PanChIP manual v.3.0.10

Hanjun Lee
hanjun@broadinstitute.org

June 20, 2022

Contents

1. Getting started.

PanChIP is a pan-ChIP-seq analysis algorithm for peak sets. The primary goal of the software is to accurately describe the protein binding characteristics of a given peak set by comparing the input dataset to the library of thousands of ChIP-seq experiments that are publicly available. The PanChIP source code can be downloaded from our GitHub repository. The current version of PanChIP supports the hg38 genome assembly.

1.1. Installation.

PanChIP is a python- and bash-based software and has been tested in UNIX-based systems. The installation of the software is done through the Python Package Index (PyPI).

Ubuntu.

```
$ sudo apt-get update
$ sudo apt-get install python3 python3-pip
$ pip3 install panchip
```

Red Hat, CentOS, Fedora.

```
$ sudo yum update
$ sudo yum install python3 python3-pip
$ pip3 install panchip
```

1.1.1. Troubleshooting.

PanChIP is designed for python 3 and is incompatible with python 2. Installation of PanChIP through the python 2 version of pip would thus generate compilation errors arising from the difference in the grammar between the two versions of python.

If the user accidentally installed PanChIP through the python 2 version of pip, the user can resolve the issue by uninstalling the python2-pip and python3-pip versions of PanChIP and reinstalling the python3-pip version of PanChIP.

UNIX.

```
$ pip2 uninstall panchip & pip3 uninstall panchip
$ pip3 install panchip
```

PanChIP has the following dependencies: argparse, gdown, pandas, scipy, bedtools.

If each python package dependency is not satisfied, python3-pip would automatically install the dependency during the installation of PanChIP. However, bedtools is written in C++ and should be preinstalled. If the user encounters dependency errors while compiling PanChIP, the user can download the dependencies individually.

Ubuntu.

```
$ sudo apt-get install bedtools
$ pip3 install argparse gdown pandas scipy
$ pip3 install panchip --upgrade
```

Red Hat, CentOS, Fedora.

```
$ sudo yum install bedtools
$ pip3 install argparse gdown pandas scipy
$ pip3 install panchip --upgrade
```

We were notified that some users have encountered errors while installing gdown. The reported cases were mostly due to the failure of python3-pip to recognize a working distribution of pyparsing. This issue can be resolved via installing pyparsing as superuser.

UNIX.

```
$ sudo pip3 install pyparsing
$ pip3 install panchip
```

While we have tested PanChIP in multiple UNIX distributions and environments, we understand that bioinformatics software often generates glitches in an unexpected manner. Active issues of PanChIP can be discussed in our GitHub issues page.

1.2. Initialization.

If the user successfully installed PanChIP to their system, the next step is to initialize the software. The initialization process can be automated via **panchip init** and will prepare the PanChIP library and the bash source codes in a target directory. This directory is designated as the **library directory** of PanChIP and will be crucial for subsequent analyses.

UNIX.

```
$ panchip init [-h] library_directory
```

The user can view the content of the **library directory** from the terminal. The total size of the **library directory** is 13.6 GB.

UNIX.

```
$ ls library_directory
    PanChIP-v.3.0.10    v.3.0
$ ls library_directory/v.3.0
    Analysis    Filter
```

1.2.1. Troubleshooting.

The **library directory** of PanChIP is version-specific and should be updated after every upgrade of the PanChIP software.

UNIX.

```
$ pip3 install panchip --upgrade
$ rm -r library_directory
$ panchip init library_directory
```

2. Bulk analysis of peak sets.

The primary function of PanChIP is to identify the list of proteins that occupy genomic loci described by the given peak set. This function of PanChIP is done through **panchip analysis**. The primary output of **panchip analysis** is the **normalized overlap** between the bulk of peak sets from the PanChIP library and the given peak set.

Normalized overlap is mathematically defined as follows:

$$normalized\ overlap\ (reference, query) = \sqrt{\frac{\{\sum_{peak\ \in\ (reference\ \cap\ query)} bin\ size_{reference}\ *\ weight_{reference}\}\cdot\{\sum_{peak\ \in\ (reference\ \cap\ query)} bin\ size_{query}\ *\ weight_{query}\}}{\{\sum_{peak\ \in\ reference} bin\ size_{reference}\ *\ weight_{reference}\}\cdot\{\sum_{peak\ \in\ query} bin\ size_{query}\ *\ weight_{query}\}}}\ .$$

The weighting formula for the calculation of the **normalized overlap** is either the quality score indicated by the fifth column in the standard BED6 format (query) or the likelihood of a given region showing up as a peak in a ChIP-seq experiment for the DNA-bound protein (reference). The mathematical representation of the weighting formula for the reference peak set is as follows:

$$weight_{reference} = P(peak \in experiment \mid peak \in \cup_{reference}\ experiment))\ .$$

The **panchip analysis** command computes the **normalized overlap** through the following three steps: **1) preliminary normalization via data permutation**, **2) overlap calculation**, and **3) normalization of the overlap**.

1) preliminary normalization.

To ensure that the denominator do not significantly vary library-wide (reference) and input-wide (query), a preliminary normalization is performed by permutating the peak set with target values.

$$target\left(\sum_{peak\ \in\ reference} bin\ size_{reference}\ *\ weight_{reference}\right) = \max\left(reference \in library \mid \sum_{peak\ \in\ reference} bin\ size_{reference}\ *\ weight_{reference}\right),$$

$$target\left(\sum_{peak\in query} bin\ size_{query} * weight_{query}\right) = \min\left(query \in input \mid \sum_{peak\in query} bin\ size_{query} * weight_{query}\right).$$

2) overlap calculation.

The selection of overlapping peaks is performed by **bedtools intersect**.

3) normalization of the overlap.

While we had implemented a preliminary normalization step that permutates the library and the input until they reach a target value by approximation, there still exists a slight variation in the exact values. In order to remove the bias created by this variation, the final value for the normalized overlap is acquired by dividing the calculated overlap by the reference- and query-specific denominator.

UNIX.

$ panchip analysis [-h] [-t threads] [-r repeats] **library_directory input_directory output_directory**

2.1. Positional arguments.

The **panchip analysis** command requires three positional arguments: 1) the path to the **library directory**, 2) the path to the **input directory**, and 3) the path to the **output directory**.

1) library_directory.

The **library_directory** is the path to the **library directory** designated by **panchip init**. The user can compare the version of the software and the library from the terminal.

UNIX.

$ panchip --version
PanChIP 3.0.10
$ ls **library_directory** | grep PanChIP-v.*.*.*
PanChIP-v.3.0.10

2) input_directory.

The **input_directory** is the path to the **input directory** that contains the BED files for the input peak sets. The **input directory** should NOT contain files that are not part of the input peak sets. The input BED files should be in a standard tab-delimited BED6 format, with fifth column being the quality score from 0 to 1,000. The fifth column should NOT be a non-numerical value. If the user is to run the analysis using a peak set that does not contain a quality score, we recommend setting the fifth column values as 1,000.

UNIX.

```
$ ls input_directory
    input_1.bed    input_2.bed    input_3.bed
$ head -n 1 input_directory/Input_1.bed
    chr1        1        100      peak1   1000    +
```

3) output_directory.

The **output_directory** is the path to the **output directory** to which the output files will be exported. The **output directory** will be automatically generated if not existent.

2.2. Optional arguments.

The **panchip analysis** command has two optional arguments: **threads** and **repeats**. Parallelization of **panchip analysis** is monitored through the **jobs** command of UNIX. The default value for **threads** is set as 1. The **repeats** argument indicates the number of technical replicates for the command. Due to the **preliminary normalization** step that permutates the peak sets, we recommend utilizing the value for **repeats** that is greater than or equal to 3 for publications. The default value for repeats is set as 1 to enable rapid preliminary analysis.

3. Statistical analysis of the results.

In contrary to the **panchip analysis** command, which compares a set of input peak sets to an aggregate peak set derived from a bulk of ChIP-seq experiments, the **panchip filter** command compares an input peak set to each individual dataset to enable statistical analysis of the **panchip analysis** results.

Individual overlap is mathematically defined as follows:

$$individual\ overlap\ (experiment, query) = \sqrt{\frac{\{\sum_{peak\ \in\ (experiment\ \cap\ query)} bin\ size_{experiment}\ *\ weight_{experiment}\} \cdot \{\sum_{peak\ \in\ (experiment\ \cap\ query)} bin\ size_{query}\ *\ weight_{query}\}}{\{\sum_{peak\ \in\ experiment}\ bin\ size_{experiment}\ *\ weight_{experiment}\} \cdot \{\sum_{peak\ \in\ query}\ bin\ size_{query}\ *\ weight_{query}\}}}.$$

Unlike the calculation of the **normalized overlap**, the calculation of **individual overlap** does not undergo the **preliminary normalization** step. The weighting formula for the calculation of the **individual overlap** is either the quality score indicated by the fifth column in the standard BED6 format (query) or a uniform distribution (reference).

UNIX.

```
$ panchip filter [-h] [-t threads] library_directory input_file output_directory
```

The **panchip filter** command measures the mean, standard deviation, signal-to-noise ratio, and Bonferroni-corrected *p*-value for each DNA-bound protein. The signal-to-noise ratio is defined as

$$signal\ to\ noise\ ratio\ (reference, query) = \frac{mean\ (experiment\ \in\ reference \mid individual\ overlap(experiment, query))}{standard\ deviation\ (experiment\ \in\ reference \mid individual\ overlap(experiment, query))}.$$

The signal-to-noise ratio is only defined for DNA-bound proteins with number of experiments greater than or equal to 2. For DNA-bound proteins with single supporting experiment, the standard deviation and signal-to-noise ratio are presented as NA. Bonferroni-corrected *p*-value was measured by an unpaired two-sample Welch's *t*-test of all measured **individual overlap** values and **individual overlap** values specific for the DNA-bound protein. Bonferroni correction was performed via multiplying the *p*-value acquired from the Welch's *t*-test by the square root of the number of DNA-bound proteins in the library. DNA-bound proteins with signal-to-noise ratio greater than 2 and Bonferroni-corrected *p*-value less than 0.05 are regarded as passing the filter.

3.1. Positional arguments.

The **panchip filter** command requires three positional arguments: 1) the path to the **library directory**, 2) the path to the **input file**, and 3) the path to the **output directory**.

1) library_directory.

The **library_directory** is the path to the **library directory** designated by **panchip init**. The user can compare the version of the software and the library from the terminal.

UNIX.

```
$ panchip --version
    PanChIP 3.0.10
$ ls library_directory | grep PanChIP-v.*.*.*
    PanChIP-v.3.0.10
```

2) input_file.

The **input_file** is the path to the **input file** that is the peak set in the form of the standard BED6 format.

UNIX.

```
$ head -n 1 input_file
    chr1    1    100    peak1    1000    +
```

3) output_directory.

The **output_directory** is the path to the **output directory** to which the output files will be exported. The **output directory** will be automatically generated if not existent.

3.2. Optional arguments.

The **panchip filter** command has single optional argument: **threads**. Parallelization of **panchip analysis** is monitored through the **jobs** command of UNIX. The default value for **threads** is set as 1. The **panchip filter** command does not require the **repeats** argument as the command does not require the **preliminary normalization** step.

4. Output files.

The output files of the PanChIP software are located in the **output directory** designated by the user during the execution of each command.

4.1. PanChIP analysis.

The **panchip analysis** command generates three files and a subdirectory. The three files include two bash scripts which were executed while running **panchip analysis** and a primary output file that is the **normalized overlap matrix**. The subdirectory generated by the command contains three files that describes the **input statistics**.

UNIX.

```
$ ls output_directory
    analysis.sh    executable.sh    input.stat    primary.output.tsv
$ ls output_directory/input.stat
    SUM.count    SUMdivbyWC.count    WC.count
```

4.1.1. Normalized overlap matrix.

The **primary.output.tsv** file in the **output directory** contains the **normalized overlap matrix**. The **normalized overlap matrix** is consisted of *N*+1 columns that are tab-delimited, with *N* being the number of input files. The first column indicates the gene symbol for each DNA-bound protein from the PanChIP library. The other columns contain the information for the calculated **normalized overlap**.

UNIX.

```
$ head -n 2 output_directory/primary.output.tsv
    TF    input_1    input_2    input_3
    ADNP    0.005    0.008    0.007
```

4.1.2. Input statistics.

The three **input statistics** files describe the statistical properties of the input peak sets and have *N* rows, with *N* being the number of input files. The SUM.count file contains the sum of {|column 3-column 2| * (column 5)} for each input file, while the WC.count files contains the number of rows for each input file. The SUMdivbyWC.count file contains the mean of {|column 3-column 2| * (column 5)} for each input file. The order for each input file is the same as in the **normalized overlap matrix**.

4.2. PanChIP filter.

The **panchip filter** command generates four files and a subdirectory. The four files include two bash scripts which were executed while running **panchip filter**, a primary output file that is the **individual overlap matrix**, and a statistics file that contains the **signal-to-noise ratio matrix**. The subdirectory generated by the command contains three files that describes the **input statistics**.

UNIX.

```
$ ls output_directory
    executable.sh    filter.sh    input.stat    primary.output.tsv    statistics.tsv
$ ls output_directory/input.stat
    SUM.count    SUMdivbyWC.count    WC.count
```

4.2.1. Individual overlap matrix.

The **primary.output.tsv** file in the **output directory** contains the **individual overlap matrix**. The **individual overlap matrix** is consisted of three columns. The first column indicates the gene symbol for the DNA-bound protein from each experiment in the PanChIP library. The second column indicates accession code for each experiment that is either the GEO accession code or the ENCODE accession code. The last column contains the information for the calculated **individual overlap**.

UNIX.

```
$ head -n 2 output_directory/primary.output.tsv
    TF    Experiment    input_file
    ESR1    GSM1325248    0.005
```

4.2.2. Signal-to-noise ratio and adjusted *p*-value.

The **statistics.tsv** file in the **output directory** contains the **signal-to-noise ratio** and **adjusted *p*-value**. The **statistics.tsv** file is consisted of six columns: 1) gene symbol, 2) mean, 3) standard deviation, 4) signal-to-noise ratio, 5) adjusted *p*-value, and 6) filter.

The signal-to-noise ratio is only defined for DNA-bound proteins with number of experiments greater than or equal to 2. For DNA-bound proteins with single supporting experiment, the standard deviation and signal-to-noise ratio are presented as NA. DNA-bound proteins with signal-to-noise ratio greater than 2 and adjusted *p*-value less than 0.05 are regarded as passing the filter.

4.2.3. Input statistics.

The three **input statistics** files describe the statistical properties of the input peak set and have single row. The SUM.count file contains the sum of {|column 3-column 2| * (column 5)} for the input file, while the WC.count files contains the number of rows for the input file. The SUMdivbyWC.count file contains the mean of {|column 3-column 2| * (column 5)} for the input file.

4.2.4. Cell-type-specific PanChIP.

For the ENCODE datasets in the PanChIP library, we have curated the cell line information for each experiment. With this information, users can perform cell-type-specific PanChIP runs.

UNIX.

```
$ wget -O cellLine.tsv
https://raw.githubusercontent.com/hanjunlee21/PanChIP/main/lib/cellLine.txt
$ cat cellLine.tsv | awk '{if($3!="unknown") {print}}' | head -n 2
    TF    Experiment    cellLine
    MNT    ENCSR261EDU_1    HepG2
$ paste output_directory/primary.output.tsv cellLine.tsv | awk
'{if(NR==1||$6=="GM12878") {printf "%s\t%s\t%s\t%s\n",$1,$2,$6,$3}}' >
output_directory/primary.output.GM12878.tsv
$ head -n 2 output_directory/primary.output.GM12878.tsv
    TF    Experiment    cellLine    input_file
    TRIM22    ENCSR835XKS_2    GM12878    0.005
```

5. Benchmark analysis.

5.1. Analysis of CTCF, REST, and ZNF143 ChIP-seq peak sets.

Peak sets from ChIP-seq experiments for **CTCF**, **REST**, **ZNF143** in GM12878 cells were analyzed by PanChIP (accession codes; **CTCF**, ENCSR000AKB; **REST**, ENCSR000BQS; **ZNF143**, ENCSR000DKT).

UNIX.

$ panchip analysis -t 24 -r 3 **library_directory input_directory output_directory**
$ panchip filter -t 24 **library_directory input_directory**/GM12878.**CTCF**.bed **output_directory**/**CTCF**
$ panchip filter -t 24 **library_directory input_directory**/GM12878.**REST**.bed **output_directory**/**REST**
$ panchip filter -t 24 **library_directory input_directory**/GM12878.**ZNF143**.bed **output_directory**/**ZNF143**

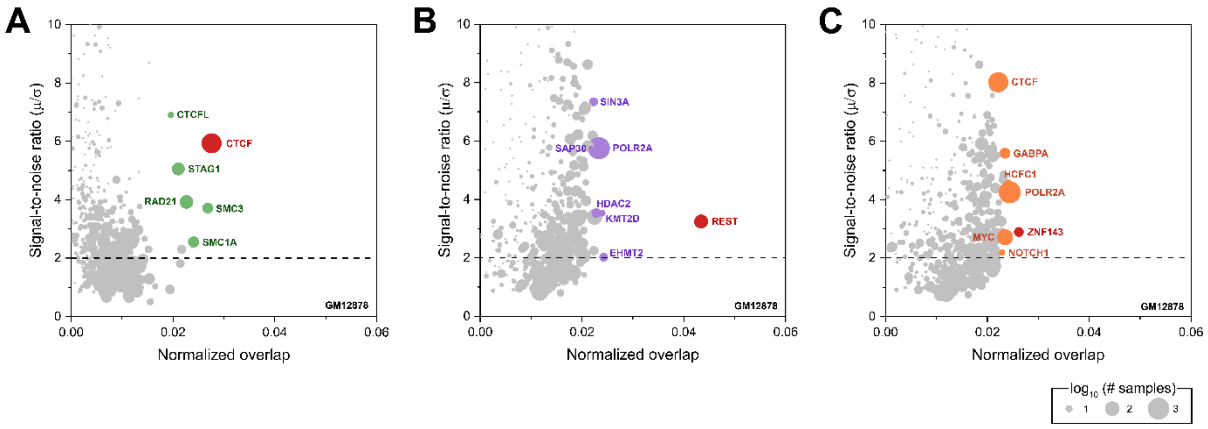

Figure SM1. Normalized overlap versus signal-to-noise ratio for CTCF, REST, and ZNF143 ChIP-seq peak sets from GM12878 cell line (**A**, CTCF; **B**, REST; and **C**, ZNF143). Top hits are indicated in colors, with the input protein itself indicated in red. The size of the individual dot is proportional to the logarithm of the number of samples. Dashed line indicates the threshold for the signal-to-noise ratio filter.

CTCF is a key component of the protein complex in the insulator region and primarily colocalizes with the components of the cohesin complex, including SMC1A, SMC3, RAD21, and STAG1 (Rubio ED et al. *Proc Natl Acad Sci*, 2008; DOI: 10.1073/pnas.0801273105). CTCFL is germ line-specific protein that often localizes in the insulator region (Loukinov DI et al. *Proc Natl Acad Sci*, 2002; DOI: 10.1073/pnas.092123699).

REST is a transcriptional repressor that silences gene transcription via recruiting corepressor SIN3A and histone deacetylases (HDACs) (Huang Y et al. *Nat Neurosci*, 1999; DOI: 10.1038/13165). SAP30 is a component of the human SIN3A complex (Zhang Y et al. *Mol Cell*, 1998; DOI: 10.1016/S1097-2765(00)80102-1), and EHMT2 is a known binding partner of **REST** that functions as a histone methyltransferase (Roopra A et al. *Mol Cell*, 2004; DOI: 10.1016/j.molcel.2004.05.026).

ZNF143 is a transcriptional activator that is known to mediate CTCF-bound promoter-enhancer loops (Zhou Q et al. *Nat Commun*, 2021; DOI: 10.1038/s41467-020-20282-1). **ZNF143** binding is known to often coincide with HCFC1 and GABP binding (Michaud J et al. *Genome Res*, 2013; DOI: 10.1101/gr.150078.112) and with NOTCH1 binding (Ngondo-Mbongo RP et al. *Nucleic Acids Res*, 2013; DOI: 10.1093/nar/gkt088).

5.2. Analysis of E2F1, CTCF, and JUN ChIP-seq peak sets.

Peak sets from ChIP-seq experiments for **E2F1**, **CTCF**, **JUN** in RPE1 cells were analyzed by PanChIP (accession codes; **E2F1**, GSM5353414; **CTCF**, GSM5353423; **JUN**, GSM5353435).

UNIX.

$ panchip analysis -t 24 -r 3 **library_directory input_directory output_directory**
$ panchip filter -t 24 **library_directory input_directory**/RPE1.**E2F1**.bed **output_directory**/**E2F1**
$ panchip filter -t 24 **library_directory input_directory**/RPE1.**CTCF**.bed **output_directory**/**CTCF**
$ panchip filter -t 24 **library_directory input_directory**/RPE1.**JUN**.bed **output_directory**/**JUN**

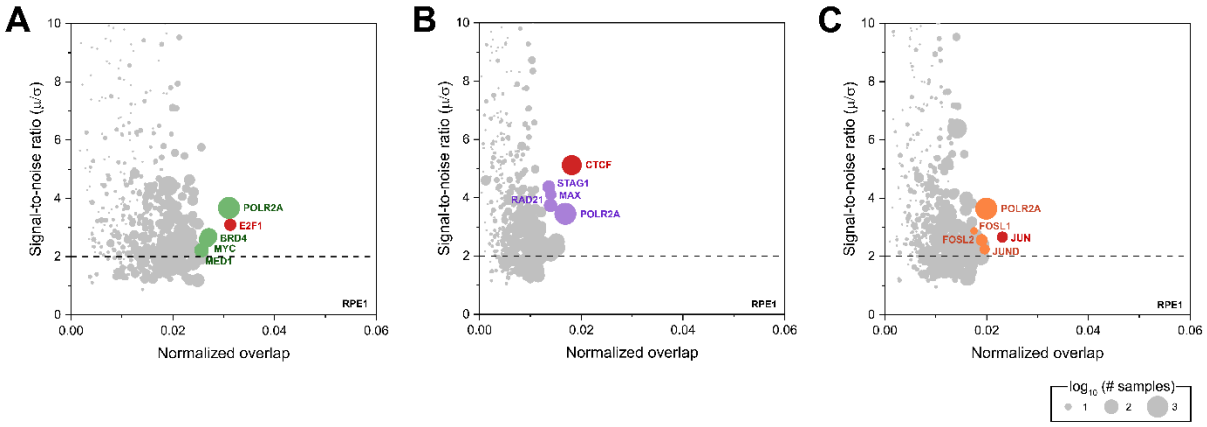

Figure SM2. Normalized overlap versus signal-to-noise ratio for E2F1, CTCF, and JUN ChIP-seq peak sets from RPE1 cell line (**A**, E2F1; **B**, CTCF; and **C**, JUN). Top hits are indicated in colors, with the input protein itself indicated in red. The size of the individual dot is proportional to the logarithm of the number of samples. Dashed line indicates the threshold for the signal-to-noise ratio filter.

E2F1 is a key transcription factor that binds to the promoter regions and regulates cell proliferation (Dyson N. *Genes Dev*, 1998; DOI: 10.1101/gad.12.15.2245). BRD4 is a well-established promoter-bound coactivator of **E2F1**-dependent transcription (Chapuy B et al. *Cancer Cell*, 2013; DOI: 10.1016/j.ccr.2013.11.003). MED1 is a key component of the mediator complex for RNA polymerase II.

CTCF is a key component of the protein complex in the insulator region and primarily colocalizes with the components of the cohesin complex, including SMC1A, SMC3, RAD21, and STAG1 (Rubio ED et al. Proc Natl Acad Sci, 2008; DOI: 10.1073/pnas.0801273105). **CTCF** is known to directly interact with POLR2A (Chernukhin I et al. *Mol Cell Biol*, 2007; DOI: 10.1128/MCB.01993-06).

JUN is a member of the AP-1 transcription factors and forms dimeric complexes with FOS, FOSL1, or FOSL2 (Karin M et al. *Curr Opin Cell Biol*, 1997; DOI: 10.1016/S0955-0674(97)80068-3).

5.3. Comparison of POLR2A ChIP-seq biological replicates.

Peak sets from ChIP-seq experiments for **POLR2A** in GM12878 cells were analyzed by PanChIP (accession code, ENCSR000DKT). Two biological replicate experiments were compared to assess the reproducibility of PanChIP outputs.

UNIX.

$ panchip analysis -t 24 -r 3 **library_directory input_directory output_directory**

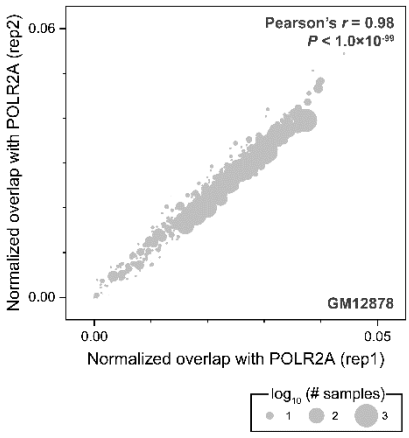

Figure SM3. Comparison of normalized overlap from two biological replicates of POLR2A ChIP-seq peak sets derived from GM12878 cell line. The size of the individual dot is proportional to the logarithm of the number of samples.

The Pearson's correlation coefficient between the two biological replicate experiments was 0.98, with a *p*-value less than 1.0×10^{-99}.

5.4. Cell-type-specific Analysis of IRF4, IRF5, SPI1, and JUND ChIP-seq peak sets.

Peak sets from ChIP-seq experiments for **IRF4**, **IRF5**, **SPI1**, and **JUND** in GM12878 cells were analyzed by PanChIP (accession codes; IRF4, ENCSR000BGY; IRF5, ENCSR976TBC; SPI1, ENCSR000BGQ; JUND, ENCSR000DYS).

UNIX.

$ wget -O cellLine.tsv
https://raw.githubusercontent.com/hanjunlee21/PanChIP/main/lib/cellLine.tsv
$ for file in **GM12878_IRF4 GM12878_IRF5 GM12878_JUND GM12878_SPI1**
$ do
$ panchip filter -t 24 **library_directory input_directory/$file**.bed
output_directory/$file
$ paste **$file**/primary.output.tsv cellLine.tsv | awk '{if(NR==1||$6=="GM12878") {printf "%s\t%s\t%s\t%s\n",$1,$2,$6,$3}}' > $file/primary.output.**GM12878**.tsv
$ done

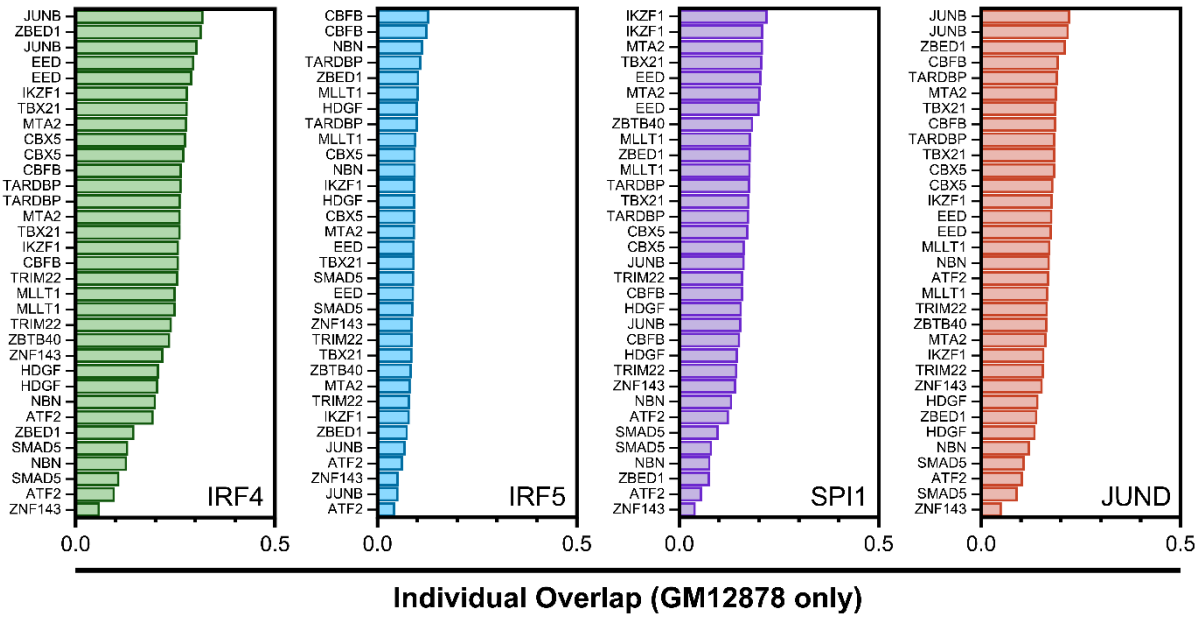

Figure SM4. Bar plot of individual overlap from IRF4, IRF5, SPI1, and JUND ChIP-seq peak sets derived from GM12878 cell line. Multiple occurrences indicate biological replicate experiments.