CHANCE-HT user guide

Aaron Diaz DiazA2@humgen.ucsf.edu

CHANCE-HT is a software for quality filtering and normalizing large ensembles of ChIP-seq data for use in comparative analyses. This document is a brief guide to using the software and interpreting its results. If you find this software useful please cite A. Diaz, “CHANCE-HT: massively parallel ChIP-seq normalization and quality control”. For the theory behind the statistical tests used by CHANCE, see Diaz et al. Statistical Applications in Genetics and Molecular Biology.11(3) March 2012 (http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3342857/). For software downloads and the sample data referred to in this guide see: https://sourceforge.net/projects/chanceht/ and for source code, wiki, bugs or requests see https://github.com/diazlab/chance/

Table of contents

Installation

CHANCE-HT runs under most 64bit Mac OSX, Windows, and Linux distributions. Start by downloading the appropriate installation package from: https://sourceforge.net/projects/chanceht/

CHANCE-HT is released under the GNU General Public License: http://www.gnu.org/licenses/. The CHANCE-HT source code can be obtained from https://github.com/diazlab/chance.

If you are running Mac OSX

1. Decompress the CHANCE-HT archive
    1. Unzip the file `CHANCE_MacOS.zip` by double clicking the `CHANCE_MacOS.zip` icon.
    2. Open the folder `CHANCE_MacOS/`.
2. Install MCR, the MATLAB Compiler Runtime:
    1. Unzip `MCRInstaller.zip` by double clicking its icon
    2. Double click `InstallForMacOSX`
    3. Follow the on screen instructions, but keep track of the install location if you change the default.
3. To start CHANCE-HT:
    1. Navigate to the `CHANCE_MacOS` folder
    2. Execute `./run_chance.sh path_to_mcr`, where `path_to_mcr` is the path to the MCR you

installed. The default path is `/Applications/MATLAB/MATLAB_Compiler_Runtime/v717/`

If you are running 64bit Linux:

1. Navigate to where you downloaded `CHANCE_Linux.zip`
2. Decompress the CHANCE-HT archive `unzip CHANCE_Linux.zip`
   `cd chance_linux`
3. Install MCR, the MATLAB Compiler Runtime: `unzip MCRInstaller.zip`
   `sudo ./install`
   Follow the on screen instructions, keep track of the install location if you change the default
4. To start CHANCE: `./run_chance.sh path_to_mcr` where `path_to_mcr` is the path to the MCR you installed,the default is `/usr/local/MATLAB/MATLAB_Compiler_Runtime/v717/`

If you are running 64bit Windows

1. To start CHANCE: type `chance.exe` at the DOS prompt.

Usage summary

```
run_chance.sh /PATH/TO/MCR batch -p parameter_file -o output_file (-b on/off)
```

Notes: 1. parameter_file and output_file should be the full paths to the respective files. 2. The -b parameter is optional, it toggles batch effects detection, the default is "on". 3. There is also an auxillary sub-routine to bin reads which can be accessed directly. The invocation is : `run_chance.sh /PATH/TO/MCR batch binData -p parameter_file`

This can be useful if you plan to run CHANCE multiple times on the same files, since the binned reads take much less memory and time to read. Also, the binary format (.mat) that this routine produces can be read by the GUI version of CHANCE. The parameter file format for this subroutine is: `alignments_file_name,output_file_name,sample_id,build,file_type`

Manifest file format

CHANCE-HT uses a parameter file manifest to specify which files to process and which IP files should be mated to which Input files. The manifest file contains comma separated values with no header. Place one line per file to process. Each line must have the following format:

```
IP_file_name,Input_file_name,IP_sample_ID,Input_sample_ID,Build,File_type
```

- `IP_file_name` : the full path to the IP sample
- `Input_file_name` : the full path to the Input sample
- `IP_sample_ID` : any string, will identify the IP sample in the CHANCE output file
- `Input_sample_ID` : any string, will identify the Input sample in the CHANCE output file
- `Build` : hg18, hg19, mm9 or tair10
- `File_type` : a string indicating the type of file storing the alignments, one of: "bam", "sam", "bowtie", "bed", "tagAlign", or "mat". BAM file format is fastest aside from MAT, which is the matlab format for samples saved from a CHANCE gui session or a call to binData

Output file format

CHANCE-HT outputs three files: `output_file`, `output_file.msg` and `output_file-dendrogram.tsv`. output_file is a tab separated values file with the following fields:

- `IP` : the ID string of the IP sample
- `Input` : the ID string of the Input sample
- `test` : sample classification string, one of:
  1. PASS - the sample shows significant signal at an FDR <= 5%
  2. WEAK - the sample shows significant signal at 5% < FDR <=10%
  3. FAIL - the sample does NOT show significant signal, FDR > 10%
- `p-value` : the p-value for the divergence test used to classify sample
- `FDR` : this is a q-value (positive FDR) defined as the minimum q-value (Fisher's method) computed over any of the 5 subsets of ENCODE training data (described below).
- `IP_strength` : percentage enrichment of IP over Input, as a fraction from 0 to 1 This and Percent_genome_enriched are essentially the test statistics which are implicitly used to determine the p-value and subsequently the FDR.
- `Percent_genome_enriched` : percentage of the genome differentially enriched
- `FDR_cancer_tfbs` : FDR for transcription factor binding site ChIPs in cancer cells
- `FDR_cancer_histone`: FDR for epigenetic mark ChIPs in cancer cells
- `FDR_normal_tfbs` : FDR for transcription factor binding site ChIPs in normal cells
- `FDR_normal_histone` : FDR for epigenetic mark ChIPs in normal cells
- `FDR_comb` : FDR for the combined dataset
- `Input_bias` : a test statistic measuring bias in the sample, when this metric is high the ChIP may have worked but the sample will have low statistical power due to bias in the library preparation. This tests for bi-modality in the frequency- response of the Input channel read-density.

Bi-modality indicates systematic bias in read-density which can be introduced for example by some chromatin sonnication methods.

- `Input_bias_pvalue` : a p-value for the Input_bias test statistic, when this is low there is a significant bias in the Input channel
- `ip_scaling_factor` : normalization factor for the IP sample, scale IP read-counts by this amount
- `input_scaling_factor` : normalization factor for the Input sample, scale Input read-counts by this amount
- `batch` : An integer from 1 to the number of samples. If the same index is repeated then there may be batch-effects and the samples with the same index may have come from the same batch. This routine only checks for batch-effects in the Input channel.

output_file.msg is a descriptive file with one entry per sample. Each entry contains error messages and warnings regarding the sample, scaling factors, and other descriptive metrics such as indicators of zero-inflation in the IP or Input channels or very high duplication levels indicating possible PCR bias. If you see a failed sample in the TSV file you might want to check that sample's entry in this file for more information.

Analysis of the sample dendrogram

The file output_file-dendrogram.tsv contains the dendrogram of samples, represented in matrix form Z. This matrix encodes the dendrogram in a form which can be read by MATLAB or parsed by a user's own code. From the MATLAB documentation:

Z is a (m – 1)-by-3 matrix, where m is the number of observations in the original data. Columns 1 and 2 of Z contain cluster indices linked in pairs to form a binary tree. The leaf nodes are numbered from 1 to m. Leaf nodes are the singleton clusters from which all higher clusters are built. Each newly-formed cluster, corresponding to row Z(I,:), is assigned the index m+I. Z(I,1:2) contains the indices of the two component clusters that form cluster m+I. There are m-1 higher clusters which correspond to the interior nodes of the clustering tree. Z(I,3) contains the linkage distances between the two clusters merged in row Z(I,:). For example, suppose there are 30 initial nodes and at step 12 cluster 5 and cluster 7 are combined. Suppose their distance at that time is 1.5. Then Z(12,:) will be [5, 7, 1.5]. The newly formed cluster will have index 12 + 30 = 42. If cluster 42 appears in a later row, it means the cluster created at step 12 is being combined into some larger cluster.