ChromHMM Tutorial

Jason Ernst  
Assistant Professor  
University of California, Los Angeles
Talk Outline

• Chromatin states analysis and ChromHMM
• Accessing chromatin state annotations for ENCODE2 and Roadmap Epigenomics
• Running the ChromHMM software
Talk Outline

• Chromatin states analysis and ChromHMM
• Accessing chromatin state annotations for ENCODE2 and Roadmap Epigenomics
• Running the ChromHMM software
Chromatin Marks for Genome Annotation

100+ histone modifications

Specificity in:
- Histone protein
- Amino acid residue
- Chemical modification (e.g. methyl, acetylation)
- Number of occurrence of the modifications

Examples
- H3K4me1 – Enhancers
- H3K4me3 – Promoters
- H3K27me3 – Repressive
- H3K9me3 – Repressive
- H3K36me3 – Transcribed

Histone Modifications can be Mapped Genome-wide with ChIP-seq

Image source: http://nihroadmap.nih.gov/epigenomics/
From ‘chromatin marks’ to ‘chromatin states’

- Learn de novo significant promoter states and transcribed states, combining and spatial patterns of chromatin marks.
- Reveal functional elements, even without looking at sequence.
- Use for genome annotation.

Ernst and Kellis, Nat Biotech 2010
Our approach: Multivariate Hidden Markov Model (ChromHMM)

**Enhancer**
- Binarized chromatin marks. Called based on a poisson distribution
- Unobserved

**Gene Starts**
- Most likely Hidden State

**Gene - Transcribed Region**
- 200 base pair interval

**DNA**

**High Probability Chromatin Marks in State**

1:
- 0.8 H3K4me1
- 0.8 K27ac

2:
- 0.9 H3K4me3
- 0.8 K4me1

3:
- 0.9 H3K4me3

4:
- 0.7 H3K4me1

5:
- 0.9 H3K36me3

6:
- 0.9 H3K36me3

Emission distribution is a product of independent Bernoulli random variables

Binarization leads to explicit modeling of mark combinations and interpretable parameters

ENCODE: Study nine marks in nine human cell lines

<table>
<thead>
<tr>
<th>9 marks</th>
<th>9 human cell types</th>
</tr>
</thead>
<tbody>
<tr>
<td>H3K4me1</td>
<td>HUVEC</td>
</tr>
<tr>
<td>H3K4me2</td>
<td>NHEK</td>
</tr>
<tr>
<td>H3K4me3</td>
<td>GM12878</td>
</tr>
<tr>
<td>H3K27ac</td>
<td>K562</td>
</tr>
<tr>
<td>H3K9ac</td>
<td>HepG2</td>
</tr>
<tr>
<td>H3K27me3</td>
<td>NHLF</td>
</tr>
<tr>
<td>H4K20me1</td>
<td>HMEC</td>
</tr>
<tr>
<td>H3K36me3</td>
<td>HSMM</td>
</tr>
<tr>
<td>CTCF</td>
<td>H1</td>
</tr>
</tbody>
</table>

Brad Bernstein ENCODE Group

<table>
<thead>
<tr>
<th>HUVEC</th>
<th>NHEK</th>
<th>GM12878</th>
<th>K562</th>
<th>HepG2</th>
<th>NHLF</th>
<th>HMEC</th>
<th>HSMM</th>
<th>H1</th>
</tr>
</thead>
<tbody>
<tr>
<td>State</td>
<td>CTCF</td>
<td>H3K27me3</td>
<td>H3K36me3</td>
<td>H4K20me1</td>
<td>H3K4me1</td>
<td>H3K4me2</td>
<td>H3K4me3</td>
<td>H3K27ac</td>
</tr>
<tr>
<td>1</td>
<td>16</td>
<td>2</td>
<td>2</td>
<td>6</td>
<td>17</td>
<td>93</td>
<td>99</td>
<td>96</td>
</tr>
<tr>
<td>2</td>
<td>12</td>
<td>2</td>
<td>6</td>
<td>9</td>
<td>53</td>
<td>94</td>
<td>95</td>
<td>14</td>
</tr>
<tr>
<td>3</td>
<td>13</td>
<td>72</td>
<td>0</td>
<td>9</td>
<td>48</td>
<td>78</td>
<td>49</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>11</td>
<td>1</td>
<td>15</td>
<td>11</td>
<td>96</td>
<td>99</td>
<td>75</td>
<td>97</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>10</td>
<td>3</td>
<td>13</td>
<td>88</td>
<td>57</td>
<td>5</td>
<td>84</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>1</td>
<td>13</td>
<td>38</td>
<td>38</td>
<td>75</td>
<td>8</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>56</td>
<td>3</td>
<td>6</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>92</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>6</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>5</td>
<td>1</td>
<td>43</td>
<td>43</td>
<td>37</td>
<td>11</td>
<td>2</td>
<td>9</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>47</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>27</td>
<td>2</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>22</td>
<td>28</td>
<td>19</td>
<td>41</td>
<td>6</td>
<td>5</td>
<td>26</td>
<td>5</td>
</tr>
<tr>
<td>15</td>
<td>85</td>
<td>85</td>
<td>91</td>
<td>88</td>
<td>76</td>
<td>77</td>
<td>91</td>
<td>73</td>
</tr>
</tbody>
</table>

Chromatin Mark Observation Frequency (%):  
- Active Promoter  
- Weak Promoter  
- Inactive/poised Promoter  
- Strong enhancer  
- Strong enhancer  
- Weak/poised enhancer  
- Weak/poised enhancer  
- Insulator  
- Transcriptional transition  
- Transcriptional elongation  
- Weak transcribed  
- Polycomb-repressed  
- Heterochrom; low signal  
- Repetitive/CNV  
- Repetitive/CNV  

Ernst et al, Nature 2011  
Learned jointly across cell types (virtual concatenation)  
State definitions are common  
State locations are dynamic
Chromatin states dynamics across nine ENCODE cell types

- Single annotation track for each cell type
- Summarize cell-type activity at a glance
- Can study 9-cell activity pattern across

Ernst et al, Nature 2011
Talk Outline

• Chromatin states analysis and ChromHMM
• Accessing chromatin state annotations for ENCODE2 and Roadmap Epigenomics
• Running the ChromHMM software
Chromatin States Defined Across 127 Cell/Tissues Types

16 epigenomes from ENCODE 2
Chromatin States Defined on Imputed Data

ChromImpute method
Ernst and Kellis, *Nature Biotech* 2015
ChromHMM Models across Many Roadmap/ENCODE Cell and Tissue Types

127 Cell/Tissue Types

<table>
<thead>
<tr>
<th>H3K4me1</th>
<th>H3K4me1</th>
</tr>
</thead>
<tbody>
<tr>
<td>H3K4me3</td>
<td>H3K4me3</td>
</tr>
<tr>
<td>H3K27me3</td>
<td>H3K27me3</td>
</tr>
<tr>
<td>H3K9me3</td>
<td>H3K9me3</td>
</tr>
<tr>
<td>H3K36me3</td>
<td>H3K36me3</td>
</tr>
<tr>
<td><strong>H3K27ac</strong></td>
<td></td>
</tr>
</tbody>
</table>

98 Cell/Tissue Types

<table>
<thead>
<tr>
<th>H3K4me1</th>
<th>H3K4me1</th>
</tr>
</thead>
<tbody>
<tr>
<td>H3K4me3</td>
<td>H3K4me3</td>
</tr>
<tr>
<td>H3K27me3</td>
<td>H3K27me3</td>
</tr>
<tr>
<td>H3K9me3</td>
<td>H3K9me3</td>
</tr>
<tr>
<td>H3K36me3</td>
<td>H3K36me3</td>
</tr>
<tr>
<td>H3K27ac</td>
<td></td>
</tr>
<tr>
<td>H2A.Z</td>
<td></td>
</tr>
<tr>
<td>DNase</td>
<td></td>
</tr>
</tbody>
</table>

127 Cell/Tissue Types

<table>
<thead>
<tr>
<th>H3K4me1</th>
<th>H3K4me1</th>
</tr>
</thead>
<tbody>
<tr>
<td>H3K4me3</td>
<td>H3K4me3</td>
</tr>
<tr>
<td>H3K27me3</td>
<td>H3K27me3</td>
</tr>
<tr>
<td>H3K9me3</td>
<td>H3K9me3</td>
</tr>
<tr>
<td>H3K36me3</td>
<td>H3K36me3</td>
</tr>
<tr>
<td>H3K27ac</td>
<td></td>
</tr>
<tr>
<td>H3K9ac</td>
<td></td>
</tr>
<tr>
<td>H4K20me1</td>
<td></td>
</tr>
<tr>
<td>H3K79me2</td>
<td></td>
</tr>
<tr>
<td>H3K4me2</td>
<td></td>
</tr>
<tr>
<td>H2A.Z</td>
<td></td>
</tr>
<tr>
<td>DNase</td>
<td></td>
</tr>
</tbody>
</table>
Roadmap Epigenomics Integrative Analysis Portal

http://compbio.mit.edu/roadmap
Roadmap Epigenomics Integrative Analysis Portal

http://compbio.mit.edu/roadmap

Chromatin state learning

In order to capture the significant combinatorial interactions between different chromatin marks in their spatial context (chromatin states) across 127 epigenomes, we used ChromHMM v1.10 (Ernst et al., 2012), which is based on a multivariate Hidden Markov Model.

Core 15-state model (5 marks, 127 epigenomes)

DATA SOURCE

- Download URL:
  http://egg2.wustl.edu/roadmap/data/bty256p15/chromhmmSegmentations/ChmmModels/coreMarks/jointModel/final

- Open in a new page (deactivate pop-up blockers)

- Summarized visualization of all 127 epigenomes using epilogs
- Emission, transition probabilities and enrichment of states relative to various genomic and functional annotations
- MNEMONICS BED FILES (Epigenome_id_15_coreMarks_mnemonics.bed.gz files)
  - Tab delimited 4 columns
  - chromosome, start (0-based), stop (1-based), state_label_mnemonic for that region
- ARCHIVE of all mnemonics.bed files
- BROWSER FRIENDLY FILES (Epigenome_id_15_coreMarks_dense.bed)
  - The dense BIGBED files will allow you to view each epigenome as a single track with regions labeled with state mnemonics and representative colors. You can stream these to UCSC Genome Browser or IGV
  - ARCHIVE of all the dense BIGBED files
  - Epigenome_id_15_coreMarks_dense.bed.gz (Same as above except in text format)
  - ARCHIVE of all dense BED files
  - Epigenome_id_15.coreMarks.expanded.bed.gz files: The expanded files will allow you to view each epigenome with each state as a separate track labeled with state mnemonics and representative colors
  - ARCHIVE of expanded BED files
- STATES FOR EACH 200bp BIN:
  - Max. posterior state label for each 200 bp bins in each chromosome for all epigenomes. The difference from the Mnemonic RFID files is that in the Mnemonic files continuous bins with the same state label are merged and a label is assigned to the
Accessing Roadmap ChromHMM through the UCSC Genome Browser Track Hubs

http://genome.ucsc.edu
Accessing Roadmap ChromHMM through the UCSC Genome Browser Track Hubs

http://genome.ucsc.edu
Accessing Roadmap ChromHMM through the UCSC Genome Browser Track Hubs

Note: Different than track hub

Roadmap Epigenomics Data Complete Collection at Wash U VizHub
Accessing Roadmap ChromHMM through the UCSC Genome Browser Track Hubs
Accessing Roadmap ChromHMM through the UCSC Genome Browser Track Hubs
Accessing Roadmap ChromHMM through the UCSC Genome Browser Track Hubs
Accessing Roadmap ChromHMM through the UCSC Genome Browser Track Hubs
Human Epigenome Browser at Washington University

http://epigenomemegateway.wustl.edu/
Talk Outline

• Chromatin states analysis and ChromHMM
• Accessing chromatin state annotations for ENCODE2 and Roadmap Epigenomics
• Running the ChromHMM software
ChromHMM: Chromatin state discovery and characterization

ChromHMM is software for learning and characterizing chromatin states. ChromHMM can integrate multiple chromatin datasets such as ChIP-seq data of various histone modifications to discover de novo the major re-occurring combinatorial and spatial patterns of marks. ChromHMM is based on a multivariate Hidden Markov Model that explicitly models the presence or absence of each chromatin mark. The resulting model can then be used to systematically annotate a genome in one or more cell types. By automatically computing state enrichments for large-scale functional and annotation datasets ChromHMM facilitates the biological characterization of each state. ChromHMM also produces files with genome-wide maps of chromatin state annotations that can be directly visualized in a genome browser.

ChromHMM software v1.11 (version Jun)

Quick instructions on running ChromHMM:
1. Install Java 1.5 or later if not already installed.
2. Unzip the file ChromHMM.zip
3. To try out ChromHMM learning a 10-state model on the sample data enter from a command line in the directory with the ChromHMM.jar file the command:
   java -mx1600M -jar ChromHMM.jar LearnModel SAMPLEDATA_HG18 OUTPUTSAMPLE 10 hg18

After termination in ~5-10 minutes a file in OUTPUTSAMPLE/webpage_10.html will be created showing output images and linking to all the output files created. If a web browser is found on the computer the webpage will automatically be opened in it.

In general binarized input for the LearnModel command can be generated by first running the BinarizeBed command on bed files with coordinates of aligned reads or the BinarizeBam command on bam files with the coordinates of aligned reads.

ChromHMM has a BinarizeBam command which allows binarizing bam files of aligned reads.

ChromHMM has the option for parallel training with multithreads leading to significantly reduced training times. Add the `-p 0` option to the LearnModel command to have ChromHMM to try to use as many processors as available or specify the maximum it should use.

The ChromHMM software is described in:

Contact Jason Ernst (jason.ernst@ucla.edu) with any questions, comments, or bug reports.
Subscribe to a mailing list for announcements of new versions
ChromHMM is released under a GPL 3 license
ChromHMM source code is available on GitHub
Funding for ChromHMM provided by NSF Postdoctoral Fellowship 0905968 to JE and grants from the National Institutes of Health (NIH 1-R21-HG005334 and NIH 1 U54 HG004570).
ChromHMM: Chromatin state discovery and characterization

ChromHMM is software for learning and characterizing chromatin states. ChromHMM can integrate multiple chromatin datasets such as ChIP-seq data of various histone modifications to discover de novo the major re-occurring combinatorial and spatial patterns of marks. ChromHMM is based on a multivariate Hidden Markov Model that explicitly models the presence or absence of each chromatin mark. The resulting model can then be used to systematically annotate a genome in one or more cell types. By automatically computing state enrichments for large-scale functional and annotation datasets ChromHMM facilitates the biological characterization of each state. ChromHMM also produces files with genome-wide maps of chromatin state annotations that can be directly visualized in a genome browser.

- ChromHMM software documentation
- ChromHMM manual

Quick instructions on running ChromHMM:
1. Install Java 1.5 or later if not already installed.
2. Unzip the file ChromHMM.zip
3. To try out ChromHMM learning a 10-state model on the sample data enter from a command line in the directory with the ChromHMM.jar file the command:

   java -mx1600M -jar ChromHMM.jar LearnModel SAMPLEDATA_HG18 OUTPUTSAMPLE 10 hg18

After termination in ~5-10 minutes a file in OUTPUTSAMPLE/webpage_10.html will be created showing output images and linking to all the output files created. If a web browser is found on the computer the webpage will automatically be opened in it.

In general, binary input for the LearnModel command can be generated by first running the BinarizeBed command on bed files with coordinates of aligned reads or the BinarizeBam command on bam files with the coordinates of aligned reads.

New in version 1.11: ChromHMM has a BinarizeBam command which allows binarizing bam files of aligned reads.

New in version 1.10: ChromHMM has the option for parallel training with multithreaded processors leading to significantly reduced training times. Add the `-p 0` option to the LearnModel command to have ChromHMM try to use as many processors as available or specify the maximum it should use.

- The ChromHMM software is described in:
  Here are links to some existing ChromHMM annotations in hg19 available for 127 Reference Epigenomes (Roadmap Epigenomics), 0-ENCODE cell types (from Ernst et al, Nature 2011), and 6-ENCODE cell types (from ENCODE Integrative Analysis).

- Contact Jason Ernst (Jason.ernst at ucla dot edu) with any questions, comments, or bug reports.
- Subscribe to a mailing list for announcements of new versions
- ChromHMM is released under a GPL 3 license.
- ChromHMM source code is available on GitHub here.
- Funding for ChromHMM provided by NSF Postdoctoral Fellowship 0905968 to JE and grants from the National Institutes of Health (NIH 1-RC1-HG005334 and NIH 1 U54 HG004570).
Try to Run ChromHMM on Sample Data on Your Computer

(Java needs to already be installed)

1. Download
   http://compbio.mit.edu/ChromHMM/ChromHMM.zip

2. Unzip ChromHMM.zip

3. Open a command line

4. Change into the ChromHMM directory

5. Enter the command:

   java -mx1600M -jar ChromHMM.jar LearnModel -p 0 SAMPLEDATA_HG18 OUTPUTSAMPLE 10 hg18
Input to ChromHMM

• ChromHMM models are learned from binarized data using its LearnModel command

• Binarized data is typically obtained starting from aligned reads.
  – Apply BinarizeBed if reads are in BED format
  – Apply BinarizeBam if reads are in BAM format
BinarizeBed

Java command ‘-mx1600M’ specifies memory to Java
BinarizeBed

File with the chromosome lengths for the assembly
BinarizeBed

DIRECTORY of BED files
BinarizeBed

Cell-mark –file table

Control data – is optional and can also be treated as a mark
BinarizeBed

```
java -mx1G -jar ChromHMM.jar BinarizeBed CHROMSIZES\hg18.txt INPUTBEDFILES cllmarkfile.txt SAMPLEDATA_HG18
```
LearnModel

'\-p 0' Use as many processors as available
'\-p N' Use up to N processors (default N=1)
LearnModel

Directory where the output goes
LearnModel

```bash
java -mx1600M -jar ChromHMM.jar LearnModel -p 0 SAMPLEDATA_HG18 OUTPUTSAMPLE 10 hg18
```
LearnModel

Genome assembly
ChromHMM Report

Model Parameters

Emission Parameters

- Emission Parameter SVG File
- Emission Parameter Tab-Delimited Text File

Transition Parameters
ChromHMM Report

Model Parameters

Emission Parameters

Transition Parameters

Input Directory: SAMPLEDATA_HG18
Output Directory: OUTPUTSAMPLE
Number of States: 10
Assembly: hg18
Full ChromHMM command: LearnModel -p 0 SAMPLEDATA_HG18 OUTPUTSAMPLE 10 hg18
Emission Parameters
Transition Parameters

ChromHMM Report

Model Parameters

Emission Parameters

Mark

State From (Emission order)

State To (Emission order)

Transition Parameters

State From (Emission order)

State To (Emission order)

- Transition Parameter SVG File
- Transition Parameter Tab-Delimited Text File
Model Parameter File

- Transition Parameter SVG File
- Transition Parameter Tab-Delimited Text File
- All Model Parameters Tab-Delimited Text File

Genome Segmentation Files
- GM12878_10 Segmentation File (Four Column Bed File)
- K562_10 Segmentation File (Four Column Bed File)

Custom Tracks for loading into the UCSC Genome Browser:
- GM12878_10 Browser Custom Track Dense File
- GM12878_10 Browser Custom Track Expanded File
- K562_10 Browser Custom Track Dense File
- K562_10 Browser Custom Track Expanded File

State Enrichments
GM12878_10 Enrichments

Fold Enrichment GM12878_10

Category

- GM12878_10 Overlap Enrichment SVG File
- GM12878_10 Overlap Enrichment Tab-Delimited Text File
Segmentation File

Genome Segmentation Files

- GM12878_10 Segmentation File (Four Column Bed File)
- K562_10 Segmentation File (Four Column Bed File)
- Custom Tracks for loading into the UCSC Genome Browser
- GM12878_10 Browser Custom Track Dense File
- GM12878_10 Browser Custom Track Expanded File
- K562_10 Browser Custom Track Dense File
- K562_10 Browser Custom Track Expanded File

State Enrichments

GM12878_10 Enrichments

Fold Enrichment GM12878_10

- GM12878_10 Overlap Enrichment SVG File
- GM12878_10 Overlap Enrichment Tab-Delimited Text File
Browser Files

Genome Segmentation Files
- GM12878_10 Segmentation File (Four Column Bed File)
- K562_10 Segmentation File (Four Column Bed File)
- Custom Tracks for loading into the UCSC Genome Browser
- GM12878_10 Browser Custom Track Dense File
- GM12878_10 Browser Custom Track Expanded File
- K562_10 Browser Custom Track Dense File
- K562_10 Browser Custom Track Expanded File

State Enrichments
GM12878_10 Enrichments

Fold Enrichment GM12878_10

Can load into browser UCSC Genome, IGV

https://www.broadinstitute.org/igv/
Enrichments

State Enrichments

GM12878_10 Enrichments

Fold Enrichment GM12878_10

Category

Fold Enrichment GM12878_10 RefSeqTES

- GM12878_10 Overlap Enrichment SVG File
- GM12878_10 Overlap Enrichment Tab-Delimited Text File

Fold Enrichment GM12878_10

Category

- GM12878_10 Overlap Enrichment SVG File
- GM12878_10 Overlap Enrichment Tab-Delimited Text File
Positional Plots

- GM12878_10_RefSeqTES_neighborhood_Enrichment_SVG_File
- GM12878_10_RefSeqTES_neighborhood_Enrichment_Tab-Delimited_Text_File

- GM12878_10_RefSeqTSS_neighborhood_Enrichment_SVG_File
- GM12878_10_RefSeqTSS_neighborhood_Enrichment_Tab-Delimited_Text_File
Enrichments for Additional Cell Types

- GM1278_10_RefSeqTES_neighborhood_Enrichment_SVG_Files
- GM1278_10_RefSeqTES_neighborhood_Enrichment_Tab-Delimited_Text_Files

K562_10 Enrichments

Fold Enrichment K562_10

Category

- K562_10_Overlap_Enrichment_SVG_Files
- K562_10_Overlap_Enrichment_Tab-Delimited_Text_Files

Fold Enrichment K562_10 RefSeqTES
Chromatin states to interpret disease variants

- Specific chromatin states enriched in GWAS catalog
  Ernst and Kellis, *Nature Biotech* 2010

- Enhancers from different cell types enriched in different traits
  Claussnitzer et al, *NEJM* 2015

- Imputation based chromatin state used in dissection FTO loci
  Claussnitzer et al, *NEJM* 2015

- Interpreting epigenetic disease associated variation in Alzheimer’s disease

- Many other examples in the literature
Collaborators and Acknowledgements

• Manolis Kellis

ENCODEx consortium
  – Brad Bernstein production group

Roadmap Epigenomics consortium

Funding
• NHGRI, NIH, NSF, HHMI, Sloan Foundation
Additional Commands

- CompareModels – the command allows the comparison of the emission parameters of a selected model to a set of models in terms of correlation.
Additional Commands

- **MakeBrowserFiles** – (re)generates browser files from segmentation files and allows specifying the coloring

  MakeBrowserFiles [-c colormappingfile][ -m labelmappingfile][ -n numstates] segmentfile segmentationname outputfileprefix
Additional Commands

• **OverlapEnrichment** – (re)computes enrichments of a segmentation for a set of annotations

OverlapEnrichment [-a cell][-b binsize][-binres][-color r,g,b][-center] [-colfields chromosome,start,end[,signal]][-e offsetend][-f coordlistfile][-m labelmappingfile][-multicount][-posterior][-s offsetstart][-signal][-t title][-uniformscale] inputsegment inputcoordddir outfileprefix

Fold Enrichment GM12878_10
Additional Commands

- NeighborhoodEnrichment – (re)computes enrichments of a segmentation around a set of anchor positions

usage NeighborhoodEnrichment [-a cell][-b binsize][-color r,g,b]
[-colfields chromosome,position[,optionalcol1[,optionalcol2,...]]]
[-l numleftintervals][-m labelmappingfile][-nostrand][-o anchoroffset]
[-posterior][-r numrightintervals][-s spacing][-signal][-t title]
inputsegment anchorpositions outfilename
Additional Commands

• Reorder – reorders the states of the model

usage: Reorder [-color r,g,b] [-f columnorderingfile] [-holdcolumnorder] [-i outfileID] [-m labelmappingfile] [-o stateorderingfile] [-stateordering emission|transition] inputmodel outputdir