# *How to calculate a $p-value$ of independence of two genome markups?*

`GenomtriCorr`: *An attempt of a cookbook*

March 24, 2011

# Outline

*Introduction*

- Introduction itself
- Different senses of correlation

*Local correlation*

- Kolmogorov-Smirnov
- The sign of correlation

*Chromosome-scale correlation*

- Absolute distance test
- Bernoulli test
- Naïve Jaccard approach

*Genomewide*

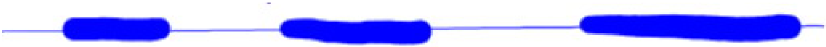
*R implementation*

- Installation
- Usage
- Technical

*Conclusions*

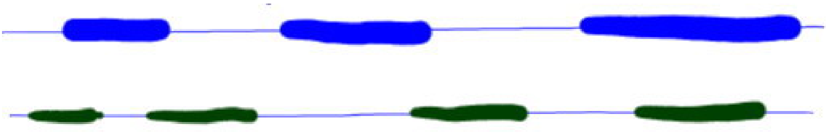
# *A markup*

What is it a markup?



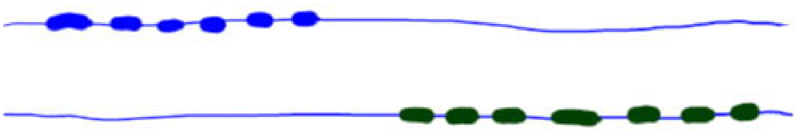
What we refer to as a markup is whatever we can represent as a set of intervals on chromosomes. In other words, it is a spatial annotation of a genome. It could be any interval annotation on genome: genes, upstreams, TFBS, clusters, CpG islands, etc...

## *Two markups*



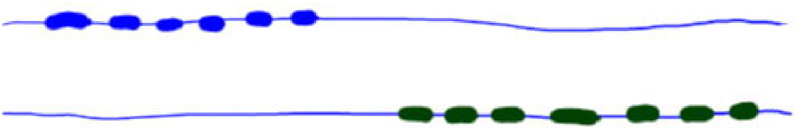
Are these two things independent? What does it mean? $p - value$? Let's say one of two markups (query) is independent from the other markup (reference) if the query is positioned in a manner that is 'blind' to the scattering of reference. The relation is asymmetric.

## *Different senses of correlation*

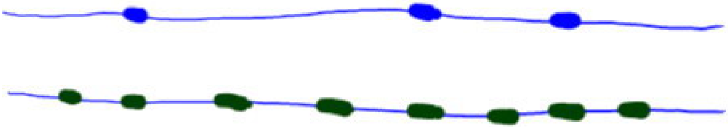


Chromosome-scale ("global") negative correlation.

# *Different senses of correlation*

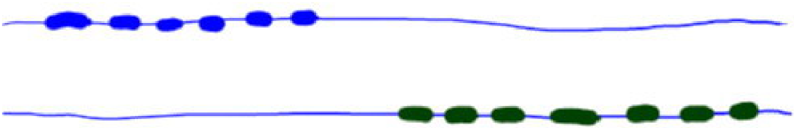


Chromosome-scale (“global”) negative correlation.

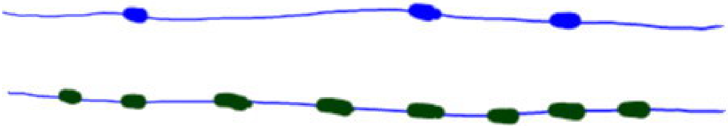


'Local' positive correlation.

# *Different senses of correlation*



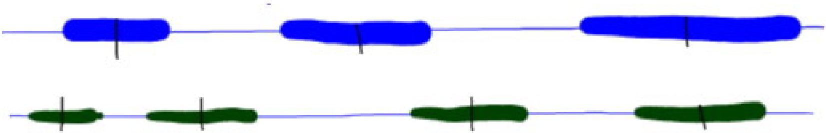
Chromosome-scale (“global”) negative correlation.



'Local' positive correlation.

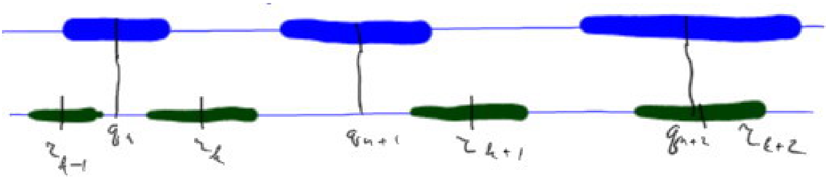
Asymmetric relation. Query and reference.

# *Local correlation: contracted intervals*



First of all, we contract all the intervals, both query and reference, into their characteristic points (middles).

# *Local correlation: relative distances and Kolmogorov-Smirnov*

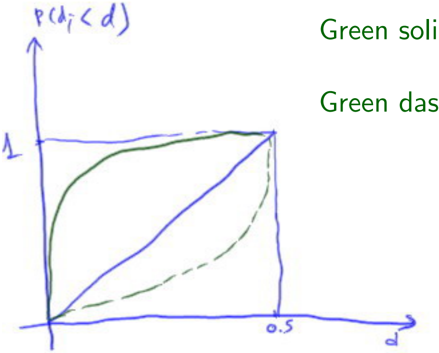


So, relative distance $d_i$ for a query point $i$ is:

$$d_i = \frac{\min\left(\left|q_i - r_k\right|, \left|r_{k+1} - q_i\right|\right)}{\left|r_{k+1} - r_k\right|}, k = \arg\min_{q_i \geq r_k}\left(q_i - r_k\right).$$

If the markups are locally independent, the $d_i$'s are to be uniformly i.i.d. (u.i.i.d) in [0..0.5]. The corresponding $p - value$ is obtained by Kolmogorov-Smirnov's test.

# *Local correlation: The sign of correlation*

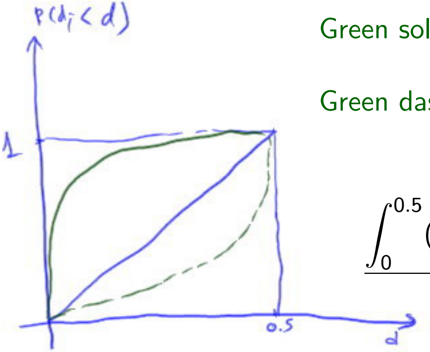


Blue line: theoretical distribution for independence (uniform)

Green solid line: they like each other

Green dash line: they dislike each other

# *Local correlation: The sign of correlation*



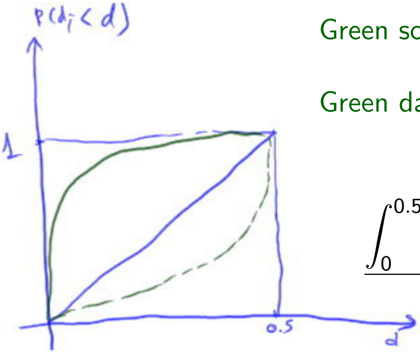
Blue line: theoretical distribution for independence (uniform)

Green solid line: they like each other

Green dash line: they dislike each other

$$Corr_{ECDF} = \frac{\int_0^{0.5} (ECDF(d) - ECDF_{ideal}(d))\, dd}{\int_0^{0.5} ECDF_{ideal}(d)\, dd}.$$

# *Local correlation: The sign of correlation*



Blue line: theoretical distribution for independence (uniform)

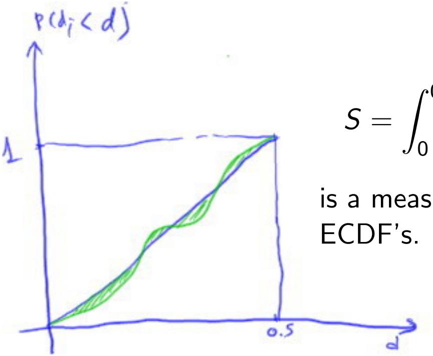
Green solid line: they like each other

Green dash line: they dislike each other

$$Corr_{ECDF} = \frac{\int_0^{0.5} (ECDF(d) - ECDF_{ideal}(d))\, dd}{\int_0^{0.5} ECDF_{ideal}(d)\, dd}.$$

Positive $Corr_{ECDF}$ shows positive local correlation (the distribution density is shifted towards 0) and vice versa.

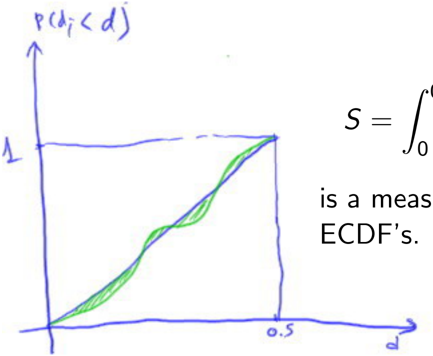
# Local correlation: ECDF area test



$$S = \int_0^{0.5} |ECDF(d) - ECDF_{ideal}(d)| \; dd$$

is a measure of discrepancy of real and ideal ECDF's.

## *Local correlation: ECDF area test*

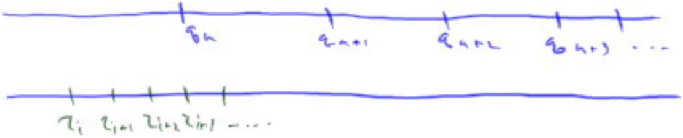


$$S = \int_0^{0.5} |ECDF(d) - ECDF_{ideal}(d)| \; dd$$

is a measure of discrepancy of real and ideal ECDF's.

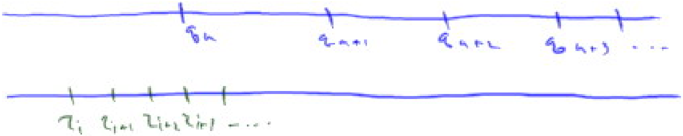
Permutations: drawing $N$ sets of $d_i$ we get $N$ outcomes for "null-hypothesis" $S$ and we get $p - value$ for S.

# *Chromosome-scale correlation: Absolute distance test*



For each query point $i$, $l_i = \min_k(q_i - r_k)$ is found.
$L = \langle l_i \rangle$ characterises the "attraction" or "repulsion" of query and reference points.

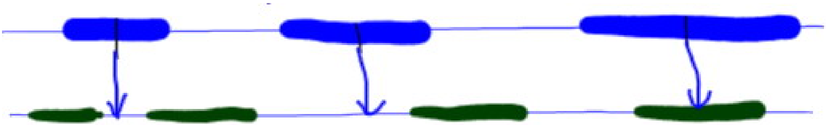
# Chromosome-scale correlation: Absolute distance test



For each query point $i$, $l_i = \min_k(q_i - r_k)$ is found. $L = \langle l_i \rangle$ characterises the "attraction" or "repulsion" of query and reference points.

Permutations: we draw $N$ pseudo-queries as sets of u.i.i.d. points, calculating "null" for $L$. The test is two-sided, it gives both $p - value$ for the real $L$ and the sign of effect if there is one.

## *Chromosome-scale correlation: Bernoulli test*

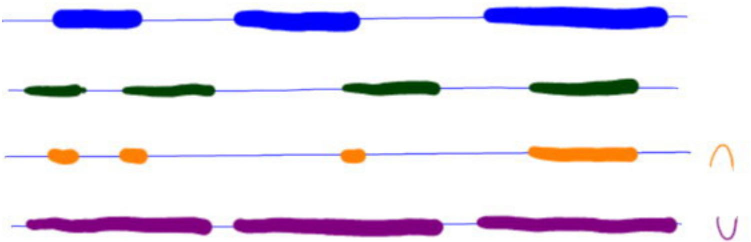


If the coverage of the reference is high, we can use Bernoulli test. We contract only the query. The probability for a query point to get into a reference interval is:

$$p = \frac{\text{coverage of the reference}}{\text{chromosome length}}.$$

The number of "successes" is approximately Bernoulli with the parameters $\#q$ and $p$. The test is two-sided; it provides both $p - value$ and the direction.

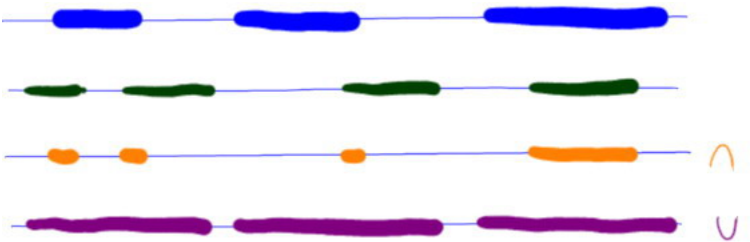
# *Chromosome-scale correlation: Naïve Jaccard approach*



The coverage is high. Now, both markups are sets of nucleotides.

$$\text{Jaccard measure (index):} \quad J(A,B) = \frac{A \cap B}{A \cup B}$$

# *Chromosome-scale correlation: Naïve Jaccard approach*



The coverage is high. Now, both markups are sets of nucleotides.

$$\text{Jaccard measure (index): } J(A,B) = \frac{A \cap B}{A \cup B}$$

Permute the query. Two kinds of permutation a) permute the starts b)permute the intervals order and permute the gaps order.

# *Genomewide tests*

All the test described above are applicable to the genome awhole. The data for the criteria is summarised over the chromosomes. The absolute distances are scaled by the expectation of the distance between adjacent reference points. Then, all the tests are run for the accumulated data in the same way as it is done for each chromosome.

# *R implementation*

- Based on IRanges

# R implementation

- Based on IRanges
- Utilities: read test files and visualise IRanges

# *R implementation*

- Based on IRanges
- Utilities: read test files and visualise IRanges
- Main procedure

# *R implementation*

- Based on IRanges
- Utilities: read test files and visualise IRanges
- Main procedure
- `GenomtriCorr package`
  http://genometricorr.sourceforge.net/

## *Let's install the package*

- In R:

```
source("http://bioconductor.org/biocLite.R")
biocLite("IRanges")
```

## *Let's install the package*

- In R:
```
source("http://bioconductor.org/biocLite.R")
biocLite("IRanges")
```

- In shell:
```
R CMD INSTALL GenometriCorr_1.02.tar.gz
```

# *Let's install the package*

- In R:
```
source("http://bioconductor.org/biocLite.R")
biocLite("IRanges")
```

- In shell:
```
R CMD INSTALL GenometriCorr_1.02.tar.gz
```

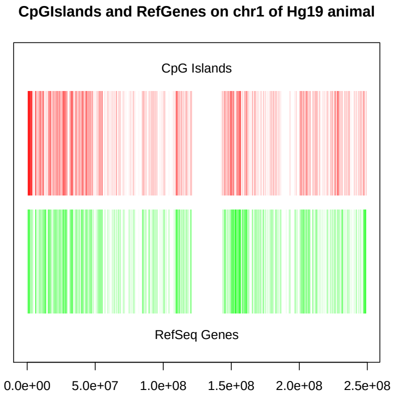
- In R:
```
library("GenometriCorr")
```

# *Utilities: read*

```
USCSrefseqgenesURL<-’http://genome.ucsc.edu/cgi-bin/hgTables?db=hg19&hgta_database=hg19&
hgta_group=genes&hgta_track=refGene&
hgta_table=refGene&hgta_regionType=genome&hgta_outputType=primaryTable&
hgta_fieldSelectTable=hg19.refGene&hgta_fs.check.hg19.refGene.chrom=1&hgta_fs.check.hg19.refGene.name=1&
hgta_fs.check.hg19.refGene.txEnd=1&hgta_fs.check.hg19.refGene.txStart=1&hgta_doPrintSelectedFields=&’

USCScpgisURL<-’http://genome.ucsc.edu/cgi-bin/hgTables?clade=mammal&command=start&db=hg19&
hgta_database=hg19&hgta_fieldSelectTable=hg19.cpgIslandExt&hgta_fs.check.hg19.cpgIslandExt.chrom=1&
hgta_fs.check.hg19.cpgIslandExt.chromEnd=1&hgta_fs.check.hg19.cpgIslandExt.chromStart=1&
hgta_fs.check.hg19.cpgIslandExt.cpgNum=0&hgta_fs.check.hg19.cpgIslandExt.gcNum=0&
hgta_fs.check.hg19.cpgIslandExt.length=0&hgta_fs.check.hg19.cpgIslandExt.name=0&
hgta_fs.check.hg19.cpgIslandExt.obsExp=0&hgta_fs.check.hg19.cpgIslandExt.perCpg=0&
hgta_fs.check.hg19.cpgIslandExt.perGc=0&hgta_group=regulation&hgta_outputType=primaryTable&
hgta_regionType=genome&hgta_table=cpgIslandExt&hgta_track=cpgIslandExt&hgta_doPrintSelectedFields=&
org=Human&’

refseq <- readTableToIRanges(USCSrefseqgenesURL, comment.char = "$", header = T)

cpgis <- readTableToIRanges(USCScpgisURL,comment.char = "$", header = T)
```

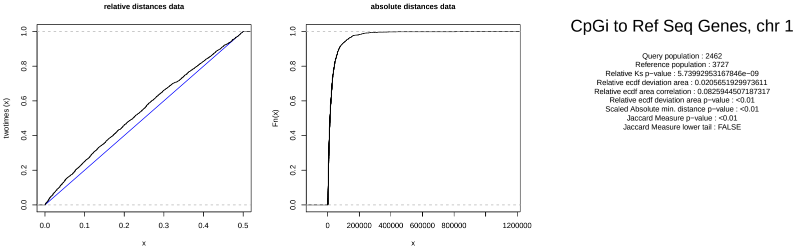
# *Utilities: visualise*

```
VisualiseTwoIRanges(cpgis["chr1"]$ranges, refseq["chr1"]$ranges, nameA = "CpG Islands", nameB =
"RefSeq Genes", chrom_length = human.chrom.length[["chr1"]], title = "CpGIslands and RefGenes on
chr1 of Hg19 animal")
```



# *Main procedure: GenometricCorrelation*

```
cpgi_to_genes <- GenometricCorrelation(cpgis, refseq, chromosomes.length = human.chrom.length,
chromosomes.to.proceed = c("chr1"), ecdf.area.permut.number = pn.area,
mean.distance.permut.number = pn.dist, jaccard.measure.permut.number = pn.jacc,
keep.distributions = TRUE, showProgressBar = FALSE)
```



CpGi to Ref Seq Genes, chr 1

```
Query population : 2462
Reference population : 3727
Relative Ks p−value : 5.73992953167846e−09
Relative ecdf deviation area : 0.0205651929973611
Relative ecdf area correlation : 0.0825944507187317
Relative ecdf deviation area p−value : <0.01
Scaled Absolute min. distance p−value : <0.01
Jaccard Measure p−value : <0.01
Jaccard Measure lower tail : FALSE
```

## *Some technical issues:R*

- In R:
  `package.skeleton()`

## Some technical issues:R

- In R:
  ```
  package.skeleton()
  ```

- In shell:
  ```
  R CMD check GenometriCorr

  R CMD build GenometriCorr
  ```

## *Some technical issues:Documentation*

- In R:

  ```
  Sweave('GenometricCorrelationPackage.Rnw')
  ```

  In shell:

  ```
  R CMD Sweave GenometricCorrelationPackage
  ```

## *Some technical issues:Documentation*

- In R:
  ```
  Sweave('GenometricCorrelationPackage.Rnw')
  ```
  In shell:
  ```
  R CMD Sweave GenometricCorrelationPackage
  ```
- In shell:
  ```
  echo "library(weaver);
  Sweave('GenometricCorrelationPackage.Rnw',
  driver=weaver())" | R --no-save --no-restore
  ```

Alexander Favorov

Loris Mularoni

Leslie Cope

Yulia Medvedeva

Vsevolod Makeev

Sarah Wheelan

# *Conclusions*