cobind Documentation

Liguo Wang

Aug 11, 2023

DOCUMENTATION

CHAPTER

ONE

INTRODUCTION

1.1 Introductionn

Collocated genomic intervals indicate biological association. Therefore, overlapping analysis of genomic intervals has been widely used to QC, integrate, and impute the function of genomic intervals.

The conventional approach of measuring the "overlap between genomic intervals" involves arbitrary thresholds to decide the total number of overlapped genomic regions, which leads to biased, non-reproducible, and incomparable results. Specifically,

- The result derived from this *threshold-and-count* approach is non-reproducible and incomparable, as different thresholds produce different results.
- The overlapping between two intervals is a continuous variable, whereas the thresholded approach reduces it into a binary variable. Casting the one-dimensional intervals as zero-dimensional points loses the information and sensitivity needed to accurately evaluate the collocation strength.
- The absolute or relative counts is biased by the size and the total number of intervals.

To address these limitations, **cobind** offers six threshold-free metrics that rigorously quantify the strength of genomic overlapping. These metrics aim to provide more reliable and comparable results without arbitrary thresholds.

- the Collocation coefficient (C)
- the Jaccard coefficient (J)
- the Sørensen–Dice coefficient (SD)
- the Szymkiewicz–Simpson coefficient (SS)
- the Pointwise Mutual Information (PMI)
- the Normalized Pointwise Mutual Information (NPMI)

CHAPTER

TWO

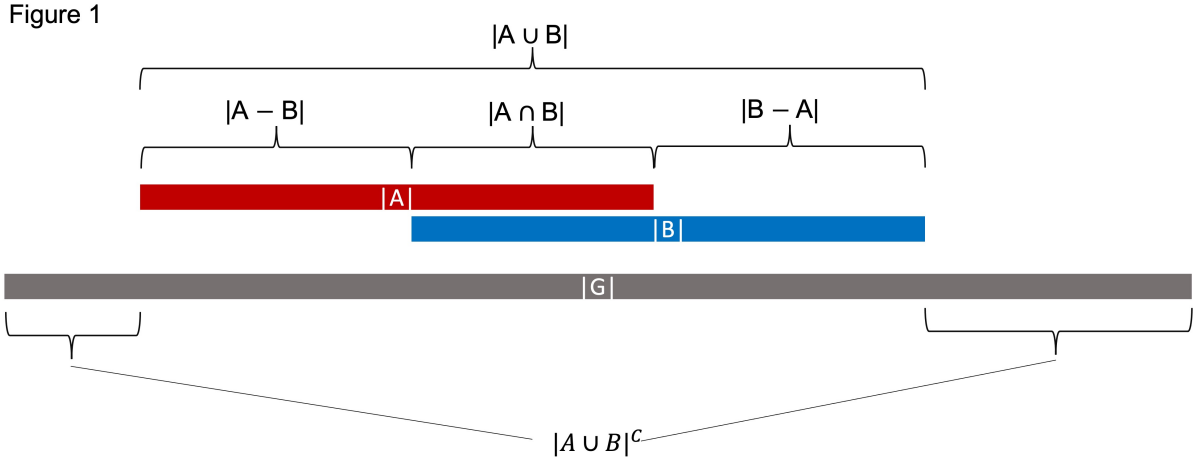
DEFINITIONS

2.1 Symbols definitions

We have two sets of genomic intervals **A** and **B**, and the genomic background is **G**. In Figure 1 below, both A and B contain only one genomic region for the purpose of clarity, but the definitions are still applicable if **A** and **B** have many intervals.

Symbols are defined as:

Figure 1



|A|
: The cardinality of **A** (i.e., all the **non-redundant** bases covered by **A**). For example, if A contains two genomic intervals: "chr1 0 10", "chr1 5 15", then |A| = 15.

|B|
: The cardinality of **B** (i.e., all the **non-redundant** bases covered by **B**).

|G|
: The genomic background. Depending on the context, this can be *the whole genome*, *all the cis-regulatory elements*, *all the promoters*, *all the TF binding sites* in the genome, etc. **A** and **B** must be the subsets of **G**.

|A B|
: Union of A and B (i.e., bases covered by A or B).

|A B|
: Intersection of A and B (i.e., bases covered by A and B simultaneously). This is commonly used to measure the *collocation* of A and B.

|A B|
: Difference (A not B) (i.e., bases covered by only A but not B).

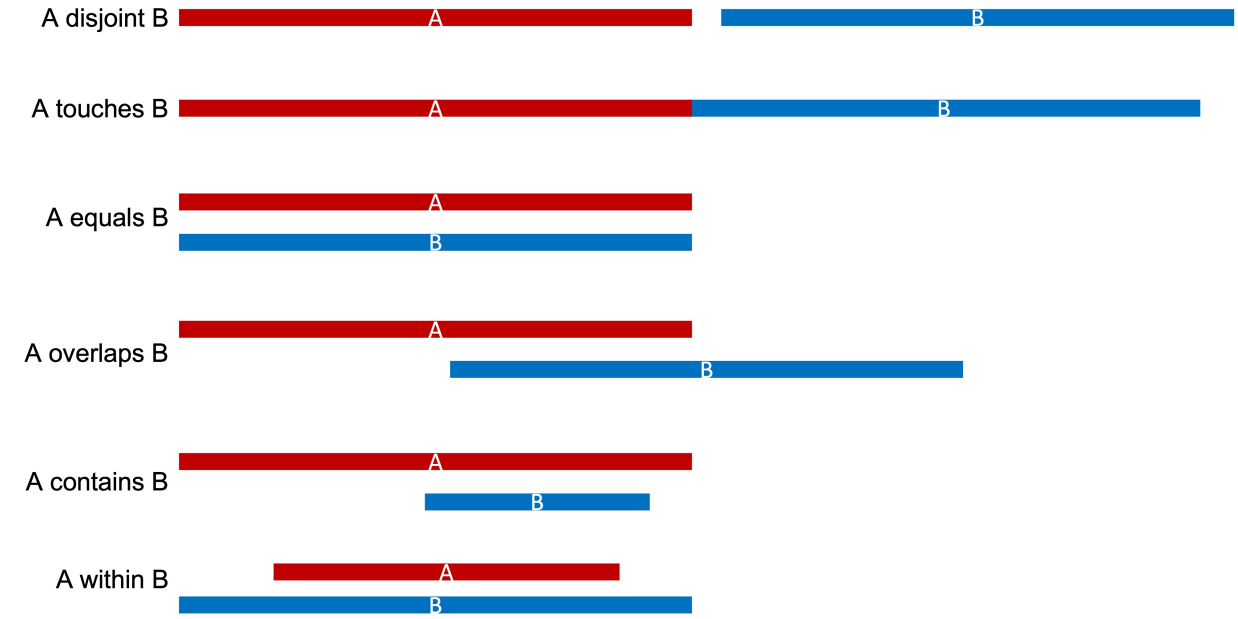
|B A|
Difference (B not A) (i.e., bases covered by only B but not A).

|A B|^
Complement of |A B| (i.e., bases NOT covered by A or B).

2.2 Spacial Relations of Genomic regions (SROG)

There are six different spacial relations between two genomic regions (A and B). As illustrated below:

Figure 2



2.3 Collocation coefficient (C)

The collocation coefficient between A and B is calculated as the ratio between |A B| and the *geometric mean of* |*A*| *and* |*B*|. C(A,B) is a value between [0, 1], with 0 indicating 'no overlap', and 1 indicating '100% overlap' (i.e., A and B are identical). C(A, B) is defined as 0 when |A| = 0 or |B| = 0, or |A| = |B| = 0.

$$C(A,B) = \frac{|A \cap B|}{\sqrt{|A| \times |B|}}$$

$$0 \leq C(A,B) \leq 1$$

Overall collocation coefficient
The collocation coefficient between two **sets** of genomic regions. For example, you can use the *overall collocation coefficient* to measure the cobindability of two transcription factors.

peakwise collocation coefficient
The collocation coefficient between **two** genomic intervals (A protein-bound genomic region is called "peak" in ChIP-seq experiment).

2.4 Jaccard coefficient (J)

The Jaccard similarity coefficient, also known as the Jaccard index. It is the ratio between **intersection** and **union**. J(A, B) is defined as 0 when |A| = 0 or |B| = 0, or |A| = |B| = 0.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

$$0 \leq J(A,B) \leq 1$$

The Jaccard distance *Dj* is calculated as:

$$\mathrm{D}_j(A,B) = 1 - J(A,B) = \frac{|A \cup B| - |A \cap B|}{|A \cup B|}$$

Similar to O(A,B), we have an **overall Jaccard coefficient** and **peakwise Jaccard coefficient**.

Note: The Jaccard coefficient implemented here is slightly different from BEDTools `jaccard` function. When calculating the union, BEDTools only use the intervals that are overlapped with each other, while we use all the intervals.

---

overall Jaccard coefficient
The Jaccard coefficient between two **sets** of genomic regions.

peakwise Jaccard coefficient
The Jaccard coefficient between **two** genomic intervals.

2.5 Sørensen–Dice coefficient (SD)

Sørensen–Dice coefficient, also called *Sørensen–Dice index*, *Sørensen index* or *Dice's coefficient*. SD(A, B) is defined as 0 when |A| = 0 or |B| = 0, or |A| = |B| = 0.

$$SD(A,B) = \frac{2|A \cap B|}{|A| + |B|}$$

$$0 \leq SD(A,B) \leq 1$$

Jaccard coefficient (J) can be converted into Sørensen–Dice coefficient (SD) and vice versa:

J = SD/(2-SD) and *SD = 2J/(1+J)*

2.6 Szymkiewicz–Simpson coefficient (SS)

Szymkiewicz–Simpson coefficient is defined as the size of the intersection divided by the smaller size of the two sets.

$$SS(A,B) = \frac{|A \cap B|}{\min(|A|,|B|)}$$

$$0 \leq SS(A,B) \leq 1$$

2.7 Pointwise mutual information (PMI)

Pointwise mutual information (PMI) is one of the standard association measures in collocation analysis. It measures how much the observed overlaps differ from what we would expect them to be. Assume A and B represent two sets of genomic regions bound by transcription factors A and B; respectively, PMI measures if A and B bind together or separately.

PMI is calculated as:

$$pmi(A \cap B) \equiv log\left(\frac{p(A \cap B)}{p(A) \times p(B)}\right)$$

where

$$\mathrm{p(A)} = \frac{|A|}{|G|}, \mathrm{p(B)} = \frac{|B|}{|G|}, p(A \cap B) = \frac{|A \cap B|}{|G|}$$

PMI = 0
: Indicates that A and B are independent.

PMI > 0
: Indicates that the overlapping between A and B is in a frequency *higher* than what we would expect if A and B are independent (i.e, A and B tend to bind together).

PMI < 0
: Indicates that the overlapping between A and B is in frequency *lower* than what we would expect if A and B were independent. (i.e., A and B tend to bind separately).

Note, PMI has no boundaries:

$$-\infty \leq pmi(A \cap B) \leq min(-log(p(A)), -log(p(B)))$$

2.8 Normalized pointwise mutual information (NPMI)

Normalized pointwise mutual information (NPMI) is calculated as:

$$npmi(A \cap B) = \frac{pmi(A \cap B)}{-log(p(A \cap B))} = \frac{log\left(\frac{p(A \cap B)}{p(A) \times p(B)}\right)}{-log(p(A \cap B))} = \frac{log(p(A) \times p(B))}{log(p(A \cap B))} - 1$$

Note, after normalization, NPMI is confined to [-1, 1]:

$$-1 \leq npmi(A \cap B) \leq 1$$

2.9 Which metric to use?

Use the Z-score approach to combine all the six metrics as an overall measure, or choose the **Collocation coefficient (C)** and **NPMI** which generally performs better than other approaches.

Metric evaluation

CHAPTER

THREE

INSTALLATION

cobind is written in Python. Python3 (v3.5.x) is required to run all programs in cobind.

3.1 Dependencies

- pandas
- numpy
- scipy
- bx-python
- pyBigWig

Note: These packages will be automatically installed when you use pip3 to install cobind.

3.2 Install pip

Please install pip (Package Installer for Python) first if you do not have it.

```
#check if pip is available.
$ pip --version
pip 23.0.1 from /Users/m102324/miniconda3/lib/python3.10/site-packages/pip (python 3.10)
```

3.3 Install to virtual environment

Python's **Virtual Environments** allow Python packages to be installed in an isolated location rather than being installed globally. If you would like to install *cobind* into a virtual environment, please follow these instructions. Specifically, follow these steps:

```
$python3 -m venv cobind
$source cobind/bin/activate
$pip install cobind
```

3.4 Install globally

Install *cobind* using pip from PyPI or GitHub

```
$ pip install cobind
#or
$ pip install git+https://github.com/liguowang/cobind.git
```

3.5 Upgrade

```
$ pip install cobind --upgrade
```

3.6 Uninstall

```
$ pip uninstall cobind
```

CHAPTER

FOUR

INPUT FILE AND DATA FORMAT

4.1 BED format

BED (Browser Extensible Data) format is commonly used to describe genomic intervals. Standard BED file has 12 columns, but **cobind** only requires the first three columns (all the other columns are optional):

```
# BED3 format (chrom, start, end)
chr1    629149    629391
chr1    629720    630165
chr1    631404    631758
...

# BED4 format (chrom, start, end, name)
chr1    629149    629391   region_1
chr1    629720    630165   region_2
chr1    631404    631758   region_3
...

# BED6 format (chrom, start, end, name, score, strand)
chr1    629149  629391 region_1    0    +
chr1    629720  630165 region_2    0    +
chr1    631404  631758 region_3    0    -
...
```

4.2 BED-like format

- bedgraph
- ENCODE narrowpeak
- ENCODE broadpeak
- ENCODE gappedpeak

4.3 bigBed

bigBed is an indexed binary format of a BED file. UCSC's `bedToBigBed` and `bigBedToBed` commands can be used to convert BED files into bigBed files or *vice versa*.

4.4 bigWig

The bigWig format is an indexed binary format of a wiggle file, which is widely used to represent genomic signals. UCSC's `wigToBigWig` and `bigWigToWig` commands can be used to convert wiggle files into bigWig files or *vice versa*.

CHAPTER

FIVE

TEST DATASET

5.1 CTCF ChIP-seq

Project	ENCODE
Lab	Michael Snyder, Stanford
TF	CTCF (CCCTC-binding factor)
Bio sample	Homo sapiens K562
Reference genome	GRCh38
narrowPeak (bed)	ENCFF660GHM.bed.gz (md5sum = 2b9e2c2ba7afe8d64f5f3549ce16cf1a)
narrowPeak (bigBed)	ENCFF400DFR.bigBed (md5sum = 15bf51e2a37b8d93b44c8746b83583b4)
signal Pvalue (bigWig)	ENCFF336UPT.bigWig (md5sum = 883eb33a975e14130e142b98070b14c0)

5.2 RAD21 ChIP-seq

Project	ENCODE
Lab	Michael Snyder, Stanford
TF	RAD21
Bio sample	Homo sapiens K562
Reference genome	GRCh38
narrowPeak (bed)	ENCFF057JFH.bed.gz (md5sum = 0e638759eb09e8d0825d3d124b2c77d6)
narrowPeak (bigBed)	ENCFF066JWO.bigBed (md5sum = 92d3e303d3d880db5c9d604823e9831d)
signal Pvalue (bigWig)	ENCFF130GMP.bigWig (md5sum = c7e73bd2fba6a21a9d02da181e303578)

5.3 Other files

These files can be used as genomic "background".

Dataset (Human, GRCh38/hg38)	md5sum
remap2022_CRM_hg38_v1_0.bed.gz	4717178cd730471f5ac897838c55847c
ENCODE_CCRE_hg38.bed.gz	a572f25b1f7a51283591f4afd8f0c3b7
GeneHancer_v4.4_hg38.bed.gz	e6fbecf8f637db49ce12d1390c6285b6
CpG_island_hg38.bed.gz	8c783529fb4a8f86b1d90d70afa6a1f7

Dataset (Mouse, GRCm39/mm39)	md5sum
remap2022_CRM_mm39_v1_0.bed.gz	9c20058b6ab324f2292029566e59993a

Dataset (Fly, dm6)	md5sum
remap2022_CRM_dm6_v1_0.bed.gz	3633180a8cba0495147682cc9b288aca

CHAPTER

SIX

RELEASE HISTORY

6.1 Version 1.0.0

Initial release

6.2 Version 1.0.1

1. add *-l* or *–log* options to save log information to the file. If not specified, log information will be printed to the screen.
2. add *–nameA* and *–nameB* to represent the two input genomic intervals. If not specified, the names of the input files will be used.
3. add the 'zscore' command to calculate the combined Z-score of the six metrics.

CHAPTER

SEVEN

OVERVIEW

7.1 Subcommands description

cobind is a python package designed to quantify the "overlapping" or "collocation" of genomic intervals.

Table 1: **subcommands provided by cobind**

Subcommand	Description
overlap	Calculate the collocation coefficient (C).
jaccard	Calculate the Jaccard similarity coefficient (J).
dice	Calculate the Sørensen–Dice coefficient (SD).
simpson	Calculate the Szymkiewicz–Simpson coefficient (SS).
pmi	Calculate the pointwise mutual information (PMI).
npmi	Calculate the normalized pointwise mutual information (NPMI).
cooccur	Evaluate if two sets of genomic regions are significantly overlapped.
covary	Calculate the covariance of binding intensities between two sets of genomic intervals.
srog	Report the code of Spatial Relation Of Genomic (SROG) regions.
stat	Wrapper function. Calculate *C*, *J*, *SD*, *SS*, *PMI*, and *NPMI*.
zscore	Calculate the overall Zscore of *C*, *J*, *SD*, *SS*, *PMI*, and *NPMI*.

7.2 Usage

Print out all the avaiable subcommands and their descriptions

`cobind.py -h` or `cobind.py --help`

```
usage: cobind.py [-h] [-v]
                 {overlap,jaccard,dice,simpson,pmi,npmi,cooccur,covary,srog,stat,zscore}
                 ...

**cobind: collocation analyses of genomic regions**

positional arguments:
  {overlap,jaccard,dice,simpson,pmi,npmi,cooccur,covary,srog,stat,zscore}
                        Sub-command description:
    overlap             Calculate the collocation coefficient (C) between two
                        sets of genomic regions. C = |A and B| /
                        (|A|*|B|)**0.5
    jaccard             Calculate the Jaccard similarity coefficient (J)
```

(continues on next page)

(continued from previous page)

```
                        between two sets of genomic regions. J = |A and B| /
                        |A or B|
    dice                Calculate the Sørensen-Dice coefficient (SD) between
                        two sets of genomic regions. SD = 2*|A and B| / (|A| +
                        |B|)
    simpson             Calculate the Szymkiewicz-Simpson coefficient (SS)
                        between two sets of genomic regions. SS = |A and B| /
                        min(|A|, |B|)
    pmi                 Calculate the pointwise mutual information (PMI)
                        between two sets of genomic regions. PMI = log(p(|A
                        and B|)) - log(p(|A|)) - log(p(|B|))
    npmi                Calculate the normalized pointwise mutual information
                        (NPMI) between two sets of genomic regions. NPMI =
                        log(p(|A|)*p(|B|)) / log(p(|A and B|)) - 1
    cooccur             Evaluate if two sets of genomic regions are
                        significantly co-occurred in given background regions.
    covary              Calculate the covariance (Pearson, Spearman and
                        Kendall coefficients) of binding intensities between
                        two sets of genomic regions.
    srog                Report the code of Spatial Relation Of Genomic (SROG)
                        regions. SROG codes include
                        'disjoint','touch','equal','overlap', 'contain',
                        'within'.
    stat                Wrapper function. Report basic statistics of genomic
                        regions, and calculate overlapping measurements
                        (including "C", "J", "SD", "SS", "PMI", "NPMI"), without
                        bootstrap resampling or generating peakwise
                        measurements.
    zscore              Calculate Z-score of six overlapping measurements
                        inlcuding ("C", "J", "SD", "SS", "PMI", "NPMI"),
                        to provide an overall measurement of the
                        collocation strength.

options:
  -h, --help            show this help message and exit
  -v, --version         show program's version number and exit
```

Run each subcommand, for example, run the **overlap** subcommand:

`cobind.py overlap -h` or `cobind.py overlap --help`

```
usage: cobind.py overlap [-h] [--nameA NAMEA] [--nameB NAMEB] [-n ITER]
                         [-f SUBSAMPLE] [-b BGSIZE] [-o] [-l log_file] [-d]
                         input_A.bed input_B.bed

positional arguments:
  input_A.bed           Genomic regions in BED, BED-like or bigBed format. The
                        BED-like format includes:'bed3', 'bed4', 'bed6',
                        'bed12', 'bedgraph', 'narrowpeak', 'broadpeak',
                        'gappedpeak'. BED and BED-like format can be plain
                        text, compressed (.gz, .z, .bz, .bz2, .bzip2) or
                        remote (http://, https://, ftp://) files. Do not
                        compress BigBed foramt. BigBed file can also be a
```

(continues on next page)

(continued from previous page)

```
                        remote file.
  input_B.bed           Genomic regions in BED, BED-like or bigBed format. The
                        BED-like format includes:'bed3', 'bed4', 'bed6',
                        'bed12', 'bedgraph', 'narrowpeak', 'broadpeak',
                        'gappedpeak'. BED and BED-like format can be plain
                        text, compressed (.gz, .z, .bz, .bz2, .bzip2) or
                        remote (http://, https://, ftp://) files. Do not
                        compress BigBed foramt. BigBed file can also be a
                        remote file.

options:
  -h, --help            show this help message and exit
  --nameA NAMEA         Name to represent 1st set of genomic interval. If not
                        specified (None), the file name ("input_A.bed") will
                        be used.
  --nameB NAMEB         Name to represent the 2nd set of genomic interval. If
                        not specified (None), the file name ("input_B.bed")
                        will be used.
  -n ITER, --ndraws ITER
                        Times of resampling to estimate confidence intervals.
                        Set to '0' to turn off resampling. For the resampling
                        process to work properly, overlapped intervals in each
                        bed file must be merged. (default: 20)
  -f SUBSAMPLE, --fraction SUBSAMPLE
                        Resampling fraction. (default: 0.75)
  -b BGSIZE, --background BGSIZE
                        The size of the cis-regulatory genomic regions. This
                        is about 1.4Gb For the human genome. (default:
                        1400000000)
  -o, --save            If set, will save peak-wise coefficients to files
                        ("input_A_peakwise_scores.tsv" and
                        "input_B_peakwise_scores.tsv").
  -l log_file, --log log_file
                        This file is used to save the log information. By
                        default, if no file is specified (None), the log
                        information will be printed to the screen.
  -d, --debug           Print detailed information for debugging.
```

CHAPTER

EIGHT

COLLOCATION COEFFICIENT (C)

8.1 Description

Calculate the collocation coefficient between two sets of genomic regions.

$$C(A,B) = \frac{|A \cap B|}{\sqrt{|A| \times |B|}}$$

$$0 \leq C(A,B) \leq 1$$

8.2 Usage

```
cobind.py overlap -h
```

```
usage: cobind.py overlap [-h] [--nameA NAMEA] [--nameB NAMEB] [-n ITER]
                         [-f SUBSAMPLE] [-b BGSIZE] [-o] [-l log_file] [-d]
                         input_A.bed input_B.bed

positional arguments:
  input_A.bed           Genomic regions in BED, BED-like or bigBed format. The
                        BED-like format includes:'bed3', 'bed4', 'bed6',
                        'bed12', 'bedgraph', 'narrowpeak', 'broadpeak',
                        'gappedpeak'. BED and BED-like format can be plain
                        text, compressed (.gz, .z, .bz, .bz2, .bzip2) or
                        remote (http://, https://, ftp://) files. Do not
                        compress BigBed foramt. BigBed file can also be a
                        remote file.
  input_B.bed           Genomic regions in BED, BED-like or bigBed format. The
                        BED-like format includes:'bed3', 'bed4', 'bed6',
                        'bed12', 'bedgraph', 'narrowpeak', 'broadpeak',
                        'gappedpeak'. BED and BED-like format can be plain
                        text, compressed (.gz, .z, .bz, .bz2, .bzip2) or
                        remote (http://, https://, ftp://) files. Do not
                        compress BigBed foramt. BigBed file can also be a
                        remote file.
```

(continues on next page)

(continued from previous page)

```
options:
  -h, --help            show this help message and exit
  --nameA NAMEA         Name to represent 1st set of genomic interval. If not
                        specified (None), the file name ("input_A.bed") will
                        be used.
  --nameB NAMEB         Name to represent the 2nd set of genomic interval. If
                        not specified (None), the file name ("input_B.bed")
                        will be used.
  -n ITER, --ndraws ITER
                        Times of resampling to estimate confidence intervals.
                        Set to '0' to turn off resampling. For the resampling
                        process to work properly, overlapped intervals in each
                        bed file must be merged. (default: 20)
  -f SUBSAMPLE, --fraction SUBSAMPLE
                        Resampling fraction. (default: 0.75)
  -b BGSIZE, --background BGSIZE
                        The size of the cis-regulatory genomic regions. This
                        is about 1.4Gb For the human genome. (default:
                        1400000000)
  -o, --save            If set, will save peak-wise coefficients to files
                        ("input_A_peakwise_scores.tsv" and
                        "input_B_peakwise_scores.tsv").
  -l log_file, --log log_file
                        This file is used to save the log information. By
                        default, if no file is specified (None), the log
                        information will be printed to the screen.
  -d, --debug           Print detailed information for debugging.
```

8.3 Example

Calculate the **overall** collocation coefficient and **peak-wise** collocation coefficients between CTCF binding sites and RAD21 binding sites.

```
python3 ../bin/cobind.py overlap CTCF_ENCFF660GHM.bed RAD21_ENCFF057JFH.bed --save
```

The overall collocation coefficient between `CTCF_ENCFF660GHM.bed` and `RAD21_ENCFF057JFH.bed` was printed to screen

```
2022-02-24 08:06:29 [INFO]  Calculate collocation coefficient (overall) ...
A.name                CTCF_ENCFF660GHM.bed
B.name               RAD21_ENCFF057JFH.bed
A.interval_count                    58684
B.interval_count                    33373
A.size                          12184840
B.size                          11130268
A_or_B.size                     18375623
A_and_B.size                     4939485
Coef                              0.4241
Coef(expected)                    0.0083
Coef(95% CI)               [0.4223,0.4275]
```

(continues on next page)

(continued from previous page)

```
dtype: object
2023-07-04 08:08:18 [INFO]  Calculate collocation coefficient (peak-wise) ...
2023-07-04 08:08:18 [INFO]  Read and union BED file: "CTCF_ENCFF660GHM.bed"
2023-07-04 08:08:18 [INFO]  Unioned regions of "CTCF_ENCFF660GHM.bed" : 58584
2023-07-04 08:08:18 [INFO]  Read and union BED file: "RAD21_ENCFF057JFH.bed"
2023-07-04 08:08:19 [INFO]  Unioned regions of "RAD21_ENCFF057JFH.bed" : 31955
2023-07-04 08:08:19 [INFO]  Build interval tree for unioned BED file: "CTCF_ENCFF660GHM.
↪bed"
2023-07-04 08:08:19 [INFO]  Build interval tree for unioned BED file: "RAD21_ENCFF057JFH.
↪bed"
2023-07-04 08:08:19 [INFO]  Calculate the overlap coefficient of each genomic region in␣
↪CTCF_ENCFF660GHM.bed ...
2023-07-04 08:08:21 [INFO]  Save peakwise scores to CTCF_ENCFF660GHM.bed_peakwise_scores.
↪tsv ...
2023-07-04 08:08:21 [INFO]  Calculate the overlap coefficient of each genomic region in␣
↪RAD21_ENCFF057JFH.bed ...
2023-07-04 08:08:22 [INFO]  Save peakwise scores to RAD21_ENCFF057JFH.bed_peakwise_
↪scores.tsv ...
```

If `--save` was specified, the peakwise collocation coefficients were saved to `CTCF_ENCFF660GHM.bed_peakwise_scores.tsv` and `RAD21_ENCFF057JFH.bed_peakwise_scores.tsv`, respectively.

```
$ head -5 CTCF_ENCFF660GHM.bed_peakwise_scores.tsv

chrom start end A.size  B.size  AB AB B.list  Score
chr12 108043  108283  240 404 240 404 chr12:107919-108323 0.770752493308062
chr12 153232  153470  238 222 222 238 chr12:153236-153458 0.965801796044974
chr12 177749  177989  240 NA  NA  NA  NA  NA
chr12 189165  189405  240 404 240 404 chr12:189072-189476 0.770752493308062
```

column 1 to 3
: The genomic coordinate of CTCF peak.

column 4 (A.size)
: The size of CTCF peak.

column 5 (B.size)
: The size (cardinality) of RAD21 peak(s) that were overlapped with this CTCF peak.

column 6 (AB)
: The size (cardinality) of intersection.

column 7 (AB)
: The size (cardinality) of union.

column 8 (B.list)
: List of RAD21 peak(s) that are overlapped with this peak. Multiple peaks will be separated by ",".

column 9 (Score)
: The peakwise collocation coefficient.

CHAPTER

NINE

JACCARD COEFFICIENT (J)

9.1 Description

Calculate the Jaccard similarity coefficient between two sets of genomic regions.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

$$0 \leq J(A,B) \leq 1$$

9.2 Usage

```
cobind.py jacard -h
```

```
usage: cobind.py jaccard [-h] [--nameA NAMEA] [--nameB NAMEB] [-n ITER]
                         [-f SUBSAMPLE] [-b BGSIZE] [-o] [-l log_file] [-d]
                         input_A.bed input_B.bed

positional arguments:
  input_A.bed           Genomic regions in BED, BED-like or bigBed format. The
                        BED-like format includes:'bed3', 'bed4', 'bed6',
                        'bed12', 'bedgraph', 'narrowpeak', 'broadpeak',
                        'gappedpeak'. BED and BED-like format can be plain
                        text, compressed (.gz, .z, .bz, .bz2, .bzip2) or
                        remote (http://, https://, ftp://) files. Do not
                        compress BigBed foramt. BigBed file can also be a
                        remote file.
  input_B.bed           Genomic regions in BED, BED-like or bigBed format. The
                        BED-like format includes:'bed3', 'bed4', 'bed6',
                        'bed12', 'bedgraph', 'narrowpeak', 'broadpeak',
                        'gappedpeak'. BED and BED-like format can be plain
                        text, compressed (.gz, .z, .bz, .bz2, .bzip2) or
                        remote (http://, https://, ftp://) files. Do not
                        compress BigBed foramt. BigBed file can also be a
                        remote file.
```

(continues on next page)

(continued from previous page)

```
options:
  -h, --help            show this help message and exit
  --nameA NAMEA         Name to represent 1st set of genomic interval. If not
                        specified (None), the file name ("input_A.bed") will
                        be used.
  --nameB NAMEB         Name to represent the 2nd set of genomic interval. If
                        not specified (None), the file name ("input_B.bed")
                        will be used.
  -n ITER, --ndraws ITER
                        Times of resampling to estimate confidence intervals.
                        Set to '0' to turn off resampling. For the resampling
                        process to work properly, overlapped intervals in each
                        bed file must be merged. (default: 20)
  -f SUBSAMPLE, --fraction SUBSAMPLE
                        Resampling fraction. (default: 0.75)
  -b BGSIZE, --background BGSIZE
                        The size of the cis-regulatory genomic regions. This
                        is about 1.4Gb For the human genome. (default:
                        1400000000)
  -o, --save            If set, will save peak-wise coefficients to files
                        ("input_A_peakwise_scores.tsv" and
                        "input_B_peakwise_scores.tsv").
  -l log_file, --log log_file
                        This file is used to save the log information. By
                        default, if no file is specified (None), the log
                        information will be printed to the screen.
  -d, --debug           Print detailed information for debugging.
```

9.3 Example

Calculate the **overall** Jaccard coefficient and **peak-wise** Jaccard coefficient between CTCF binding sites and RAD21 binding sites.

```
python3 ../bin/cobind.py jaccard CTCF_ENCFF660GHM.bed RAD21_ENCFF057JFH.bed --save
```

The overall Jaccard coefficient between `CTCF_ENCFF660GHM.bed` and `RAD21_ENCFF057JFH.bed` was printed to screen

```
2022-01-16 08:24:12 [INFO]  Calculate Jaccard coefficient (overall) ...
A.name                CTCF_ENCFF660GHM.bed
B.name              RAD21_ENCFF057JFH.bed
A.interval_count                    58684
B.interval_count                    33373
A.size                           12184840
B.size                           11130268
A_or_B.size                      18375623
A_and_B.size                      4939485
Coef                               0.2688
Coef(expected)                     0.0042
Coef(95% CI)               [0.2672,0.2713]
dtype: object
```

(continues on next page)

(continued from previous page)

```
2022-01-16 08:24:40 [INFO]  Calculate Jaccard coefficient (peakwise) ...
2022-01-16 08:24:40 [INFO]  Read and union BED file: "CTCF_ENCFF660GHM.bed"
2022-01-16 08:24:40 [INFO]  Unioned regions of "CTCF_ENCFF660GHM.bed" : 58584
2022-01-16 08:24:40 [INFO]  Read and union BED file: "RAD21_ENCFF057JFH.bed"
2022-01-16 08:24:41 [INFO]  Unioned regions of "RAD21_ENCFF057JFH.bed" : 31955
...
```

If `--save` was specified, the peakwise coefficients were saved to `CTCF_ENCFF660GHM.bed_peakwise_scores.tsv` and `RAD21_ENCFF057JFH.bed_peakwise_scores.tsv`, respectively.

```
$ head -5 CTCF_ENCFF660GHM.bed_peakwise_scores.tsv

chrom start end A.size  B.size  AB AB B.list  Score
chr12 108043  108283  240 404 240 404 chr12:107919-108323 0.594059405940594
chr12 153232  153470  238 222 222 238 chr12:153236-153458 0.9327731092436975
chr12 177749  177989  240 NA  NA  NA  NA  NA
chr12 189165  189405  240 404 240 404 chr12:189072-189476 0.594059405940594
```

column 1 to 3
: The genomic coordinate of CTCF peak.

column 4 (A.size)
: The size of CTCF peak.

column 5 (B.size)
: The size (cardinality) of RAD21 peak(s) that were overlapped with this CTCF peak.

column 6 (AB)
: The size (cardinality) of intersection.

column 7 (AB)
: The size (cardinality) of union.

column 8 (B.list)
: List of RAD21 peak(s) that are overlapped with this peak. Multiple peaks will be separated by ",".

column 9 (Score)
: The peakwise Jaccard coefficient.

CHAPTER

TEN

DICE COEFFICIENT (SD)

10.1 Description

Calculate the Sørensen–Dice coefficient between two sets of genomic regions.

$$SD(A,B) = \frac{2|A \cap B|}{|A| + |B|}$$

$0 \leq SD(A,B) \leq 1$

10.2 Usage

```
cobind.py dice -h
```

```
usage: cobind.py dice [-h] [--nameA NAMEA] [--nameB NAMEB] [-n ITER]
                      [-f SUBSAMPLE] [-b BGSIZE] [-o] [-l log_file] [-d]
                      input_A.bed input_B.bed

positional arguments:
  input_A.bed           Genomic regions in BED, BED-like or bigBed format. The
                        BED-like format includes:'bed3', 'bed4', 'bed6',
                        'bed12', 'bedgraph', 'narrowpeak', 'broadpeak',
                        'gappedpeak'. BED and BED-like format can be plain
                        text, compressed (.gz, .z, .bz, .bz2, .bzip2) or
                        remote (http://, https://, ftp://) files. Do not
                        compress BigBed foramt. BigBed file can also be a
                        remote file.
  input_B.bed           Genomic regions in BED, BED-like or bigBed format. The
                        BED-like format includes:'bed3', 'bed4', 'bed6',
                        'bed12', 'bedgraph', 'narrowpeak', 'broadpeak',
                        'gappedpeak'. BED and BED-like format can be plain
                        text, compressed (.gz, .z, .bz, .bz2, .bzip2) or
                        remote (http://, https://, ftp://) files. Do not
                        compress BigBed foramt. BigBed file can also be a
                        remote file.

options:
  -h, --help            show this help message and exit
```

(continues on next page)

(continued from previous page)

```
  --nameA NAMEA         Name to represent 1st set of genomic interval. If not
                        specified (None), the file name ("input_A.bed") will
                        be used.
  --nameB NAMEB         Name to represent the 2nd set of genomic interval. If
                        not specified (None), the file name ("input_B.bed")
                        will be used.
  -n ITER, --ndraws ITER
                        Times of resampling to estimate confidence intervals.
                        Set to '0' to turn off resampling. For the resampling
                        process to work properly, overlapped intervals in each
                        bed file must be merged. (default: 20)
  -f SUBSAMPLE, --fraction SUBSAMPLE
                        Resampling fraction. (default: 0.75)
  -b BGSIZE, --background BGSIZE
                        The size of the cis-regulatory genomic regions. This
                        is about 1.4Gb For the human genome. (default:
                        1400000000)
  -o, --save            If set, will save peak-wise coefficients to files
                        ("input_A_peakwise_scores.tsv" and
                        "input_B_peakwise_scores.tsv").
  -l log_file, --log log_file
                        This file is used to save the log information. By
                        default, if no file is specified (None), the log
                        information will be printed to the screen.
  -d, --debug           Print detailed information for debugging.
```

10.3 Example

Calculate the **overall** Dice coefficient and **peak-wise** Dice coefficient between CTCF binding sites and RAD21 binding sites.

```
python3 ../bin/cobind.py dice CTCF_ENCFF660GHM.bed RAD21_ENCFF057JFH.bed --save
```

The overall Dice coefficient between `CTCF_ENCFF660GHM.bed` and `RAD21_ENCFF057JFH.bed` was printed to screen

```
2022-01-16 08:43:40 [INFO]  Calculate Sørensen-Dice coefficient (overall) ...
A.name                CTCF_ENCFF660GHM.bed
B.name               RAD21_ENCFF057JFH.bed
A.interval_count                     58684
B.interval_count                     33373
A.size                            12184840
B.size                            11130268
A_or_B.size                       18375623
A_and_B.size                       4939485
Coef                                0.4237
Coef(expected)                      0.0083
Coef(95% CI)                [0.4222,0.4275]
dtype: object
2022-01-16 08:44:08 [INFO]  Calculate Sørensen-Dice coefficient (peakwise) ...
2022-01-16 08:44:08 [INFO]  Read and union BED file: "CTCF_ENCFF660GHM.bed"
2022-01-16 08:44:08 [INFO]  Unioned regions of "CTCF_ENCFF660GHM.bed" : 58584
```

(continues on next page)

(continued from previous page)

```
2022-01-16 08:44:08 [INFO]  Read and union BED file: "RAD21_ENCFF057JFH.bed"
2022-01-16 08:44:09 [INFO]  Unioned regions of "RAD21_ENCFF057JFH.bed" : 31955
...
```

If `--save` was specified, the peakwise coefficients were saved to `CTCF_ENCFF660GHM.bed_peakwise_scores.tsv` and `RAD21_ENCFF057JFH.bed_peakwise_scores.tsv`, respectively.

```
$ head -5 CTCF_ENCFF660GHM.bed_peakwise_scores.tsv

chrom start end A.size  B.size  AB AB B.list  Score
chr12 108043  108283  240 404 240 404 chr12:107919-108323 0.7453416149068323
chr12 153232  153470  238 222 222 238 chr12:153236-153458 0.9652173913043478
chr12 177749  177989  240 NA  NA  NA  NA  NA
chr12 189165  189405  240 404 240 404 chr12:189072-189476 0.7453416149068323
```

column 1 to 3
The genomic coordinate of CTCF peak.

column 4 (A.size)
The size of CTCF peak.

column 5 (B.size)
The size (cardinality) of RAD21 peak(s) that were overlapped with this CTCF peak.

column 6 (AB)
The size (cardinality) of intersection.

column 7 (AB)
The size (cardinality) of union.

column 8 (B.list)
List of RAD21 peak(s) that are overlapped with this peak. Multiple peaks will be separated by ",".

column 9 (Score)
The peakwise Dice coefficient.

CHAPTER

ELEVEN

SZYMKIEWICZ–SIMPSON COEFFICIENT (SS)

11.1 Description

Calculate the Szymkiewicz–Simpson coefficient (SS coefficient or Simpson coefficient) between two sets of genomic regions.

$$SS(A,B) = \frac{|A \cap B|}{\min(|A|,|B|)}$$

$$0 \leq SS(A,B) \leq 1$$

11.2 Usage

```
cobind.py simpson -h
```

```
usage: cobind.py simpson [-h] [--nameA NAMEA] [--nameB NAMEB] [-n ITER]
                         [-f SUBSAMPLE] [-b BGSIZE] [-o] [-l log_file] [-d]
                         input_A.bed input_B.bed

positional arguments:
  input_A.bed           Genomic regions in BED, BED-like or bigBed format. The
                        BED-like format includes:'bed3', 'bed4', 'bed6',
                        'bed12', 'bedgraph', 'narrowpeak', 'broadpeak',
                        'gappedpeak'. BED and BED-like format can be plain
                        text, compressed (.gz, .z, .bz, .bz2, .bzip2) or
                        remote (http://, https://, ftp://) files. Do not
                        compress BigBed foramt. BigBed file can also be a
                        remote file.
  input_B.bed           Genomic regions in BED, BED-like or bigBed format. The
                        BED-like format includes:'bed3', 'bed4', 'bed6',
                        'bed12', 'bedgraph', 'narrowpeak', 'broadpeak',
                        'gappedpeak'. BED and BED-like format can be plain
                        text, compressed (.gz, .z, .bz, .bz2, .bzip2) or
                        remote (http://, https://, ftp://) files. Do not
                        compress BigBed foramt. BigBed file can also be a
                        remote file.
```

(continues on next page)

(continued from previous page)

```
options:
  -h, --help            show this help message and exit
  --nameA NAMEA         Name to represent 1st set of genomic interval. If not
                        specified (None), the file name ("input_A.bed") will
                        be used.
  --nameB NAMEB         Name to represent the 2nd set of genomic interval. If
                        not specified (None), the file name ("input_B.bed")
                        will be used.
  -n ITER, --ndraws ITER
                        Times of resampling to estimate confidence intervals.
                        Set to '0' to turn off resampling. For the resampling
                        process to work properly, overlapped intervals in each
                        bed file must be merged. (default: 20)
  -f SUBSAMPLE, --fraction SUBSAMPLE
                        Resampling fraction. (default: 0.75)
  -b BGSIZE, --background BGSIZE
                        The size of the cis-regulatory genomic regions. This
                        is about 1.4Gb For the human genome. (default:
                        1400000000)
  -o, --save            If set, will save peak-wise coefficients to files
                        ("input_A_peakwise_scores.tsv" and
                        "input_B_peakwise_scores.tsv").
  -l log_file, --log log_file
                        This file is used to save the log information. By
                        default, if no file is specified (None), the log
                        information will be printed to the screen.
  -d, --debug           Print detailed information for debugging.
```

11.3 Example

Calculate the **overall** Simpson coefficient and **peak-wise** Simpson coefficient between CTCF binding sites and RAD21 binding sites.

```
python3 ../bin/cobind.py simpson CTCF_ENCFF660GHM.bed RAD21_ENCFF057JFH.bed --save
```

The overall Simpson coefficient between `CTCF_ENCFF660GHM.bed` and `RAD21_ENCFF057JFH.bed` was printed to screen

```
2022-01-16 08:52:41 [INFO]  Calculate Szymkiewicz-Simpson coefficient (overall) ...
A.name               CTCF_ENCFF660GHM.bed
B.name              RAD21_ENCFF057JFH.bed
A.interval_count                    58684
B.interval_count                    33373
A.size                          12184840
B.size                          11130268
A_or_B.size                     18375623
A_and_B.size                     4939485
Coef                              0.4438
Coef(expected)                    0.0087
Coef(95% CI)              [0.4413,0.4475]
```

(continues on next page)

(continued from previous page)

```
dtype: object
2022-01-16 08:53:09 [INFO]  Calculate Szymkiewicz-Simpson coefficient (peakwise) ...
2022-01-16 08:53:09 [INFO]  Read and union BED file: "CTCF_ENCFF660GHM.bed"
2022-01-16 08:53:10 [INFO]  Unioned regions of "CTCF_ENCFF660GHM.bed" : 58584
2022-01-16 08:53:10 [INFO]  Read and union BED file: "RAD21_ENCFF057JFH.bed"
2022-01-16 08:53:10 [INFO]  Unioned regions of "RAD21_ENCFF057JFH.bed" : 31955
...
```

If `--save` was specified, the peakwise coefficients were saved to `CTCF_ENCFF660GHM.bed_peakwise_scores.tsv` and `RAD21_ENCFF057JFH.bed_peakwise_scores.tsv`, respectively.

```
$ head -5 CTCF_ENCFF660GHM.bed_peakwise_scores.tsv

chrom start end A.size  B.size  AB AB B.list  Score
chr12 108043  108283  240 404 240 404 chr12:107919-108323 1.0
chr12 153232  153470  238 222 222 238 chr12:153236-153458 1.0
chr12 177749  177989  240 NA  NA  NA  NA  NA
chr12 189165  189405  240 404 240 404 chr12:189072-189476 1.0
```

column 1 to 3
: The genomic coordinate of CTCF peak.

column 4 (A.size)
: The size of CTCF peak.

column 5 (B.size)
: The size (cardinality) of RAD21 peak(s) that were overlapped with this CTCF peak.

column 6 (AB)
: The size (cardinality) of intersection.

column 7 (AB)
: The size (cardinality) of union.

column 8 (B.list)
: List of RAD21 peak(s) that are overlapped with this peak. Multiple peaks will be separated by ",".

column 9 (Score)
: The peakwise Simpson coefficient.

CHAPTER

TWELVE

POINTWISE MUTUAL INFORMATION (PMI)

12.1 Description

Calculate the Pointwise mutual information (PMI)[1] between two sets of genomic regions.

$$pmi(A \cap B) \equiv log\left(\frac{p(A \cap B)}{p(A) \times p(B)}\right)$$

$$-\infty \leq pmi(A \cap B) \leq min(-log(p(A)), -log(p(B)))$$

where

$$\text{p(A)} = \frac{|A|}{|G|}, \text{p(B)} = \frac{|B|}{|G|}, p(A \cap B) = \frac{|A \cap B|}{|G|}$$

12.2 Usage

```
cobind.py pmi -h
```

```
usage: cobind.py pmi [-h] [--nameA NAMEA] [--nameB NAMEB] [-n ITER]
                     [-f SUBSAMPLE] [-b BGSIZE] [-o] [-l log_file] [-d]
                     input_A.bed input_B.bed

positional arguments:
  input_A.bed           Genomic regions in BED, BED-like or bigBed format. The
                        BED-like format includes:'bed3', 'bed4', 'bed6',
                        'bed12', 'bedgraph', 'narrowpeak', 'broadpeak',
                        'gappedpeak'. BED and BED-like format can be plain
                        text, compressed (.gz, .z, .bz, .bz2, .bzip2) or
                        remote (http://, https://, ftp://) files. Do not
                        compress BigBed foramt. BigBed file can also be a
                        remote file.
  input_B.bed           Genomic regions in BED, BED-like or bigBed format. The
                        BED-like format includes:'bed3', 'bed4', 'bed6',
                        'bed12', 'bedgraph', 'narrowpeak', 'broadpeak',
```

(continues on next page)

[1] The natural log was used when calculating PMI.

(continued from previous page)

```
                        'gappedpeak'. BED and BED-like format can be plain
                        text, compressed (.gz, .z, .bz, .bz2, .bzip2) or
                        remote (http://, https://, ftp://) files. Do not
                        compress BigBed foramt. BigBed file can also be a
                        remote file.

options:
  -h, --help            show this help message and exit
  --nameA NAMEA         Name to represent 1st set of genomic interval. If not
                        specified (None), the file name ("input_A.bed") will
                        be used.
  --nameB NAMEB         Name to represent the 2nd set of genomic interval. If
                        not specified (None), the file name ("input_B.bed")
                        will be used.
  -n ITER, --ndraws ITER
                        Times of resampling to estimate confidence intervals.
                        Set to '0' to turn off resampling. For the resampling
                        process to work properly, overlapped intervals in each
                        bed file must be merged. (default: 20)
  -f SUBSAMPLE, --fraction SUBSAMPLE
                        Resampling fraction. (default: 0.75)
  -b BGSIZE, --background BGSIZE
                        The size of the cis-regulatory genomic regions. This
                        is about 1.4Gb For the human genome. (default:
                        1400000000)
  -o, --save            If set, will save peak-wise coefficients to files
                        ("input_A_peakwise_scores.tsv" and
                        "input_B_peakwise_scores.tsv").
  -l log_file, --log log_file
                        This file is used to save the log information. By
                        default, if no file is specified (None), the log
                        information will be printed to the screen.
  -d, --debug           Print detailed information for debugging.
```

12.3 Example

Calculate the **overall** PMI and **peak-wise** PMI between CTCF binding sites and RAD21 binding sites.

```
python3 ../bin/cobind.py pmi CTCF_ENCFF660GHM.bed RAD21_ENCFF057JFH.bed --save
```

The overall PMI between `CTCF_ENCFF660GHM.bed` and `RAD21_ENCFF057JFH.bed` was printed to screen

```
2022-01-16 09:01:34 [INFO]  Calculate the pointwise mutual information (PMI) ...
A.name                CTCF_ENCFF660GHM.bed
B.name               RAD21_ENCFF057JFH.bed
A.interval_count                    58684
B.interval_count                    33373
A.size                           12184840
B.size                           11130268
A_or_B.size                      18375623
A_and_B.size                      4939485
```

(continues on next page)

(continued from previous page)

```
Coef                              3.9316
Coef(expected)                    0.0000
Coef(95% CI)                [3.9230,3.9343]
dtype: object
2022-01-16 09:02:02 [INFO]  Read and union BED file: "CTCF_ENCFF660GHM.bed"
2022-01-16 09:02:03 [INFO]  Unioned regions of "CTCF_ENCFF660GHM.bed" : 58584
2022-01-16 09:02:03 [INFO]  Read and union BED file: "RAD21_ENCFF057JFH.bed"
2022-01-16 09:02:03 [INFO]  Unioned regions of "RAD21_ENCFF057JFH.bed" : 31955
...
```

If `--save` was specified, the peakwise PMI were saved to `CTCF_ENCFF660GHM.bed_peakwise_scores.tsv` and `RAD21_ENCFF057JFH.bed_peakwise_scores.tsv`, respectively.

```
$ head -5 CTCF_ENCFF660GHM.bed_peakwise_scores.tsv

chrom start end A.size  B.size  AB AB B.list  Score
chr12 108043  108283  240 404 240 404 chr12:107919-108323 15.058323195606475
chr12 153232  153470  238 222 222 238 chr12:153236-153458 15.58746739989615
chr12 177749  177989  240 NA  NA  NA  NA  NA
chr12 189165  189405  240 404 240 404 chr12:189072-189476 15.058323195606475
```

column 1 to 3
: The genomic coordinate of CTCF peak.

column 4 (A.size)
: The size of CTCF peak.

column 5 (B.size)
: The size (cardinality) of RAD21 peak(s) that were overlapped with this CTCF peak.

column 6 (AB)
: The size (cardinality) of intersection.

column 7 (AB)
: The size (cardinality) of union.

column 8 (B.list)
: List of RAD21 peak(s) that are overlapped with this peak. Multiple peaks will be separated by ",".

column 9 (Score)
: The peakwise PMI.

CHAPTER

THIRTEEN

NORMALIZED POINTWISE MUTUAL INFORMATION (NPMI)

13.1 Description

Calculate the Normalized pointwise mutual information (NPMI)[1] between two sets of genomic regions.

$$npmi(A \cap B) = \frac{pmi(A \cap B)}{-log(p(A \cap B))} = \frac{log\left(\frac{p(A \cap B)}{p(A) \times p(B)}\right)}{-log(p(A \cap B))} = \frac{log(p(A) \times p(B))}{log(p(A \cap B))} - 1$$

$$-1 \leq npmi(A \cap B) \leq 1$$

where

$$\mathrm{p(A)} = \frac{|A|}{|G|},\ \mathrm{p(B)} = \frac{|B|}{|G|},\ p(A \cap B) = \frac{|A \cap B|}{|G|}$$

13.2 Usage

```
cobind.py npmi -h
```

```
usage: cobind.py npmi [-h] [--nameA NAMEA] [--nameB NAMEB] [-n ITER]
                      [-f SUBSAMPLE] [-b BGSIZE] [-o] [-l log_file] [-d]
                      input_A.bed input_B.bed

positional arguments:
  input_A.bed           Genomic regions in BED, BED-like or bigBed format. The
                        BED-like format includes:'bed3', 'bed4', 'bed6',
                        'bed12', 'bedgraph', 'narrowpeak', 'broadpeak',
                        'gappedpeak'. BED and BED-like format can be plain
                        text, compressed (.gz, .z, .bz, .bz2, .bzip2) or
                        remote (http://, https://, ftp://) files. Do not
                        compress BigBed foramt. BigBed file can also be a
                        remote file.
  input_B.bed           Genomic regions in BED, BED-like or bigBed format. The
                        BED-like format includes:'bed3', 'bed4', 'bed6',
```

(continues on next page)

[1] The natural log was used when calculating NPMI.

(continued from previous page)

```
                        'bed12', 'bedgraph', 'narrowpeak', 'broadpeak',
                        'gappedpeak'. BED and BED-like format can be plain
                        text, compressed (.gz, .z, .bz, .bz2, .bzip2) or
                        remote (http://, https://, ftp://) files. Do not
                        compress BigBed foramt. BigBed file can also be a
                        remote file.

options:
  -h, --help            show this help message and exit
  --nameA NAMEA         Name to represent 1st set of genomic interval. If not
                        specified (None), the file name ("input_A.bed") will
                        be used.
  --nameB NAMEB         Name to represent the 2nd set of genomic interval. If
                        not specified (None), the file name ("input_B.bed")
                        will be used.
  -n ITER, --ndraws ITER
                        Times of resampling to estimate confidence intervals.
                        Set to '0' to turn off resampling. For the resampling
                        process to work properly, overlapped intervals in each
                        bed file must be merged. (default: 20)
  -f SUBSAMPLE, --fraction SUBSAMPLE
                        Resampling fraction. (default: 0.75)
  -b BGSIZE, --background BGSIZE
                        The size of the cis-regulatory genomic regions. This
                        is about 1.4Gb For the human genome. (default:
                        1400000000)
  -o, --save            If set, will save peak-wise coefficients to files
                        ("input_A_peakwise_scores.tsv" and
                        "input_B_peakwise_scores.tsv").
  -l log_file, --log log_file
                        This file is used to save the log information. By
                        default, if no file is specified (None), the log
                        information will be printed to the screen.
  -d, --debug           Print detailed information for debugging.
```

13.3 Example

Calculate the **overall** NPMI and **peak-wise** NPMI between CTCF binding sites and RAD21 binding sites.

```
python3 ../bin/cobind.py npmi CTCF_ENCFF660GHM.bed RAD21_ENCFF057JFH.bed --save
```

The overall NPMI between `CTCF_ENCFF660GHM.bed` and `RAD21_ENCFF057JFH.bed` was printed to screen

```
2022-01-16 09:26:50 [INFO]  Calculate the normalized pointwise mutual information (NPMI)␣
↪...
A.name               CTCF_ENCFF660GHM.bed
B.name              RAD21_ENCFF057JFH.bed
A.interval_count                   58684
B.interval_count                   33373
A.size                          12184840
B.size                          11130268
```

(continues on next page)

(continued from previous page)

```
A_or_B.size                       18375623
A_and_B.size                       4939485
Coef                                0.6962
Coef(expected)                      0.0000
Coef(95% CI)                [0.6945,0.6977]
dtype: object
2022-01-16 09:27:18 [INFO]  Read and union BED file: "CTCF_ENCFF660GHM.bed"
2022-01-16 09:27:19 [INFO]  Unioned regions of "CTCF_ENCFF660GHM.bed" : 58584
2022-01-16 09:27:19 [INFO]  Read and union BED file: "RAD21_ENCFF057JFH.bed"
2022-01-16 09:27:19 [INFO]  Unioned regions of "RAD21_ENCFF057JFH.bed" : 31955
...
```

If `--save` was specified, the peakwise coefficients were saved to `CTCF_ENCFF660GHM.bed_peakwise_scores.tsv` and `RAD21_ENCFF057JFH.bed_peakwise_scores.tsv`, respectively.

```
$ head -5 CTCF_ENCFF660GHM.bed_peakwise_scores.tsv

chrom start end A.size  B.size  AB AB B.list  Score
chr12 108043  108283  240 404 240 404 chr12:107919-108323 0.9665721394030915
chr12 153232  153470  238 222 222 238 chr12:153236-153458 0.9955551496433741
chr12 177749  177989  240 NA  NA  NA  NA  NA
chr12 189165  189405  240 404 240 404 chr12:189072-189476 0.9665721394030915
```

column 1 to 3
: The genomic coordinate of CTCF peak.

column 4 (A.size)
: The size of CTCF peak.

column 5 (B.size)
: The size (cardinality) of RAD21 peak(s) that were overlapped with this CTCF peak.

column 6 (AB)
: The size (cardinality) of intersection.

column 7 (AB)
: The size (cardinality) of union.

column 8 (B.list)
: List of RAD21 peak(s) that are overlapped with this peak. Multiple peaks will be separated by ",".

column 9 (Score)
: The peakwise NPMI.

CHAPTER

FOURTEEN

COOCCURRENCE

14.1 Description

Use Fisher's exact test to evaluate if two sets of genomic intervals (A and B) are significantly cooccured[1]. Genomic intervals (**g**) in the background BED file will be divided into 4 groups: **a** (A specific), **b** (B specific), **c** (A and B cooccur), and **n** (neith A nor B).

	Not A	A	Total
Not B	n	a	n+a
B	b	c	b+c
Total	n+b	a+c	g = a + b + c + n

Fisher's exact test p-value is calculated as:

$$p = \frac{(n+a)!\,(b+c)!\,(n+b)!\,(a+c)!}{g!\,a!\,b!\,c!\,n!}$$

Odds ratio is calculated as:

$$OR = \frac{\frac{n}{a}}{\frac{b}{c}} = \frac{nc}{ab}$$

14.2 Usage

```
cobind.py cooccur -h
```

```
usage: cobind.py cooccur [-h] [--nameA NAMEA] [--nameB NAMEB] [--ncut N_CUT]
                         [--pcut P_CUT] [-l log_file] [-d]
                         input_A.bed input_B.bed background.bed output.tsv

positional arguments:
```

(continues on next page)

[1] Note: "cooccur" does NOT necessarily mean "overlap" or "cobinding". For example, two transcription factors could bind to the same promoter region without touching each other.

(continued from previous page)

```
  input_A.bed           Genomic regions in BED, BED-like or bigBed format. The
                        BED-like format includes:'bed3', 'bed4', 'bed6',
                        'bed12', 'bedgraph', 'narrowpeak', 'broadpeak',
                        'gappedpeak'. BED and BED-like format can be plain
                        text, compressed (.gz, .z, .bz, .bz2, .bzip2) or
                        remote (http://, https://, ftp://) files. Do not
                        compress BigBed foramt. BigBed file can also be a
                        remote file.
  input_B.bed           Genomic regions in BED, BED-like or bigBed format. The
                        BED-like format includes:'bed3', 'bed4', 'bed6',
                        'bed12', 'bedgraph', 'narrowpeak', 'broadpeak',
                        'gappedpeak'. BED and BED-like format can be plain
                        text, compressed (.gz, .z, .bz, .bz2, .bzip2) or
                        remote (http://, https://, ftp://) files. Do not
                        compress BigBed foramt. BigBed file can also be a
                        remote file.
  background.bed        Genomic regions as the background (e.g., all
                        promoters, all enhancers).
  output.tsv            For each genomic region in the "background.bed" file,
                        add another column indicating if this region is
                        "input_A specific (i.e., A+B-)", "input_B specific
                        (i.e., A-B+)", "co-occur (i.e., A+B+)" or "neither
                        (i.e, A-B-)".

options:
  -h, --help            show this help message and exit
  --nameA NAMEA         Name to represent 1st set of genomic interval. If not
                        specified "A" will be used.
  --nameB NAMEB         Name to represent 2nd set of genomic interval. If not
                        specified "B" will be used.
  --ncut N_CUT          The minimum overlap size. (default: 1)
  --pcut P_CUT          The minimum overlap percentage. (default: 0.000000)
  -l log_file, --log log_file
                        This file is used to save the log information. By
                        default, if no file is specified (None), the log
                        information will be printed to the screen.
  -d, --debug           Print detailed information for debugging.
```

14.3 Example

cobind.py cooccur CTCF_ENCFF660GHM.bed RAD21_ENCFF057JFH.bed hg38_gene_hancer_v4.4.bed output.tsv

```
2022-01-20 01:24:40 [INFO]  Calculate the co-occurrence of two sets of genomic intervals␣
↪...
2022-01-20 01:24:40 [INFO]  Read and union BED file: "CTCF_ENCFF660GHM.bed"
2022-01-20 01:24:41 [INFO]  Read and union BED file: "RAD21_ENCFF057JFH.bed"
2022-01-20 01:24:41 [INFO]  Read and union background BED file: "hg38_gene_hancer_v4.4.
↪bed"
2022-01-20 01:24:42 [INFO]  Build interval tree for : "CTCF_ENCFF660GHM.bed"
```

(continues on next page)

(continued from previous page)

```
2022-01-20 01:24:42 [INFO]  Build interval tree for: "RAD21_ENCFF057JFH.bed"
A.name          CTCF_ENCFF660GHM.bed
B.name        RAD21_ENCFF057JFH.bed
A.count                       58584
B.count                       31955
G.count                      218099
A+,B-                         11545
A-,B+                          2525
A+,B+                         19602
A-,B-                        184427
odds-ratio                124.0137
p-value                      0.0000
Name: Fisher's exact test result, dtype: object
```

A.count
: Number of unique genomic intervals in “CTCF_ENCFF660GHM.bed”.

B.count
: Number of unique genomic intervals in “RAD21_ENCFF057JFH.bed”.

G.count
: Number of unique genomic intervals in background “hg38_gene_hancer_v4.4.bed” (**g**).

A+,B-
: Number of unique genomic intervals that are overlapped with A not B (**a**).

A-,B+
: Number of unique genomic intervals that are overlapped with B not A (**b**).

A+,B+
: Number of unique genomic intervals that are overlapped with both A and B (**c**).

A-,B-
: Number of unique genomic intervals that are overlapped with neither A nor B (**n**).

CHAPTER

FIFTEEN

COVARY

15.1 Description

Evaluate the signal correlations (Pearson's r , Spearman's , and Kendall's) between two sets of genomic intervals.

15.2 Usage

```
cobind.py covary -h
```

```
usage: cobind.py covary [-h] [--nameA NAMEA] [--nameB NAMEB] [--na NA_LABEL]
                        [--type {mean,min,max}] [--topx TOP_X]
                        [--min_sig MIN_SIGNAL] [--exact] [--keepna]
                        [-l log_file] [-d]
                        input_A.bed input_A.bw input_B.bed input_B.bw
                        output_prefix

positional arguments:
  input_A.bed           Genomic regions in BED, BED-like or bigBed format. The
                        BED-like format includes:'bed3', 'bed4', 'bed6',
                        'bed12', 'bedgraph', 'narrowpeak', 'broadpeak',
                        'gappedpeak'. BED and BED-like format can be plain
                        text, compressed (.gz, .z, .bz, .bz2, .bzip2) or
                        remote (http://, https://, ftp://) files. Do not
                        compress BigBed foramt. BigBed file can also be a
                        remote file.
  input_A.bw            Input bigWig file matched to 'input_A.bed'. BigWig
                        file can be local or remote. Note: the chromosome IDs
                        must be consistent between BED and bigWig files.
  input_B.bed           Genomic regions in BED, BED-like or bigBed format. The
                        BED-like format includes:'bed3', 'bed4', 'bed6',
                        'bed12', 'bedgraph', 'narrowpeak', 'broadpeak',
                        'gappedpeak'. BED and BED-like format can be plain
                        text, compressed (.gz, .z, .bz, .bz2, .bzip2) or
                        remote (http://, https://, ftp://) files. Do not
                        compress BigBed foramt. BigBed file can also be a
                        remote file.
  input_B.bw            Input bigWig file matched to 'input_B.bed'. BigWig
                        file can be local or remote. Note: the chromosome IDs
                        must be consistent between BED and bigWig files.
```

(continues on next page)

(continued from previous page)

```
  output_prefix         Prefix of output files. Three files will be generated:
                        "output_prefix_bedA_unique.tsv" (input_A.bed specific
                        regions and their bigWig scores),
                        "output_prefix_bedB_unique.tsv" (input_B.bed specific
                        regions and their bigWig scores), and
                        "output_prefix_common.tsv"(input_A.bed and input_B.bed
                        overlapped regions and their bigWig scores).

options:
  -h, --help            show this help message and exit
  --nameA NAMEA         Name of the 1st set of genomic interval, if not
                        proviced, "bedA" will be used. Only affects the name
                        of output file.
  --nameB NAMEB         Name of the 2nd set of genomic interval, if not
                        proviced, "bedB" will be used. Only affects the name
                        of output file.
  --na NA_LABEL         Symbols used to represent the missing values.
                        (default: nan)
  --type {mean,min,max}
                        Summary statistic score type ('min','mean' or 'max')
                        of a genomic region. (default: mean)
  --topx TOP_X          Fraction (if 0 < top_X <= 1) or number (if top_X > 1)
                        of genomic regions used to calculate Pearson,
                        Spearman, Kendall's correlations. If TOP_X == 1 (i.e.,
                        100%), all the genomic regions will be used to
                        calculate correlations. (default: 1.0)
  --min_sig MIN_SIGNAL  Genomic region with summary statistic score <= this
                        will be removed. (default: 0)
  --exact               If set, calculate the "exact" summary statistic score
                        rather than "zoom-level" score for each genomic
                        region.
  --keepna              If set, a genomic region will be kept even it does not
                        have summary statistical score in either of the two
                        bigWig files. This flag only affects the output TSV
                        files.
  -l log_file, --log log_file
                        This file is used to save the log information. By
                        default, if no file is specified (None), the log
                        information will be printed to the screen.
  -d, --debug           Print detailed information for debugging.
```

15.3 Example

```
cobind.py covary CTCF_ENCFF660GHM.bed3 CTCF_ENCFF682MFJ_FC.bigWig RAD21_ENCFF057JFH.bed3
RAD21_ENCFF130GMP.bigWig output
```

```
2022-01-20 02:56:53 [INFO]  Read and union BED file: "CTCF_ENCFF660GHM.bed3"
2022-01-20 02:56:54 [INFO]  Unioned regions of "CTCF_ENCFF660GHM.bed3" : 58584
2022-01-20 02:56:54 [INFO]  Read and union BED file: "RAD21_ENCFF057JFH.bed3"
```

(continues on next page)

(continued from previous page)

```
2022-01-20 02:56:54 [INFO]  Unioned regions of "RAD21_ENCFF057JFH.bed3" : 31955
...
                Correlation  P-value
Pearson_cor:         0.6378   0.0000
Spearman_rho:        0.6355   0.0000
Kendall_tau:         0.4406   0.0000
2022-01-20 02:57:06 [INFO]  Calculate covariabilities of "CTCF_ENCFF660GHM.bed3"
                            unique regions ...
2022-01-20 02:57:16 [INFO]  Sort dataframe by summary statistical scores ...
2022-01-20 02:57:16 [INFO]  Save dataframe to: "output_bedA_unique.tsv"
2022-01-20 02:57:16 [INFO]  Select 30347 regions ...
                Correlation  P-value
Pearson_cor:         0.3356   0.0000
Spearman_rho:        0.3667   0.0000
Kendall_tau:         0.2489   0.0000
2022-01-20 02:57:16 [INFO]  Calculate covariabilities of "RAD21_ENCFF057JFH.bed3"
                            unique regions ...
2022-01-20 02:57:18 [INFO]  Sort dataframe by summary statistical scores ...
2022-01-20 02:57:18 [INFO]  Save dataframe to: "output_bedB_unique.tsv"
2022-01-20 02:57:18 [INFO]  Select 3822 regions ...
                Correlation  P-value
Pearson_cor:         0.2511   0.0000
Spearman_rho:        0.2261   0.0000
Kendall_tau:         0.1534   0.0000
```

CHAPTER

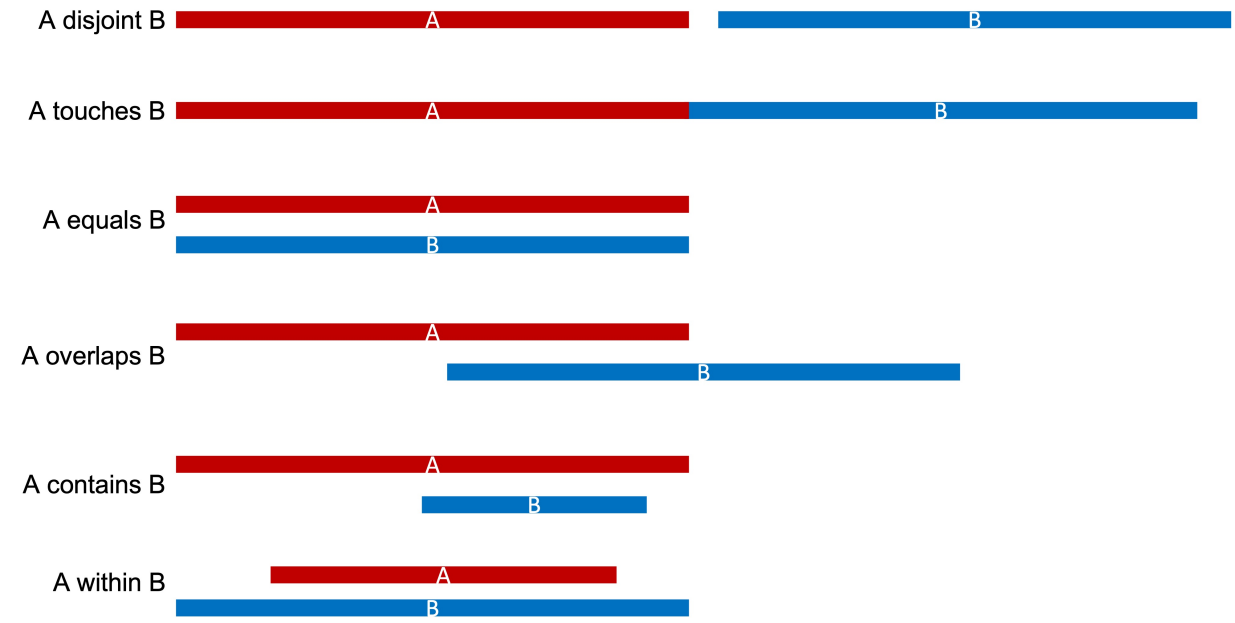
SIXTEEN

SPATIAL RELATION OF GENOMIC (SROG) INTERVALS

16.1 Description

Match up two sets of genomic intervals, and report the code of Spatial Relation Of Genomic (SROG). SROG codes include `disjoint`, `touch`, `equal`, `overlap`, `contain`, `within`.

Figure 2



16.2 Usage

```
cobind.py srog -h
```

```
usage: cobind.py srog [-h] [--dist MAX_DIST] [-l log_file] [-d]
                      input_A.bed input_B.bed output.tsv

positional arguments:
  input_A.bed           Genomic regions in BED, BED-like or bigBed format. If
```

(continues on next page)

(continued from previous page)

```
                        'name' (the 4th column) is not provided, the default
                        name is "chrom:start-end". If strand (the 6th column)
                        is not provided, the default strand is "+".
  input_B.bed           Genomic regions in BED, BED-like or bigBed format. If
                        'name' (the 4th column) is not provided, the default
                        name is "chrom:start-end". If strand (the 6th column)
                        is not provided, the default strand is "+".
  output.tsv            Generate spatial relation code (disjoint, touch,
                        equal, overlap, contain, within) for each genomic
                        interval in "input_A.bed".

options:
  -h, --help            show this help message and exit
  --dist MAX_DIST       When intervals are disjoint, find the closest up- and
                        down-stream intervals that are no further than
                        `max_dist` away. default: 250000000)
  -l log_file, --log log_file
                        This file is used to save the log information. By
                        default, if no file is specified (None), the log
                        information will be printed to the screen.
  -d, --debug           Print detailed information for debugging.
```

16.3 Example

`cobind.py srog CTCF_ENCFF660GHM.bed3 RAD21_ENCFF057JFH.bed3 output.tsv`

```
2022-01-20 09:01:17 [INFO]  Determine the spacial realtions of genomic (SROG) intervals .
↪..
2022-01-20 09:01:17 [INFO]  Build interval tree from file: "RAD21_ENCFF057JFH.bed3"
2022-01-20 09:01:17 [INFO]  Reading BED file: "CTCF_ENCFF660GHM.bed3"
disjoint    30419
overlap      4341
contain      1695
within      23214
touch           0
equal           1
other           0
dtype: int64
```

Match up results were saved to `output.tsv`

```
$head -10 output.tsv

chr12 53676079  53676369  within  chr12:53676060-53676382
chr12 57905364  57905661  within  chr12:57905272-57905699
chr22 20564334  20564661  contain chr22:20564370-20564581
chr16 57649065  57649362  within  chr16:57649007-57649370
chr17 45135294  45135610  overlap chr17:45135296-45135642
chr15 40274737  40275016  within  chr15:40274714-40275018
chr1  114346538 114346847 within  chr1:114346526-114346903
```

(continues on next page)

(continued from previous page)

```
chr7  151172578 151172888 overlap chr7:151172565-151172865
chr1  225474965 225475268 within  chr1:225474919-225475330
chr5  179668464 179668730 contain chr5:179668495-179668674
...
chr22   23128466        23128723        disjoint        UpInterval=chr22:22651059-
↪22651463,
                                                        DownInterval=chr22:23385972-
↪23386169
...
```

Column 1-3
Genome intervals from "CTCF_ENCFF660GHM.bed3".

Column 4
SROG code. When SORG = `disjoint`, two closest intervals (up- and down-stream) from `RAD21_ENCFF057JFH.bed3` were reported.

column 5
Genomic intervals from `RAD21_ENCFF057JFH.bed3`.

CHAPTER

SEVENTEEN

STAT

17.1 Description

Wrapper function. Report basic statistics of genomic intervals, including

- count
- total size
- unique size
- mean size
- median size
- min size
- max size
- Standard deviation

and calculate overlapping measurements, including

- collocation coefficient (C)
- Jaccard similarity coefficient (J)
- Sørensen–Dice coefficient (SD)
- Szymkiewicz–Simpson coefficient (SS)
- pointwise mutual information (PMI)
- normalized pointwise mutual information (NPMI)

17.2 Usage

```
cobind.py stat -h
```

```
usage: cobind.py stat [-h] [--nameA NAMEA] [--nameB NAMEB] [-b BGSIZE]
                      [-l log_file] [-d]
                      input_A.bed input_B.bed

positional arguments:
  input_A.bed           Genomic regions in BED, BED-like or bigBed format. The
                        BED-like format includes:'bed3', 'bed4', 'bed6',
```

(continues on next page)

(continued from previous page)

```
                        'bed12', 'bedgraph', 'narrowpeak', 'broadpeak',
                        'gappedpeak'. BED and BED-like format can be plain
                        text, compressed (.gz, .z, .bz, .bz2, .bzip2) or
                        remote (http://, https://, ftp://) files. Do not
                        compress BigBed foramt. BigBed file can also be a
                        remote file.
  input_B.bed           Genomic regions in BED, BED-like or bigBed format. The
                        BED-like format includes:'bed3', 'bed4', 'bed6',
                        'bed12', 'bedgraph', 'narrowpeak', 'broadpeak',
                        'gappedpeak'. BED and BED-like format can be plain
                        text, compressed (.gz, .z, .bz, .bz2, .bzip2) or
                        remote (http://, https://, ftp://) files. Do not
                        compress BigBed foramt. BigBed file can also be a
                        remote file.

options:
  -h, --help            show this help message and exit
  --nameA NAMEA         Name to represent 1st set of genomic interval. If not
                        specified (None), the file name ("input_A.bed") will
                        be used.
  --nameB NAMEB         Name to represent the 2nd set of genomic interval. If
                        not specified (None), the file name ("input_B.bed")
                        will be used.
  -b BGSIZE, --background BGSIZE
                        The size of the cis-regulatory genomic regions. This
                        is about 1.4Gb For the human genome. (default:
                        1400000000)
  -l log_file, --log log_file
                        This file is used to save the log information. By
                        default, if no file is specified (None), the log
                        information will be printed to the screen.
  -d, --debug           Print detailed information for debugging.
```

17.3 Example

cobind.py stat CTCF_ENCFF660GHM.bed RAD21_ENCFF057JFH.bed

```
2022-07-09 09:44:12 [INFO]  Gathering information for "CTCF_ENCFF660GHM.bed" ...
2022-07-09 09:44:12 [INFO]  Gathering information for "RAD21_ENCFF057JFH.bed" ...
A.name                    CTCF_ENCFF660GHM.bed
A.interval_count                         58684
A.interval_total_size                 12190325
A.interval_mean_size                  207.7283
A.interval_median_size                240.0000
A.interval_min_size                         60
A.interval_max_size                        576
A.interval_size_SD                     51.5489
B.name                   RAD21_ENCFF057JFH.bed
B.interval_count                         33373
B.interval_total_size                 11381586
```

(continues on next page)

(continued from previous page)

```
B.interval_mean_size                341.0417
B.interval_median_size              404.0000
B.interval_min_size                      101
B.interval_max_size                      553
B.interval_size_SD                   96.8607
G.size                         1400000000.0000
A.size                             12184840
Not_A.size                     1387815160.0000
B.size                             11130268
Not_B.size                     1388869732.0000
A_not_B.size                        7245355
B_not_A.size                        6190783
A_and_B.size                        4939485
A_and_B.exp_size                  96871.8105
A_or_B.size                        18375623
Neither_A_nor_B.size           1381624377.0000
coef.Collocation                       0.4241
coef.Jaccard                           0.2688
coef.Dice                              0.4237
coef.SS                                0.4438
A_and_B.PMI                            3.9316
A_and_B.NPMI                           0.6962
dtype: object
```

CHAPTER

EIGHTEEN

Z-SCORE

18.1 Description

Calculate Z-score as an overall measurement for these six metrics. The Z-score approach becomes valuable , for example, when we are comparing a query TF with multiple other TFs to identify potential co-factors.

- collocation coefficient (C)
- Jaccard similarity coefficient (J)
- Sørensen–Dice coefficient (SD)
- Szymkiewicz–Simpson coefficient (SS)
- pointwise mutual information (PMI)
- normalized pointwise mutual information (NPMI)

First, values of the six metrics were converted into Z-scores by Z_i = (x –)/, where and are the average and standard deviation of the score, and i belongs to {C, J, SD, SS, PMI, NPMI}. Then, the combined Z-score is defined as:

$$Z = \frac{\sum Z_i}{\sqrt{6}}$$

18.2 Usage

```
cobind.py zscore -h
```

```
usage: cobind.py zscore [-h] [-l log_file] [-d] input_file.tsv output_file.tsv

positional arguments:
  input_file.tsv        Input dataframe with row names and column names. Must
                        separate different columns with tab. If "C", "J",
                        "SD", "SS", "PMI", "NPMI" are used as the column
                        names, only these six columns will be used to
                        calculate the Z-score, otherwise, all numerical
                        columns in the dataframe will be used.
```

(continues on next page)

(continued from previous page)

```
  output_file.tsv       Output dataframe with Z-scores as the last column.

options:
  -h, --help            show this help message and exit
  -l log_file, --log log_file
                        This file is used to save the log information. By
                        default, if no file is specified (None), the log
                        information will be printed to the screen.
  -d, --debug           Print detailed information for debugging.
```

18.3 Example

First, download the test file: CTCF_vs_ReMap.tsv

```
cobind.py zscore CTCF_vs_ReMap.tsv output.tsv
```

```
2023-07-06 10:20:35 [INFO]  Calculate Z-scores from "CTCF_vs_ReMap.tsv"

              C      J     SD     SS     PMI    NPMI
 TF_name
 RAD21    0.1446 0.0224 0.0438 0.9326  2.0074  0.3417
 SMC3     0.1430 0.0214 0.0420 0.9525  2.0285  0.3428
 SMC1A    0.1413 0.0211 0.0413 0.9462  2.0219  0.3407
 TRIM22   0.1400 0.0214 0.0419 0.9127  1.9858  0.3355
 STAG1    0.1368 0.0191 0.0375 0.9787  2.0556  0.3407
 ...         ...    ...    ...    ...     ...     ...
 SVIL     0.0017 0.0000 0.0000 0.1376  0.0936  0.0073
 ZNF212   0.0014 0.0000 0.0000 0.1473  0.1616  0.0122
 ZNF570   0.0012 0.0000 0.0000 0.0955 -0.2710 -0.0205
 SIRT3    0.0011 0.0000 0.0000 0.1249 -0.0033 -0.0002
 GLI1     0.0003 0.0000 0.0000 0.0267 -1.5442 -0.1054

 [1207 rows x 6 columns]

 2023-07-06 10:20:35 [INFO]  Save Z-scores to "output.tsv"
               C       J      SD      SS     PMI    NPMI  Zscore
 TF_name
 RAD21     3.5704  3.3312  3.2881  3.8229  2.0169  2.7221  7.6553
 SMC3      3.5114  3.1213  3.0964  3.9639  2.0615  2.7375  7.5493
 SMC1A     3.4488  3.0584  3.0218  3.9192  2.0475  2.7082  7.4317
 TRIM22    3.4009  3.1213  3.0857  3.6819  1.9711  2.6355  7.3062
 STAG1     3.2830  2.6387  2.6172  4.1494  2.1189  2.7082  7.1506
 ...          ...     ...     ...     ...     ...     ...     ...
 SVIL     -1.6949 -1.3698 -1.3765 -1.8082 -2.0337 -1.9490 -4.1772
 ZNF212   -1.7059 -1.3698 -1.3765 -1.7395 -1.8898 -1.8806 -4.0670
 ZNF570   -1.7133 -1.3698 -1.3765 -2.1064 -2.8054 -2.3373 -4.7801
 SIRT3    -1.7170 -1.3698 -1.3765 -1.8982 -2.2388 -2.0538 -4.3495
 GLI1     -1.7465 -1.3698 -1.3765 -2.5937 -5.5001 -3.5233 -6.5768

 [1207 rows x 7 columns]
```

CHAPTER

NINETEEN

COMPARE DIFFERENT METRICS

The table below gives the lower and upper bounds of the 6 metrics and their major drawbacks if any.

Metric	*Lower bound*	*Upper bound*	*Comments*
C(A,B)	0 (no overlap)	1 (A = B)	
J(A,B)	0 (no overlap)	1 (A = B)	Bias towards the **larger** interval
SD(A,B)	0 (no overlap)	1 (A = B)	Bias towards the **larger** interval
SS(A,B)	0 (no overlap)	1 (A = B, A B, or B A)	Bias towards the **smaller** interval
PMI	-inf (no overlap)	min(-log(p(A)), -log(p(B)))	No fixed bound
NPMI	-1 (no overlap)	1 (A = B)	

The table below compares the intersection-based metrics. **C**, **J**, **SD**, and **SS**. All the four metrics are bounded by 0 and 1. When the size of the two genomic intervals are significanlty different, **C** is less sensitive to the extreme, and gives a compromised score compared to **J/SD** and **SS**.

Table 1: **C(A,B)** vs **J(A,B)** vs **SD(A,B)** vs **SS(A,B)**

SROG	|A|	|B|	|A B|	|A B|	*C*	*J*	*SD*	*SS*
A equals B	1000	1000	1000	1000	1	1	1	1
A disjoint B	1000	1000	0	2000	0	0	0	0
A overlaps B	100	1000	50	1050	0.158	0.0476	0.0909	0.5
A within B	100	1000	100	1000	0.316	0.1	0.182	1

CHAPTER

TWENTY

CTCF: DEMONSTRATION

70-95% of CTCF binding sites are also bound by cohesin complex (including SMC1, SMC3, RAD21, STAG1, and STAG2) to establish chromatin loops and regulate gene expression[1],[2].

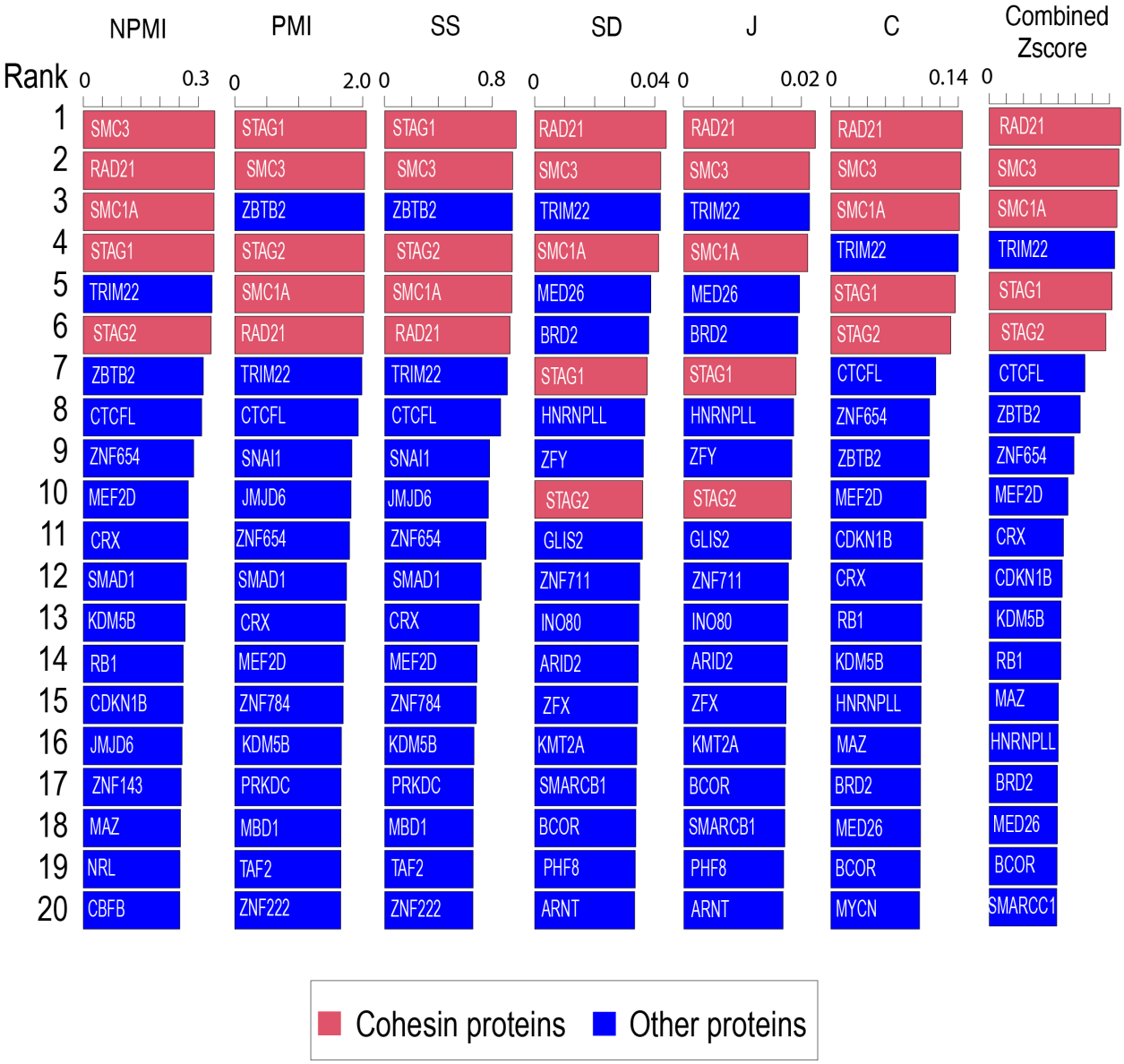
We used CTCF-cohesin as a positive control to evaluate the performance of the six collocation measurements (including C, J, SD, SS, PMI and NPMI). We first calculated the scores of these metrics between all the binding sites (defined as cistrome) of CTCF with those cistromes of 1207 TFs curated in the ReMap database. Then, we calculate the Zscore as an overall measurement of the cobindability. Please note, TRIM22 is not part of the cohesin complex, but multiple studies have identified TRIM22 as a critical regulator of chromatin structure. TRIM22 bindings are highly enriched at chromatin contact domain boundaries[3],[4].

[1] Pugacheva EM, Kubo N, Loukinov D, et al. CTCF mediates chromatin looping via N-terminal domain-dependent cohesin retention. Proc Natl Acad Sci U S A. 2020;117(4):2020-2031. doi:10.1073/pnas.1911708117

[2] Xiao T, Li X, Felsenfeld G. The Myc-associated zinc finger protein (MAZ) works together with CTCF to control cohesin positioning and genome organization. Proc Natl Acad Sci U S A. 2021;118(7):e2023127118. doi:10.1073/pnas.2023127118

[3] Chen F, Li G, Zhang MQ, Chen Y. HiCDB: a sensitive and robust method for detecting contact domain boundaries. Nucleic Acids Res. 2018;46(21):11239-11250. doi:10.1093/nar/gky789

[4] Di Pierro M, Cheng RR, Lieberman Aiden E, Wolynes PG, Onuchic JN. De novo prediction of human chromosome structures: Epigenetic marking patterns encode genome architecture. Proc Natl Acad Sci U S A. 2017;114(46):12126-12131. doi:10.1073/pnas.1714980114



Collocation between CTCF binding sites and the binding sites of 1207 TFs were evaluated uing the six measurements as well as the zscore. Only the top 20 TFs were displayed.

CHAPTER

TWENTYONE

PERFORMANCE (CPU & MEMORY USAGE)

The CPU time & memory usage for running cobind.py stat between two bed file with different number of intervals.

CPU model: Intel(R) Xeon(R) Gold 6248 CPU @ 2.50GHz

# of intervals in A	*# of intervals in B*	*Real time (seconds)*	*Max memory (GB)*
100,000	100,000	7.040	0.822
200,000	200,000	9.384	0.938
300,000	300,000	11.798	0.957
400,000	400,000	14.307	1.002
500,000	500,000	16.563	1.049
600,000	600,000	18.893	1.057
700,000	700,000	21.599	1.097
800,000	800,000	23.874	1.146
900,000	900,000	27.262	1.166
1,000,000	1,000,000	29.472	1.220

CHAPTER

TWENTYTWO

LICENSE

CHAPTER

TWENTYTHREE

ACKNOWLEDGEMENTS

Cobind is funded in part by the **bioinformatics program** and the **epigenomics program** of the Center for Individualized Medicine (CIM) in Mayo Clinic.

CHAPTER

TWENTYFOUR

CONTACT

Bugs report to github

CHAPTER

TWENTYFIVE

REFERENCE

Ma T, Guo L, Yan H, Wang L. Cobind: quantitative analysis of the genomic overlaps. **Bioinformatics Advances**. 2023; vbad104.