Genomic coverage by ENCODE data. Computational processing documentation

Georgi K. Marinov[1]

[1] *Division of Biology, California Institute of Technology, Pasadena, CA, United States*

February 11, 2014

This document describes the computational steps used to estimate the genomic coverage by ENCODE data presented in:

Kellis M, Hardison RC, Wold BJ, Snyder MP, Bernstein BE, Kundaje A, Marinov GK, Ward LD, Birney E, Crawford GE, Dekker J, Dunham I, Elnitski L, Farnham PJ, Feingold EA, Gerstein M, Giddings MC, Gilbert DM, Gingeras TR, Green ED, Guigo R, Hubbard TJP, Kent WJ, Lieb JD, Myers RM, Pazin MJ, Ren B, Stamatoyannopoulos J, Weng Z, White KP, Members of the ENCODE Consortium. Defining functional DNA elements in the human genome. *Proc Natl Acad Sci U S A*, 2014.

1 General principle

For a set of input files containing genomic regions with a score associated with each, a bedGraph track is created containing the maximum score across all regions for each base pair in the genome. For example, if a genomic position $\{c, i\}$, where c is the chromosome name and i the position on the chromosomes, is covered by 5 regions $\{r_1, r_2, r_3, r_4, r_5\}$ in different datasets with scores $\{s_1, s_2, s_3, s_4, s_5\}$, then it receives a score of $max(s_1, s_2, s_3, s_4, s_5)$.

2 Creating maximum score tracks

This is done using the `MultipleBEDsToWig.py` script. The script works as follows:

```
usage: python MultipleBEDsToWig.py list_of_bed_files outputfilename [-scores]
        list_of_bed_files format: file_name      chrFieldID
        list_of_bed_files format if you use the [-scores] option:
                        file_name      chrFieldID      ScoreFieldID;
                        the maximum score will then be outputted
        use - instead of a filename if you want to print the output to stdout
```

If the `-scores` option is not used, the script will output the number of regions covering each position instead of the maximum score. In this case the option was used.

The `chrFieldID` and `ScoreFieldID` parameters refer to the column ID containing the chromosome name and the scores respectively, and are 0-based.

3 Calculating the distribution of scores

After the bedGraph file is created, the distribution of scores over the genome is calculated using the `makehistogram.py` script:

```
usage: python makehistogram.py datafilename FieldID outfilename
        [-bins size min max] [-specificbins (0),number1,number2,number3...,numberN]
```

```
[-fields ID1,ID2,..IDN] [-splitby string]
note use _s_ if you want to split by "
use - for input if you want to read from standard input
```

which by default takes a tab-delimited data file (the `-splitby` option can be used to define columns using some other character) and a column ID (again, 0-based) and returns the number of instances of each thus defined element in the file. Alternatively, unique elements can be defined over multiple columns (using the `-fields` option). The distribution of numeric values over user-specified bins can be determined with the `-bins` and `-specificbins` options. Here, the `-specificbins` option was used, as shown in the example.

4 Example

1. `python MultipleBEDsToWig.py wgEncodeCshlLongRnaSeq.CellPap.FPKM.files - -scores | bzip2 > wgEncodeCshlLongRnaSeq.CellPap.FPKM.wig.bz2`

2. `bzip2 -cd | wgEncodeCshlLongRnaSeq.CellPap.FPKM.wig.bz2 | python makehistogram.py - 3 wgEncodeCshlLongRnaSeq.CellPap.FPKM.wig.hist -specificbins 0,0.1,0.5,1,5,10,50,100,500`

Where the `wgEncodeCshlLongRnaSeq.FPKM.files` file looks like this:

```
wgEncodeCshlLongRnaSeq/wgEncodeCshlLongRnaSeqA549CellPapContigs.bedRnaElements 0 6
wgEncodeCshlLongRnaSeq/wgEncodeCshlLongRnaSeqAg04450CellPapContigs.bedRnaElements 0 6
wgEncodeCshlLongRnaSeq/wgEncodeCshlLongRnaSeqBjCellPapContigs.bedRnaElements 0 6
wgEncodeCshlLongRnaSeq/wgEncodeCshlLongRnaSeqCd20CellPapContigs.bedRnaElements 0 6
wgEncodeCshlLongRnaSeq/wgEncodeCshlLongRnaSeqGm12878CellPapContigs.bedRnaElements 0 6
wgEncodeCshlLongRnaSeq/wgEncodeCshlLongRnaSeqH1hescCellPapContigs.bedRnaElements 0 6
wgEncodeCshlLongRnaSeq/wgEncodeCshlLongRnaSeqHelas3CellPapContigs.bedRnaElements 0 6
wgEncodeCshlLongRnaSeq/wgEncodeCshlLongRnaSeqHepg2CellPapContigs.bedRnaElements 0 6
wgEncodeCshlLongRnaSeq/wgEncodeCshlLongRnaSeqHmecCellPapContigs.bedRnaElements 0 6
wgEncodeCshlLongRnaSeq/wgEncodeCshlLongRnaSeqHmecCellPapContigsV2.bedRnaElements 0 6
wgEncodeCshlLongRnaSeq/wgEncodeCshlLongRnaSeqHsmmCellPapContigs.bedRnaElements 0 6
wgEncodeCshlLongRnaSeq/wgEncodeCshlLongRnaSeqHuvecCellPapContigs.bedRnaElements 0 6
wgEncodeCshlLongRnaSeq/wgEncodeCshlLongRnaSeqImr90CellPapContigs.bedRnaElements 0 6
wgEncodeCshlLongRnaSeq/wgEncodeCshlLongRnaSeqK562CellPapContigs.bedRnaElements 0 6
wgEncodeCshlLongRnaSeq/wgEncodeCshlLongRnaSeqMcf7CellPapContigs.bedRnaElements 0 6
wgEncodeCshlLongRnaSeq/wgEncodeCshlLongRnaSeqMonocd14CellPapContigs.bedRnaElements 0 6
wgEncodeCshlLongRnaSeq/wgEncodeCshlLongRnaSeqNhekCellPapContigs.bedRnaElements 0 6
wgEncodeCshlLongRnaSeq/wgEncodeCshlLongRnaSeqNhekCellPapContigsV2.bedRnaElements 0 6
wgEncodeCshlLongRnaSeq/wgEncodeCshlLongRnaSeqNhlfCellPapContigs.bedRnaElements 0 6
wgEncodeCshlLongRnaSeq/wgEncodeCshlLongRnaSeqSknshCellPapContigs.bedRnaElements 0 6
wgEncodeCshlLongRnaSeq/wgEncodeCshlLongRnaSeqSknshraCellPapContigs.bedRnaElements 0 6
```