

# Real world analyses with bedtools.

**Applied Computational Genomics, Lecture 18**

<https://github.com/quinlan-lab/applied-computational-genomics>

**Aaron Quinlan**

**Departments of Human Genetics and Biomedical Informatics**

**USTAR Center for Genetic Discovery**

**University of Utah**

**quinlanlab.org**

Let's use bedtools to answer some real(ish) research questions.

Q1. Which enhancers in human embryonic stem cells are under the most genetic "constraint" (that is, have the least genetic variation)?

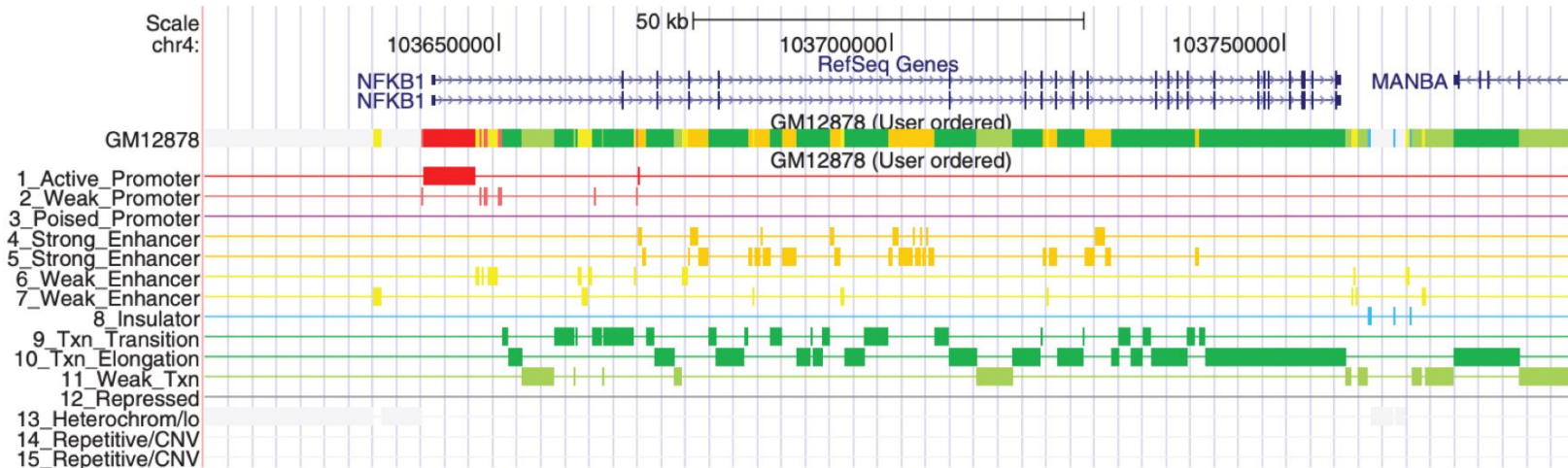
Q1. Which enhancers in human embryonic stem cells are under the most genetic "constraint" (that is, have the least genetic variation)?

Experimental design: what materials do we need?

1. We need a set of predicted enhancers in hESCs. ENCODE?
2. We need a deep catalog of genetic variation in the human genome. 1000 genomes?
3. We then need to count the number of genetic variants observed in each enhancer.

# chromHMM: <https://www.nature.com/articles/nmeth.1906.pdf>

a



Q1. Which enhancers in human embryonic stem cells are under the most genetic "constraint" (that is, have the least genetic variation)?

1. We need a set of predicted enhancers in hESCs. Let's focus on chr22 for simplicity.

```
$ wget https://s3.amazonaws.com/bedtools-tutorials/web/hesc.chromHmm.bed
```

```
$ grep "^chr22" hesc.chromHmm.bed > hesc.chromHmm.chr22.bed
```

```
$ head -n 5 hesc.chromHmm.chr22.bed
```

```
chr22 16050000 16075600 13_Heterochrom/lo
chr22 16075600 16076000 8_Insulator
chr22 16076000 16084200 13_Heterochrom/lo
chr22 16084200 16084600 8_Insulator
chr22 16084600 16156800 13_Heterochrom/lo
```

Not at all enhancers.



Q1. Which enhancers in human embryonic stem cells are under the most genetic "constraint" (that is, have the least genetic variation)?

1. We need a set of predicted enhancers in hESCs. How are the enhancers labeled?

```
$ cut -f 4 hesc.chromHmm.chr22.bed | sort | uniq -c
```

```
444 10_Txn_Elongation
2875 11_Weak_Txn
514 12_Repressed
1213 13_Heterochrom/lo
58 14_Repetitive/CNV
29 15_Repetitive/CNV
251 1_Active_Promoter
671 2_Weak_Promoter
297 3_Poised_Promoter
149 4_Strong_Enhancer
335 5_Strong_Enhancer
1656 6_Weak_Enhancer
2840 7_Weak_Enhancer
1218 8_Insulator
493 9_Txn_Transition
```

Let's focus on the strong enhancers



Q1. Which enhancers in human embryonic stem cells are under the most genetic "constraint" (that is, have the least genetic variation)?

1. We need a set of predicted enhancers in hESCs.

```
$ grep "Strong_Enhancer" hesc.chromHmm.chr22.bed > hesc.chromHmm.chr22.strong.enh.bed
```

```
# sanity check
```

```
$ wc -l hesc.chromHmm.chr22.strong.enh.bed  
484
```





Q1. Which enhancers in human embryonic stem cells are under the most genetic "constraint" (that is, have the least genetic variation)?

2. We need a deep catalog of genetic variation in the human genome. 1000 genomes

```
# warning. 1.8 gigabytes. Will take a few minutes to download from 1000G FTP site
```

```
$ wget
```

```
ftp://ftp-trace.ncbi.nlm.nih.gov/1000genomes/ftp/release/20130502/ALL.wgs.phase3\_shapeit2\_mvncall\_integrated\_v5b.20130502.sites.vcf.gz
```

```
# download the tabix index of the VCF file.
```

```
$ wget
```

```
ftp://ftp-trace.ncbi.nlm.nih.gov/1000genomes/ftp/release/20130502/ALL.wgs.phase3_shapeit2_mvncall_integrated_v5b.20130502.sites.vcf.gz.tbi
```

```
# extract just the genetic variants for chromosome 22 with tabix
```

```
$ tabix -h ALL.wgs.phase3_shapeit2_mvncall_integrated_v5b.20130502.sites.vcf.gz 22 \  
> 1000g.chr22.vcf
```

Extract just chromosome 22 variants

Retain the VCF header in the output with -h

Q1. Which enhancers in human embryonic stem cells are under the most genetic "constraint" (that is, have the least genetic variation)?

3. We then need to count the number of genetic variants observed in each enhancer.

Well, now we have our enhancer file (`hesc.chromHmm.chr22.enh.bed`) and our variants file (`1000g.chr22.vcf`). How do we count the number of variants in each enhancer?

```
$ bedtools intersect -a hesc.chromHmm.chr22.strong.enh.bed -b 1000g.chr22.vcf -c | head
```

```
***** WARNING: File hesc.chromHmm.chr22.strong.enh.bed has inconsistent naming convention for record:
```

```
chr22 17675000 17675600 5_Strong_Enhancer
```



Q1. Which enhancers in human embryonic stem cells are under the most genetic "constraint" (that is, have the least genetic variation)?

3. We then need to count the number of genetic variants observed in each enhancer.

*What happened?*

```
$ head -n 5 hesc.chromHmm.chr22.strong.enh.bed
chr22    17675000 17675600 5_Strong_Enhancer
chr22    17679200 17679600 4_Strong_Enhancer
chr22    17680000 17681000 4_Strong_Enhancer
chr22    17681000 17681400 5_Strong_Enhancer
chr22 ← 17714800 17717000 5_Strong_Enhancer
```


There is greatness in the world, but there is also this.

```
$ grep -v "^#" 1000g.chr22.vcf | head -n 5 | cut -f 1-6
22    16050075 rs587697622    A    G    100
22 ← 16050115 rs587755077    G    A    100
22    16050213 rs587654921    C    T    100
22    16050319 rs587712275    C    T    100
22    16050527 rs587769434    C    A    100
```

Q1. Which enhancers in human embryonic stem cells are under the most genetic "constraint" (that is, have the least genetic variation)?

3. We then need to count the number of genetic variants observed in each enhancer. Now we need to make the chromosome labels the same. Let's remove "chr" from the BED file with `sed`

```
$ sed -e 's/^chr//' hesc.chromHm.chr22.strong.enh.bed > hesc.chromHm.chr22.enh.strong.nochr.bed
```

- 
1. Give `sed` (stream editor) an expression with `-e`
  2. The "s" operator is to switch one pattern with another. In this case, we switch "chr" at the beginning of each line (^) with nothing "//". That is, remove it.

Q1. Which enhancers in human embryonic stem cells are under the most genetic "constraint" (that is, have the least genetic variation)?

3. We then need to count the number of genetic variants observed in each enhancer.

*Now we are ready for prime time!*

```
$ bedtools intersect -a hesc.chromHmm.chr22.strong.nochr.enh.bed -b 1000g.chr22.vcf -c  
| head
```

22	17675000	17675600	5_Strong_Enhancer	25
22	17679200	17679600	4_Strong_Enhancer	16
22	17680000	17681000	4_Strong_Enhancer	41
22	17681000	17681400	5_Strong_Enhancer	11
22	17714800	17717000	5_Strong_Enhancer	60
22	17737800	17738000	5_Strong_Enhancer	8
22	17738000	17739200	4_Strong_Enhancer	50
22	17739200	17740000	5_Strong_Enhancer	40
22	17741600	17742400	4_Strong_Enhancer	29
22	17744800	17745000	5_Strong_Enhancer	2

200bp versus 2200bp

Oooooooh, look. Constraint!!!!

Wait a second...

Q1. Which enhancers in human embryonic stem cells are under the most genetic "constraint" (that is, have the least genetic variation)?

3. We then need to count the number of genetic variants observed in each enhancer. We need to compute the density of genetic variation, not the count.

```
$ bedtools intersect -a hesc.chromHmm.chr22.enh.strong.nochr.bed -b 1000g.chr22.vcf -c \  
  | awk -v OFS="\t" '{print $0, $5 / ($3-$2)}' \  
> hesc.chromHmm.chr22.enh.nochr.strong.vardensity.bedgraph
```

Density is variant count (column 5) divided by the length of the region (end - start)

Q1. Which enhancers in human embryonic stem cells are under the most genetic "constraint" (that is, have the least genetic variation)?

3. We then need to count the number of genetic variants observed in each enhancer. We need to compute the density of genetic variation, not the count.

```
$ head hesc.chromHmm.chr22.enh.nochr.vardensity.bedgraph
22 17675000 17675600 5_Strong_Enhancer 25 0.0416667
22 17679200 17679600 4_Strong_Enhancer 16 0.04
22 17680000 17681000 4_Strong_Enhancer 41 0.041
22 17681000 17681400 5_Strong_Enhancer 11 0.0275
22 17714800 17717000 5_Strong_Enhancer 60 0.0272727
22 17737800 17738000 5_Strong_Enhancer 8 0.04
22 17738000 17739200 4_Strong_Enhancer 50 0.0416667
22 17739200 17740000 5_Strong_Enhancer 40 0.05
22 17741600 17742400 4_Strong_Enhancer 29 0.03625
22 17744800 17745000 5_Strong_Enhancer 2 0.01
```

Q1. Which enhancers in human embryonic stem cells are under the most genetic "constraint" (that is, have the least genetic variation)?

3. We then need to count the number of genetic variants observed in each enhancer. We need to compute the density of genetic variation, not the count.

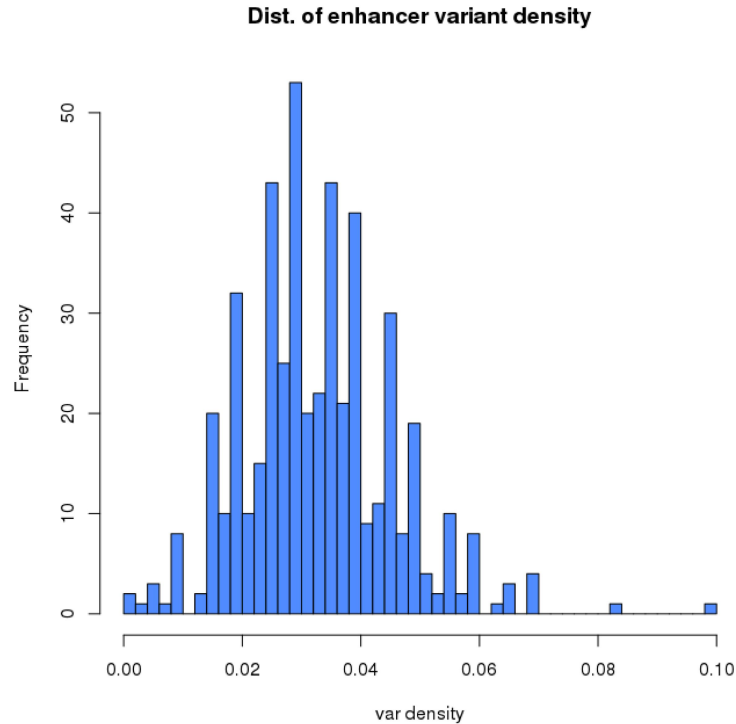
Sort the enhancers by the column starting with density and ending with that column. Treat the values as numbers.

```
$ sort -k6,6n hesc.chromHm.chr22.enh.nochr.strong.vardensity.bedgraph | head
22 28245600 28247400 5_Strong_Enhancer 0 0
22 21846400 21847600 5_Strong_Enhancer 1 0.000833333
22 20402800 20403600 5_Strong_Enhancer 2 0.0025
22 29331400 29331600 5_Strong_Enhancer 1 0.005
22 36310454 36310654 4_Strong_Enhancer 1 0.005
22 48205736 48205936 5_Strong_Enhancer 1 0.005
22 20725800 20726200 4_Strong_Enhancer 3 0.0075
22 21795400 21797400 5_Strong_Enhancer 17 0.0085
22 17744800 17745000 5_Strong_Enhancer 2 0.01
22 29225800 29226000 5_Strong_Enhancer 2 0.01
```

Whoa! No variants from 2,504 genomes in a span of 1,800 bp!



# Is approx. 0.0 variant density in an embryonic stem cell enhancer atypical?



\$ R

```
> enh_constraint <- read.table('hesc.chromHmm.chr22.enh.nochr.strong.vardensity.bedgraph', header=FALSE)
> hist(enh_constraint[,6], breaks=50, col='dodgerblue', xlab='var density', main="Dist. of enhancer variant density")
```

What sanity checks should we conduct to identify potential artifacts?

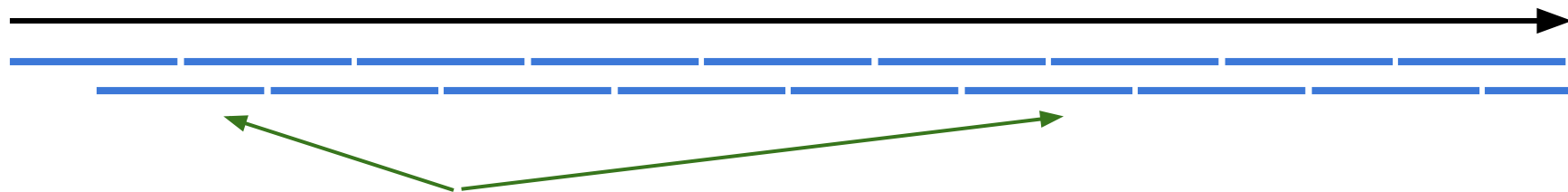
Q2. How could we compute genetic constraint for overlapping 10000bp windows along all of chromosome 22, and not just enhancers?

Q2. How could we compute genetic constraint for overlapping 10000bp windows along *all* of chromosome 22, and not just enhancers?

Experimental design: what materials do we need?

1. Now we need to create a BED file of overlapping, 10000bp windows tiled across chromosome 22.
2. We need a deep catalog of genetic variation in the human genome. 1000 genomes?
3. We then need to count the number of genetic variants observed in window.

Q2. How could we compute genetic constraint for overlapping 10000bp windows along *all* of chromosome 22, and not just enhancers?



Make equally-sized, yet overlapping windows (intervals in BED format) along each chromosome.

To do this, one **MUST** know the length of each chromosome.  
This is the purpose of a so-called "genome" file in BEDTOOLS

Q2. How could we compute genetic constraint for overlapping 1000bp windows along *all* of chromosome 22, and not just enhancers?

```
$ curl https://s3.amazonaws.com/bedtools-tutorials/web/genome.txt > human.grch37.txt
```

```
$ head human.grch37.txt | column -t
```

chr1	249250621
chr10	135534747
chr11	135006516
chr11_gl000202_random	40103
chr12	133851895
chr13	115169878
chr14	107349540
chr15	102531392
chr16	90354753
chr17	81195210

Q2. How could we compute genetic constraint for overlapping 1000bp windows along *all* of chromosome 22, and not just enhancers?

```
$ bedtools makewindows -w 10000 -s 5000 -g human.grch37.txt | head
chr1 0 10000
chr1 5000 15000
chr1 10000 20000
chr1 15000 25000
chr1 20000 30000
chr1 25000 35000
chr1 30000 40000
chr1 35000 45000
chr1 40000 50000
chr1 45000 55000
```

10000bp windows.  
5000bp "step"



Q2. How could we compute genetic constraint for overlapping 1000bp windows along *all* of chromosome 22, and not just enhancers?

*Let's retain windows for just chromosome 22. And we now know we need to remove the "chr" from the chromosome label...*

```
$ bedtools makewindows -w 10000 -s 5000 -g human.grch37.txt \  
  | grep -w "^chr22" \  
  | sed -e 's/chr//' \  
> human.grch37.22.10000w.5000s.bed
```

```
$ head human.grch37.22.10000w.5000s.bed  
22 0 10000  
22 5000 15000  
22 10000 20000  
22 15000 25000  
22 20000 30000  
22 25000 35000  
22 30000 40000  
22 35000 45000  
22 40000 50000  
22 45000 55000
```



Q2. How could we compute genetic constraint for overlapping 1000bp windows along *all* of chromosome 22, and not just enhancers?

*Let's retain windows for just chromosome 22. And we now know we need to remove the "chr" from the chromosome label...*

```
$ bedtools intersect -a human.grch37.22.10000w.5000s.bed -b 1000g.chr22.vcf -c \  
| awk '{OFS="\t"; print $0, $4 / ($3-$2)}' \  
> human.grch37.22.10000w.5000s.vardensity.bedgraph
```

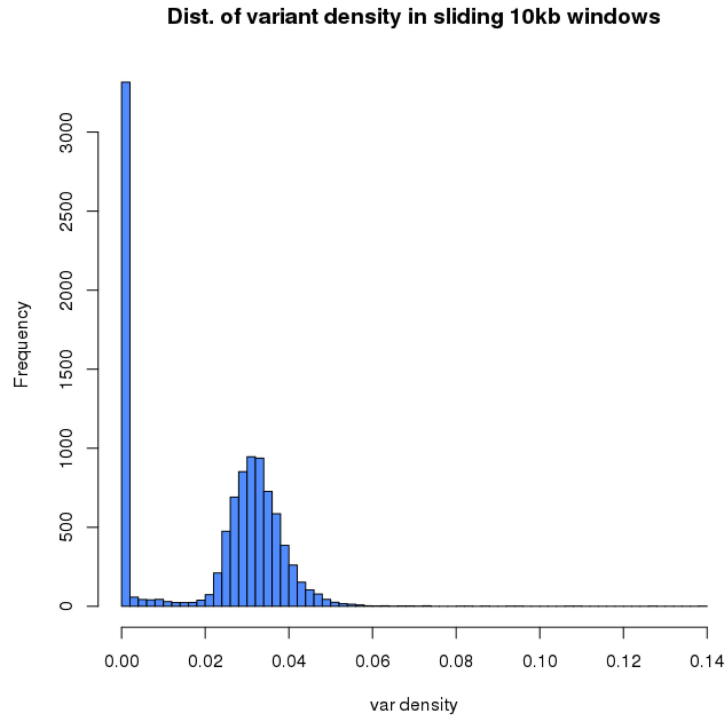
```
$ head human.grch37.22.10000w.5000s.vardensity.bedgraph
```

```
22 0 10000 0 0  
22 5000 15000 0 0  
22 10000 20000 0 0  
22 15000 25000 0 0  
22 20000 30000 0 0  
22 25000 35000 0 0  
22 30000 40000 0 0  
22 35000 45000 0 0  
22 40000 50000 0 0  
22 45000 55000 0 0
```

*Number of variants in 10000bp window, then density of variants in 10000bp window.*

*Are these windows really constrained?*

# Hmmmm. Why are there so many windows with 0.0 variant density? Ideas?



\$ R

```
> win_constraint <- read.table('human.grch37.22.10000w.5000s.vardensity.bedgraph', header=FALSE)
> hist(win_constraint[,5], breaks=50, col='dodgerblue', xlab='var density', main="Dist. of variant density in sliding 10kb windows")
```

Idea. Maybe there is something about the repeat content of these windows that prevents variant detection...

*To test this idea, let's download the FASTA file for chromosome 22 (GRCh37).*

```
$ curl http://hgdownload.soe.ucsc.edu/goldenPath/hg38/chromosomes/chr22.fa.gz > chr22.fa.gz
```

```
$ gzip -d chr22.fa.gz
```

```
$ head -n 1 chr22.fa  
>chr22
```

```
# we need to remove the "chr" from the header of the FASTA file as well.
```

```
$ sed -e 's/chr//' chr22.fa > chr22.numchrom.fa
```

```
$ head -n 1 chr22.numchrom.fa  
>22
```

Idea. Maybe there is something about the repeat content of these windows that prevents variant detection...

*Now that we have the chr22 FASTA, let's investigate the nucleotide content of each 10kb window with the "getfasta" tool.*

**FASTA** ACAGACTGGTATGAAGGTGGCCACAATTCAGAAAGAAAAAAGAAGAGC

**BED**



**getfasta**

GACT

TGAAGGT

AAAAAAG



Idea. Maybe there is something about the repeat content of these windows that prevents variant detection...

*Aha. Looks like masked sequence is our culprit. Let's use the "nuc" tool to just `_count_` the number of Ns in the FASTA file for each 10kb window.*

```
$ bedtools nuc -bed human.grch37.22.10000w.5000s.vardensity.bedgraph -fi chr22.numchrom.fa | head | column -t
```

#1_usercol	2_usercol	3_usercol	4_usercol	5_usercol	6_pct_at	7_pct_gc	8_num_A	9_num_C	10_num_G	11_num_T	12_num_N	13_num_oth	14_seq_len
22	0	10000	0	0	0.000000	0.000000	0	0	0	0	10000	0	10000
22	5000	15000	0	0	0.000000	0.000000	0	0	0	0	10000	0	10000
22	10000	20000	0	0	0.000000	0.000000	0	0	0	0	10000	0	10000
22	15000	25000	0	0	0.000000	0.000000	0	0	0	0	10000	0	10000
22	20000	30000	0	0	0.000000	0.000000	0	0	0	0	10000	0	10000
22	25000	35000	0	0	0.000000	0.000000	0	0	0	0	10000	0	10000
22	30000	40000	0	0	0.000000	0.000000	0	0	0	0	10000	0	10000
22	35000	45000	0	0	0.000000	0.000000	0	0	0	0	10000	0	10000
22	40000	50000	0	0	0.000000	0.000000	0	0	0	0	10000	0	10000

Zero variant density because there are 10000 Ns in these regions. Therefore, one cannot align reads to these regions and thus one has no power to detect variation in these regions.

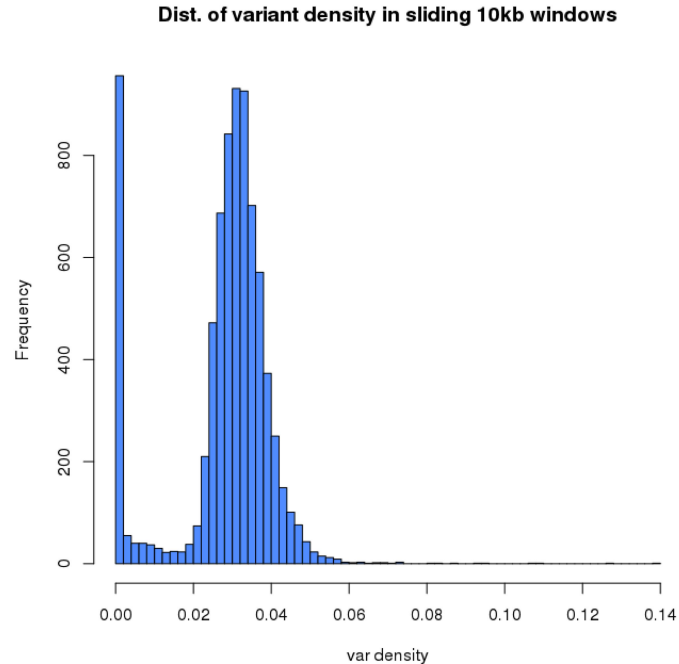
Idea. Maybe there is something about the repeat content of these windows that prevents variant detection...

*Let's use bioawk to filter out regions that have 1 or more "N" in the FASTA sequence. Recall that bioawk allows us to refer to columns in a file by name rather than solely by number in regular awk. Thanks Heng Li!*

```
$ bedtools nuc -bed human.grch37.22.10000w.5000s.vardensity.bedgraph -fi chr22.numchrom.fa \  
  | bioawk -H -t -c bed '$12_num_N ==0' \  
  > human.grch37.22.10000w.5000s.vardensity.bedgraph.noN.bed  
$ head human.grch37.22.10000w.5000s.vardensity.bedgraph.noN.bed | column -t
```

#1_usercol	2_usercol	3_usercol	4_usercol	5_usercol	6_pct_at	7_pct_gc	8_num_A	9_num_C	10_num_G	11_num_T	12_num_N	13_num_oth	14_seq_len
22	10510000	10520000	0	0	0.653900	0.346100	3800	1627	1834	2739	0	0	10000
22	10515000	10525000	0	0	0.625600	0.374400	3240	1789	1955	3016	0	0	10000
22	10520000	10530000	0	0	0.532900	0.467100	2350	2478	2193	2979	0	0	10000
22	10525000	10535000	0	0	0.568100	0.431900	2691	2093	2226	2990	0	0	10000
22	10530000	10540000	0	0	0.673900	0.326100	3280	1475	1786	3459	0	0	10000
22	10535000	10545000	0	0	0.675800	0.324200	3245	1607	1635	3513	0	0	10000
22	10540000	10550000	0	0	0.670500	0.329500	3659	1549	1746	3046	0	0	10000
22	10545000	10555000	0	0	0.656700	0.343300	3665	1598	1835	2902	0	0	10000
22	10550000	10560000	0	0	0.640300	0.359700	3369	1771	1826	3034	0	0	10000

# Much better. Now bimodal. Any ideas?



\$ R

```
> win_constraint <- read.table('human.grch37.22.10000w.5000s.vardensity.bedgraph.noN.bed', header=FALSE)
> hist(win_constraint[,5], breaks=50, col='dodgerblue', xlab='var density', main="Dist. of variant density in sliding 10kb windows")
```



Q3. How do we know if an observed number of intersections between two datasets is statistically significant? That is, is the observation more extreme than what we would expect by chance?

Q3. Do GWAS SNPs overlap enhancers in hESCs more often than expected by chance?

# Do GWAS SNPs overlap enhancers in hESCs more often than expected by chance?

## Experimental design: what materials do we need?

1. We need hESC enhancers.
2. We need a catalog of significant SNPs from Genome Wide Association Studies (GWAS).
3. We need to measure the number of observed overlaps between the two.
4. Lastly, we need to compare the observed to what we expect to assess the significance of the relationship between these two genomic "features".

# Q3. Do GWAS SNPs overlap enhancers in hESCs more often than expected by chance?

## 1. We need hESC enhancers. This time, let's include strong and weak enhancers

```
$ grep "Enhancer" hesc.chromHmm.bed | sed -e 's/chr//' > hesc.chromHmm.allenh.nochr.bed
```

```
$ head hesc.chromHmm.allenh.nochr.bed
```

```
1 27537 27737 6_Weak_Enhancer
```

```
1 30337 30537 6_Weak_Enhancer
```

```
1 34737 34937 7_Weak_Enhancer
```

```
1 35737 35937 7_Weak_Enhancer
```

```
1 35937 36137 6_Weak_Enhancer
```

```
1 36137 36337 7_Weak_Enhancer
```

```
1 36337 36537 5_Strong_Enhancer
```

```
1 36537 37537 6_Weak_Enhancer
```

```
1 37537 37737 7_Weak_Enhancer
```

```
1 56337 56537 6_Weak_Enhancer
```

Q3. Do GWAS SNPs overlap enhancers in hESCs more often than expected by chance?

2. We need a catalog of significant SNPs from Genome Wide Association Studies (GWAS).

```
# obtained from the UCSC Genome Browser's Table Browser
```

```
$ wget https://s3.amazonaws.com/bedtools-tutorials/web/gwas.phenotype.bed
```

```
$ sed -e 's/chr//' gwas.phenotype.bed > gwas.phenotype.nochr.bed
```

```
$ head gwas.phenotype.nochr.bed | column -t
```

#chrom	chromStart	chromEnd	name	pubMedID	trait
1	780396	780397	rs141175086	26955885	Morning vs. evening chronotype
1	1005805	1005806	rs3934834	19851299	Body mass index
1	1079197	1079198	rs11260603	23382691	IgG glycosylation
1	1247493	1247494	rs12103	26192919	Ulcerative colitis
1	1247493	1247494	rs12103	26192919	Inflammatory bowel disease
1	1247493	1247494	rs12103	26192919	Crohn's disease
1	1247493	1247494	rs12103	23128233	Inflammatory bowel disease
1	1287054	1287055	rs186507655	27197191	Cancer (pleiotropy)
1	1723030	1723031	rs9660180	25673413	Body mass index

Q3. Do GWAS SNPs overlap enhancers in hESCs more often than expected by chance?

3. We need to measure the number of observed overlaps between the GWAS SNPs and hESC enhancers.

```
$ wc -l gwas.phenotype.nochr.bed  
39335
```

```
$ wc -l hesc.chromHmm.allenh.nochr.bed  
240136
```

```
$ bedtools intersect -a gwas.phenotype.nochr.bed -b hesc.chromHmm.allenh.nochr.bed -u | wc -l  
2040
```

So 5.2% (2,040 of 39335) GWAS SNPs overlap at least 1 predicted enhancer in human embryonic stem cells.  
Is that interesting? How would we know?

Q3. Do GWAS SNPs overlap enhancers in hESCs more often than expected by chance?

4. Lastly, we need to compare the observed to what we expect to assess the significance of the relationship between these two genomic "features".

```
# observed = 2,040
```

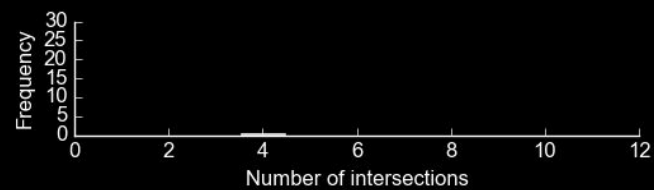
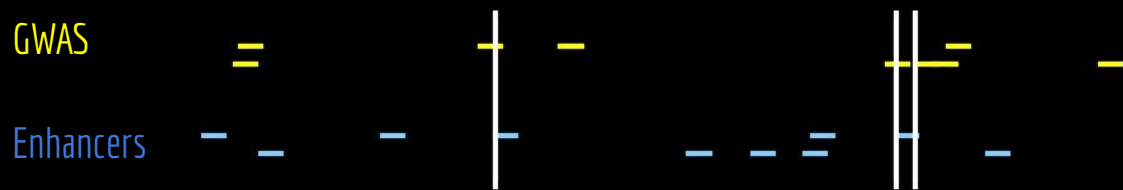
```
$ bedtools intersect -a gwas.phenotype.nochr.bed -b hesc.chromHm.allenh.nochr.bed -u | wc -l  
2040
```

How do we derive an expectation (that is, a null hypothesis)? One way is to do a Monte Carlo simulation.

Experimental design:

1. Repeatedly (1000s of times) shuffle intervals randomly throughout the genome.
2. For each "shuffling", measure how many intersections there are.
3. Build up a distribution of the number of intersections observed for each shuffling.

# Monte Carlo simulation by shuffling intervals and measuring intersections that occur "by chance"



Simulation animation from Ryan Layer (@ryanlayer)



Q3. Do GWAS SNPs overlap enhancers in hESCs more often than expected by chance?

4. Lastly, we need to compare the observed to what we expect to assess the significance of the relationship between these two genomic "features".

```
# this will take a few minutes
```

```
for i in `seq 1 100`;
```

```
do
```

```
    bedtools shuffle -i gwas.phenotype.nochr.bed -g genome.txt \  
    | bedtools intersect -a - -b hesc.chromHmm.allenh.nochr.bed -u \  
    | wc -l \  
    >> random.intersections.txt
```

```
done
```

```
sort random.intersections.txt | uniq -c  
100 0
```

">>" Append the results of each of the 100 experiments to a file.

All 100 experiments yielded 0 intersections!!!

One must be careful about defining the "domain" of an experiment

# The dilemma of choosing the ideal permutation strategy while estimating statistical significance of genome-wide enrichment

*Subhajoti De, Brent S. Pedersen and Katerina Kechris*

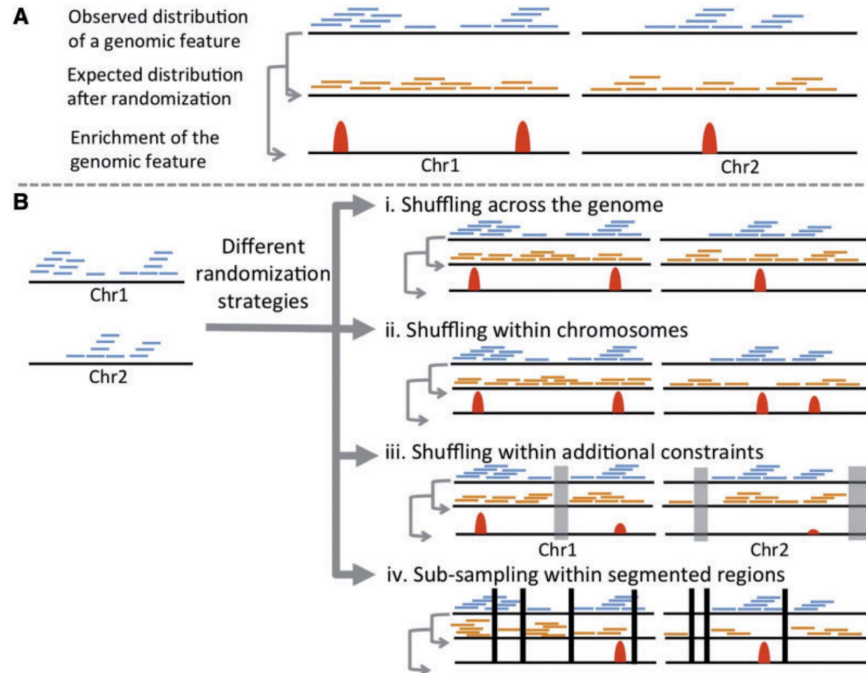
Submitted: 30th April 2013; Received (in revised form): 13th June 2013

## **Abstract**

Integrative analyses of genomic, epigenomic and transcriptomic features for human and various model organisms have revealed that many such features are nonrandomly distributed in the genome. Significant enrichment (or depletion) of genomic features is anticipated to be biologically important. Detection of genomic regions having enrichment of certain features and estimation of corresponding statistical significance rely on the expected null distribution generated by a permutation model. We discuss different genome-wide permutation approaches, present examples where the permutation strategy affects the null model and show that the confidence in estimating statistical significance of genome-wide enrichment might depend on the choice of the permutation approach. In those cases, where biologically relevant constraints are unclear, it is preferable to examine whether key conclusions are consistent, irrespective of the choice of the randomization strategy.

**Keywords:** *genome-wide enrichment; statistical significance; permutation strategy; null distribution*

# One must be careful about defining the "domain" of an experiment



**Figure 1:** (A) The basic principle behind permutation analysis to determine genome-wide enrichment of a genomic or epigenomic feature. (B) A different randomization strategy can produce a different expected distribution, and hence affect statistical significance of enrichment of the feature. In (Biii), disallowed regions are masked (gray) while shuffling with additional constraints. The displayed list does not represent the exhaustive list of possible randomization strategies. A colour version of this figure is available at BIB online: <http://bib.oxfordjournals.org>.

Use the bedtools -incl (include regions) and -excl (exclude regions)

Input (I) 

---

shuffle I



shuffle I  
(-incl)



shuffle I  
(-excl)

