HiC1Dmetrics

Release 0.0.1

Jiankang Wang

Oct 23, 2022

CONTENTS:

CHAPTER

ONE

1. OVERVIEW AND INSTALLATION

1.1 1.1 Introduction

We presented "HiC1Dmetrics", a pipeline that are able to calculate and analysis one-dimensional (1D) metrics for Hi-C data. HiC1Dmetrics is a Python3-based program (https://github.com/wangjk321/HiC1Dmetrics) and provide command line interface for UNIX system.

1.2 1.2 Installation

HiC1Dmetrics were released on PyPI, and could be accessed by:

```
pip3 install h1d
```

After installation, try:

```
$ h1d -V
h1d version 0.1.20
```

1.3 1.3 Requirements

HiC1Dmetrics is based on python3 and it requires:

- Python (>= 3.6) packages:
  - pandas
  - numpy
  - scipy
  - scikit-learn
  - statsmodels
  - matplotlib
  - seaborn
  - fithic==2.0.7
  - multiprocess
  - cooler == 0.8.2

- Others:
  - bedtools >= 2.29.2

All required python packages will be automatically installed when use `pip install h1d`

1.4 1.4 Test file

In this tutorial, we prepared servel files for test. **All results in this tutorial can be reproduced using test data**. We mainly used the data from GSE104334.

- Zipped intra-chromosomal contacts (can be find in the ./test_data/ of GitHub page)
  - GSE104334_Ctrl.chr21.matrix.gz
  - GSE104334_Rad21KD.chr21.matrix.gz
- Hi-C file in .hic (Juicer) format (the download link is in ./test_data/ of GitHub page)
  - GSE104334_Ctrl.hic
  - GSE104334_Rad21KD.hic
- Hi-C file in .cool (Cooler) format (the download link is in ./test_data/ of GitHub page)
  - GSE104334_Ctrl.cool
  - GSE104334_Rad21KD.cool
- Two folder for the test of multisamples as described here:
  - multisample1
  - multisample2
- Other files that support the test (can be find in the ./test_data/ of GitHub page):
  - hg19_genome_table.txt
  - hg19_geneDensity50000.txt

1.5 1.5 Input format

HiC1Dmetrics support:

- raw `.hic` defined by juicer software.
- OR raw `cool` defined by cooler software
- OR dense matrix (raw or zipped) of intra-chromosomal contacts, like:

Note 1: when use `.hic` file, a genome table file (tab-separated) must be prepared, which described the length of each chromosome for your genome reference:

Note 2: Calculation of Interaction Frequency (IF) can only accept raw `.hic` file

1.6 1.6 Overall usage

The function of HiC1Dmetrics include serveral sub-command:

```
$ h1d -h
positional arguments:
  {basic,one,two,multitypes,multisamples,call}
                        Choose the mode to use sub-commands
    basic               Provide basic functions to visualize and handle Hi-C
                        data.
    one                 1D metrics designed for one Hi-C sample
    two                 1D metrics designed for comparison of two Hi-C samples
    multitypes          Various types of 1D metrics for the same sample
    multisamples        The same metrics for muliple samples
    call                Extract secondary information from metrics (dTAD,
                        stripeTAD, et.al)
```

In the next section, we mainly use a high-resolution Hi-C data by Rao et,al. (GSE104334) to make this tutorial

1.7 1.7 Memory requirement

- HiC1Dmetrics uses acceptable memory (serveral gigabytes) on high-resolution (i.e. >=5kb) Hi-C
- Current HiC1Dmetrics does not assume the Hi-C data with super-high resolution (e.g. <1kb). The application on fine resolution Hi-C will use much more memory. i.e., for 1kb resolution Hi-C data (GSE63525, chromosome 21), '.hic' input uses ~69 Gb memory, while 'dense matrix' input uses ~10 Gb memory; For mouse chromosome 1, 1kb-data uses >200 Gb memory.
- One reason is that we extract matrix from .hic file using Juicertools (the first step), which requires hundreds of gigabytes of memory (>300 Gb at chromosome 1) in 1kb resolution. The other reason is that h1d still need to load and deal with (the second step) the huge 2D matrix (e.g., the file size of 1kb dense matrix of chromosome 1 is ~ 233 Gb).
- We recommend using the dense matrix as the input instead of .hic/ .cool file to reduce the memory usage.

CHAPTER

TWO

2. CALCULATE 1D METRICS FOR ONE SAMPLE

2.1 2.1 Quick start

```
h1d one IS ./test_data/GSE104334_Ctrl.chr21.matrix.gz \
                50000 chr21 -o Control_IS_chr21
```

This command would generate a bedGraph file (`Control_IS_chr21.bedGraph`) for Insulation Score. An example file can be download fron here

2.2 2.2 Usage

The analysis of one-sample metrics cound be run by `h1d one` sub-command :

```
$ h1d one -h # type -h for help
usage: h1d one [-h] [-p PARAMETER] [-o OUTNAME] [-d] [-s START] [-e END]
               [--datatype DATATYPE] [--gt GT] [--prefix PREFIX]
               [--maxchr MAXCHR] [-n NPROCESSER] [-t TADFILE]
               type data resolution chromosome

1D metrics designed for one Hi-C sample.

positional arguments:
  type                  Type of 1D metrics,,should be one of
                        {IS,CI,DI,SS,DLR,PC1,IES,IAS,IF}.
  data                  Path of matrix or rawhic file.
  resolution            Resolution of input matrix.
  chromosome            Chromosome number.

optional arguments:
  -h, --help            show this help message and exit
  -p PARAMETER, --parameter PARAMETER
                        Parameter for indicated metrics.
  -o OUTNAME, --outname OUTNAME
                        output name (default: 'metrics').
  -d, --draw            Plot figure for candidate region.
  -s START, --start START
                        Start sites for plotting.
```

(continues on next page)

(continued from previous page)

```
  -e END, --end END     End sites for plotting.
  --datatype DATATYPE   Type of input data: [matrix(default),rawhic,cool]
  --msi MSI             Method for significant interactions: [fithic2,hiccups]
  --gt GT               genome_table file.
  --prefix PREFIX       ${prefix}chr1.matrix.gz
  --maxchr MAXCHR       Maximum index of chromosome (human genome is 22,i.e.)
  -n NPROCESSER, --nProcesser NPROCESSER
                        Number of processors
  -t TADFILE, --TADfile TADFILE
                        Give a TAD file, instead of using building-in TAD
                        calling method
```

- `type` : type of 1D metrics could be one of {IS,CI,DI,SS,DLR,PC1,IES,IAS,IF}:
  - Directional Index (**DI**) (PMID: 22495300)
  - Insulation Score (**IS**) (PMID: 26030525)
  - Contrast Index (**CI**) (PMID: 24981874)
  - TAD separation score (**SS**) (PMID: 26431028)
  - Distal-to-Local Ratio (**DLR**) (PMID: 30146161)
  - Compartment PC1 (**PC1**) (PMID: 19815776)
  - IntraTADscore (**IAS**) (Original metric)
  - InterTADscore (**IES**) (Original metric)
  - Interaction Frequency (**IF**)(Original metric)

  Details is shown in our paper: *link in the future*
- `-p, --parameter` for each 1D metric is :

Note !! : The sign of PC1 value is arbitrary unless provide a geneDensity file.

- `-t TADFILE`, specify a TAD file (.bed format) to replace the built-in TAD calling method.
- `--msi`, specify the method to calculate significant interactions. 'fithic2' (default) or 'hiccups' is supported.

2.3 2.3 Calculate 1D metrics (one-sample)

- Use contact matrix:

```
h1d one CI ./test_data/GSE104334_Ctrl.chr21.matrix.gz \
        50000 chr21 -p 300000 -o control_CI_chr21 --datatype matrix
```

- Use raw `.hic` file:

```
h1d one CI ./test_data/GSE104334_Ctrl.hic \
        50000 chr21  -p 300000 -o control_CI_chr21 --datatype rawhic \
        --gt ./test_data/hg19_genome_table.txt
```

- Use raw `.cool` file

```
h1d one CI ./test_data/GSE104334_Ctrl.50000.cool \
        50000 chr21 -p 300000 -o control_CI_chr21 --datatype cool
        --gt ./test_data/hg19_genome_table.txt
```

Output will be `control_CI_chr21.bedGraph` as described before.

2.3.1 Multiprocessing for all chromomes:

To deal with multiple chromosomes, you should first use dump function in h1d.

To run all chromosomes parallel, do:

```
h1d one IS ./test/Control/ 50000 all
        --maxchr 22 --prefix observed.KR. -n 30 -o control
```

- `chromosome` , set chromosome to "all" will compute metrics for all chromosomes.
- `data`, if calculating for all chromosomes, the input file should be absolute folder of contact matrix.
- `-maxchr`, Maximum index of chromosome (human genome is 22,i.e.). It will compute chromosome 1~maxchr plus chromosome X.
- `--prefix`, the prefix of matrix file, please modify the name of zipped matrix to `${prefix}chr1.matrix.gz`. If you used our dump function, the file should be:

```
├── observed.KR.chr1.matrix.gz
├── observed.KR.chr10.matrix.gz
├── observed.KR.chr11.matrix.gz
├── observed.KR.chr12.matrix.gz
├── observed.KR.chr13.matrix.gz
├── observed.KR.chr14.matrix.gz
```

  so the prefix is `observed.KR.`

- `-n`, Number of processors

Output would be `control_IS_allchr.csv`.

2.4 2.4 Visulize 1D metrics (one-sample)

- Use contact matrix:

```
h1d one CI ./test_data/GSE104334_Ctrl.chr21.matrix.gz \
        50000 chr21 -p 300000 -o Control_CI_chr21 --datatype matrix \
  --draw -s 26000000 -e 33000000
```

- Use raw hic:

```
h1d one CI ./test_data/GSE104334_Ctrl.hic \
        50000 chr21 -p 300000 -o Control_CI_chr21 --datatype rawhic \
        --gt ./test_data/hg19_genome_table.txt --draw -s 26000000 -e 33000000
```

- Use cool file:

```
h1d one CI ./test_data/GSE104334_Ctrl.50000.cool \
        50000 chr21 -p 300000 -o Control_CI_chr21 --datatype cool \
        --gt ./test_data/hg19_genome_table.txt --draw -s 26000000 -e 33000000
```

The output will be `control_CI_chr21.bedGraph` and `control_CI_chr21.pdf`:

CHAPTER

THREE

3. CALCULATE 1D METRICS FOR COMPARISON OF TWO SAMPLES

3.1 3.1 Quick start

```
h1d two ISC ./test_data/GSE104334_Rad21KD.chr21.matrix.gz \
        ./test_data/GSE104334_Ctrl.chr21.matrix.gz \
        50000 chr21 -o treat_vs_control_ISC
```

This command would generate a bedGraph file (`treat_vs_control_ISC.bedGraph`) for ISC. An example result can be checked from here

3.2 3.2 Usage

The analysis of two-sample metrics cound be run by `h1d two` sub-command :

```
$ h1d two -h # type -h for help
usage: __main__.py two [-h] [-p PARAMETER] [-o OUTNAME] [-d] [-s START]
                       [-e END] [--datatype DATATYPE] [--gt GT]
                       type matrix controlmatrix resolution chromosome

1D metrics designed for comparison of two Hi-C samples

positional arguments:
  type                  Type of 1D metrics for two-sample comparison,should be
                        one of {ISC,CIC,SSC,deltaDLR,CD,IESC,IASC,IFC,DRF}
  matrix                Path of treated file (matrix or rawhic).
  controlmatrix         Path of control file (matrix or rawhic).
  resolution            Resolution of input matrix
  chromosome            Chromosome number ('chr21',i.e).

optional arguments:
  -h, --help            show this help message and exit
  -p PARAMETER, --parameter PARAMETER
                        Parameter for indicated metrics
  -o OUTNAME, --outname OUTNAME
                        output name (default: metricsChange)
  -d, --draw            Plot figure for candidate region
  -s START, --start START
                        Start sites for plotting
  -e END, --end END     End sites for plotting
```

(continues on next page)

(continued from previous page)

```
  --datatype DATATYPE   matrix or rawhic
  --gt GT               genome table file
```

- `type` : type of 1D metrics could be one of {ISC,CIC,SSC,deltaDLR,CD,IESC,IASC,IFC,DRF}:
  - Insulation Score Change (**ISC**) (PMID: 31495782)
  - Contrast Index Change(**CIC**)
  - Separation Score Change(**SSC**)
  - Delta Distal-to-Local Ratio (**deltaDLR**) (PMID: 30146161)
  - Correlation Difference (**CD**) (PMID: 20513432)
  - IntraScore Change(**IASC**)
  - InterScore Change (**IESC**)
  - Interaction Frequency Change (**IFC**)
  - Directional Relative Frequency (**DRF**)(Original metric)

  Details is shown in our paper: *link in the future*
- `-p` or `--parameter` for each metrics:

3.3 3.3 Calculate 1D metrics (two-sample)

- Use contact matrix:

```
h1d two ISC ./test_data/GSE104334_Rad21KD.chr21.matrix.gz \
        ./test_data/GSE104334_Ctrl.chr21.matrix.gz 50000 chr21 \
        --datatype matrix -p 300000 -o treat_vs_control_ISC
```

- Use raw hic:

```
h1d two ISC ./test_data/GSE104334_Rad21KD.hic \
        ./test_data/GSE104334_Ctrl.hic 50000 chr21 \
        --datatype rawhic --gt ./test_data/hg19_genome_table.txt \
        -p 300000 -o treat_vs_control_ISC
```

- Use cool file

```
h1d two ISC ./test_data/GSE104334_Rad21KD.50000.cool \
        ./test_data/GSE104334_Ctrl.50000.cool 50000 chr21 \
        --datatype cool --gt ./test_data/hg19_genome_table.txt \
        -p 300000 -o treat_vs_control_ISC
```

3.3.1 3.3.1 Multiprocessing for all chromomes:

- `chromosome` , set chromosome to "all" will compute metrics for all chromosomes.
- `data`, if calculating for all chromosomes, the input file should be absolute folder of contact matrix.
- `-maxchr`, Maximum index of chromosome (human genome is 22,i.e.). It will compute chromosome 1~maxchr plus chromosome X.
- `--prefix`, the prefix of matrix file, please modify the name of zipped matrix to `${prefix}chr1.matrix.gz`. If you used our dump function, the file should be:

```
├── observed.KR.chr1.matrix.gz
├── observed.KR.chr10.matrix.gz
├── observed.KR.chr11.matrix.gz
├── observed.KR.chr12.matrix.gz
├── observed.KR.chr13.matrix.gz
├── observed.KR.chr14.matrix.gz
```

  so the prefix is `observed.KR.`

- `-n`, Number of processors

To run all chromosomes parallel (treat vs control), do:

```
h1d one ISC ./test/Treat/ ./test/Control/  50000 all
        --maxchr 22 --prefix observed.KR. -n 30 -o treat_vs_control
```

Output would be `treat_vs_control_IS_allchr.csv`.

3.4 3.4 Visualize 1D metrics (two-sample)

- Use contact matrix:

```
h1d two CIC ./test_data/GSE104334_Rad21KD.chr21.matrix.gz \
        ./test_data/GSE104334_Ctrl.chr21.matrix.gz 50000 chr21 \
    --datatype matrix -p 300000 -o treat_vs_control_ISC \
        --draw -s 26000000 -e 33000000
```

- Use raw `.hic` file

```
h1d two CIC ./test_data/GSE104334_Rad21KD.hic \
        ./test_data/GSE104334_Ctrl.hic 50000 chr21 \
        --datatype rawhic --gt ./test_data/hg19_genome_table.txt \
        -p 300000 -o treat_vs_control_ISC \
        --draw -s 26000000 -e 33000000
```

- Use `.cool` file

```
h1d two CIC ./test_data/GSE104334_Rad21KD.50000.cool \
        ./test_data/GSE104334_Ctrl.50000.cool 50000 chr21 \
        --datatype cool --gt ./test_data/hg19_genome_table.txt \
        -p 300000 -o treat_vs_control_ISC \
        --draw -s 26000000 -e 33000000
```

CHAPTER

FOUR

4. EXTRACT SECONDARY INFORMATION FROM METRICS (DTAD, STRIPETAD, ETC.)

4.1 4.1 Quick start

```
# dense matrix
h1d call stripe ./test_data/GSE104334_Ctrl.chr21.matrix.gz \
        50000 chr21 -o testname

# .hic
h1d call stripe ./test_data/GSE104334_Ctrl.hic \
        50000 chr21 -o testname --datatype rawhic --gt ./test_data/hg19_genome_table.txt

# .cool
h1d call stripe ./test_data/GSE104334_Ctrl.50000.cool \
        50000 chr21 -o testname --datatype cool --gt ./test_data/hg19_genome_table.txt
```

The output will be: `testname_stripe.csv`

4.2 4.2 Parameters

```
h1d call -h
usage: __main__.py call [-h] [-o OUTNAME] [-c CONTROLMATRIX]
                        [--datatype DATATYPE] [--gt GT] [-p PARAMETER]
                        mode matrix resolution chromosome
```

- Required parameters:
  - `mode`, Running mode,,should be one of {dTAD,stripe,stripeTAD, TAD,hubs}
  - `data`, Path of matrix file or raw .hic file.
  - `resolution`, resolution (50000, i.e.) of given contact matrix, or choosed resolution for analyzing `.hic` file.
  - `chromosome`, selected chromosome to be analyzed.
- Optional parameters:
  - `-o`, output name, default: defaultname
  - `-c`, contact matrix or .hic file of control sample, which is required when using "dTAD" mode.
  - `--datatype`, type of input data: "matrix" (default) or "rawhic".

- `--gt`, genome table file when using raw .hic data.
- `-p`, parameters for particular mode:

!! Please note that "stripe" is different from "stripeTAD": Stripe is the regions with "stripe" structure, whereas stripe-TAD is asymmetric TAD (may contain many stripes). Thus stripe-TAD is the classfication from all TAD.

4.3 4.3 dTAD

h1d provide the function to call dTAD as

```
h1d call dTAD ./test_data/GSE104334_Rad21KD.chr21.matrix.gz \
        50000 chr21 -c ./test_data/GSE104334_Ctrl.chr21.matrix.gz \
  --datatype matrix -o testname -p 200000-5000000
```

The output will be `testname_leftdTAD.csv` and `testname_rightdTAD.csv`, as:

4.4 4.4 stripe

This function automatically identify all regions with stripe 'structure'. The key idea is to find the sharp, strong IAS peaks. We used the similar strategy which use insulation score to identify TAD boundaries. For the stripe calling, after extracting the local maximum positions of IAS, only positions with IAS > IASmean is retained. Then, similar to Crane et.al, Nature 2015, we calculate a delta vector of IAS for each bin to extract only strong IAS peaks. To avoid clustered small peaks, the IAS value of a 'stripe' position should be higher than any position around 100kb.

```
h1d call stripe ./test_data/GSE104334_Ctrl.chr21.matrix.gz 50000 chr21 \
        --datatype matrix -o testname
```

This will output the summit of IAS signal, i.e. the stripes:

4.5 4.5 stripe-TAD

This function simply divide all TAD into "loop", "left-stripe", "right-stripe" and "other" TAD:

```
h1d call stripeTAD ./test_data/GSE104334_Ctrl.chr21.matrix.gz 50000 chr21 \
        --datatype matrix -o testname -p 300000
```

The output will be `testname_stripe.csv`, as:

4.6 4.6 Hubs

This function extract chromatin Hubs as described in PMID: 26272203

```
h1d call hubs ./test_data/GSE104334_Ctrl.hic 50000 chr21 \
        --datatype rawhic --gt ./test_data/hg19_genome_table.txt -o testname -p 0.05
```

The output will be `testname_hubs.csv` in `.bed` style, as :

4.7 4.7 TAD

This function will use Insulation Score to simply call TAD:

```
h1d call TAD ./test_data/GSE104334_Ctrl.chr21.matrix.gz 50000 chr21 \
        --datatype matrix -o testname -p 300000
```

CHAPTER

FIVE

5. OBTAIN THE SAME METRICS FOR MULIPLE SAMPLES

5.1 5.1 Calculation

```
h1d multisamples IS ./test_data/multisample1/metadata.txt \
                50000 chr21 -o multisamples_metrics
```

The output would be `multisamples_metrics.csv`:

5.2 5.2 Parameters

```
$ h1d multisamples
usage: h1d multisamples [-h] [--datatype DATATYPE]
                        [--samplelist SAMPLELIST] [--labels LABELS]
                        [-p PARAMETER] [-o OUTNAME] [--gt GT] [--corr]
                        [--heat] [--line] [--anova] [--discrete] [-s START] [-e END]
                        type data resolution chromosome
```

5.2.1 Required parameters:

- `type`, type of 1D metrics could be one of {IS,CI,DI,SS,DLR,PC1,IES,IAS,IF}, as described
- `data`, a tsv (Tab-separated) file contain name and paths for all samples (below). Used file could be contact matrix or raw `.hic` as introduced here
- `resolution`, resolution (50000, i.e.) of given contact matrix, or choosed resolution for analyzing `.hic` file.
- `chromosome`, selected chromosome to be analyzed.

5.2.2 Optional parameters:

- `--datatype`, type of input data: "matrix" (default) or "rawhic".
- `-p`, parameters for one-sample metrics.
- `-o`, output name, default: multisamples_metrics
- `--gt`, genome table file when using raw .hic data.

5.3 5.3 Visualization

- `--corr`, Plot correlation for all samples.
- `--heat`, Plot raw heatmap for all samples.
- `--line`, Plot line chart for all samples.
- `-s`, start site of plot when using `--corr` and `--line`.
- `-e`, end site of plot when using `--corr` and `--line`.

5.3.1 5.3.1 Correlation map of all samples

```
h1d multisamples IS ./test_data/multisample1/metadata.txt 50000 chr21 \
        -o multisamples_metrics --corr
```

5.3.2 5.3.2 Line chart or multiple samples:

```
h1d multisamples IS ./test_data/multisample1/metadata.txt 50000 chr21 -o multisamples_
↪metrics \
        --line -s 24500000 -e 34500000
```

5.3.3 5.3.3 Heatmap or multiple samples:

```
h1d multisamples IS ./test_data/multisample1/metadata.txt 50000 chr21 -o multisamples_
↪metrics \
        --heat -s 24500000 -e 34500000
```

5.3.4 5.3.4 Discrete heatmap

Some metrics such as PC1, can not be quantitatively compared, thus we convert it to discrete value to draw a heatmap:

```
h1d multisamples PC1 ./test_data/multisample2/metadata.txt 50000 chr19 -o multisamples2_
↪metrics \
        --discrete -s 22500000 -e 32500000 -p test_data/multisample2/mm10_
↪geneDensity50000.txt
```

5.4 5.4 Statistically comparison

As we mentioned in the manuscript, most of (except PC1 and DI) 1D metrics could be quantitatively compared. Considering most of them exhibited the unimodal distribution without obvious skew, we use ANOVA-like test to statistically compare multiple Hi-C samples. Each bin and its surrounding bins (default=2) are considered to to run such test. The obtained p-values is then adjusted via Benjamini/Hochberg method, to get the qvalue.

To run this, please specify a genomic region and type:

```
h1d multisamples IS ./test_data/multisample1/metadata.txt 50000 chr21 -o ptest --anova -
↪s 24500000 -e 34500000
```

, where the `test.txt` is described above. The out put will be:

CHAPTER

SIX

6. OBTAIN VARIOUS TYPES OF 1D METRICS FOR THE SAME SAMPLE

6.1 6.1 Calculation

```
# dense matrix
h1d multitypes IS,CI,DI \
        ./test_data/GSE104334_Ctrl.chr21.matrix.gz \
        50000 chr21 -p 300000,300000,1000000 \
        --datatype matrix -o multi_types_metric

# .hic input
h1d multitypes IS,CI,DI \
        ./test_data/GSE104334_Ctrl.hic  \
        50000 chr21 -p 300000,300000,1000000 \
        --datatype rawhic -o multi_types_metric --gt ./test_data/hg19_genome_table.txt

# .cool input
h1d multitypes IS,CI,DI \
        ./test_data/GSE104334_Ctrl.50000.cool  \
        50000 chr21 -p 300000,300000,1000000 \
        --datatype cool -o multi_types_metric --gt ./test_data/hg19_genome_table.txt
```

This will output a csv file `multi_types_metric.csv` like

6.2 6.2 Parameters

```
$ h1d multitypes
usage: h1d multitypes [-h] -p PARAMETER [-c CONTROLMATRIX]
                      [-o OUTNAME] [--datatype DATATYPE] [--gt GT]
                      [-d] [-s START] [-e END]
                      typelist data resolution chromosome
```

- **Required parameters**:
  - `typelist` the list of types separated by comma.
    - When calculating one-sample metrics, it should be subset of [IS,CI,DI,SS,DLR,PC1,IES,IAS,IF].
    - When calculating two-sample metrics, it should be subset of [ISC,CIC,SSC,deltaDLR,CD,IESC,IASC,IFC,DRF]

  - `data`, Path of matrix file or raw `.hic` file
  - `resolution`, resolution (50000, i.e.) of given contact matrix, or choosed resolution for `.hic` file
  - `chromosome`, selected chromosome to be analyzed.
  - `-p`, list of parameters (comma-separated). Please refer to one-sample or two-sample metrics.
- **Optional parameters**:
  - `-o`, output name, default: multitypes_metrics
  - `-c`, contact matrix or .hic file of control sample, which is required when calculating two-sample metrics.
  - `--datatype`, type of input data: "matrix" (default) or "rawhic".
  - `--gt`, genome table file when using raw .hic data.

6.3 6.3 Visuliazation

`-d`, decide whether to plot.

`-s`, start site for plot.

`-e`, end site for plot.

- Visulize one-sample metrics (output will be `all_onesample.csv` and `all_onesample.pdf`)

```
h1d multitypes IS,CI,DI,SS,DLR,PC1,IAS,IES,IF \
        ./test_data/GSE104334_Ctrl.hic 50000 chr21 \
        -p 300000,300000,1000000,300000,3000000,./test_data/hg19_geneDensity50000.txt,
↪300000,300000,0.05
        --datatype rawhic --gt ./test_data/hg19_genome_table.txt
        -o all_onesample -d -s 24500000 -e 34500000
```

- Visulize two-sample metrics: (output will be `all_twosample.csv` and `all_twosample.pdf`)

```
h1d multitypes ISC,CIC,SSC,deltaDLR,CD,IASC,IESC,IFC,DRF \
        ./test_data/GSE104334_Rad21KD.hic 50000 chr21 \
        -c ./test_data/GSE104334_Ctrl.hic \
        -p 300000,300000,300000,3000000,pearson,300000,300000,0.05,200000-5000000 \
        --datatype rawhic --gt ./test_data/hg19_genome_table.txt \
        -o all_twosample -d -s 24500000 -e 34500000
```

CHAPTER

SEVEN

7. PROVIDE BASIC FUNCTIONS TO VISUALIZE AND HANDLE HI-C DATA

```
$ h1d basic
usage: __main__.py basic [-h] [-o OUTNAME] [-c CONTROLMATRIX]
                         [--datatype DATATYPE] [--gt GT] [--plottype PLOTTYPE]
                         [-s START] [-e END] [--normalize NORMALIZE]
                         mode data resolution chromosome
```

- `mode`, Running mode,,should be one of {plot,dump}
- `data`, Path of matrix file or raw .hic file.
- `resolution`, resolution (50000, i.e.) of given contact matrix, or choosed resolution for analyzing `.hic` file.
- `chromosome`, selected chromosome to be analyzed.
- `--datatype`, type of input data: "matrix" (default) or "rawhic".
- `--gt`, genome table file when using raw .hic data.
- `-o`, output name, default: unname

7.1 7.1 Plot indicated region

This function provide basic visualization for Hi-C data.

- `--plottype`, Type of plot, could be one of {tri,square,tad}
- `-s`, start site for plot.
- `-e`, end site for plot.
- `-c`, File of control sample. If provided, it will plot differential matrix of treat_vs_control.

```
h1d basic plot ./test_data/GSE104334_Ctrl.chr21.matrix.gz \
        50000 chr21 --datatype matrix -o testplot --plottype square \
        -s 27500000 -e 32500000
```

If use `--plottype tri`:

Differential matrix is plotted when the control data is provided:

```
h1d basic plot ./test_data/GSE104334_Ctrl.chr21.matrix.gz \
        50000 chr21 --datatype matrix -o testplot --plottype square \
        -s 27500000 -e 32500000
```

7.2 7.2 Make contact matrix.

`--normalize`, Normalize methods {NONE/VC/VC_SQRT/KR}

```
h1d basic dump ./test_data/GSE104334_Ctrl.hic 50000 chr21 \
        --datatype rawhic -o testdump --gt ./test_data/hg19_genome_table.txt
        --normalize KR
```

The output will be:

```
testdump
└── 50000
    └── observed.KR.chr21.matrix.gz
```

which is dense matrix (zipped) of intra-chromosomal contacts, like:

7.2.1 Dump all chromosomes

- `chromosome` , set chromosome to "all" will compute metrics for all chromosomes.
- `data`, must be `.hic` data
- `-maxchr`, Maximum index of chromosome (human genome is 22,i.e.). It will compute chromosome 1~maxchr plus chromosome X.
- `-n`, Number of processors

```
h1d basic dump ./test_data/GSE104334_Ctrl.hic 50000 all \
        --gt ./test_data/hg19_genome_table.txt --normalize KR -o justtest \
        --datatype rawhic --maxchr 22 -n 30
```

Output would be:

```
justtest
└── 50000
    ├── observed.KR.chr1.matrix.gz
    ├── observed.KR.chr10.matrix.gz
    ├── observed.KR.chr11.matrix.gz
    ├── observed.KR.chr12.matrix.gz
    ├── observed.KR.chr13.matrix.gz
    ├── observed.KR.chr14.matrix.gz
    ├── observed.KR.chr15.matrix.gz
    ├── observed.KR.chr16.matrix.gz
    ├── observed.KR.chr17.matrix.gz
    ├── observed.KR.chr18.matrix.gz
    ├── observed.KR.chr19.matrix.gz
    ├── observed.KR.chr2.matrix.gz
    ├── observed.KR.chr20.matrix.gz
    ├── observed.KR.chr21.matrix.gz
    ├── observed.KR.chr22.matrix.gz
    ├── observed.KR.chr3.matrix.gz
    ├── observed.KR.chr4.matrix.gz
    ├── observed.KR.chr5.matrix.gz
    ├── observed.KR.chr6.matrix.gz
```

(continues on next page)

(continued from previous page)

```
    ├── observed.KR.chr7.matrix.gz
    ├── observed.KR.chr8.matrix.gz
    ├── observed.KR.chr9.matrix.gz
    └── observed.KR.chrX.matrix.gz
```

7.3 7.3 Make gene density file for PC1

```
h1d basic gd refFlat.hg19.txt 50000 \
                ./test_data/hg19_genome_table.txt -o hg19.geneDensity.txt
```

- `refFlat.hg19.txt` is defined by UCSC and should be at least 6 columns as (The first column geneName must not be NA):
- `50000` is the resolution for PC1 analysis.
- `./test_data/hg19_genome_table.txt` is genome table file (tab-separated) which described the length of each chromosome for your genome reference:
- `-o` is the output name

CHAPTER

EIGHT

INDICES AND TABLES

- genindex
- modindex
- search