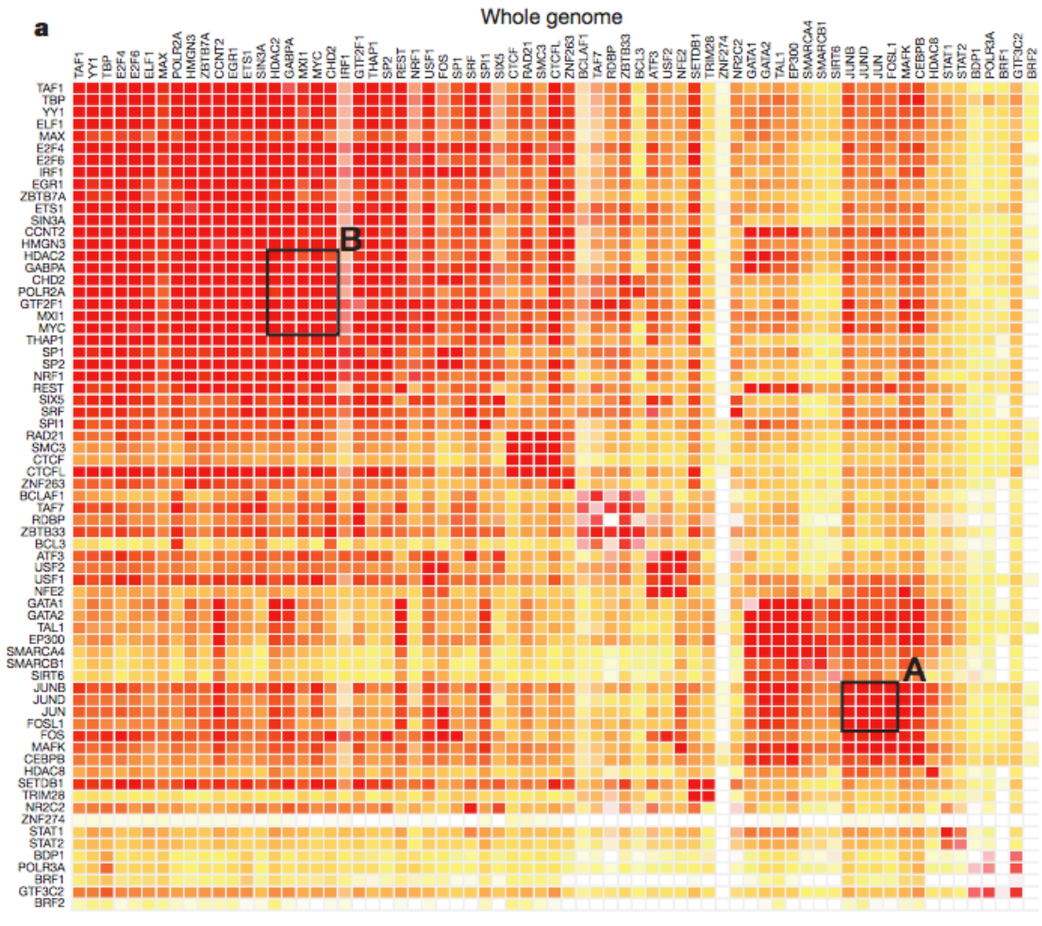


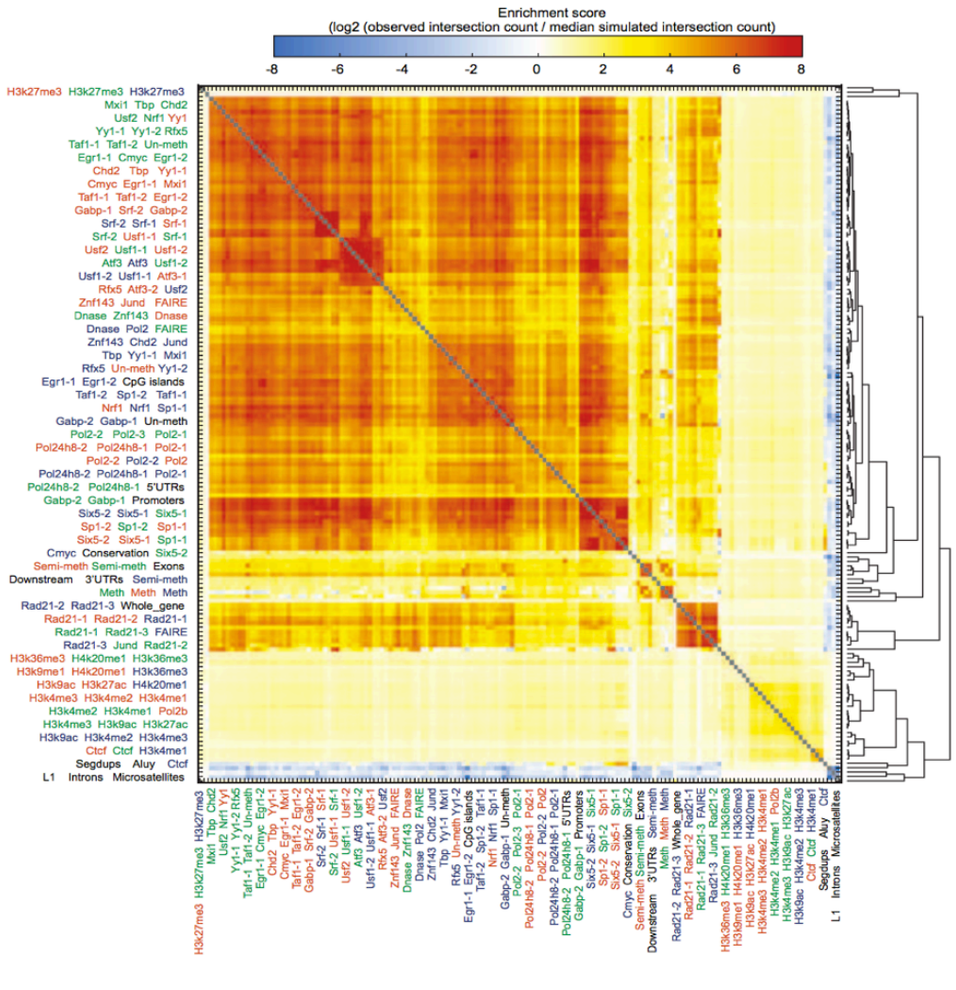
ARTICLE

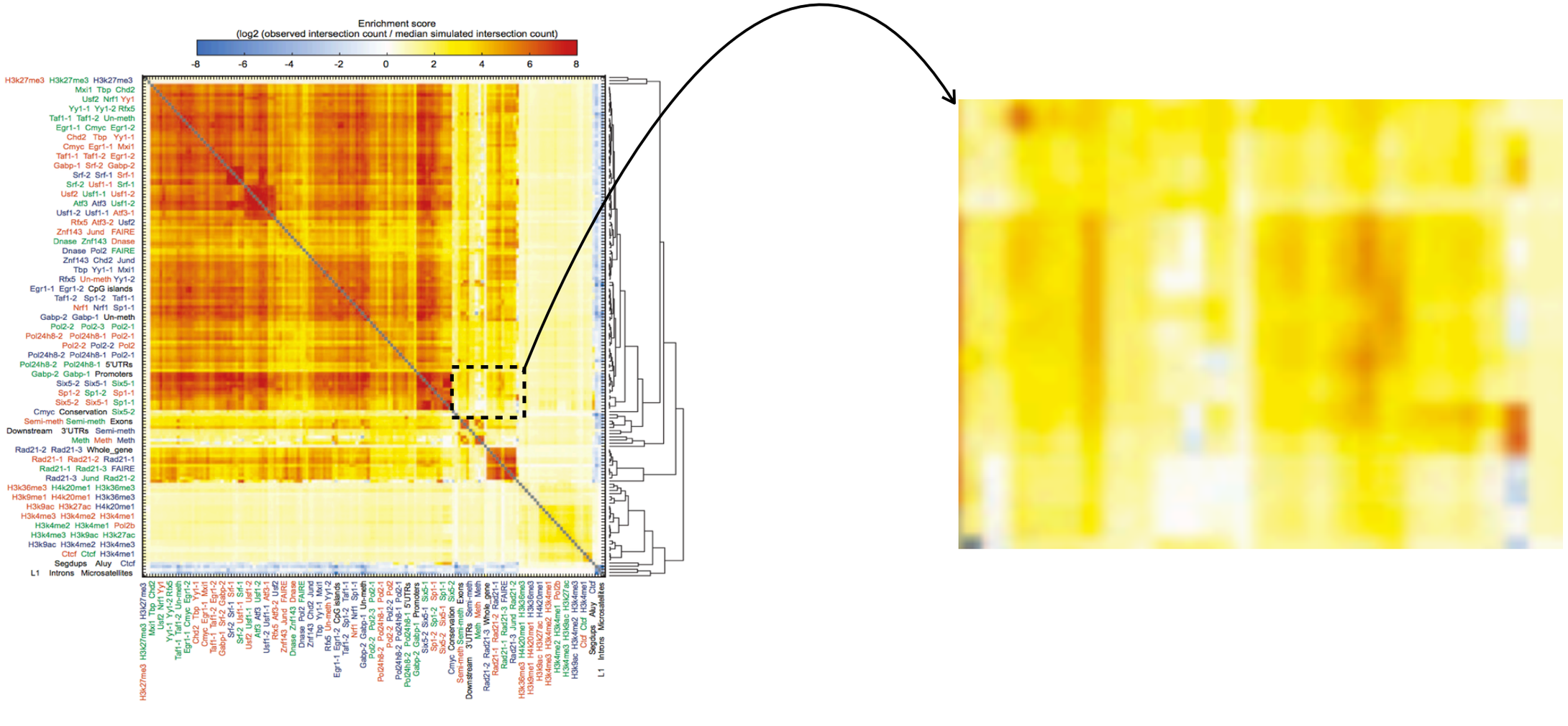
doi:10.1038/nature11247

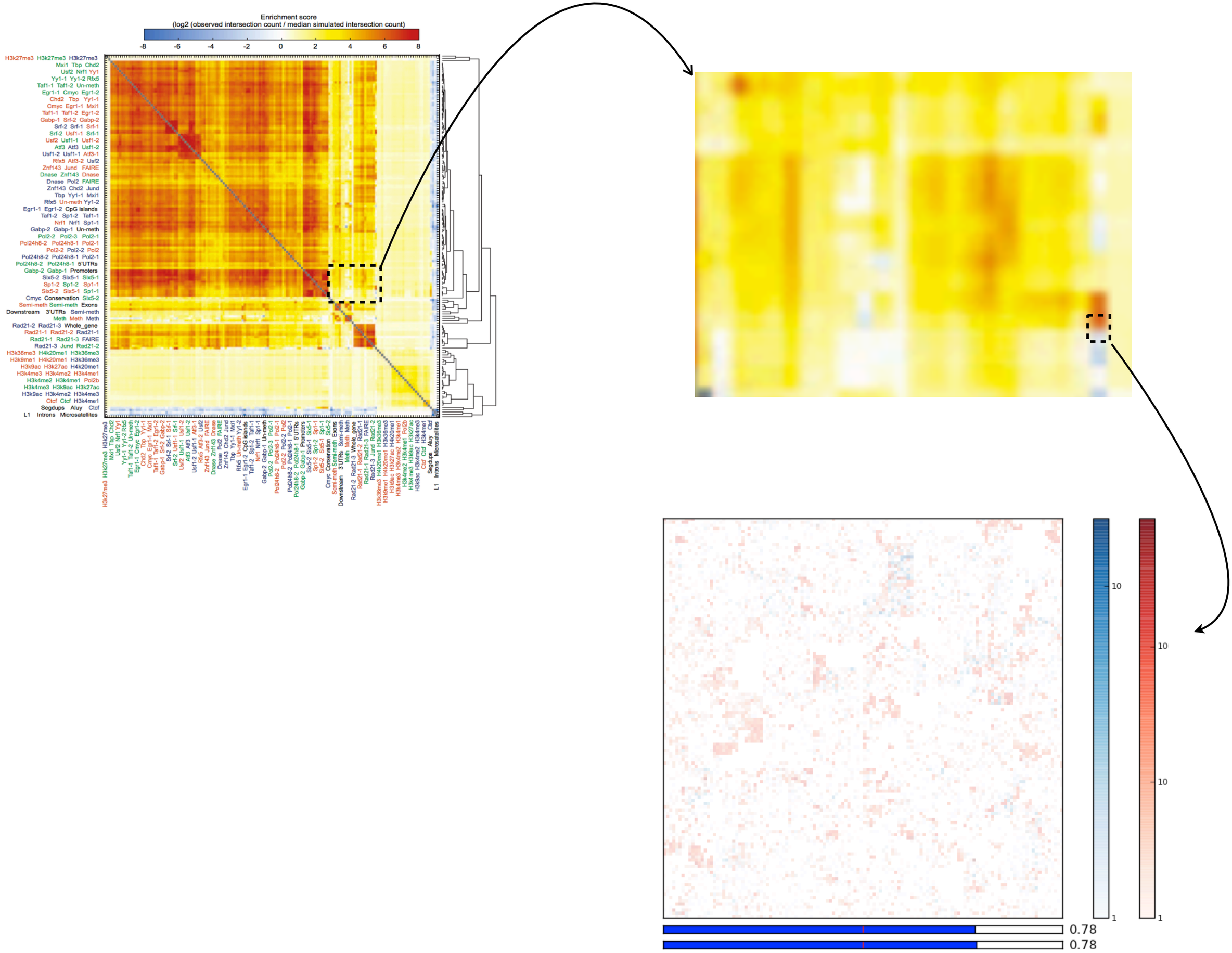
An integrated encyclopedia of DNA elements in the human genome

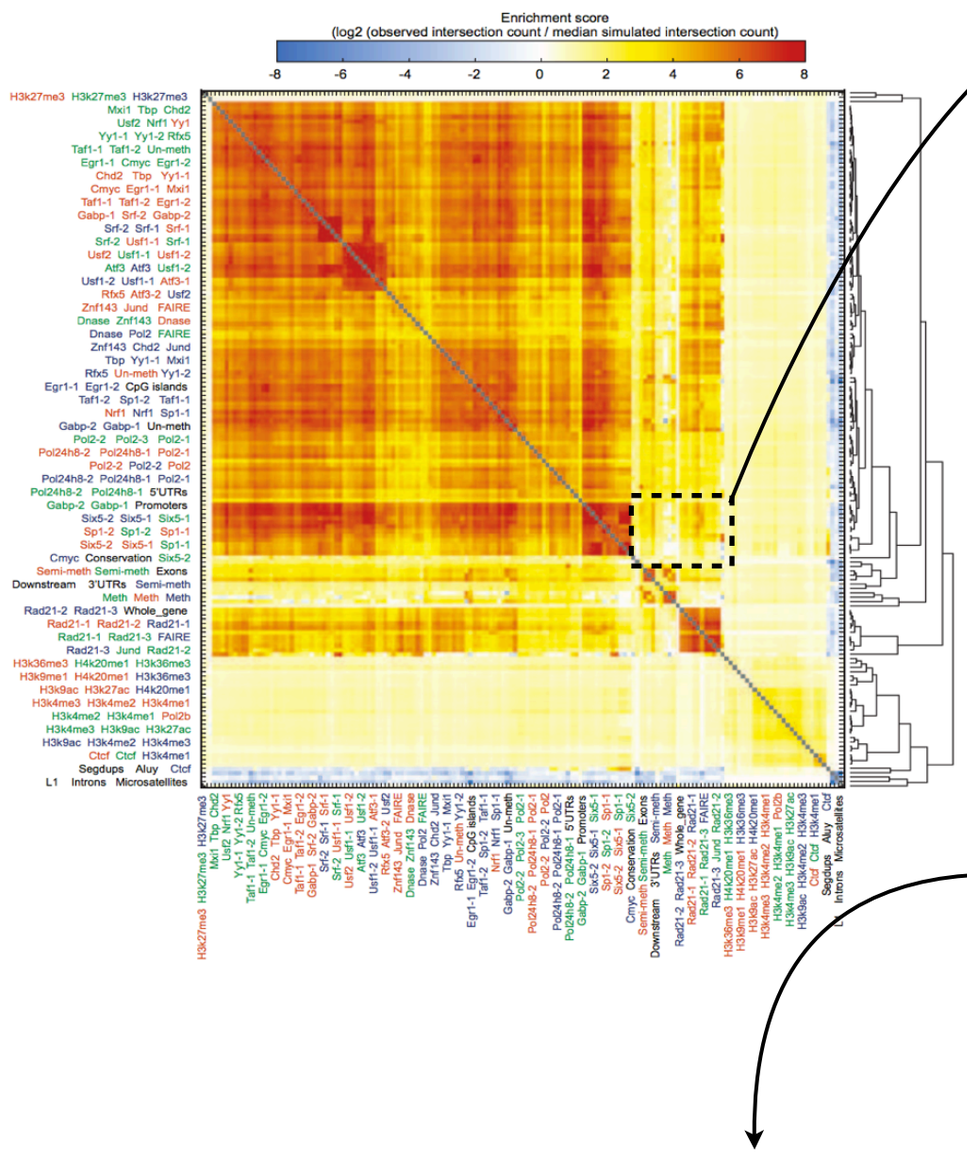
The ENCODE Project Consortium*

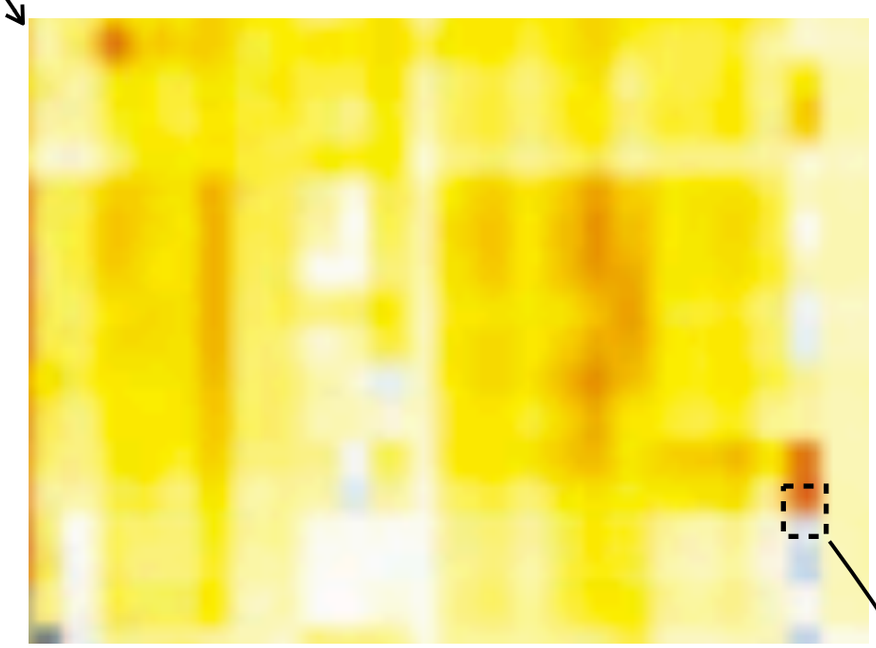


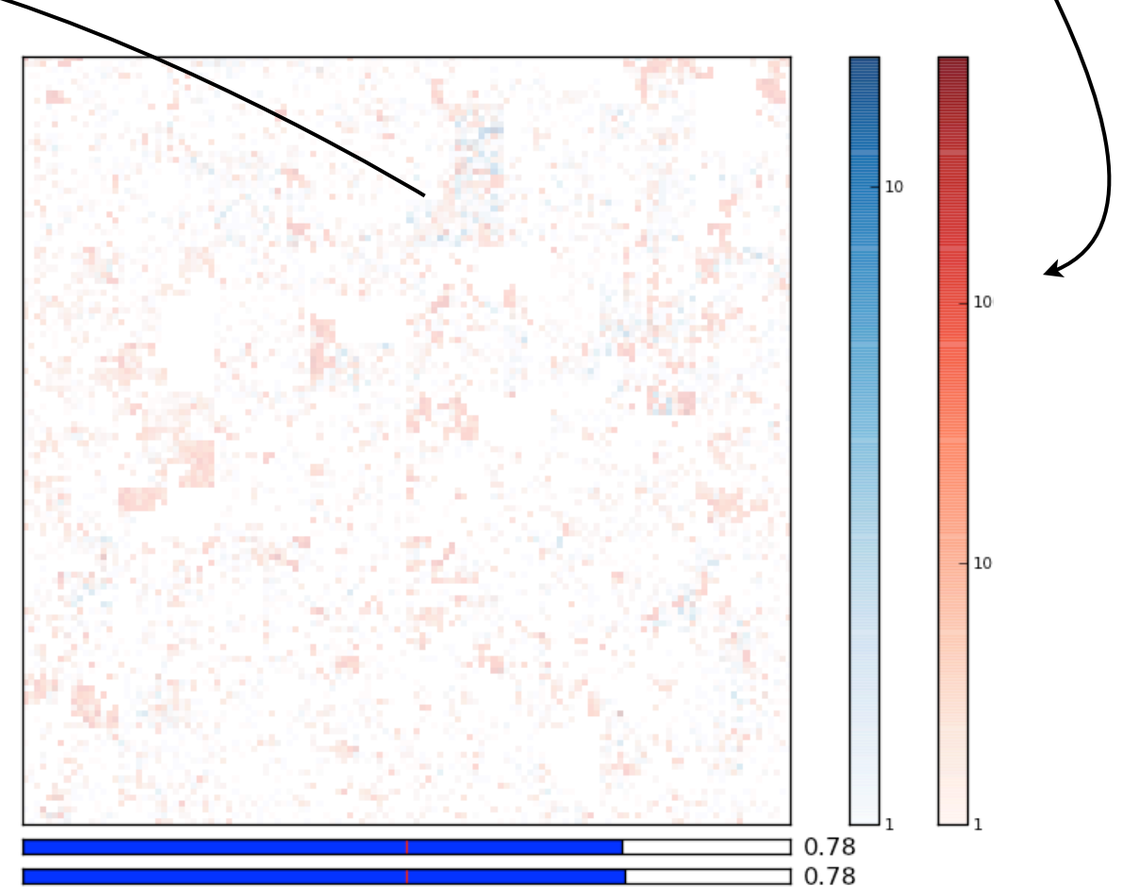

Enrichment score
(log2 (observed intersection count / median simulated intersection count)
-8
-6
-4
-2
0
2
4
6
8
H3k27me3 H3k27me3 H3k27me3
Mxi1 Tbp Chd2
Usf2 Nrf1 Yy1
Yy1-1 Yy1-2 Rfx5
Taf1-1 Taf1-2 Un-meth
Egr1-1 Cmyc Egr1-2
Chd2 Tbp Yy1-1
Cmyc Egr1-1 Mxi1
Taf1-1 Taf1-2 Egr1-2
Gabp-1 Srf-2 Gabp-2
Srf-2 Srf-1 Srf-1
Srf-2 Usf1-1 Srf-1
Usf2 Usf1-1 Usf1-2
Atf3 Atf3 Usf1-2
Usf1-2 Usf1-1 Atf3-1
Rfx5 Atf3-2 Usf2
Znf143 Jund FAIRE
Dnase Znf143 Dnase
Dnase Pol2 FAIRE
Znf143 Chd2 Jund
Tbp Yy1-1 Mxi1
Rfx5 Un-meth Yy1-2
Egr1-1 Egr1-2 CpG islands
Taf1-2 Sp1-2 Taf1-1
Nrf1 Nrf1 Sp1-1
Gabp-2 Gabp-1 Un-meth
Pol2-2 Pol2-3 Pol2-1
Pol24h8-2 Pol24h8-1 Pol2-1
Pol2-2 Pol2-2 Pol2
Pol24h8-2 Pol24h8-1 Pol2-1
Pol24h8-2 Pol24h8-1 5'UTRs
Gabp-2 Gabp-1 Promoters
Six5-2 Six5-1 Six5-1
Sp1-2 Sp1-2 Sp1-1
Six5-2 Six5-1 Sp1-1
Cmyc Conservation Six5-2
Semi-meth Semi-meth Exons
Downstream 3'UTRs Semi-meth
Meth Meth Meth
Rad21-2 Rad21-3 Whole_gene
Rad21-1 Rad21-2 Rad21-1
Rad21-1 Rad21-3 FAIRE
Rad21-3 Jund Rad21-2
H3k36me3 H4k20me1 H3k36me3
H3k9me1 H4k20me1 H3k36me3
H3k9ac H3k27ac H4k20me1
H3k4me3 H3k4me2 H3k4me1
H3k4me2 H3k4me1 Pol2b
H3k4me3 H3k9ac H3k27ac
H3k9ac H3k4me2 H3k4me3
Ctcf Ctcf H3k4me1
Segdups Aluy Ctcf
L1 Introns Microsatellites

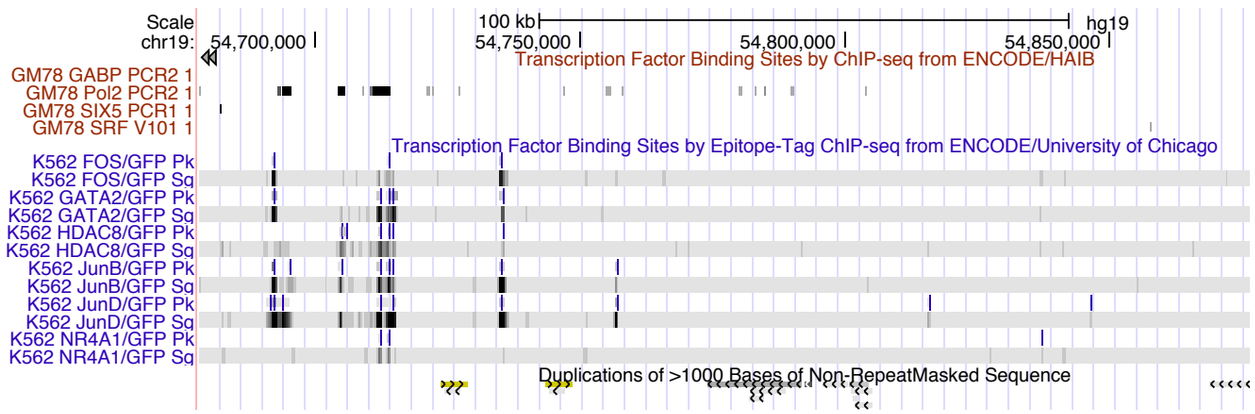

Enrichment score
(log2 (observed intersection count / median simulated intersection count)
-8
-6
-4
-2
0
2
4
6
8
H3k27me3 H3k27me3 H3k27me3
Mxi1 Tbp Chd2
Usf2 Nrf1 Yy1
Yy1-1 Yy1-2 Rfx5
Taf1-1 Taf1-2 Un-meth
Egr1-1 Cmyc Egr1-2
Chd2 Tbp Yy1-1
Cmyc Egr1-1 Mxi1
Taf1-1 Taf1-2 Egr1-2
Gabp-1 Srf-2 Gabp-2
Srf-2 Srf-1 Srf-1
Srf-2 Usf1-1 Srf-1
Usf2 Usf1-1 Usf1-2
Atf3 Atf3 Usf1-2
Usf1-2 Usf1-1 Atf3-1
Rfx5 Atf3-2 Usf2
Znf143 Jund FAIRE
Dnase Znf143 Dnase
Dnase Pol2 FAIRE
Znf143 Chd2 Jund
Tbp Yy1-1 Mxi1
Rfx5 Un-meth Yy1-2
Egr1-1 Egr1-2 CpG islands
Taf1-2 Sp1-2 Taf1-1
Nrf1 Nrf1 Sp1-1
Gabp-2 Gabp-1 Un-meth
Pol2-2 Pol2-3 Pol2-1
Pol24h8-2 Pol24h8-1 Pol2-1
Pol2-2 Pol2-2 Pol2
Pol24h8-2 Pol24h8-1 Pol2-1
Pol24h8-2 Pol24h8-1 5'UTRs
Gabp-2 Gabp-1 Promoters
Six5-2 Six5-1 Six5-1
Sp1-2 Sp1-2 Sp1-1
Six5-2 Six5-1 Sp1-1
Cmyc Conservation Six5-2
Semi-meth Semi-meth Exons
Downstream 3'UTRs Semi-meth
Meth Meth Meth
Rad21-2 Rad21-3 Whole_gene
Rad21-1 Rad21-2 Rad21-1
Rad21-1 Rad21-3 FAIRE
Rad21-3 Jund Rad21-2
H3k36me3 H4k20me1 H3k36me3
H3k9me1 H4k20me1 H3k36me3
H3k9ac H3k27ac H4k20me1
H3k4me3 H3k4me2 H3k4me1
H3k4me2 H3k4me1 Pol2b
H3k4me3 H3k9ac H3k27ac
H3k9ac H3k4me2 H3k4me3
Ctcf Ctcf H3k4me1
Segdups Aluy Ctcf
L1 Introns Microsatellites


Enrichment score
(log2 (observed intersection count / median simulated intersection count)
-8
-6
-4
-2
0
2
4
6
8
H3k27me3 H3k27me3 H3k27me3
Mxi1 Tbp Chd2
Usf2 Nrf1 Yy1
Yy1-1 Yy1-2 Rfx5
Taf1-1 Taf1-2 Un-meth
Egr1-1 Cmyc Egr1-2
Chd2 Tbp Yy1-1
Cmyc Egr1-1 Mxi1
Taf1-1 Taf1-2 Egr1-2
Gabp-1 Srf-2 Gabp-2
Srf-2 Srf-1 Srf-1
Srf-2 Usf1-1 Srf-1
Usf2 Usf1-1 Usf1-2
Atf3 Atf3 Usf1-2
Usf1-2 Usf1-1 Atf3-1
Rfx5 Atf3-2 Usf2
Znf143 Jund FAIRE
Dnase Znf143 Dnase
Dnase Pol2 FAIRE
Znf143 Chd2 Jund
Tbp Yy1-1 Mxi1
Rfx5 Un-meth Yy1-2
Egr1-1 Egr1-2 CpG islands
Taf1-2 Sp1-2 Taf1-1
Nrf1 Nrf1 Sp1-1
Gabp-2 Gabp-1 Un-meth
Pol2-2 Pol2-3 Pol2-1
Pol24h8-2 Pol24h8-1 Pol2-1
Pol2-2 Pol2-2 Pol2
Pol24h8-2 Pol24h8-1 Pol2-1
Pol24h8-2 Pol24h8-1 5'UTRs
Gabp-2 Gabp-1 Promoters
Six5-2 Six5-1 Six5-1
Sp1-2 Sp1-2 Sp1-1
Six5-2 Six5-1 Sp1-1
Cmyc Conservation Six5-2
Semi-meth Semi-meth Exons
Downstream 3'UTRs Semi-meth
Meth Meth Meth
Rad21-2 Rad21-3 Whole_gene
Rad21-1 Rad21-2 Rad21-1
Rad21-3 Jund Rad21-2
H3k36me3 H4k20me1 H3k36me3
H3k9me1 H4k20me1 H3k36me3
H3k9ac H3k27ac H4k20me1
H3k4me3 H3k4me2 H3k4me1
H3k4me2 H3k4me1 Pol2b
H3k4me3 H3k9ac H3k27ac
H3k9ac H3k4me2 H3k4me3
Ctcf Ctcf H3k4me1
Segdups Aluy Ctcf
L1 Introns Microsatellites
10
10
10
1
1
0.78
0.78


Enrichment score
(log2 (observed intersection count / median simulated intersection count)
-8 -6 -4 -2 0 2 4 6 8




10
1
10
1
0.78
0.78


Scale
chr19:
54,700,000
100 kb
54,750,000
54,800,000
54,850,000
hg19
Transcription Factor Binding Sites by ChIP-seq from ENCODE/HAIB
GM78 GABP PCR2 1
GM78 Pol2 PCR2 1
GM78 SIX5 PCR1 1
GM78 SRF V101 1
Transcription Factor Binding Sites by Epitope-Tag ChIP-seq from ENCODE/University of Chicago
K562 FOS/GFP Pk
K562 FOS/GFP Sq
K562 GATA2/GFP Pk
K562 GATA2/GFP Sq
K562 HDAC8/GFP Pk
K562 HDAC8/GFP Sq
K562 JunB/GFP Pk
K562 JunB/GFP Sq
K562 JunD/GFP Pk
K562 JunD/GFP Sq
K562 NR4A1/GFP Pk
K562 NR4A1/GFP Sq
Duplications of >1000 Bases of Non-RepeatMasked Sequence

scurgen GUI

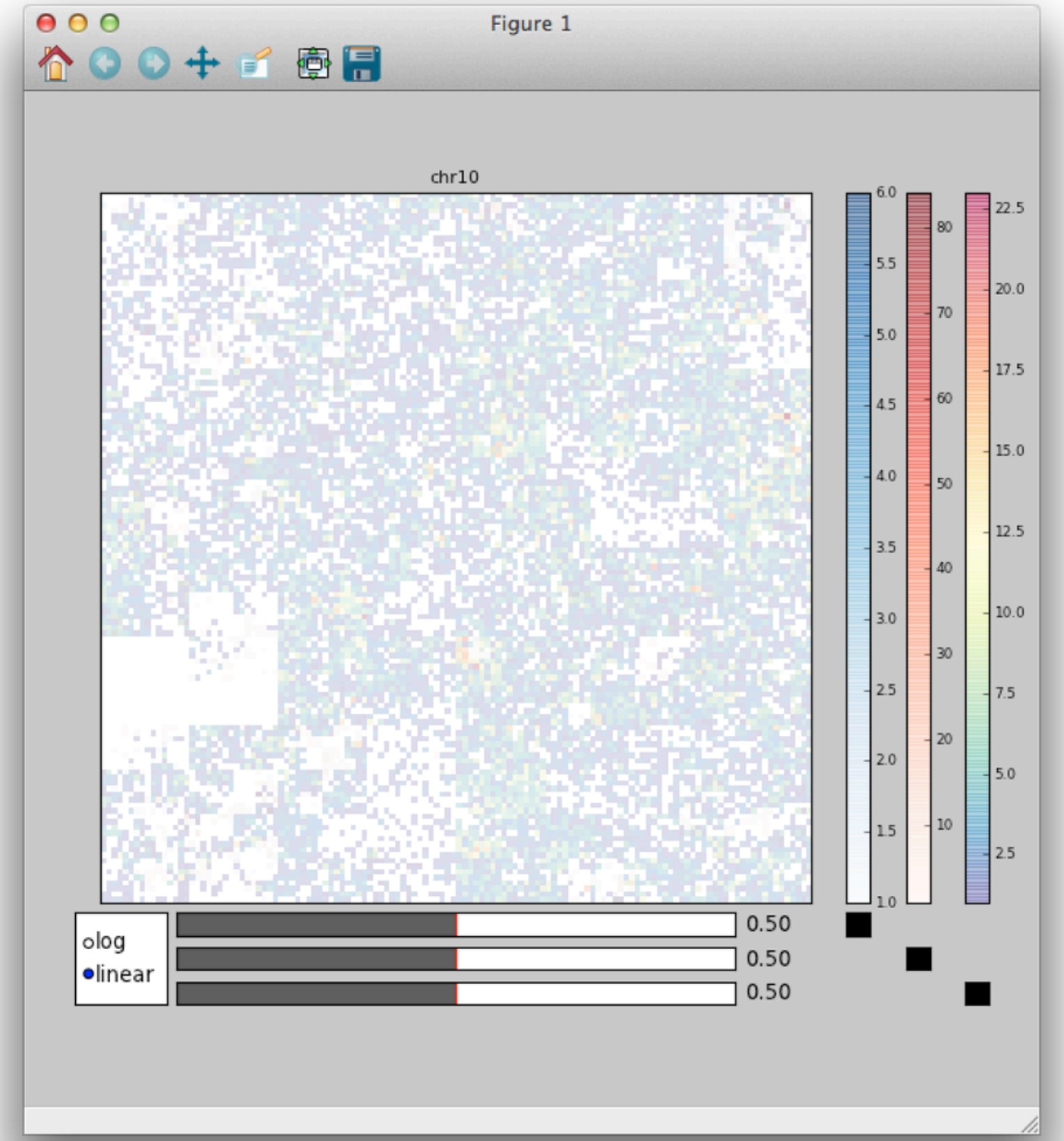


Figure 1
chr10
6.0
5.5
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
80
70
60
50
40
30
20
10
22.5
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
olog
linear
0.50
0.50
0.50

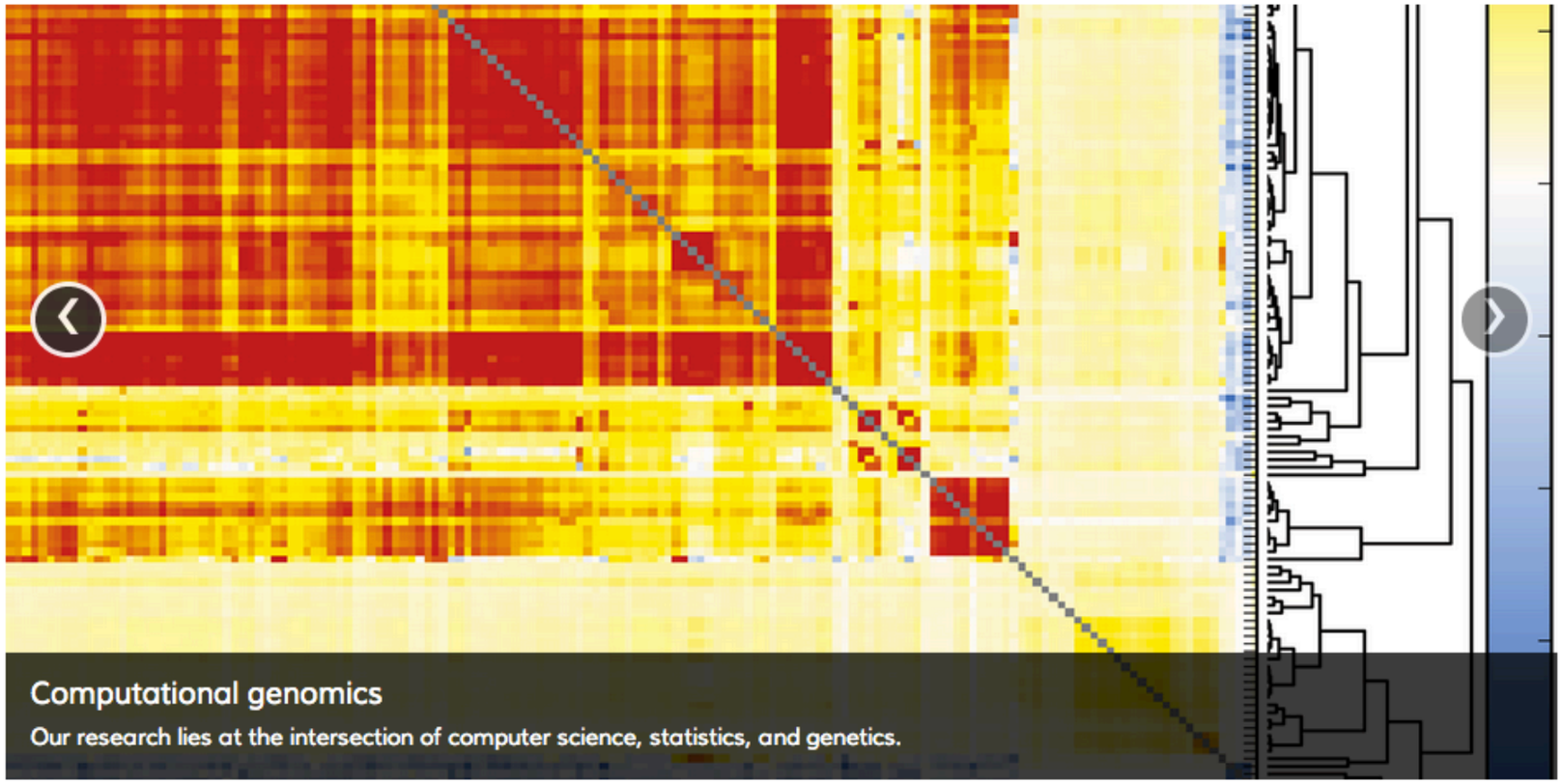
Ideal for comparative analyses:

- Spatial genomic relationships among multiple experiments: ChIP-seq, RNA-seq
- Spatial and quantitative
- Before and after comparisons
- Comparing replicates
- Unbiased data mining

Acknowledgements

Ira Hall	*Univ. of Virginia*
Ankit Malhotra	*Univ. of Virginia*
Michael Lindberg	*Univ. of Virginia*
Royden Clark	*Univ. of Virginia*
Svetlana Sokolova	*Univ. of Virginia*
Mitchell Leibowitz	*Univ. of Virginia*
Pat Concannon	Univ. of Virginia
Steve Rich	Univ. of Virginia
Suna Onengut-Gumuscu	Univ. of Virginia
Chris Moskaluk	Univ. of Virginia
Shu-Man Fu	Univ. of Virginia
Gabor Marth	Boston College
Jim Robinson	Broad Institute
James Taylor	Emory
Nick Navin	MD Anderson
Kristin Baldwin	Scripps
Nik Krumm	*Univ. of Washington*
Evan Eichler	*Univ. of Washington*
Debbie Nickerson	*Univ. of Washington*
Chris Carlson	*Fred Hutchinson CRC*
Mark Rieder	*Univ. of Washington*
Josh Smith	*Univ. of Washington*
Peter Sudmant	*Univ. of Washington*

quinlanlab.org



The Quinlan Lab at UVa.

We are a new computational genomics group in the Center for Public Health Genomics at the University of Virginia. Our research marries genetics and genomics techniques with computer science, machine learning, and engineering to develop new ways of gaining insight into genome biology and the genetic basis of traits. We try to tackle problems with practical importance to understanding genome variation, chromosome evolution and mining genetic variation for improved understanding of human disease. Understanding the genome is a hard problem; we try to develop new approaches to make genomic research easier.

Learn more...

We are grateful to be funded by:

National Human Genome Research Institute

University of Virginia

Mining the genome.

Aaron R. Quinlan

quinlanlab.org

University of Virginia, Charlottesville VA

Center for Public Health Genomics

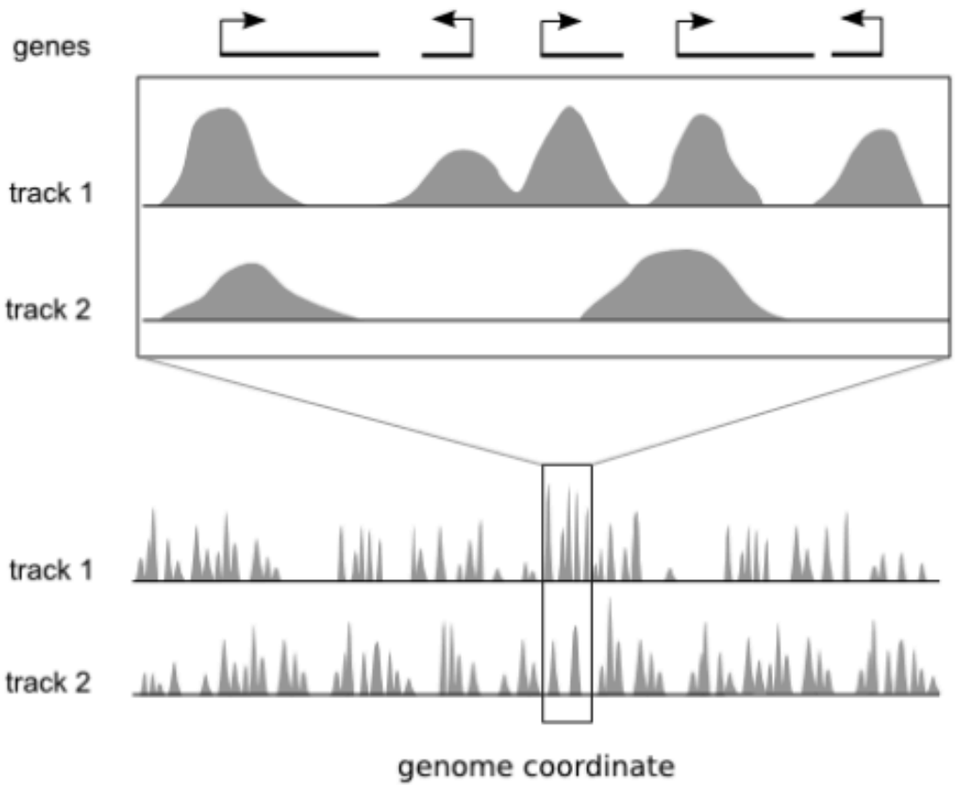
Biochemistry and Molecular Genetics

Spark.

BC Cancer Agency
CARE + RESEARCH
An agency of the Provincial Health Services Authority

CANADA'S MICHAEL SMITH GENOME SCIENCES CENTRE

Motivation



Genome browsers are ideal for viewing local regions of interest.

Many genomics techniques produce measurements that have both a value and a position on a reference genome, for example ChIP-sequencing. Popular genome browsers arrange the linear genome coordinate along the x axis and express the data value range on the y axis. This approach enables integration of diverse data sets by plotting them as vertically stacked tracks across a common genomic x axis. Genome browsers are designed for viewing local regions of interest (e.g. an individual gene) and are frequently used during the initial data inspection and exploration phases.

But they do not provide a global overview of these regions.

Most genome browsers support zooming along the genome coordinate. This type of overview is not always useful because it produces a summary across a continuous genomic range (e.g. chromosome 1) and not across the subset of regions that are of interest (e.g. genes on chromosome 1). There is a need for tools that help answer questions like: "What are the common data patterns across genes start sites in my data set?"

http://www.sparkinsight.org/

Spark.

Step 1: Pre-processing

A Spark analysis begins with two user inputs: (i) one or more data tracks, and (ii) a set of regions of interest. Spark extracts a data matrix for each input region and orients it according to strand. Rows in these matrices correspond to data tracks and columns represent data bins along the genomic x-axis (two bins per region are used the diagram). The values are then normalized between 0 and 1 that can be visualized as heatmaps.

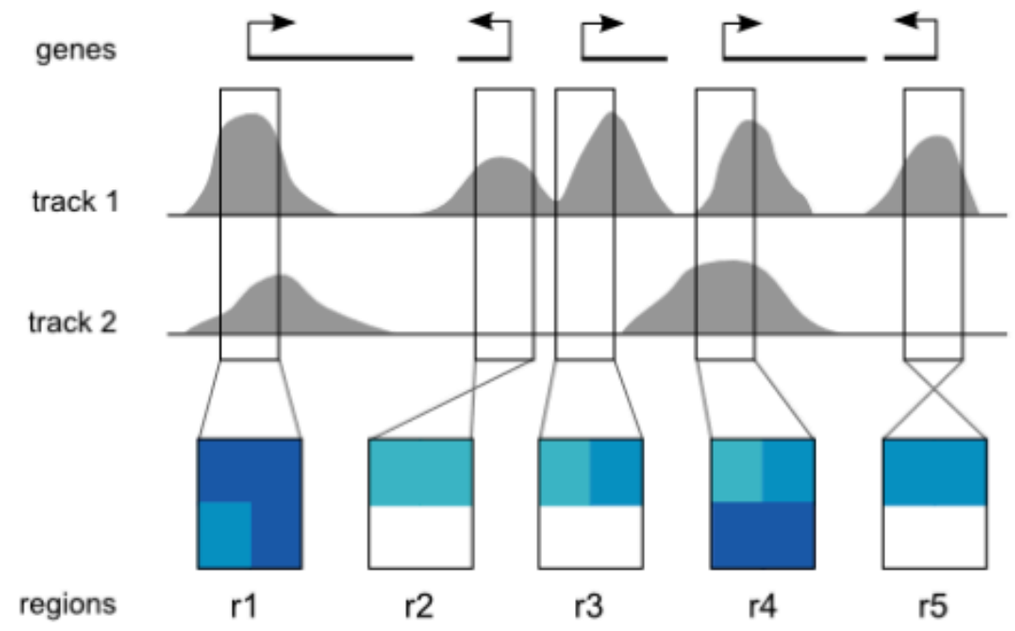
Step 2: Clustering

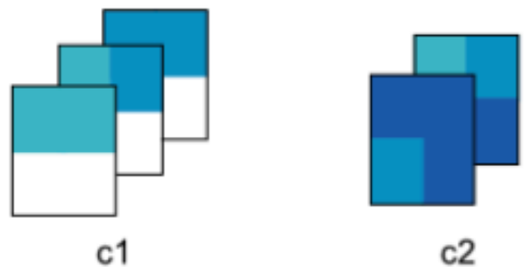
The preprocessed data are then clustered using k-means clustering using a user specified number of clusters (k). This technique was chosen for its effectiveness, relative simplicity, and runtime speed.

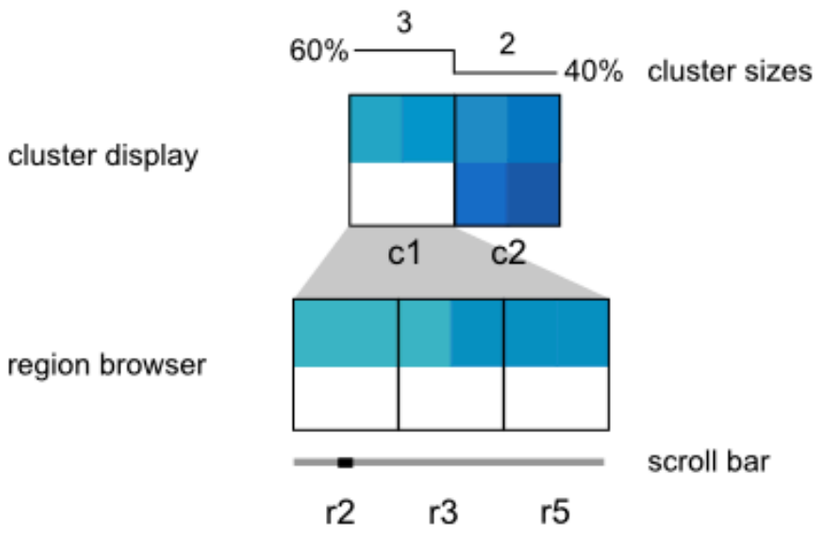
Step 3: Interactive visualization

The visualization is composed of two components: the Cluster Display, which provides a summary of each cluster, and the Region Browser, which displays individual cluster members. The user can therefore see both a high-level picture of the patterns in their data, while also being able to drill-down to individual regions of interest. The low-level region view is supported by links out to the UCSC Genome Browser, and the high-level cluster view supports interactive analyses such as manual cluster refinement and links out to the DAVID gene ontology tool set.

Approach





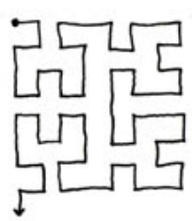


MAP OF THE INTERNET

THE IPv4 SPACE, 2006

THIS CHART SHOWS THE IP ADDRESS SPACE ON A PLANE USING A FRACTAL MAPPING WHICH PRESERVES GROUPING -- ANY CONSECUTIVE STRING OF IPs WILL TRANSLATE TO A SINGLE COMPACT, CONTIGUOUS REGION ON THE MAP. EACH OF THE 256 NUMBERED BLOCKS REPRESENTS ONE /8 SUBNET (CONTAINING ALL IPs THAT START WITH THAT NUMBER). THE UPPER LEFT SECTION SHOWS THE BLOCKS SOLD DIRECTLY TO CORPORATIONS AND GOVERNMENTS IN THE 1990's BEFORE THE RIRs TOOK OVER ALLOCATION.

0 1 14 15 16 19 →
3 2 13 12 17 18
4 7 8 11
5 6 9 10



= UNALLOCATED BLOCK

http://xkcd.com/195/