
regulation of gene expression by methylation

development of a software framework to integrate genomic data

Brent S. Pedersen

Sequence Data

```
brentp@compbio:~/src/bwa-meth$ zless /proj/Schwartz/brentp/2013/ken-rrbs/pilot/38372_ACAGTG_L003_R1_001.fastq.gz | head -n 12
@HISEQ:105:C2UE1ACXX:3:1101:1338:2021 1:N:0:ACAGTG
NTTTTTTTAGGTTTTTTTATTGTGGGGTAGGGGAGGTTTTTGGAAGTGTATGTTTTTTTTGGAGTGATTGGTAAGGTTTAAATATTAGGTGTTTTA
+
#0<FFFFFF<0BFFFFIIFI<F<BBFFFFIIFIFFBFBFF00<BBBBFBFFF7BFFBBBBBBB ' '07B<BBBF '07<B0<BBBBBBBBBBBF< ' '00<BBBB
@HISEQ:105:C2UE1ACXX:3:1101:1365:2029 1:N:0:ACAGTG
NTAATGAATAAGGATTGTTGTATTGGAATTATAAATTAGAGAGTGGGGATTATTGAAAGAAGTTAAATGAATAAAAGTTGAAAATTGTGTGTTTTTAATA
+
#0<FF<BFFFFBFFFBFF<FFFIB<FFFIFFFFFIII<FBFFFIFIIIFFIIBFFFBFFBFFIIIII<BFFFFFFF7<B7BBFFFFF7<<B<BBBFFBFB<
@HISEQ:105:C2UE1ACXX:3:1101:1425:2074 1:N:0:ACAGTG
NTTTAAGTAGTTTGGGGTATGGTGGTTTTATATTGGGGATAGGAAAAATGCGGAAGGAGTTATGGTTTGTATTTGGTATTGATTGCGTTAAGGTTGGTATT
+
#0<FFFFFFFFFFFFFF<FFFBFIFIIFFIIBFFFIIB0 '0<BFF<FFFFFI<B7BBB<BFFB BBB<7BBB<BFFFF70<BBB ' <BB<<<BB<B#####
brentp@compbio:~/src/bwa-meth$ zless /proj/Schwartz/brentp/2013/ken-rrbs/pilot/38372_ACAGTG_L003_R1_001.fastq.gz | wc -l
```

Outline

- detecting gene expression
 - detecting methylation
 - analyzing expression and methylation
 - traditional methods
 - in-development methods/ideas
-

Gene Expression

- DNA -> RNA -> Protein
- We measure (m)RNA as proxy for protein
 - cheaper than measuring protein
 - high-throughput methods (none for protein)

In asthmatics we expect to see genes related to immune response expressed at higher levels than in healthy individuals

De

image from: <http://voer.edu.vn/module/dna-replication>

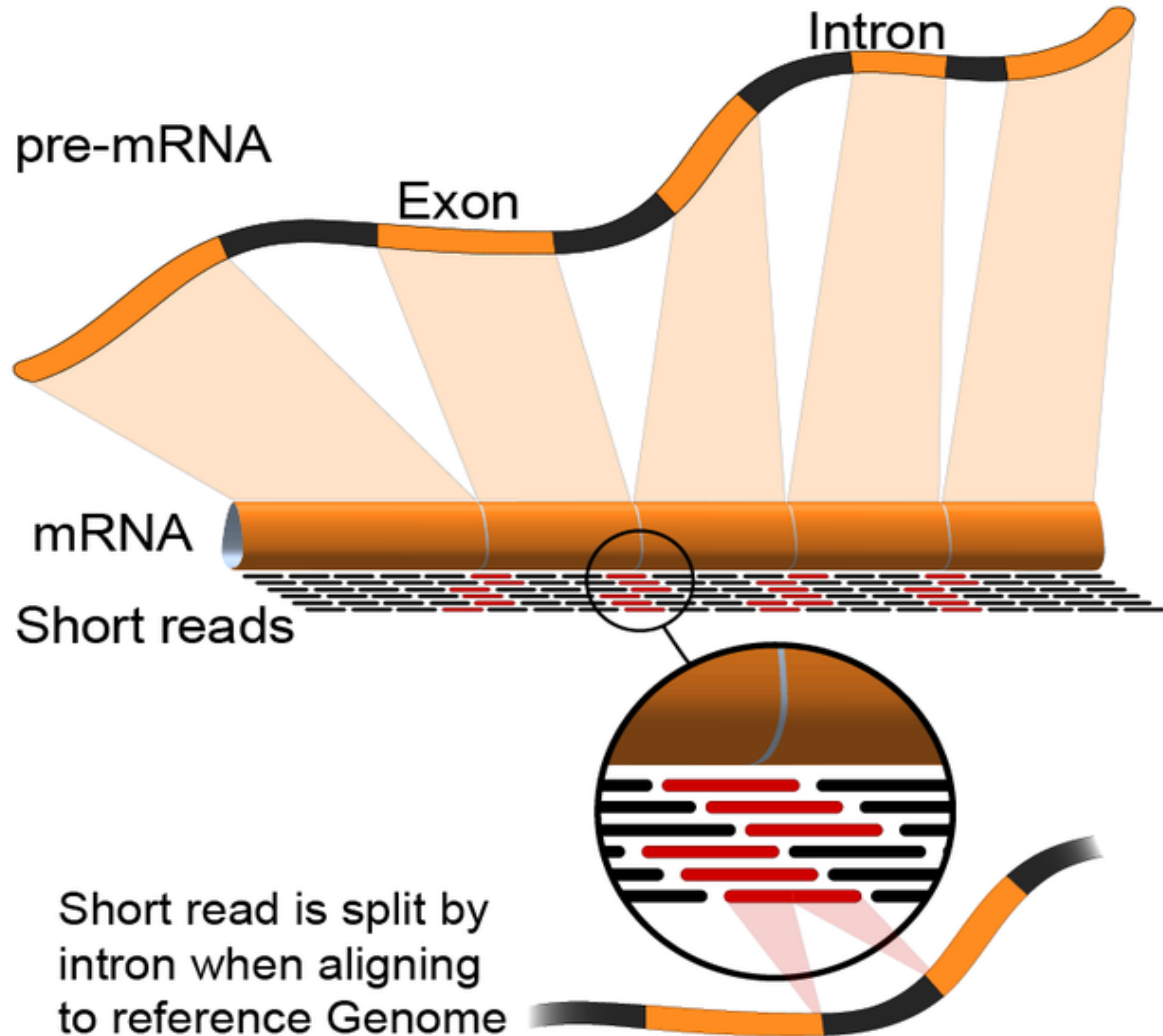
```
brentp@compbio:~/src/denver-bio/jian-tolerance-2013$ wc -l data/expr.norm.txt
23134 data/expr.norm.txt
brentp@compbio:~/src/denver-bio/jian-tolerance-2013$ head -15 data/expr.norm.txt | cut -f 1-10 | cols
probe          tol-8  tol-2  control-8  nontol-8  tol-7  tol-1  control-5  nontol-3  nontol-4
A_51_P414243   6.36   5.87   7.60       6.79      6.52   6.24   7.45       6.71      6.50
A_30_P01024440 5.64   5.31   5.97       5.58      5.79   5.36   5.00       5.13      5.74
A_30_P01025554 9.73   9.88   9.87       9.55      9.59   9.34   9.66       8.94      9.37
A_51_P328014    7.70   7.75   7.56       7.20      7.73   7.54   7.46       7.56      7.55
A_55_P2056220   8.38   8.06   8.46       7.84      8.28   8.19   8.27       8.59      8.45
A_55_P1985764  11.33  11.10  11.43      11.21     11.32  10.88  11.39      10.79     11.05
A_52_P108321    7.12   5.87   5.93       5.42      5.62   5.69   6.79       5.78      5.77
A_52_P123354    7.58   7.65   8.26       7.12      7.67   7.56   7.91       7.69      7.78
A_55_P2061724   6.45   5.54   6.86       7.23      6.43   6.64   7.06       6.43      7.01
A_55_P2049122   4.76   4.43   5.30       5.02      4.86   4.94   4.46       4.88      5.08
A_51_P385099    10.69  11.24  6.91       13.54     9.77   11.29  5.96       14.49     12.89
A_55_P2111005   8.35   8.22   8.06       7.67      8.09   7.87   7.75       8.09      8.42
A_55_P2041509   9.68   9.41   9.73       9.23      9.52   9.31   9.64       9.35      9.62
A_30_P01029765 5.09   4.64   5.14       4.98      4.99   5.04   4.18       4.72      4.90
brentp@compbio:~/src/denver-bio/jian-tolerance-2013$
```

RNA-Seq: Detection

Aligning spliced reads back to genome is hard!

Every paper detects new, undiscovered transcripts.

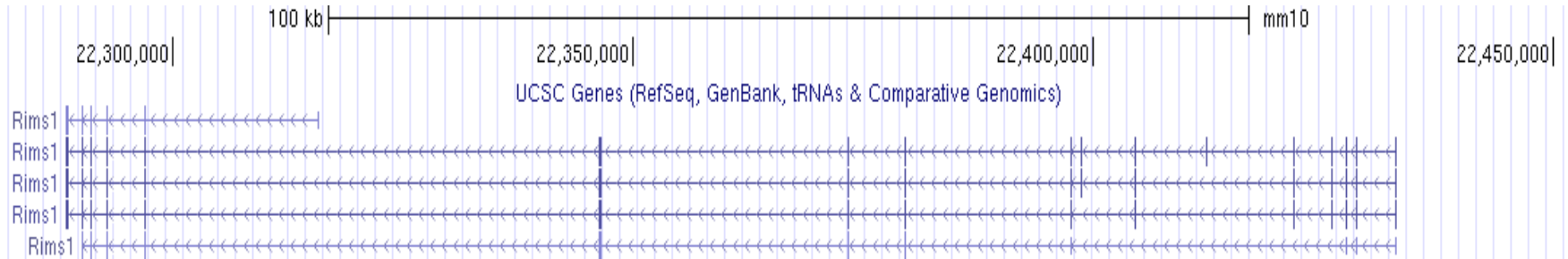
Many methods allow aligning to set of known transcripts.



RNA-Seq: Detecting Gene Expression

Much more information:

- Alternative splicing (different transcript of gene)
- Allele-specific expression
- Genetic Variants
- Finer resolution of differences in expression



-
- Diagram illustrating a DNA sequence with four overlapping reading frames. The top three frames show stop codons (T) in red, indicating premature termination. The bottom frame shows the full sequence of amino acids in green: G C C C A A G A T C G G A G A T T T C G G G C T C G A A A G G C T T C T C G C C G G G G A T A C T A G.

Bisulfite Sequencing

BIOINFORMATICS APPLICATIONS NOTE

Vol. 27 no. 17 2011, pages 2435–2436
doi:10.1093/bioinformatics/btr394

Sequence analysis

Advance Access publication June 30, 2011

MethylCoder: software pipeline for bisulfite-treated sequences

Brent Pedersen*, Tzung-Fu Hsieh, Christian Ibarra and Robert L. Fischer

Department of Plant and Microbial Biology, University of California, Berkeley, CA 94720, USA

Associate Editor: Martin Bishop

ABSTRACT

Motivation: MethylCoder is a software program that generates per-base methylation data given a set of bisulfite-treated reads. It provides the option to use either of two existing short-read aligners, each with different strengths. It accounts for soft-masked alignments and overlapping paired-end reads. MethylCoder outputs data in text and binary formats in addition to the final alignment in SAM format, so that common high-throughput sequencing tools can be used on the resulting output. It is more flexible than existing software and competitive in terms of speed and memory use.

Availability: MethylCoder requires only a python interpreter and a C compiler to run. Extensive documentation and the full source code are available under the MIT license at: <https://github.com/brentp/methylcode>.

Contact: bpederse@gmail.com

are limited to the bowtie aligner and do not support color space reads. Bisulfite-treated reads analysis tool (BRAT; Harris *et al.*, 2009) also uses a hashing approach and is the only other aligner that avoids double-counting overlapping paired-end reads.

We introduce MethylCoder, a fast, memory-efficient BS-Seq pipeline. It supports both paired- and single-end reads in color space or nucleotide formats. MethylCoder provides a single entry point and common output formats for the bowtie (Langmead *et al.*, 2009) and genomic short-read nucleotide alignment program (GSNAP) (Wu and Nacu, 2010) aligners. Each of these aligners has different strengths; GSNAP has no limitation on the size of the reference, but does not consider quality information with the reads. Bowtie can only map to references <4 Gb in total length, but considers quality and can map color space reads. Utilizing these short-read aligners, while providing access to their arguments, ensures that MethylCoder

- Use
- Me
- Un-
- BS-
- de

T
T T T
A
A G C C C

- pot

T
A
A G C T

Mapping BS-Seq Reads

We don't know which **T**'s in the reads are actual **T**'s and which are unmethylated (and therefore converted) **C**'s.

We can't use traditional aligners to map reads back to the genome because of C=>T mismatches. So:

- Convert* genome C => T, reverse-complement, G => A.
- Convert* reads C => T
- Map converted reads to converted reference
- For each alignment, recover original (un-converted) read and compare to un-converted reference to calculate Methylation.

* *where "convert" is In silico*

Output

- Per-base report on conversions:

====	=====	=====	===	===
chr	context	bp-position	C's	T's
====	=====	=====	===	===
chr1	CG+	106	1	7
chr1	CG-	107	6	2
chr1	CG+	108	7	1
chr1	CG-	109	11	0
chr1	CHG+	113	0	9
chr1	CG+	114	9	0
chr1	CG-	115	12	1

- % Methylation calculated as $(C / (C + T))$

Methylation: What it does (?)

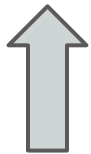
Regulates Nearby Gene Expression !!



Methylation



Expression

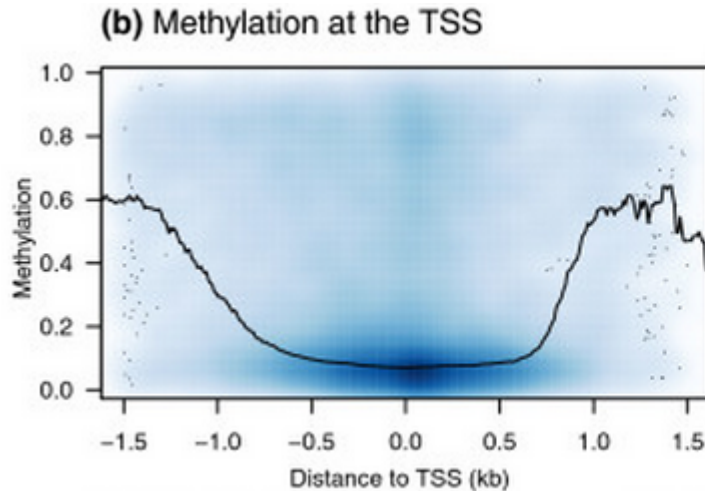


Methylation

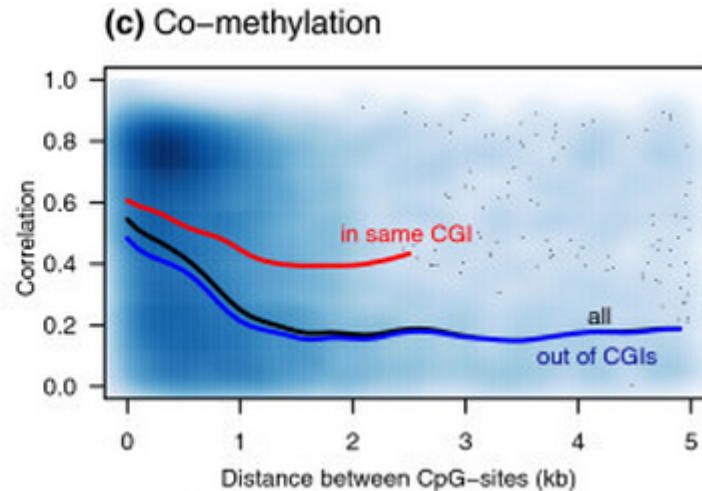


Expression

Methylation: What it does (where)



Promoter Methylation

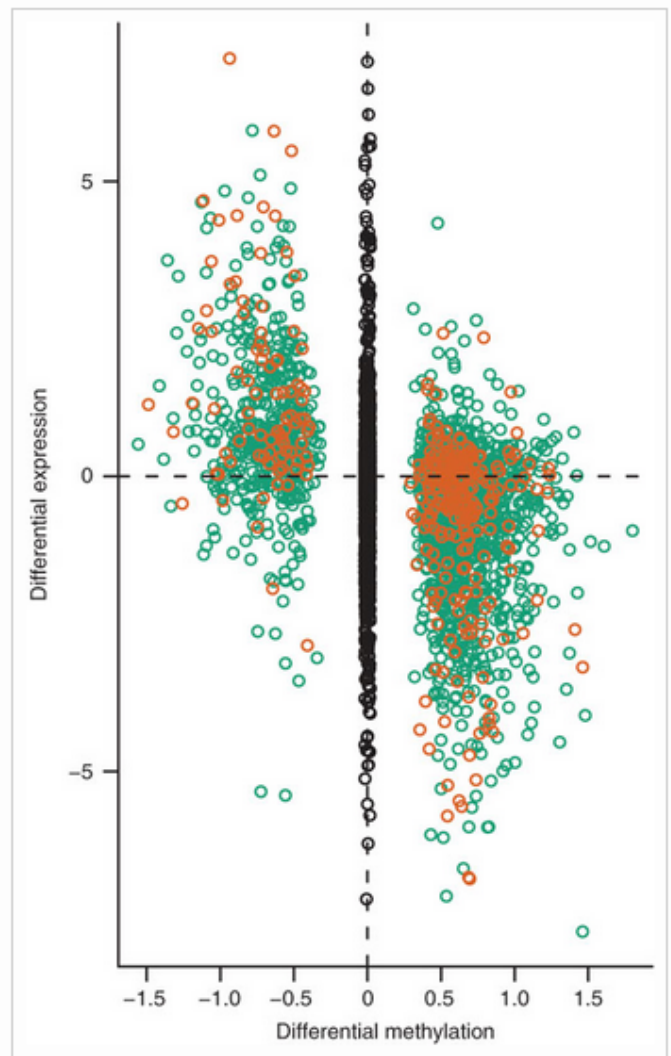


Methylation
Auto-Correlation

Methylation: What it does (sometimes)

Plotted are log (base 2) ratios of liver to brain expression against DeltaM values for liver and brain DNAm. **Orange** dots represent T-DMRs located within **300 bp** from the corresponding gene's transcriptional start site (TSS). **Green** dots represent T-DMRs that are located from **300 to 2,000 bp** from the TSS of an annotated gene. **Black** dots, in the middle, represent log ratios for all genes **further than 2 kb** from an annotated TSS

Nature Genetics 41, 178 - 186 (2009) . Irizzary et al



Methylation: What it does

But we (and many others) find:

- Expression changes from genome-wide experiments
- Methylation changes from genome-wide experiments

→ low overlap

Methylation changes explain very few and very little of the expression changes

WHY?

Independent Analysis

List of Methylation Sites
associated with disease

gene-123
gene-456
gene-777
gene-678
gene-222
gene-BBB
gene-342

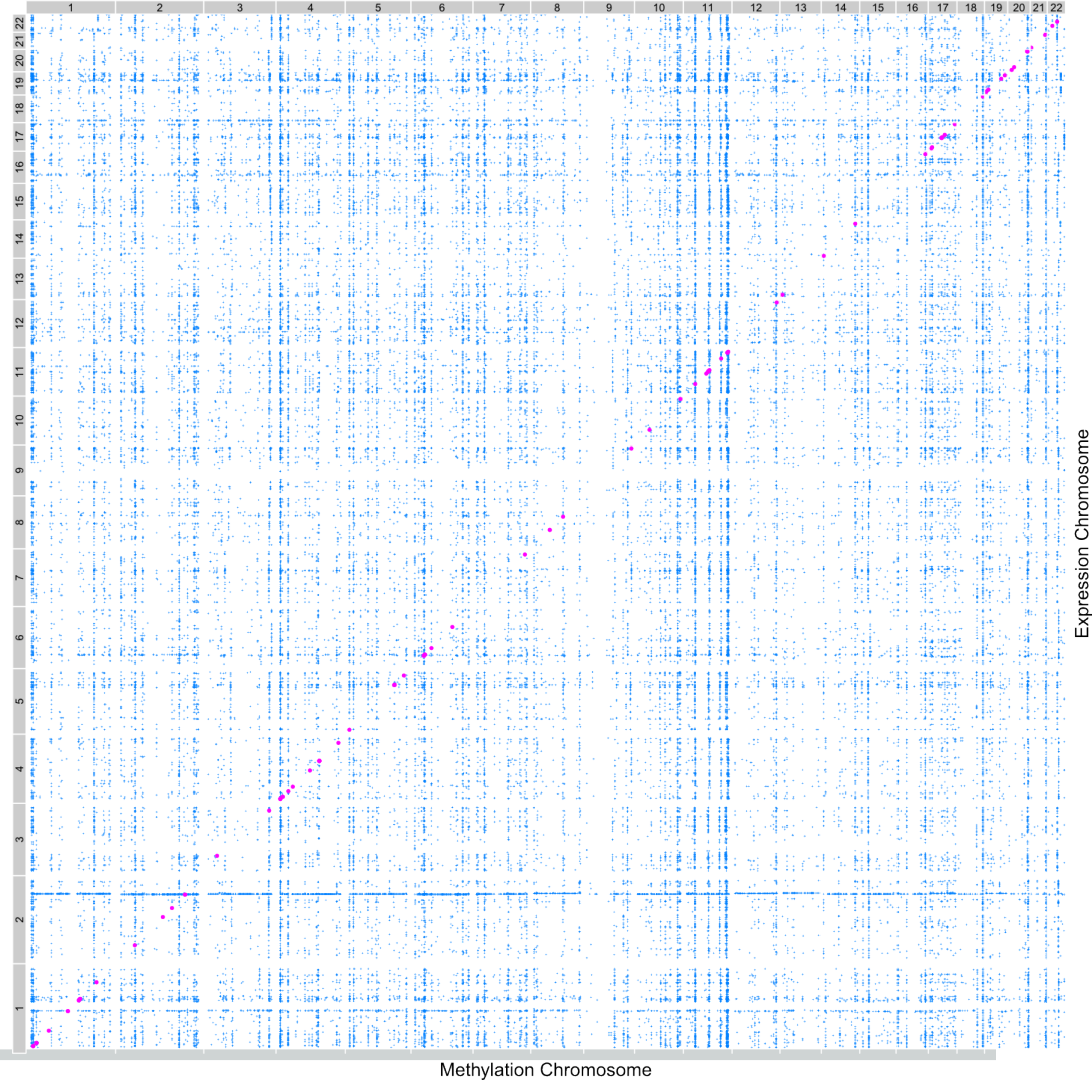
List of Expression Sites
associated with disease

gene-AAZ
gene-FGA
gene-GCG
gene-BBB
gene-KKQ
gene-LLS
gene-MMM



Post-Hoc Comparisons of Independent Analyses.

Exhaustive single-probe analysis



Traditional Analyses

Fit a linear model at each probe:

expression ~ disease + age + gender

Extract p-value for disease parameter.

Report genes with **corrected** p-value < 0.05

May account for batch effects and/or study design.

Side Note: Multiple Testing

Test 40K sites, how to determine which are truly “different”?

p-value of 0.05 on 40,000 tests on random data (with no true differences) will give about 2,000 false positives.

Traditional Analyses For Methylation

Fit a linear model at each probe:

methylation \sim disease + age + gender

But, methylation arrays now have 480K probes.

$$0.05 / 480K == 1.04 \times 10^{-7}$$

Most methods now aggregate across probes to increase *power* since adjacent methylation sites are often highly correlated.

Traditional Analyses For Methylation

Aggregate information
across probes to find:

DMR:
Differentially
Methylated
Region

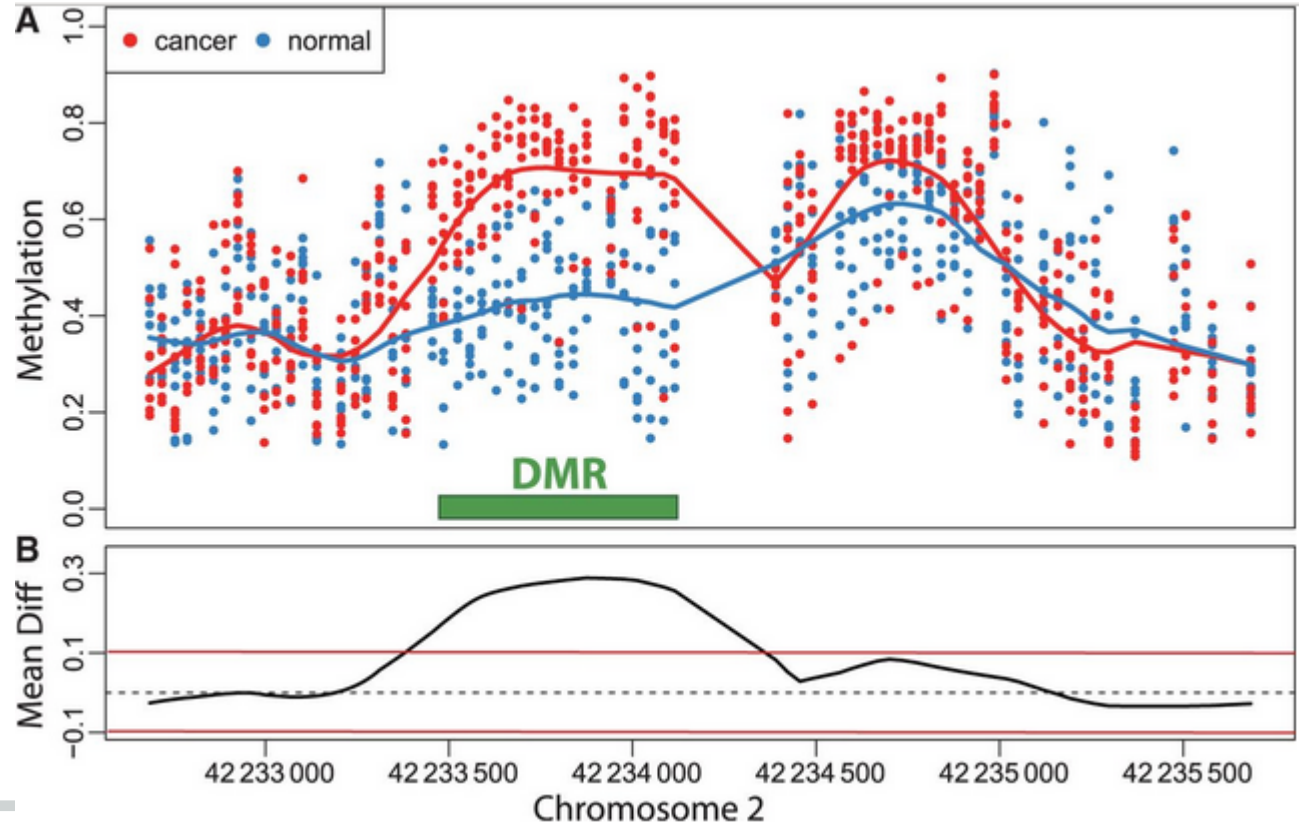
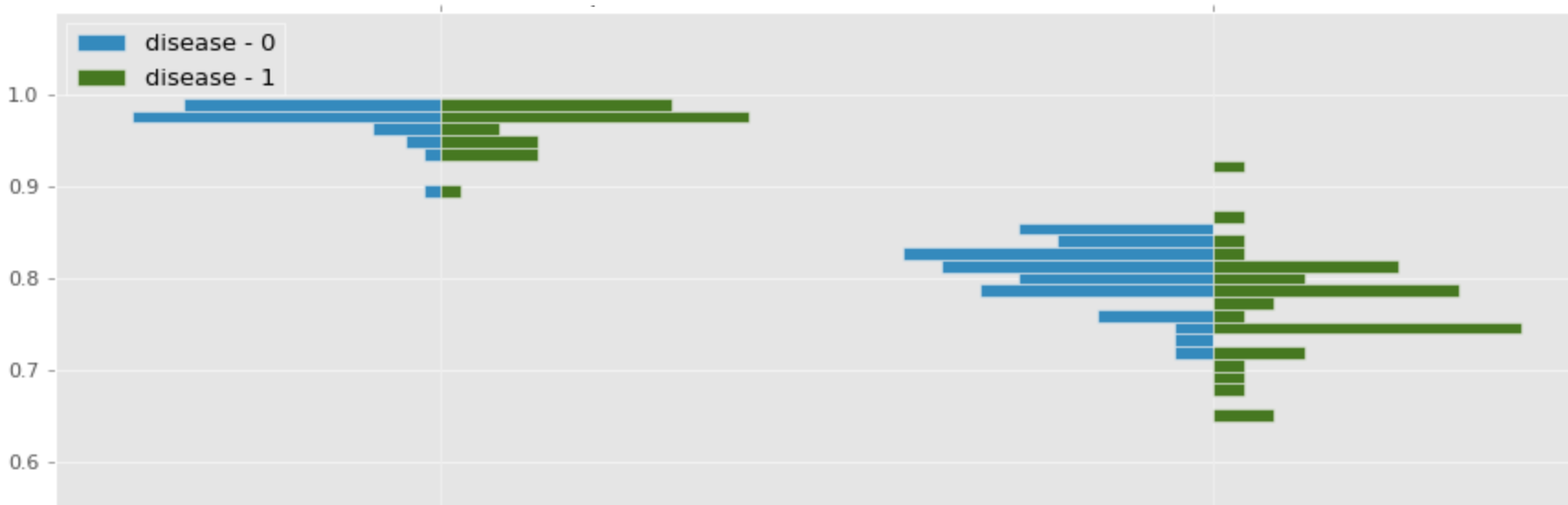


Image From Jaffe et al.
Bumphunting paper.

Methylation: Finding DMRs

GOAL: find even subtle, 2-probe DMR's while minimizing false +



DMR-finding methods

Bumphunting:

1. find coefficient from a linear model for every site (e.g. for disease)
 - a. (these form the putative bumps)
2. generate simulated data by shuffling the residuals of the null model (without disease) and adding them to the predictions for the null model.
3. Fit full model to simulated data from 2. to generate null distribution of betas.
4. compare observed betas to simulated to get significance

include locally weighted smoothing and **sum** the coefficients in a region above some cutoff.

From Jaffe et al.
Bumphunting paper.

Comb-p: software for combining, analyzing, grouping and correcting spatially correlated *P*-values

Liptak (1958)

Brent S. Pedersen^{1,*}, David A. Schwartz¹, Ivana V. Yang^{1,†} and Katerina J. Kechris^{2,†}

¹Department of Medicine and ²Department of Biostatistics and Informatics, University of Colorado, Denver, Anschutz Medical Campus, Aurora, CO 80045, USA

Associate Editor: Alex Bateman

ABSTRACT

Summary: *comb-p* is a command-line tool and a python library that manipulates BED files of possibly irregularly spaced *P*-values and (1) calculates auto-correlation, (2) combines adjacent *P*-values, (3) performs false discovery adjustment, (4) finds regions of enrichment (i.e. series of adjacent low *P*-values) and (5) assigns significance to those regions. In addition, tools are provided for visualization and assessment. We provide validation and example uses on bisulfite-seq with *P*-values from Fisher's exact test, tiled methylation probes using a linear model and Dam-ID for chromatin binding using moderated *t*-statistics. Because the library accepts input in a simple, standardized format and is unaffected by the origin of the *P*-values, it can be used for a wide variety of applications.

Availability: *comb-p* is maintained under the BSD license. The documentation and implementation are available at <https://github.com/brentp/combined-pvalues>.

Contact: bpedersen@gmail.com

2 APPROACH

Tiling array studies relying on two-sample comparisons may be amenable to the calculation of sliding window averages of log ratios or two-sample test statistics. However, more complex study designs often require covariates and report *P*-values from linear models or other statistical tests.

We utilize a 'moving averages' method of *P*-value correction that does not depend on the test used to generate the *P*-values. Fisher (1948) developed an approach of combining *P*-values from independent tests to get a single meta-analysis test statistic with a χ^2 distribution and degrees of freedom based on the number of tests being combined. A similar method developed by Stouffer *et al.* (1949) and Liptak (1958) first converts *P*-values to Z-scores which are then summed and scaled to create a single, combined Z-score. The Stouffer–Liptak method lends itself to the addition of weights on each *P*-value. Zaykin *et al.* (2002) introduced a method to use weights to perform a

Problem

DMI

Gene expression

A-clustered
methylationTamar Sofer¹Andrea A. Baccarelli²¹Department of Biostatistics, Harvard School of Public Health, Boston, MA 02115, USA, ²Department of Epidemiology, Harvard School of Public Health, Boston, MA 02115, USA³NIEHS, Epidemiology Branch, National Institutes of Health, Research Triangle Park, NC 27709, USA

of Preventive Medicine, University of Chicago, Chicago, IL 60611, USA

of Public Health, Harvard School of Public Health, Boston, MA 02115, USA

Associate Editor: Martin J. Blaser

a.

ABSTRACT

b.

Motivation: DNA methylation is a process that affects gene expression and is associated with many diseases (e.g. cancer, other diseases). Current methods for analyzing DNA methylation data (e.g. Illumina arrays) measure the proportion of methylated CpG sites in a sample, but do not provide information on the relative abundance of individual CpG sites.

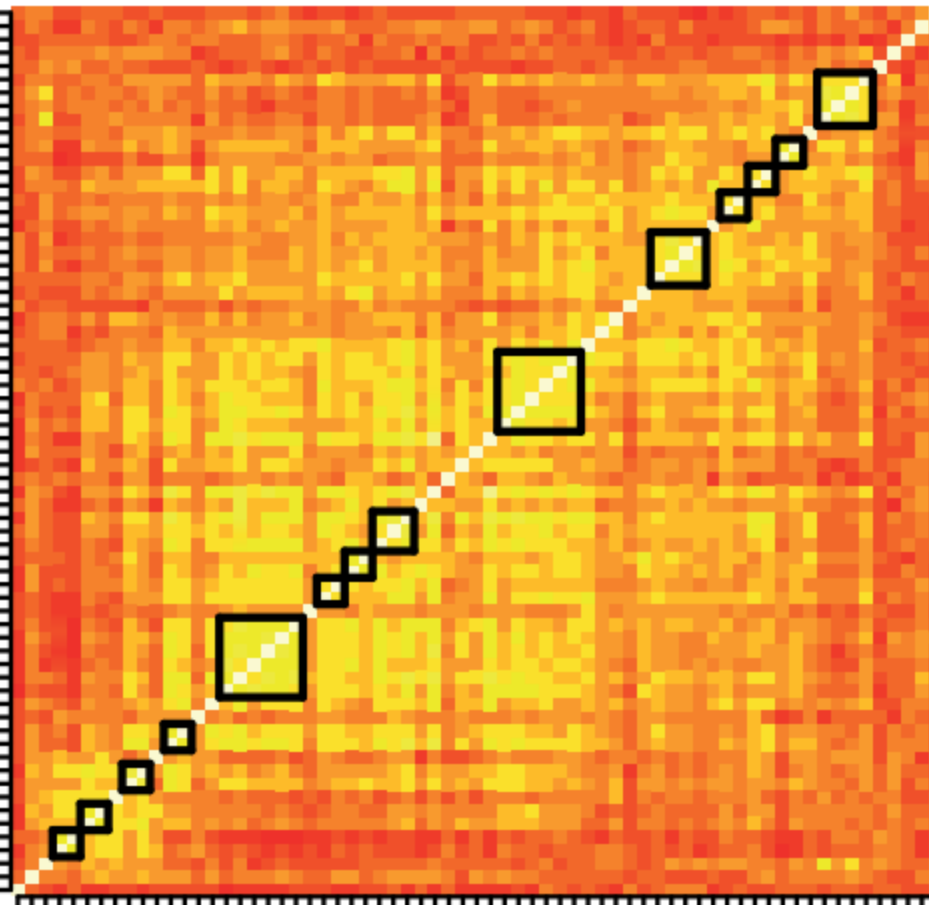
Current methods for analyzing DNA methylation data (e.g. Illumina arrays) measure the proportion of methylated CpG sites in a sample, but do not provide information on the relative abundance of individual CpG sites.

Current methods for analyzing DNA methylation data (e.g. Illumina arrays) measure the proportion of methylated CpG sites in a sample, but do not provide information on the relative abundance of individual CpG sites.

Current methods for analyzing DNA methylation data (e.g. Illumina arrays) measure the proportion of methylated CpG sites in a sample, but do not provide information on the relative abundance of individual CpG sites.

Current methods for analyzing DNA methylation data (e.g. Illumina arrays) measure the proportion of methylated CpG sites in a sample, but do not provide information on the relative abundance of individual CpG sites.

Current methods for analyzing DNA methylation data (e.g. Illumina arrays) measure the proportion of methylated CpG sites in a sample, but do not provide information on the relative abundance of individual CpG sites.



on August 29, 2013

ulated
ure

r, Boston, MA

i269, USA,

⁴Department

e, Suite 1400

rvard School

chemicals (Anttila

3). Modern arrays

tens of thousands

are methylation in

d as a continuous

the proportion of

measured tissue.

ites, whether asso-

ected by environ-

h sets of CpG sites

ithin

act p-

of

nilar to

regions

A-Cluster

1. find

100

2. Fit

val

a.

b.

Sofer, Tamar


associated with exposure.

A-Clustering Python Module

<https://github.com/brentp/aclust> | <https://pypi.python.org/pypi/aclust>

Streaming, agglomerative clustering

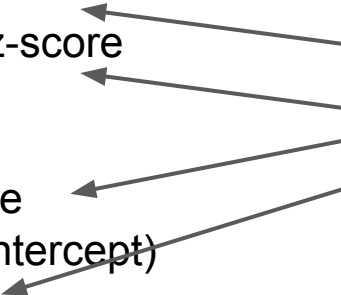
since we know objects are sorted, we can find local clusters without reading into memory.



```
for cluster in aclust(sorted_objs, max_dist, max_skip=1,  
                      linkage='single', multi_member=False):  
    result = test_cluster(cluster, covariates, model)  
    yield result
```

Finding DMRs: Proposed Method(s)

1. **Find clusters** of similar probes as in A-clustering
 - a. unbiased selection discards single probes without consideration of study-design
 - b. reduce multiple-testing burden by testing N regions instead of I CpG's
2. **Transform data** as needed
 - a. logit/inverse logit
 - b. outlier removal (**)
3. Apply any method to **assign significance**:
 - a. bump-hunting (sort of)
 - b. combine p-values with liptak or z-score
 - c. GEE
 - i. any correlation structure
 - ii. cluster by CpG or by sample
 - d. mixed-model (random slope or intercept)
 - e. SKAT



Provide **all** of these methods with the same interface and compare them.

Implementation: clustermodel

P

-
-

R

```
> library(devtools)
```

```
> install_github("brentp/clustermodelr")
```

-
-

P

Py, per-committee is the central directory, then \clustermodelr, files).

Example Usage

```
python -m clustercorr \  
    'methylation ~ case + age_delivery + insulin_ever' \  
    --gee-args ar,id \  
    covariates.txt \  
    methylation.txt \  
    --min-cluster-size 4 \  
    --rho-min 0.4 --outlier-sds 3 > dmrs.output.bed
```

Simulating Methylation Data

Things that are true:

- We Simulate data so we can tune our algorithms for detecting methylation differences.
 - simulating correlated data is hard (let's go shopping)
 - Assumptions in the simulations drive how we tune the algorithms
-

Simulating Methylation Data

Method 1: Sofer, Tamar, et al. "A-clustering: a novel method for the detection of co-regulated methylation regions, and regions associated with exposure." Bioinformatics (2013): btt498.

1. Real methylation data with 100 samples
2. Find site with multi-CpG correlation
3. weighted random selection of $2 * 20$ samples
 - a. group **H**: weight increases likelihood of choosing sample with high methylation
 - b. group **L**: weight decreases likelihood of choosing sample with high methylation
4. contrast 20 in group **H** vs 20 in group **L** to find differentially methylated region
5. if **weight is 0**, group H should not be different from group L => **random data**
6. measure true+, false+ [where truth is determined by the weight parameter]

Simulating Methylation Data

Method 2: Jaffe, Andrew E., et al. "Bump hunting to identify differentially methylated regions in epigenetic epidemiology studies." International journal of epidemiology 41.1 (2012): 200-209.

1. Choose a CpG-to-CpG correlation
 2. Generate-data with an auto-regressive moving average model (ARMA(1))
 - a. utilize a t-distribution with 5 df (simulates outliers)
 3. insert DMRs at a given beta
 4. They use longer DMRs (10 or 20 probes). But we want to find down to 2 probe DMRs.
-

Simulating Methylation Data

Method 3:

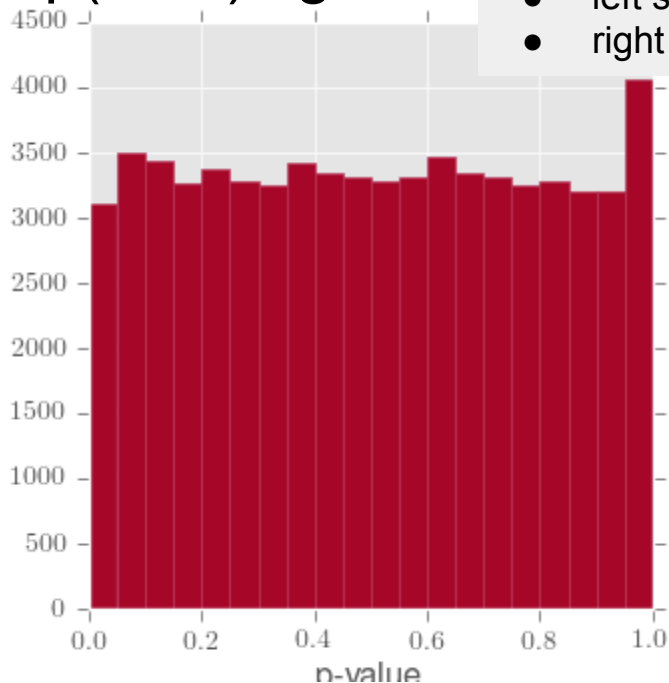
Take existing data and:

- randomize the case/control status
- fit reduced model (with other covariates), shuffle residuals, then randomize case-control status

and check false positives.

Simulation Results (Random Data)

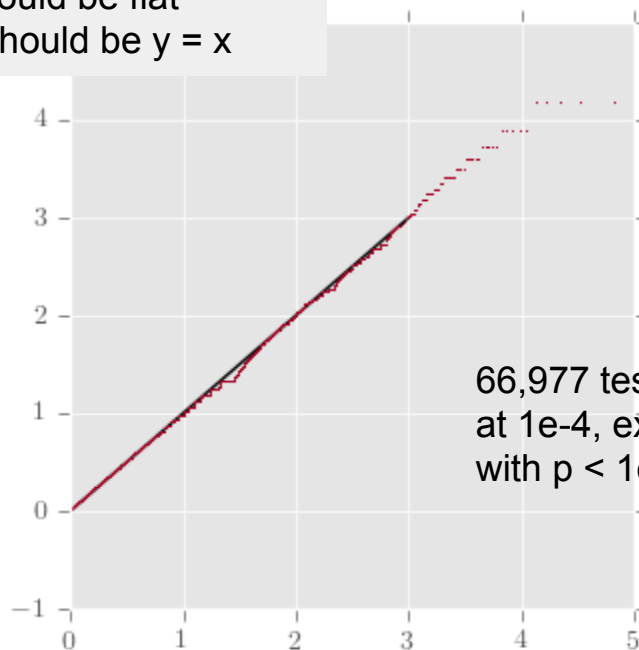
Bump(hunt)ing



Ideally:

- left should be flat
- right should be $y = x$

$$6 < 1e-4$$

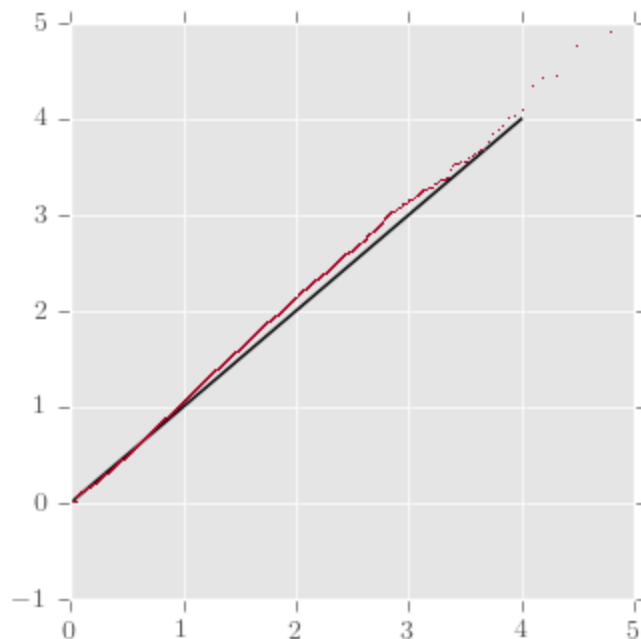
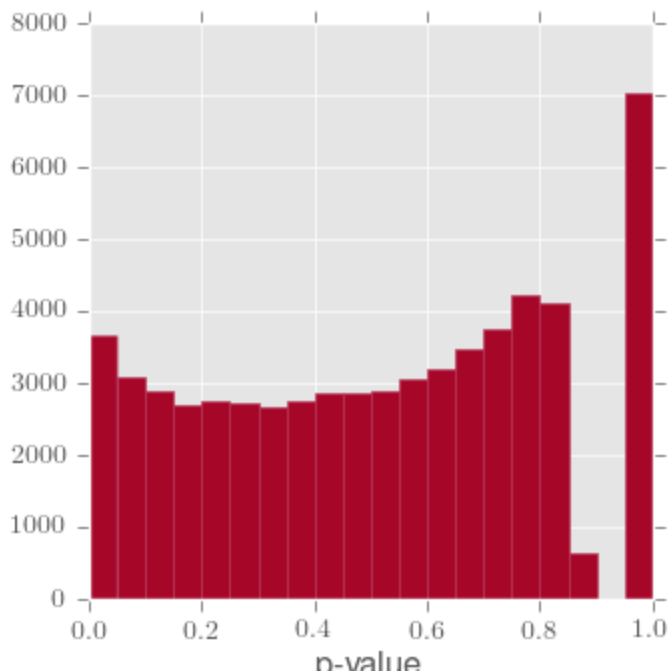


66,977 tested regions
at $1e-4$, expect **6.698**
with $p < 1e-4$

Simulation Results (Random Data)

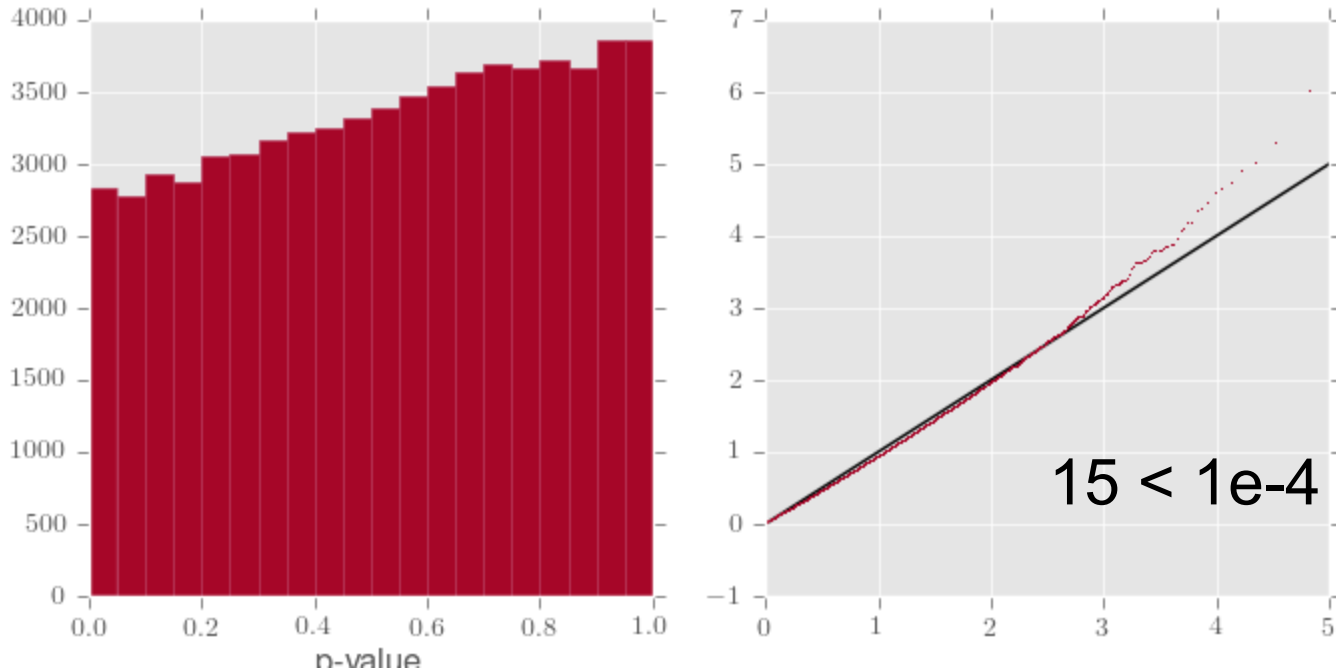
SKAT-O

$8 < 1e-4$



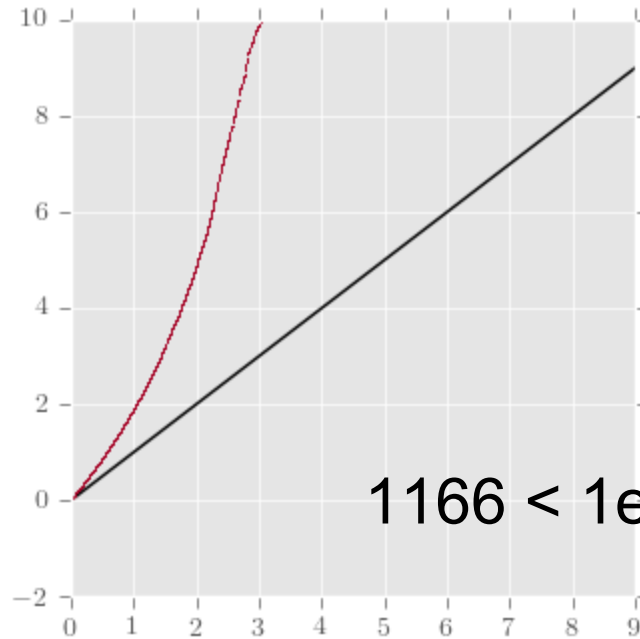
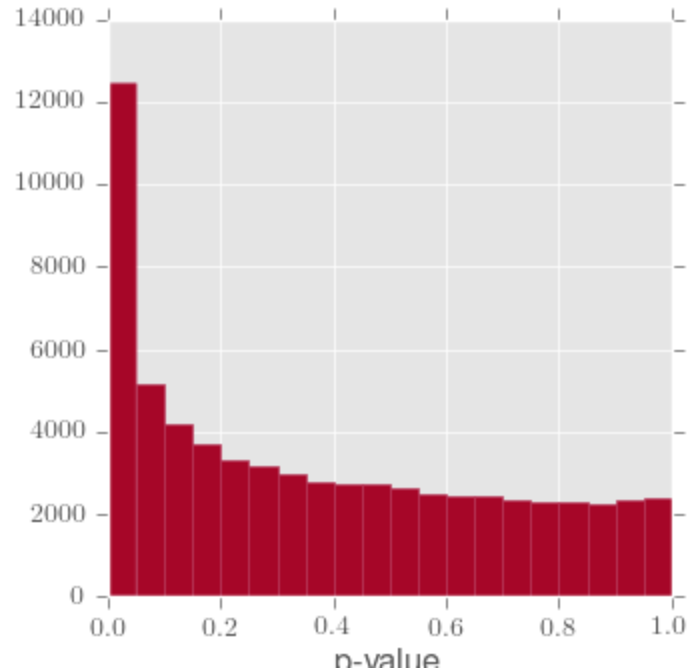
Simulation Results (Random Data)

mixed model: methylation intercept by sample



Simulation Results (Random Data)

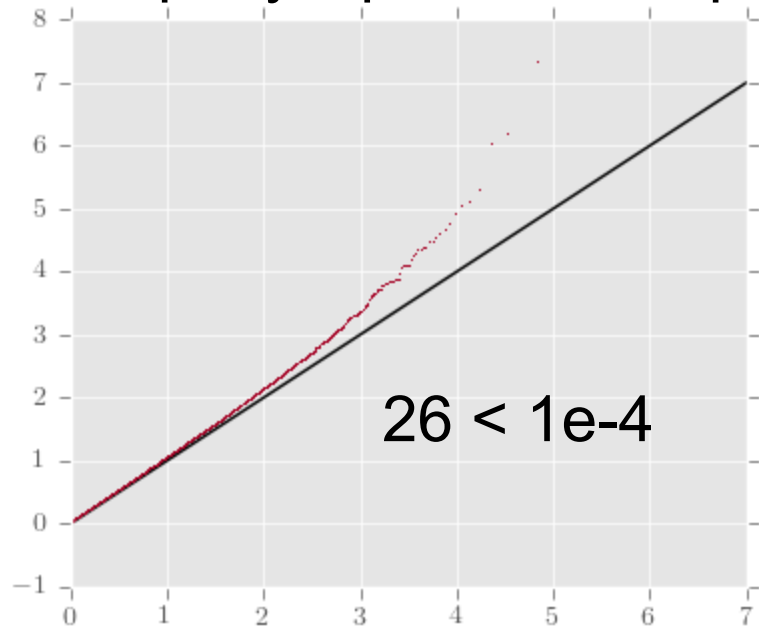
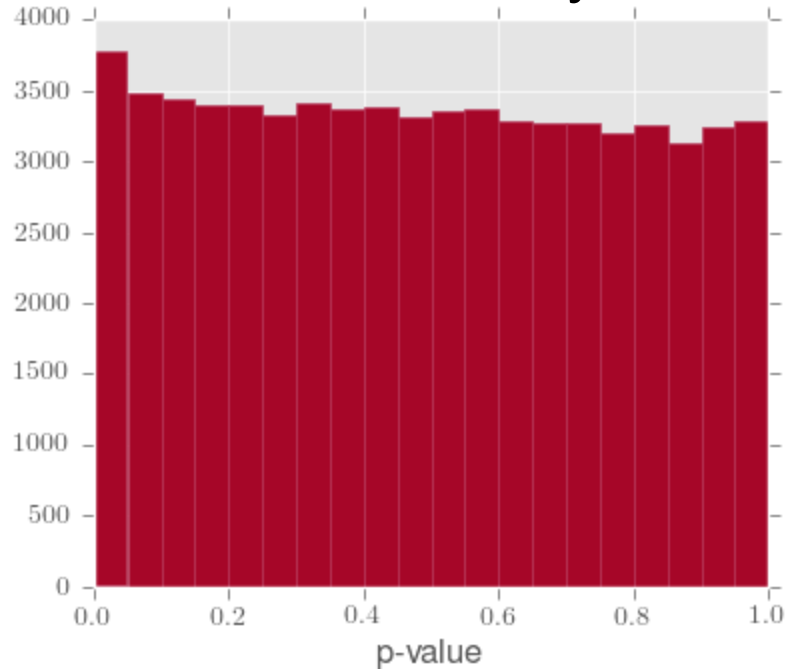
mixed model: methylation intercept by CpG



$1166 < 1e-4$

Simulation Results (Random Data)

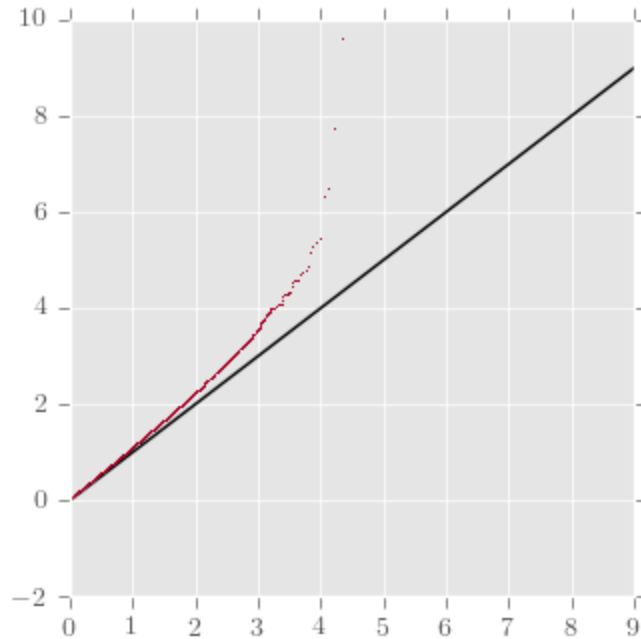
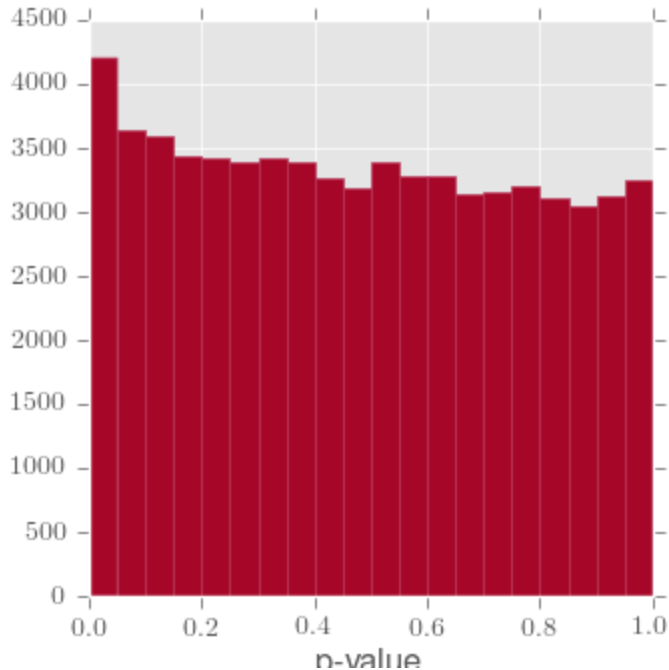
mixed model: methylation intercept by CpG and sample



Simulation Results (Random Data)

GEE: auto-regressive by sample

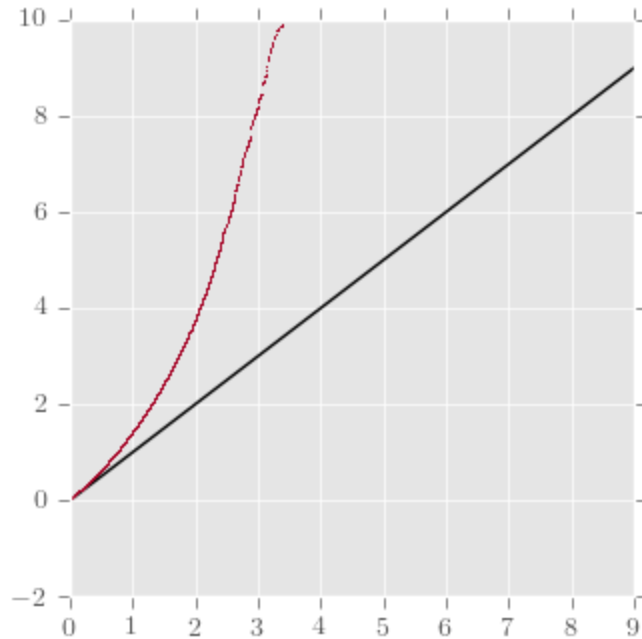
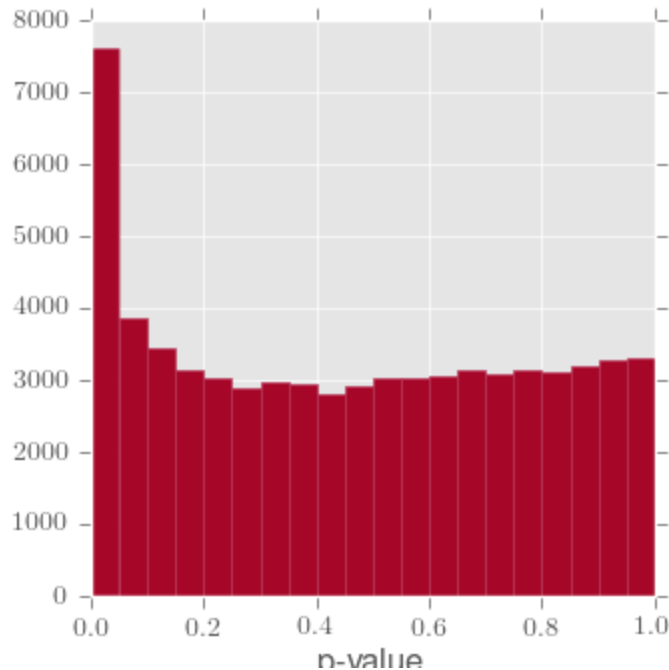
$35 < 1e-4$



Simulation Results (Random Data)

GEE: exchangeable by CpG

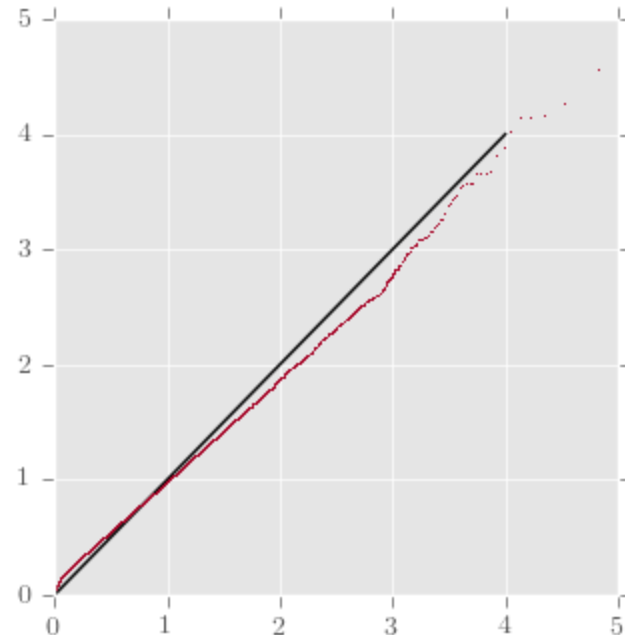
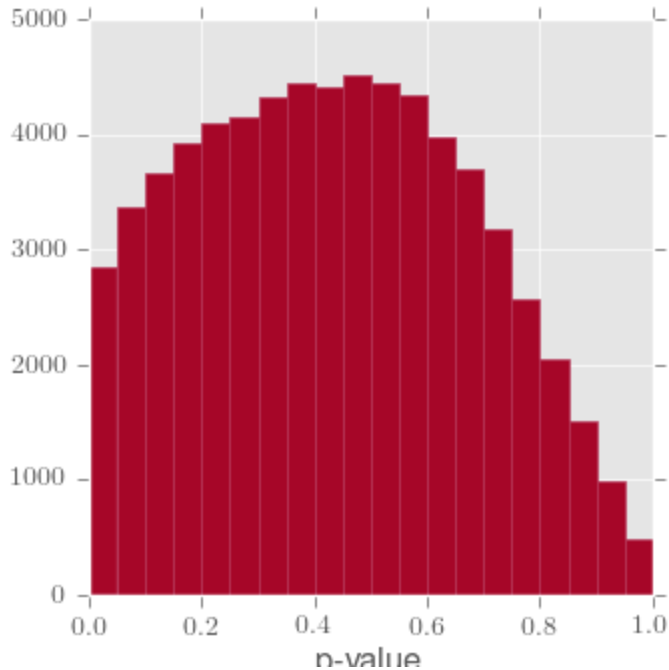
$566 < 1e-4$



Simulation Results (Random Data)

Liptak

$7 < 1e-4$



Simulation Results (Random Data)

How do those change:

- with the size of the DMR?
 - size in simulated data doesn't seem to affect false+ rate
- with the assumptions about correlation? **
- with the number of samples simulated?
 - fewer false+ with more samples?
- with a more complex study design?

Evaluating Methods

false +

true +

BUT:

- Still not using methylation and expression
- Conclusions from output depend on assumptions

0 -

10 samples

20 samples

40 samples

0 -

10 samples

20 samples

40 samples

Remember Example Usage

```
python -m clustermodel \  
    'methylation ~ case + (1|CpG) + (1|id)' \  
    covariates.txt \  
    methylation.txt \  
    > dmrs.output.bed
```

Expression Example Usage

```
python -m clustermodel \  
    'methylation ~ expression + (1|CpG) + (1|id)' \  
    covariates.txt \  
    methylation.txt \  
> dmrs.output.bed
```



Contains an “expression” column.

But, how to test a lot of expression sites
against a lot of methylation sites?

Expression Example Usage

python -m c

'methylation

covariate

methylation

--X expression

--X-locs e

--X-dist 5

> dmrs.out

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
27868	brentp	20	0	217m	120m	4108	R	100	0.5	0:08.47	R
27869	brentp	20	0	217m	120m	4108	R	100	0.5	0:08.47	R
27870	brentp	20	0	217m	120m	4108	R	100	0.5	0:08.46	R
27875	brentp	20	0	220m	123m	4108	R	100	0.5	0:08.45	R
27878	brentp	20	0	216m	120m	4108	R	100	0.5	0:08.46	R
27872	brentp	20	0	217m	121m	4108	R	100	0.5	0:08.44	R
27873	brentp	20	0	217m	121m	4108	R	100	0.5	0:08.45	R
27874	brentp	20	0	217m	121m	4108	R	100	0.5	0:08.44	R
27877	brentp	20	0	218m	122m	4108	R	100	0.5	0:08.45	R
27879	brentp	20	0	217m	121m	4108	R	100	0.5	0:08.41	R
27876	brentp	20	0	217m	120m	4108	R	99	0.5	0:08.41	R
27871	brentp	20	0	216m	120m	4108	R	99	0.5	0:08.40	R

Parallelization important because we're testing up to:
~25K expression probes * 100K methylation regions

2,500,000,000 tests

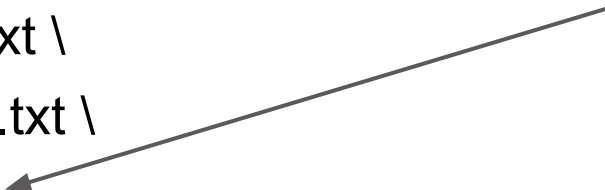
we probably want to reduce the number of tests by looking at local relationships.

matrix.txt
each
methylation.

elized)

Expression Example Usage

```
python -m clustermodel \  
    'methylation ~ (1|CpG) + (1|id)' \  
    covariates.txt \  
    methylation.txt \  
    --X ??? \  
    > dmrs.output.bed
```



Doesn't have to be expression,
can be any matrix of data to
test against.

- OTU's from microbiome
- Long list of covariates
- Genetic Variants

Deep Thoughts

We know that local changes in methylation should affect expression.

Use that to compare methods by finding how many *significant* expression::methylation associations we find:

- in real data (**true +**)
 - in data where expression samples are shuffled relative to methylation (removes expression::methylation association (**false +**))
-

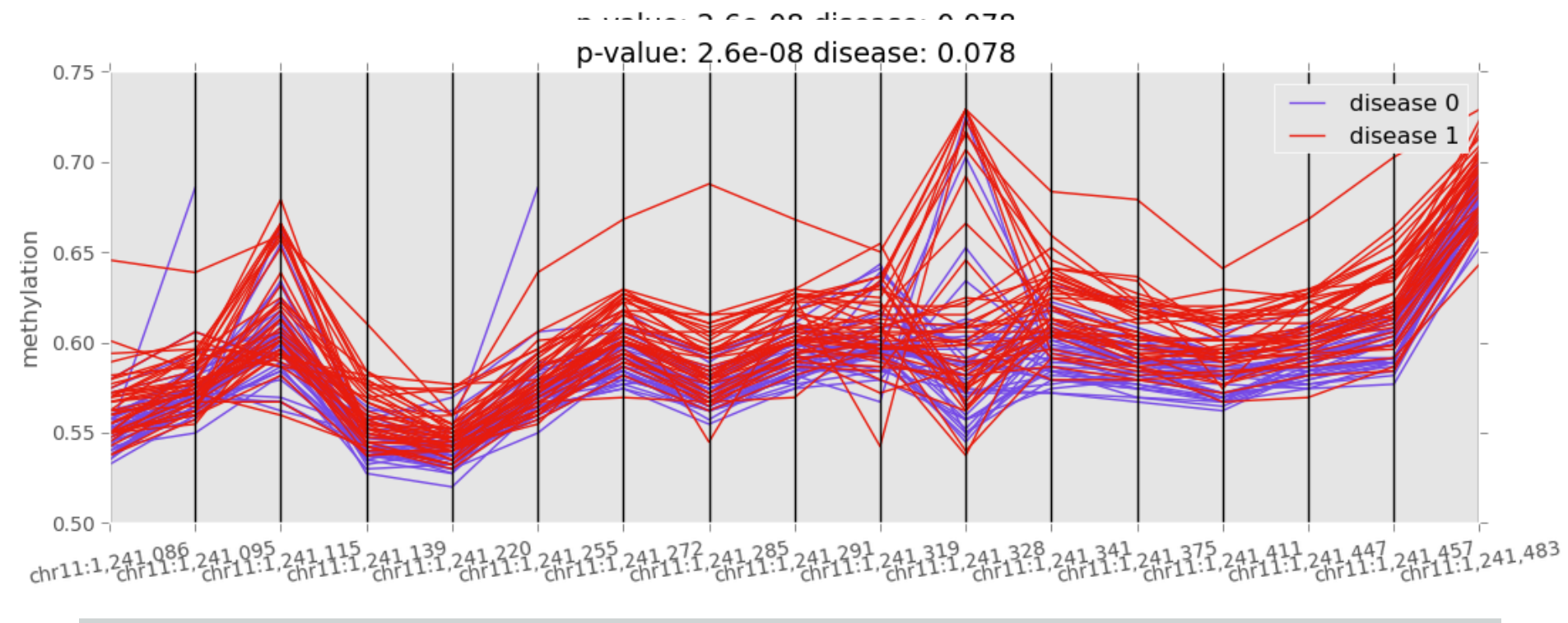
Comparison of Methods

method	outlier_removal	false+	true+	ratio
combine-liptak	NO	117	223523	0.0005234361
combine-z-score	NO	120	235145	0.0005103234
GEE autoregressive	NO	147	274863	0.0005348119
GEE exchangeable	NO	11330	431703	0.0262448952
mixed-effect	NO	112	297993	0.0003758478
combine-liptak	YES	31	174846	0.0001772989
combine-z-score	YES	21	235589	8.913829E-005
GEE autoregressive	YES	175	308030	0.0005681265
GEE exchangeable	YES	9424	423291	0.0222636437
mixed-effect	YES	57	321299	0.0001774048

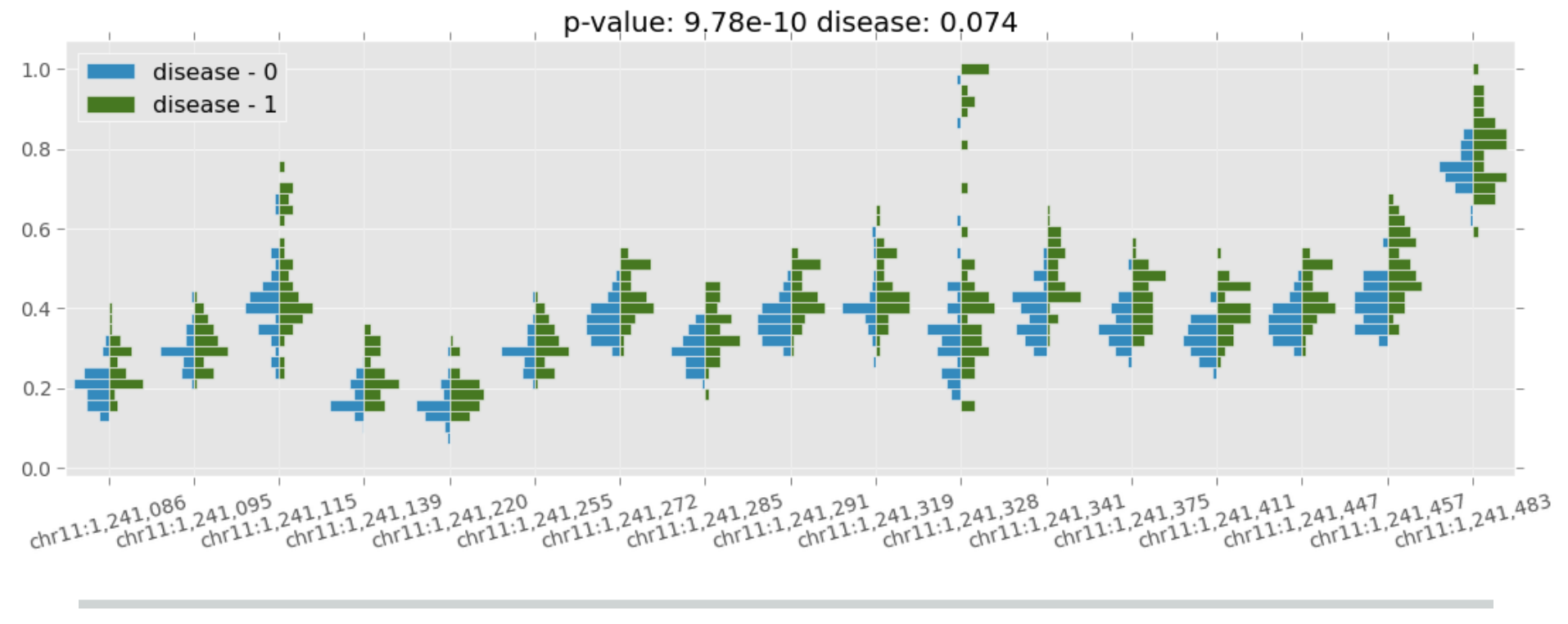
Example Output

#chrom	start	end	coef	p	n_probes	method
chr1	795626	795849	-0.00854791240868	0.294913446774	5	mixed-model
chr1	838227	838399	-0.0110692993468	0.469833850996	6	mixed-model
chr1	841953	842134	0.00983936802679	0.583288570285	6	mixed-model
chr1	847691	847830	0.0113425659212	0.437455916454	5	mixed-model
chr1	860485	860799	0.00207611617104	0.896598686966	5	mixed-model
chr1	876686	876828	0.00926762379034	0.414417426125	5	mixed-model
chr1	880713	880855	-0.0127340057724	0.568003009012	5	mixed-model
chr1	895867	896042	0.00116539019903	0.951352372135	6	mixed-model
chr1	922530	922669	-0.0108648006754	0.374373866703	5	mixed-model

Example Output



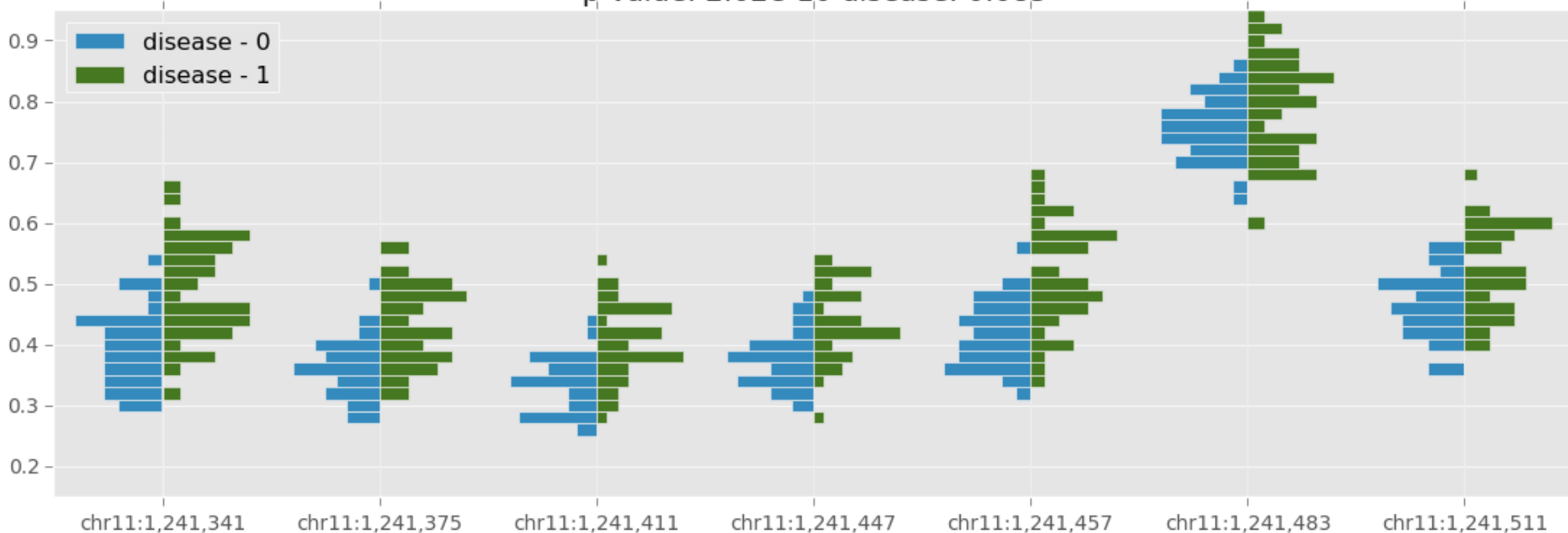
Example Output:



Example Output:

p-value: 4.6e-08 disease: 0.057

p-value: 2.02e-10 disease: 0.085



Example Output:

```
python -m clustercorr \  
    'methylation ~ disease + log2CT + smoking + SNP' \  
    bcovs.txt \  
    bmeth.txt \  
    --gee-args ex,id --rho-min 0.2 --png show
```

Other Uses: genotypes

```
python -m clustercorr \  
  'disease ~ gender' \  
  --skat cluster/ipf.covs.txt \  
  cluster/ipf.genotypes.txt \  
  --min-cluster-size 5 \  
  --max-dist 20000 \  
  --rho-min 0.5 \  
  --linkage complete
```

SKAT compares this null model against one which includes the genotypes. Weights each SNP by AF.

SKAT requires groups of variants, this allows to specify the groups as clusters of correlated sites.

Future Work

- test methods on data simulated by Jaffee *et al* method
 - test and optimize for small samples
 - handle bisulfite-sequencing data (in progress)
 - ensemble methods?
 - **Handle multiple X tests**
-

Thanks

<https://github.com/brentp>
