BIO392 genomic variation file formats and tools, 2022 Exercises

izaskun.mallona@mls.uzh.ch

Learning objectives

1. Understand the standard NGS -> QC -> mapping -> QC -> variant calling flow
2. Recognize a handful of the standard file formats (FASTQ, SAM, GTF, BED, VCF)
3. Be aware of the caveats of such formats (e.g. coordinate systems, 0/1 counting)
4. Manipulate standard file formats using UNIX tools
5. Run and interpret results: genomic analysis of human genetic variation due to mobile elements insertions

Schedule

1. Thursday 22nd Sept
   - Lecture on Unix, data generation, data formats, and data handling
     - Introduction to the commandline
     - Sequencing, alignment, coordinate files, variant files
   - Hands-on session on UNIX and file handling
     - Working with the terminal
     - Data streams
     - Basics of awk, sed, cut
   - Hands-on session on data formats
     - FASTA, FASTQ, SAM, GTF, BED, VCF
2. Friday 23rd Sept
   - Hands-on session on data formats (cont)
   - Small project regarding human genetic variation and interpretation
     - Retrotransposition events in 1000 genomes data

Today's structure

1. Slides
    - Software management
    - Renku
    - Genome arithmetics (bedtools suite)
    - Human variation arising from mobile elements
    - ChromHMM
2. Activities
    - Finish the SIB UNIX course
    - Run the exercises

Reproducibility for software

- In bioinformatics, software versionin needs to be tracked
- Installs are run command-line, specifying versions and paths
- Handled through:
    - Package managers (apt, yum)
    - Package + environoment managers (conda)
    - Binary download
    - Compilation from source code
    - (Wrap everything into containers, e.g. docker/singularity)

Software installs

```
cd soft/kent

## downloads a pre-compilled binary
curl http://hgdownload.soe.ucsc.edu/admin/exe/macOSX.x86_64/bedToBigBed \
   > bedToBigBed

## adding exec permissions to the binary
chmod a+x bedToBigBed

## running the actual command
./bedToBigBed example.bed hg19.genome out.bb
```

Software compilation

- Translate source code to executables (or other artifacts)
- Optimize software execution for the host OS
- Frequently, orchestrated through Makefiles (run using `make`)
- Extra read https://en.wikipedia.org/wiki/Makefile

Compiling bedtools

```
cd
cd soft

curl -L https://github.com/arq5x/bedtools2/releases/download/v2.25.0/bedtools-2.2
  > bedtools-2.25.0.tar.gz

tar zxvf bedtools-2.25.0.tar.gz
cd bedtools2
make
## why do we set an alias? Discuss $PATH and root installs
alias bedtools='./bin/bedtools'
```

The PATH variable

- PATH (`echo $PATH`) is an environmental variable
- the PATH indicates which directories to search for executable files
- /usr/bin, /usr/local/bin etc are normally owned by root
- often circumvented using `aliased` commands: `alias R41='/home/imallona/src/R4.1-bC3.13/bin/R'`

Tricky parts of the exercises

- https://github.com/compbiozurich/UZH-BIO392/blob/master/course-material/2022/imallona/exercises.md#executable-software-download Defining the OS: Mac, Linux (includes renku)
- Compilation might fail in our local iMacs. Alternatives
  - Use the system's bedtools and vcftools (installed by Tina)
  - Switch to renku

Alternative for running the exercises: renku

- https://renkulab.io/projects/izaskun.mallona/uzh-bio392
- Please run on 0.5 cpus, 1 GB RAM, 1 GB disk space
- Free computing environment generously (freely) provided by SDSC, even without login
- Slides (Rok Roskar and team) https://www.openscience-summerschool.uzh.ch/dam/jcr:b0890e9e-6fb6-4f23-a952-54acb899e41c/renku-OS-summer-school.pdf

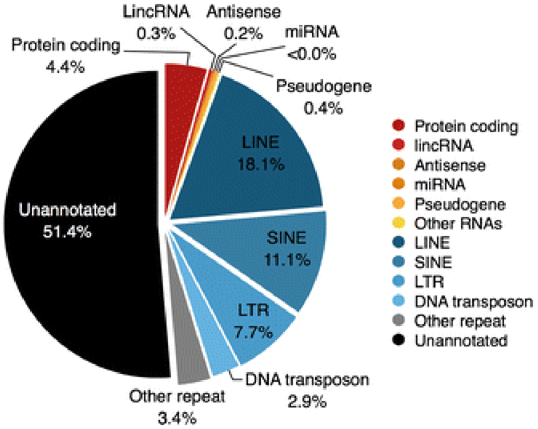
Using renku

- The renku platform interface
  https://renkulab.io/projects/izaskun.mallona/uzh-bio392
- The gitlab interface
- The makings of a repo (files, check the Dockerfile)
- Starting an instance and selecting a graphical user interface
  - rstudio
  - jupyter nb

Exercise 33 and following

- We'll explore mobile elements insertions as a source of human genetic variation
- Data integration: 1000 Genomes, ENCODE's Epigenomics data, gene annotations (exon locations)

Genomes are often repeat-laden



Human genome (Hutchins and Pei, 2015)

- CDS (coding sequences) are a small part of the human genome
- Repeats are abundant

Repeat types (Richard...Dujon, 2008)

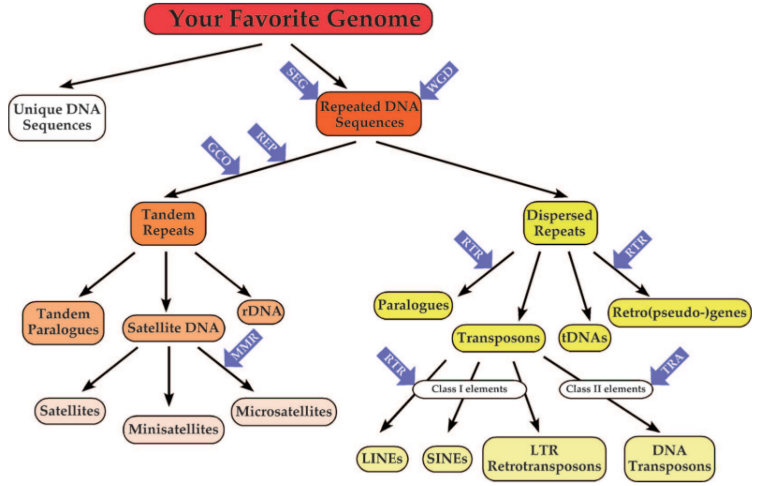
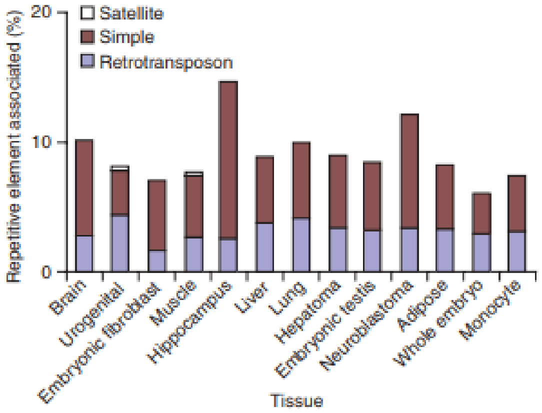


FIG. 1. Repeated DNA sequences in eukaryotic genomes and mechanisms of evolution. The two main categories of repeated elements (tandem repeats and dispersed repeats) are shown, along with subcategories, as described in the text. Blue arrows point to molecular mechanisms that are involved in propagation and evolution of repeated sequences. REP, replication slippage; GCO, gene conversion; WGD, whole-genome duplication; SEG, segmental duplications; RTR, reverse transcription; TRA, transposition.

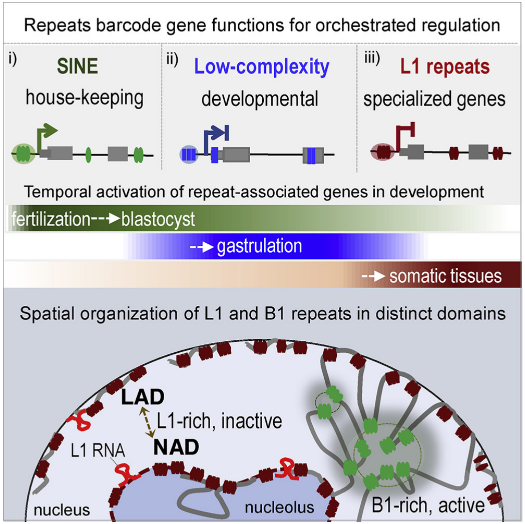
Are repetitive elements expressed (bulk analysis)?



Y axis: percentage of CAGE tags associated to repeats

- In CAGE experiments in bulk (PHANTOM), ≈ 10% of the CAGE tags map to repetitive elements (in mouse)
- Similarly in human (not shown here)
- (Faulkner...Carninci, 2009)

Why so many repetitive elements? They play a role in genome architecture

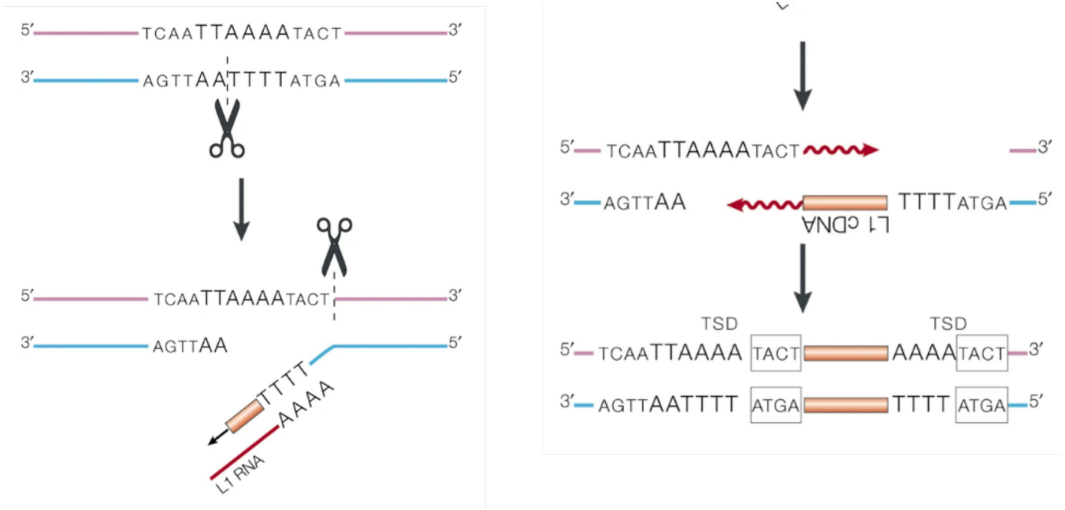

Repeats barcode gene functions for orchestrated regulation
i) SINE
house-keeping
ii) Low-complexity
developmental
iii) L1 repeats
specialized genes
Temporal activation of repeat-associated genes in development
fertilization--➔blastocyst
--➔ gastrulation
--➔ somatic tissues
Spatial organization of L1 and B1 repeats in distinct domains
LAD
L1-rich, inactive
NAD
L1 RNA
nucleus
nucleolus
B1-rich, active

- Repeats barcode genes with distinct functions
- Repeats orchestrate gene expression timing during development
- (Lu..Shen, 2020)

Mobile elements self-sufficiency

- Transposable elements can be classified into:
  - By nature: DNA transposons or RNA (retro) transposons
  - By self-sufficiency: autonomous (capable of self-mobilization), or passenger-like
- Around half of the human genome is composed by immobile transposable sequences
- 40-60 copies of the L1 retrotransposon are active (Luning Prak and Kazazian Jr, 2020)
- L1s can jump autonommously, and they can mobilize other passengers (e.g. some SINEs)

L1 insertion mechanism (AT rich)



(Luning Prak and Kazazian Jr, 2020)

Mobile elements as a source of variation

- Repeats insertion into key genes (e.g. APC gene in colon cancer) can drive disease
  - Somatic insertion: tumor has the insertion, adjacent normal doesn't
- (Also, some cancers show repeats deregulation (low DNA methylation))
- Novel insertion might play a role in brain development
  - *massive* LINE-1 associated to neuron differentiation (Muotri et al., 2010)

Mobile elements sum-up

- Abundant elements of the genome (=junk DNA)
- Regulatory activities
- If active:
  - Insertions in disease (i.e. cancer)
  - Insertions in healthy individuals (1000 Genomes, we'll use their VCF today)

The mobile insertions data from 1000 genomes

- Data source http://ftp.1000genomes.ebi.ac.uk/vol1/ftp/pilot_data/paper_data_sets/a_map_of_human_variation/low_coverage/sv/
- Extra reading/paper https://journals.plos.org/plosgenetics/article?id=10.1371/journal.pgen.1002236
- Quick overview on populations: CEU, YRI etc at HapMap

First round of questions:

- Which family of mobile elements has the most insertions?
- How many insertions are there in exons? what do we expect?
- How can we quantify those?

First round of questions:

- Which family of mobile elements has the most insertions?
- How many insertions are there in exons? what do we expect?
- How can we quantify those?
    - Bedtools intersect on exons? documentation

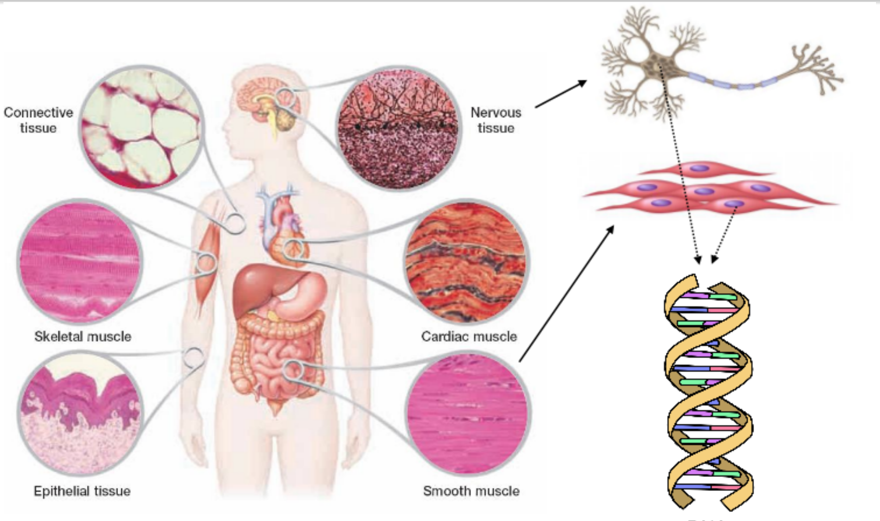
Other approaches to the data

- Given the low abundance in exons, where do they jump?
- Promoters? enhancers? heterochromatin? other repetitive elements?
- What can we use as a reference, how do we assign ‘functions’ to all nucleotides in a genome?

Regulatory genomics and epigenomics

- We'd benefit from integrating the variation data with regulatory genomics (Epigenomics) layers
- Epigenomics data: mostly, histone marks and DNA methylation

Regulatory diversity: same DNA, different phenotypes

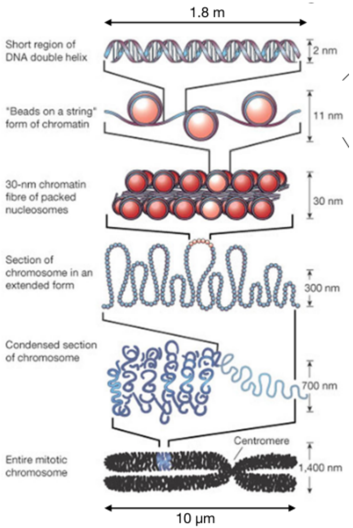


~200 cell types with different properties

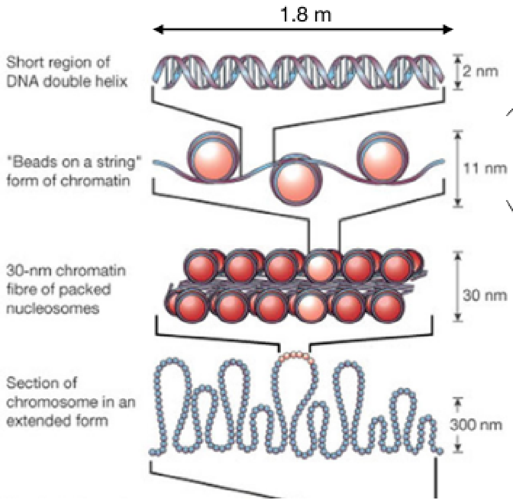
same genetic material

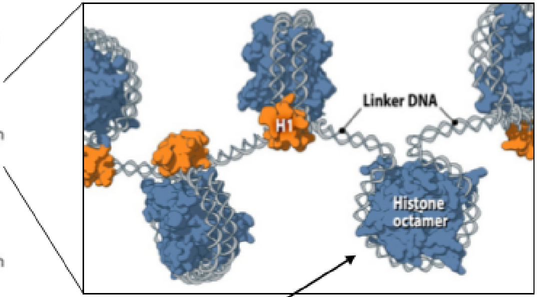
DNA folding levels

- Genome has a 3D organization inside the nucleus: folding 1.8 m of DNA vs nucleus diameter of 4-6 microns
- DNA folding (chromatin accessibility) influences binding of transcription factors to DNA
- With folding, what makes an enhacer close to a gene is not (only) being next in DNA sequence


1.8 m
Short region of DNA double helix
2 nm
"Beads on a string" form of chromatin
11 nm
30-nm chromatin fibre of packed nucleosomes
30 nm
Section of chromosome in an extended form
300 nm
Condensed section of chromosome
700 nm
Centromere
Entire mitotic chromosome
1,400 nm
10 µm

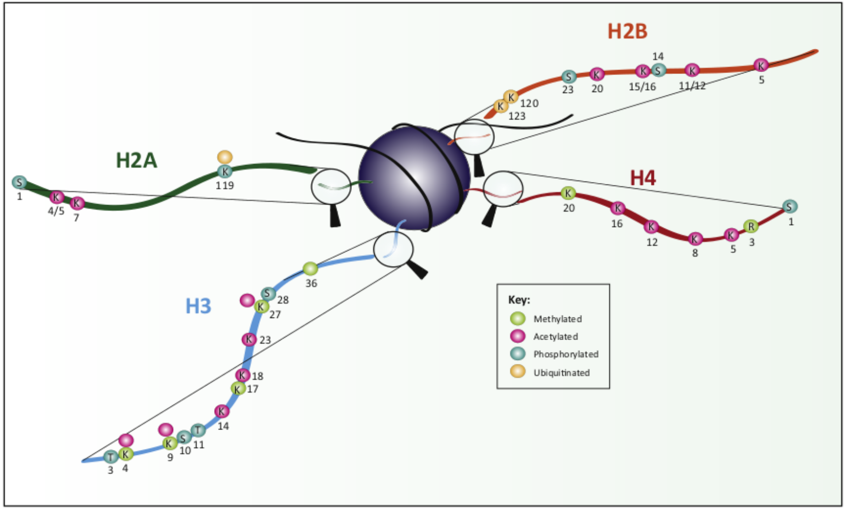
Nucleosomes 3D


1.8 m
Short region of DNA double helix
2 nm
"Beads on a string" form of chromatin
11 nm
30-nm chromatin fibre of packed nucleosomes
30 nm
Section of chromosome in an extended form
300 nm



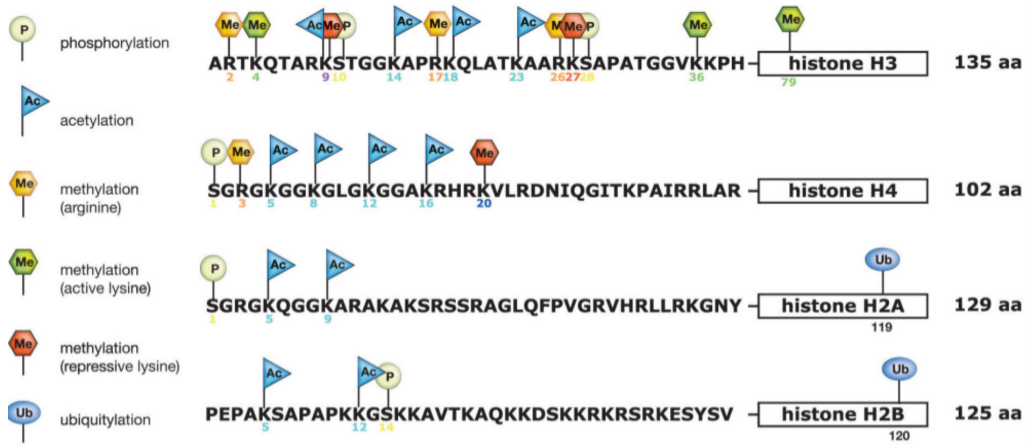
Nucleosome: basic subunit of chromatin
- contains DNA and histones (proteins)

Histone tails



(Lawrence, Daujat, Schneider 2015)

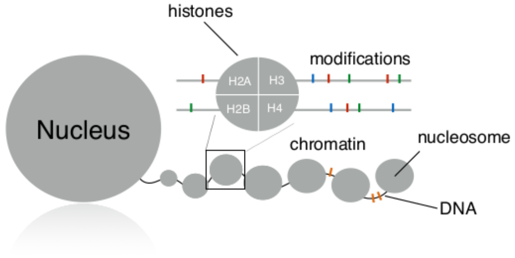
Chromatin compaction and modifications: histones


P
phosphorylation
Ac
acetylation
Me
methylation (arginine)
Me
methylation (active lysine)
Me
methylation (repressive lysine)
Ub
ubiquitylation
ARTKQTARKSTGGKAPRKQLATKAARKSAPATGGVKKPH
histone H3
135 aa
2 4 9 10 14 17 18 23 26 27 28 36 79
SGRGKGGKGLGKGGAKRHRKVLRDNIQGITKPAIRRLAR
histone H4
102 aa
1 3 5 8 12 16 20
SGRGKQGGKARAKAKSRSSRAGLQFPVGRVHRLLRKGNY
histone H2A
129 aa
1 5 9 119
PEPAKSAPAPKKGSKKAVTKAQKKDSKKRKRSRKESYSV
histone H2B
125 aa
5 12 14 120

Chromatin compaction and modifications: histones

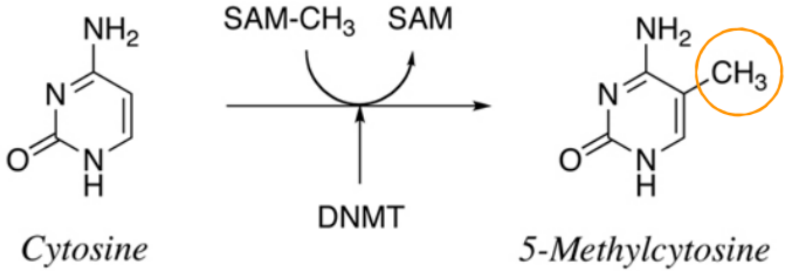
- Naming: Histone name - aminoacid number - decoration
- e.g. H3K4me3: histone 3, lysine 4, trimethyl (has 3 methyl groups)

Chromatin compaction and modifications: DNA



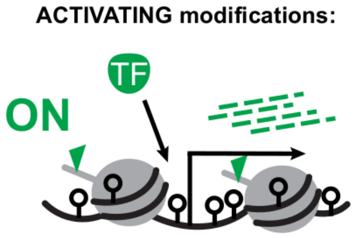
on histones:
- majority on histone H3 and H4 N-terminal tail
- depending on the type of modification and AA residue: activating or repressing

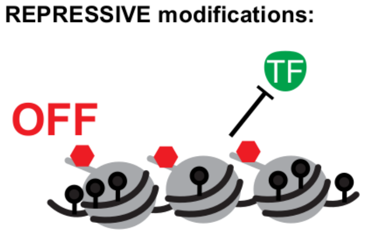
on DNA:
- methylation of cytosines
- repressive mark



Chromatin compaction impact in regulation

- Chromatin packaging and modifications influence the accessibility and transcriptional output of the genome
- How?

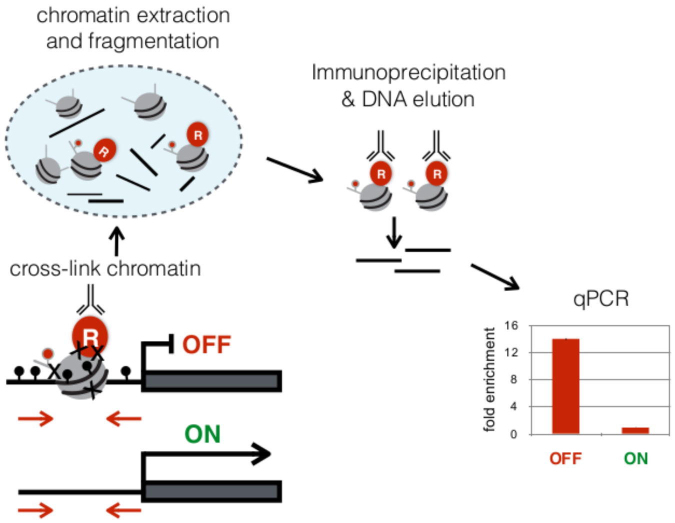




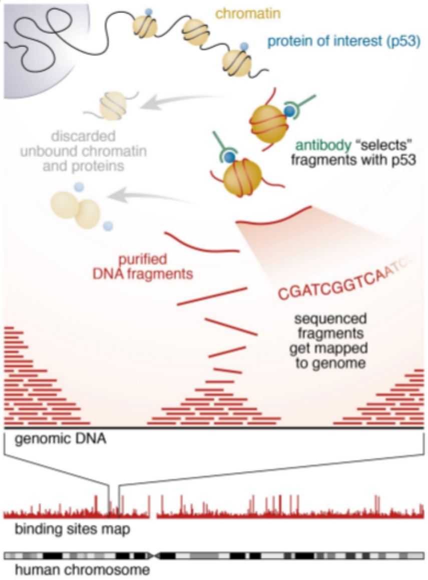
Histone acetylation
H3K4me3
no DNA methylation

H3K27me3 (Polycomb)
H3K9me3
DNA methylation

Localisation analysis by ChIP

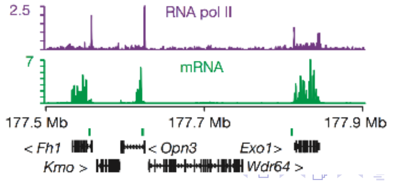

chromatin extraction and fragmentation
Immunoprecipitation & DNA elution
cross-link chromatin
R
OFF
ON
qPCR
fold enrichment
16
12
8
4
0
OFF
ON

Localisation analysis by ChIP-Seq


chromatin
protein of interest (p53)
discarded unbound chromatin and proteins
antibody "selects" fragments with p53
purified DNA fragments
CGATCGGTCAATC
sequenced fragments get mapped to genome
genomic DNA
binding sites map
human chromosome

Chromatin Immunoprecipitation - sequencing
method to identify genomic location of protein of interest (e.g. TF, RNA Pol2) or histone modifications

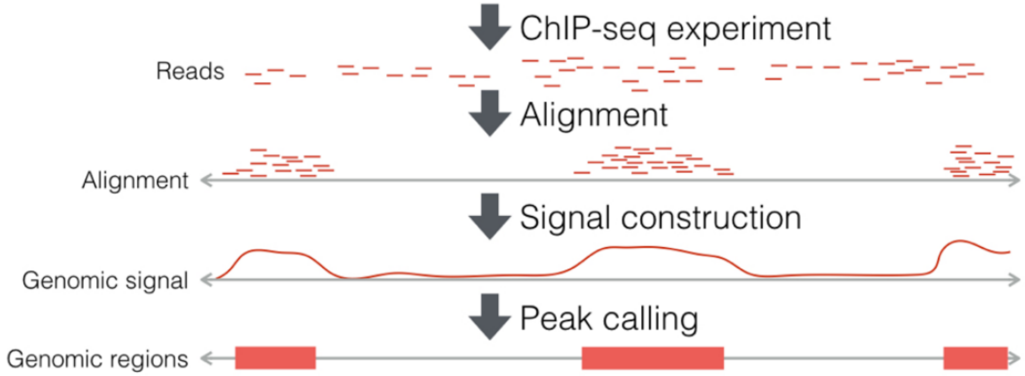
1. proteins are fixed to chromatin by formaldehyde (crosslinking)
2. chromatin is sheared to 100-300bp (ultrasound or enzymes)
3. specific antibodies enrich pieces of DNA bound by protein of interest
4. enriched DNA is purified and sequenced
5. usually 20-100 mio sequences are obtained from one experiment = "reads"
6. sequences reads are "mapped" back to the genome to identify their position along the chromosome
7. signal intensity indicates localisation frequency



ChIP-Seq data flow

- High-throughput short read sequencing
- Read quality control
- Mapping the sequences back to the reference genome
- Mapping quality control
- Summarization into coordinate-based files

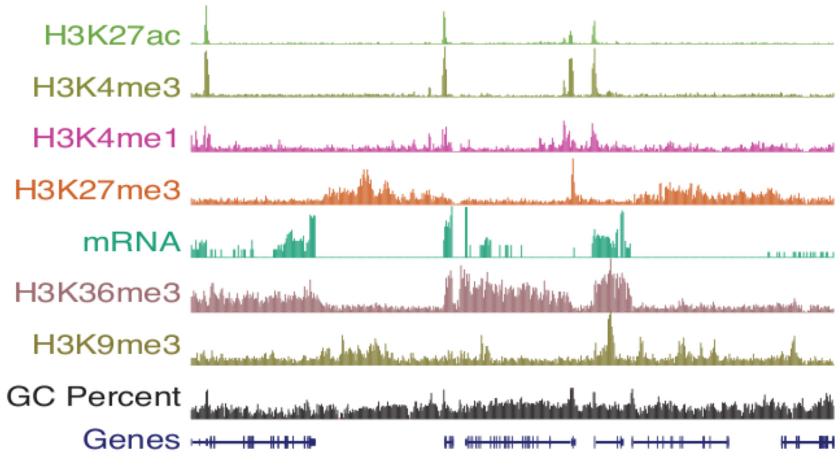
ChIP-Seq data flow



regulatory-genomics.org

Visual inspection of the data

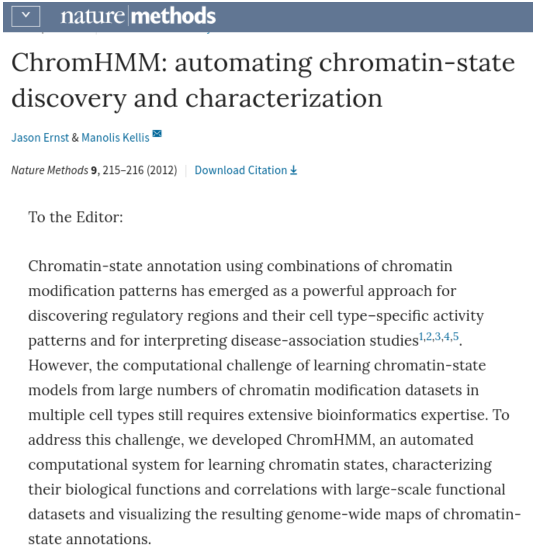
visual inspection (seq. counts along chromosome)



Machine learning

- Can we learn chromatin states and correlate with regulatory functions using machine learning methods?
- Which data can we feed into them?

Machine learning: genome segmentation using ChromHMM



nature methods

ChromHMM: automating chromatin-state discovery and characterization

Jason Ernst & Manolis Kellis

Nature Methods **9**, 215–216 (2012) | Download Citation

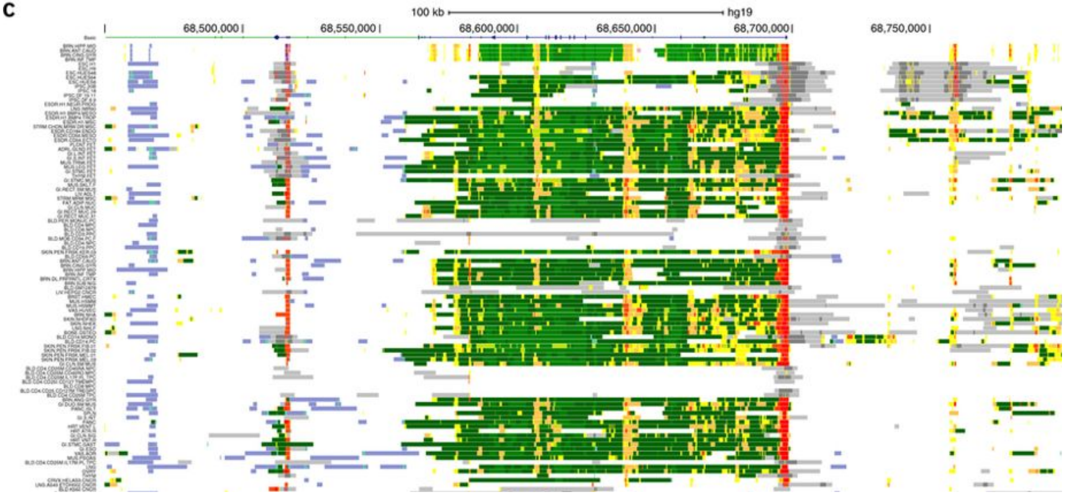
To the Editor:

Chromatin-state annotation using combinations of chromatin modification patterns has emerged as a powerful approach for discovering regulatory regions and their cell type–specific activity patterns and for interpreting disease-association studies[1,2,3,4,5]. However, the computational challenge of learning chromatin-state models from large numbers of chromatin modification datasets in multiple cell types still requires extensive bioinformatics expertise. To address this challenge, we developed ChromHMM, an automated computational system for learning chromatin states, characterizing their biological functions and correlations with large-scale functional datasets and visualizing the resulting genome-wide maps of chromatin-state annotations.

- ChromHMM: released in 2012
- Still widely used: e.g. latest 2020 ENCODE's release preprint, and resource

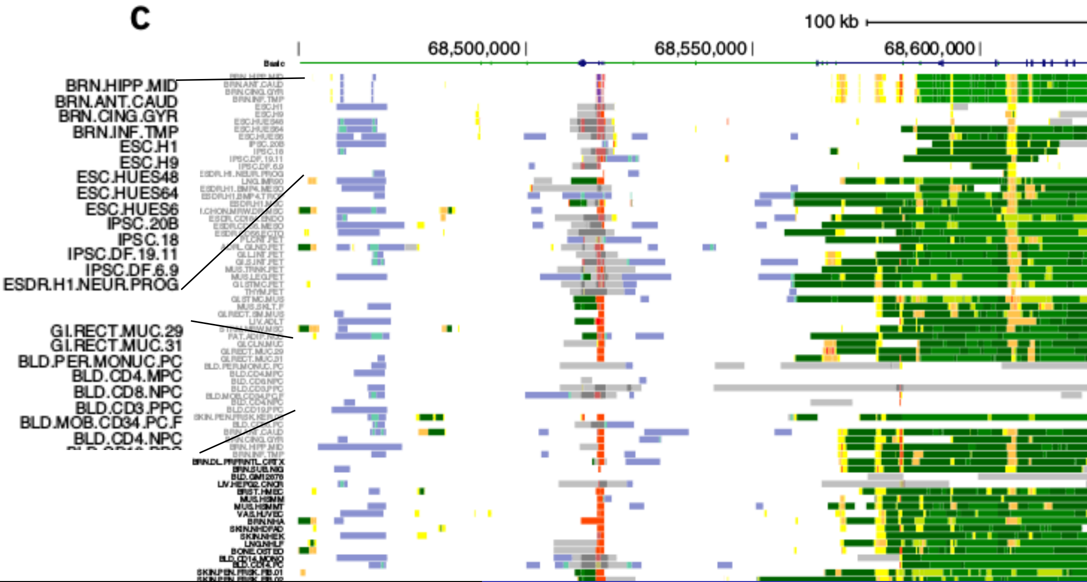
Cell types segmentation by ChromHMM

c

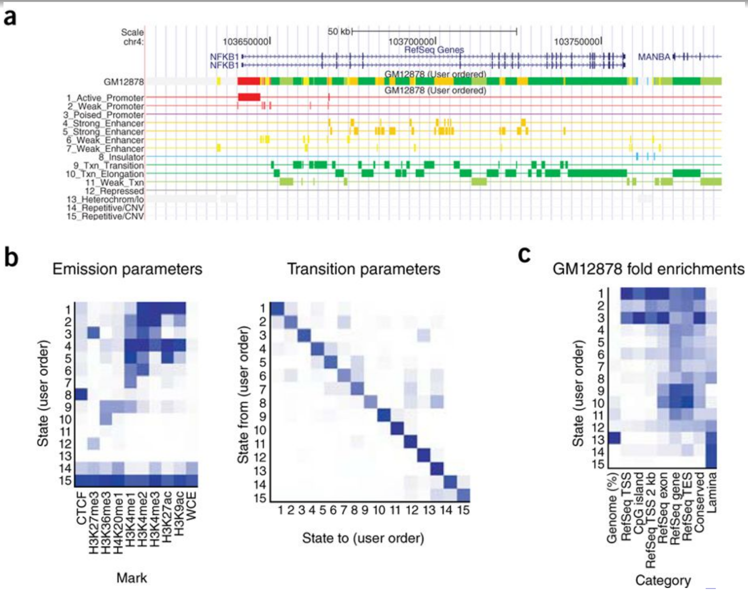


- Rows: cell. Darkgreen, transcription; red, TSS; blue, heterochromatin; yellow-green, enhancer; gray, repressed.

Zooming in the regulatory genome (segmentation by ChromHMM)

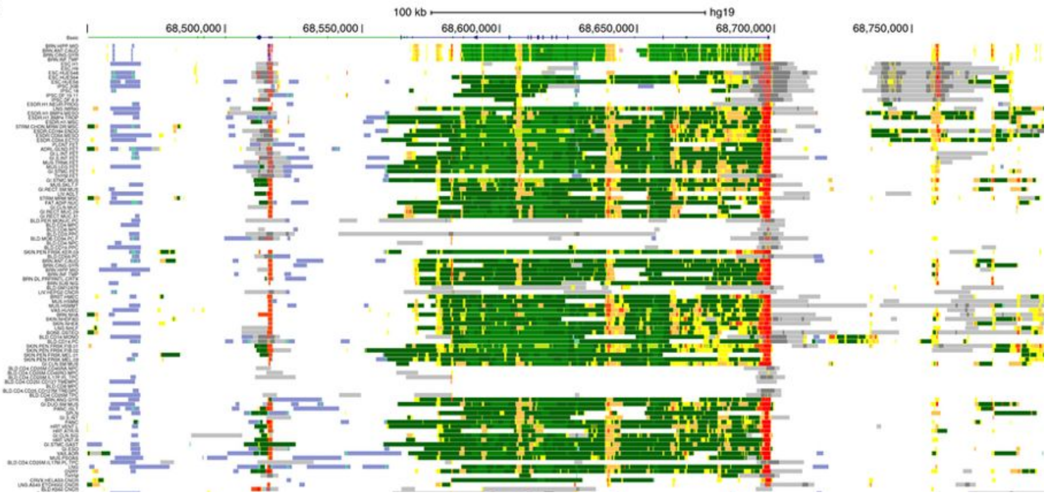


Machine learning: genome segmentation using ChromHMM



Cell types segmentation by ChromHMM

c



- Rows: cell. Darkgreen, transcription; red, TSS; blue, heterochromatin; yellow-green, enhancer; gray, repressed.

chromHMM segmentations (BED3 + 1 column)

```
$ head hesc.chromHmm.bed
chr1 10000 10600 15_Repetitive/CNV
chr1 10600 11137 13_Heterochrom/lo
chr1 11137 11537 8_Insulator
chr1 11537 11937 11_Weak_Txn
chr1 11937 12137 14_Repetitive/CNV
chr1 12137 14137 11_Weak_Txn
chr1 14137 27537 9_Txn_Transition
chr1 27537 27737 6_Weak_Enhancer
chr1 27737 28537 2_Weak_Promoter
chr1 28537 30137 1_Active_Promoter
```

Activity

- Integrate chromHMM segmentations and mobile element insertion points to retrieve where do they ocurr: promoters? enhancers? expressed parts of the genome?
- Toolset: bedtools intersect

Caveats/challenges

- This was a prototype/pilot exercise, full of weaknesses/caveats
- Criticisms?

Caveats/challenges

- This was a prototype/pilot exercise, full of weaknesses/caveats
- Criticisms?
- Are we taking into account genome assemblies when integrating data layers? Shall we liftOver some files?
- Do we trust the VCF file? Are mobile elements insertions easy to detect? Are people chimeras (genetic mosaics)? Does this pose a challenge when running variant detection (on blood)?
- (other criticisms you might have)

Caveats/challenges II

- Where does ‘exons.bed‘ come from?! From which GTF? from which assembly? Is using a file named ‘exons.bed‘ from somewhere a good practice?
- Why are we using a stem cell line (H1) chromHMM segmentations? Is this representative of the chromatin in an embryonic tissue (where mobile elements insertion happen)?
- What does it mean the variants were from the pilot/low coverage? Can we trust them?
- Why did we use CEU/European variants?
- To compare insertions vs background, are we counting BED6 records (lines) and not total sequence breadths (nt)!? does this make sense? how to fix it?