UNIVERSITY OF CALIFORNIA,
IRVINE

Temporal Bioinformatic Analysis of Epigenetics through Histone Modifications

THESIS

submitted in partial satisfaction of the requirements
for the degree of

MASTER OF SCIENCE

in Biomedical Engineering

by

Robert Thomas West

Thesis Committee:
Assistant Professor Timothy Downing, Chair
Professor Frithjof Kruggel
Assistant Professor Zeba Wunderlich

2019

DEDICATION

To

my wife, family and friends

"…if a process of embryonic development is disturbed, it usually returns to normality some time before reaching the adult condition. Its trajectory, that is to say, converges not merely to the normal end state, but to some earlier point on the path leading towards the steady state.

….

This is well symbolized by the epigenetic landscape. If a ball, running down one of the valleys, were pushed partway up the hillside, it might well reach the valley bottom again before the slope of the valley flattens out as it reaches the adult steady state."

Conrad H. Waddington
"*The Strategy of Genes*"

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to express my deepest appreciation to the committee chair, Professor Timothy Downing, who has provided me with the opportunity and guidance to continually pursue and advance knowledge of epigenetics and stem cell generation. I would also like to thank my lab members and mentors, specifically Zachary Weber-Reitz, for his guidance and help with my work. Without Dr. Downing agreeing to take me into the program, his help and lab member's collaboration, this thesis would not have been possible.

I would also like to give huge thanks to the UCI HPC group of individuals who provided the platform and support for countless GBs of data and processing power. The Biostars.org community also provided numerous insights to issues throughout my code production. Additionally, I would like to thank the Cacchiarelli lab for their contributions to the field by providing their data and research as well as help answering questions.

I thank all the professors in the Biomedical Engineering Department at UCI. In particular, my committee members, Dr. Frithjof Kruggel, who initially taught me the basics and power of the R programming language and Zeba Wunderlich for her work in genome expression.

I thank the University of California, Irvine for providing me the opportunity and place to pursue higher education. Financial support was provided by the California National Guard Education Assistance Award Program.

ABSTRACT OF THE THESIS

Temporal Bioinformatic Analysis of Epigenetics through Histone Modifications

By

Robert Thomas West

Biomedical Engineering, M.S.

University of California, Irvine, 2019

Professor Timothy Downing, Chair

Gathering information on DNA activity during development remains problematic as DNA regulation continues to periodically change, creating unknown downstream variations in final cell heterogeneity. The fluidity and complexity create extreme difficulty and impedes understanding of the process of epigenetics. Enormous amounts of modifications happen within DNA replication; current research only begins to realize the significance of histone modifications, the role they play in mRNA expression and epigenetics of cell fate. With approximately 3 billion base pairs[13], histone modifications are nearly endless, each one can change the regulation of DNA in diverse ways, silencing or enhancing the expression of the DNA. With current analyses, such as Chromatin immunoprecipitation sequencing (ChIP-Seq), the ability to utilize high-powered computing and differing downstream analysis techniques, insightful observations can help realize the complexities of epigenetics and reprogramming. We converted raw ChIP-seq data from Cacchiarelli, et al. into heatmaps and developed computing techniques to analyze six histone modifications through reprogramming with different sets of transcription factors[4]. Heatmaps were generated and sorted two dissimilar ways for visualization. Some histone modifications

depicted their typical roles while we saw some interesting clusters that developed over time. We were also able to discover which genes had the biggest peak enrichment differential. With insight into how each histone modification acts on the DNA over time and how certain promoters change expression, methods for mechanisms and stimulus affecting the expression and speed of reprogramming can be determined. This can then lessen reprogramming time and further enhance the importance of epigenetics in cell fate.

INTRODUCTION

Reprogramming a mature cell into a pluripotent stem cell has opened many opportunities and avenues to study the human developmental process. Studying the reverse developmental process through reprogramming has advanced the discoveries of differentiation pathways, cellular mechanisms and therapeutic techniques. Those critical breakthroughs originated through the factors OCT4/KLF4/c-MYC/SOX2 (OKMS) discovered by Shinya Yamanaka. By increasing the expression of those four key transcription factors for visualization purposes in Figure 1, a Mouse Embryonic Fibroblast (MEF), slowly transforms through different phases of cell state. This happens through epigenetic changes to the DNA, silencing genes that maintain cell homeostasis and activating genes pushing the cell through waves of reprogramming.

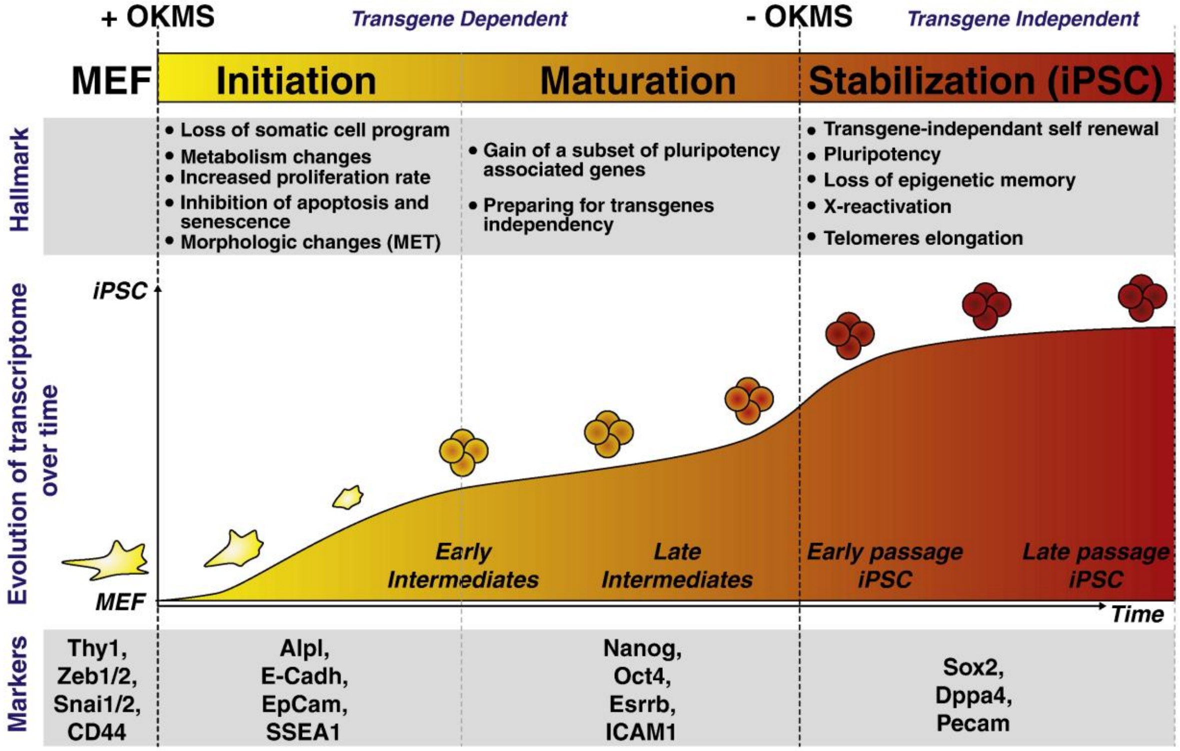

+ OKMS
Transgene Dependent
- OKMS
Transgene Independent
MEF
Initiation
Maturation
Stabilization (iPSC)
Hallmark
• Loss of somatic cell program
• Metabolism changes
• Increased proliferation rate
• Inhibition of apoptosis and senescence
• Morphologic changes (MET)
• Gain of a subset of pluripotency associated genes
• Preparing for transgenes independency
• Transgene-independant self renewal
• Pluripotency
• Loss of epigenetic memory
• X-reactivation
• Telomeres elongation
Evolution of transcriptome over time
iPSC
MEF
Early Intermediates
Late Intermediates
Early passage iPSC
Late passage iPSC
Time
Markers
Thy1, Zeb1/2, Snai1/2, CD44
Alpl, E-Cadh, EpCam, SSEA1
Nanog, Oct4, Esrrb, ICAM1
Sox2, Dppa4, Pecam

Figure 1: Evolution of cell after OKMS introduction. By introducing OKMS transcription factors, DNA activation/repression aspects are altered in a somatic cell over time until induced Pluripotent Stem Cell iPSC state is reached. Specific indicators and transition periods of each phase of reprogramming are indicated. iPSC cell colonies form during late passage[7].

To further study the effects of the OKMS transcription factors on cellular expression, scientists began scrutinizing chromosome histone modifications, specific cell state markers and their silenced or activated epigenetic expression. Environmental influences on cell fate and resulting epigenetic state and cell function happens to be a relatively new concept compared to the 1D DNA blueprint script. Environment plays a much bigger role on an organism's development and thus a significant role in reprogramming.

In the study by Cacchiarelli, they discuss cells passing through transient cell states during reprogramming and expressing altered epigenetic signatures. They question if those transient cell states can be bypassed through unique chemical inhibitors or transcription factors[4]. Cellular stages through reprogramming can be altered and regulated through different chromatin states via the chemical inhibition or activation[5]. Successful reprogramming from fibroblasts to an induced pluripotent stem cell colony stands as a delicate process and can fail for seemingly insignificant reasons. Figure 2 depicts the cell morphologies from human induced Fibroblasts Telomerase, immortalized (hiF-T), cells to the distinct iPSC colonies. There are noticeable changes to the cell morphology as well as significant losses in the number of cells. Cacchiarelli et al. reprogrammed fibroblasts with a DOX-inducible, polycistronic OKMS lentiviral vector, giving rise to an initial hIPSC line. Then the cells were differentiated in vitro to a hiF, an hTERT lentivirus was introduced to create hiF-T and then reprogrammed again into hIPSC-T. The secondary reprogramming displayed more identifiable transient waves as well as more efficiently, with also an unlimited expansion potential.

Many studies are attempting to improve those drastic changes and increase both the speed and efficiency in order to save money and time. In order to accomplish this task, we must know what to target. We know the DNA is either silenced or activated based on the histone modification and the state of the cells are determined through cell markers. Those histones can be modified through transcription factors and by changing the DNA activation state, the cell will change. By studying the expression of genes and if a histone modification is present, we can determine what to target to affect the speed and efficiency of reprogramming.

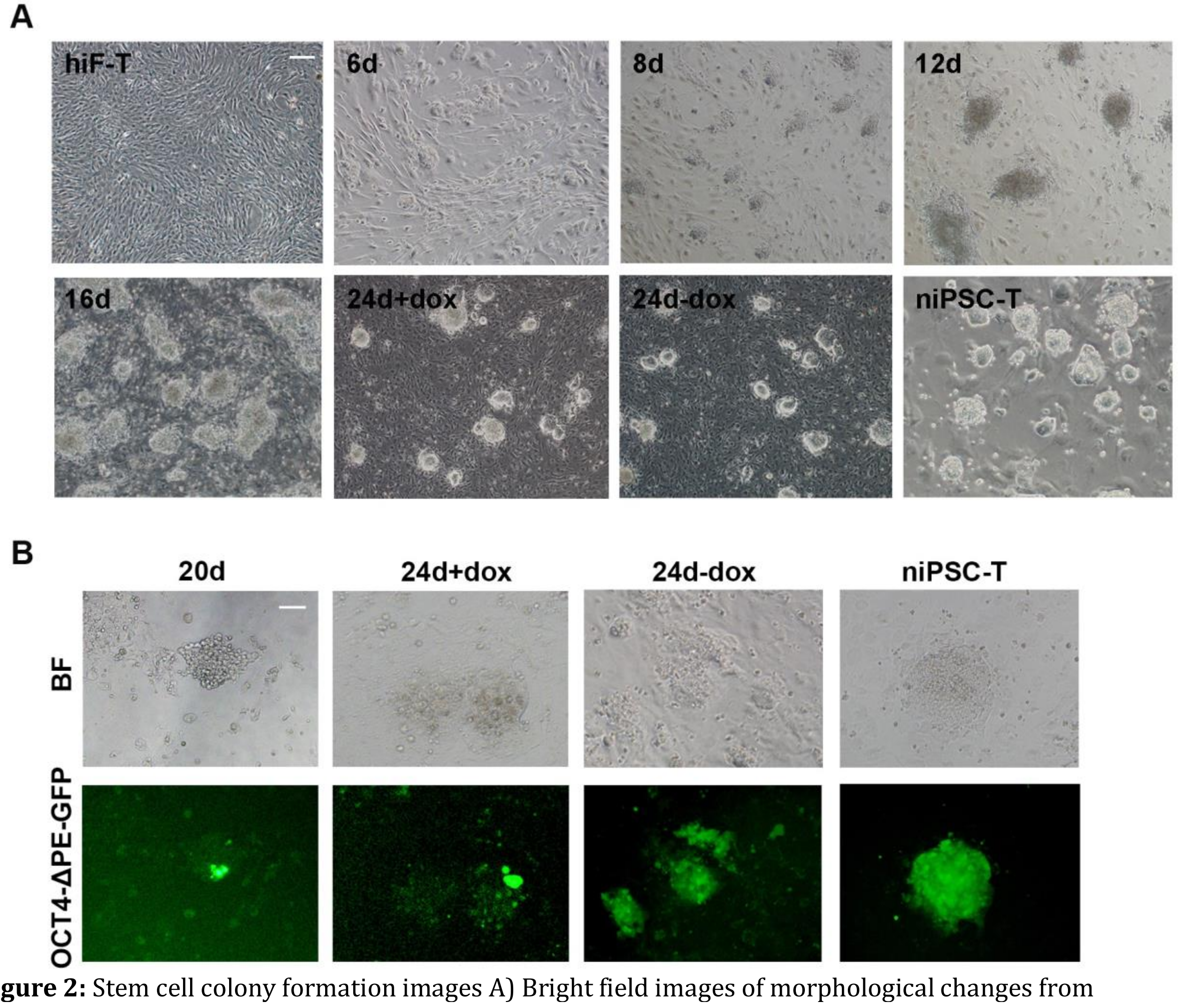


Figure 2: Stem cell colony formation images A) Bright field images of morphological changes from immortalized hiF-Ts to naïve induced pluripotent stem cell-TERTs(niPSC-Ts) for different time points of reprogramming cells. B) Time points of bright field images and GFP of OCT4 change from late reprogramming cells to niPSC-Ts using doxycyckine (DOX) inducible gene expression of OKMS for "secondary" reprogramming. Scale bars, 100 μm[41].

Facilitation through either expression or repression of transcription factors which affect histone modifications enhances the efficiency of reprogramming[22]. In order to accurately determine cause for epigenetic change, profiling of chromatin states before and after reprogramming can help identify regulatory genes that dictate the cell state[18]. Induced pluripotent stem cells undergo a series of changes to molecular pathways during reprogramming and most of the cell's distinctiveness form roadblocks in order to maintain somatic identity. Pathways, such as the p53 pathway, a central component regulating cell cycle preventing cancer, are continually studied to find a way to increase efficiency and speed of reprogramming[39]. The p53 pathway is commonly knocked down to breach an analogous "Stress and senescence" barrier as seen in Figure 3. We are focusing our

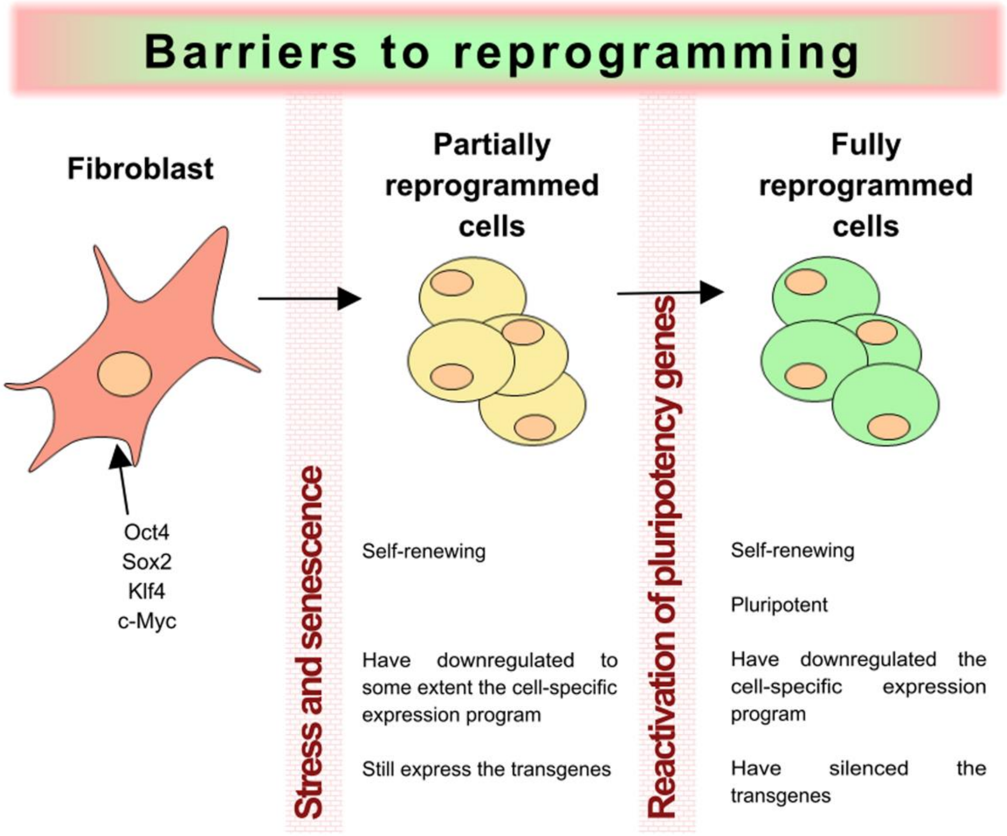

Barriers to reprogramming
Fibroblast
Oct4
Sox2
Klf4
c-Myc
Stress and senescence
Partially reprogrammed cells
Self-renewing
Have downregulated to some extent the cell-specific expression program
Still express the transgenes
Reactivation of pluripotency genes
Fully reprogrammed cells
Self-renewing
Pluripotent
Have downregulated the cell-specific expression program
Have silenced the transgenes

Figure 3: Identified cellular barriers that preserve cell homeostasis and their identity. Increased expression of the OKMS factors create stress, decrease of normal cell division and stimulate apoptosis. The chromatin environment creates silenced sections of DNA after the first barrier is crossed. This environment of partial cell specific expression and partial iPSC expression depend on the OKMS factors because they are not yet self-expressing the pluripotency factors: the second barrier. Once the DNA histone modifications are poised for the expression of pluripotency genes, the second barrier is crossed, and the cells maintain their iPSC state[3].

research on certain genes that determine those transient states and ways to potentially breakdown or skip those barriers altogether.

When the OKMS transcription factors are introduced, they specifically bind to target sites, change chromatin state and reactivate pluripotency genes to then generate hIPSCs. Transcription factors such as Transcriptional Enhancer Factor (TEAD1), Serum Response Factor (SRF), the Yes-associated protein (YAP)/Transcriptional coactivator with PDZ-binding motif (TAZ) and the genes they target affect development and are also affected by environmental influences[8,21]. Additionally, Adhesome binding sites, a combination of Adherins and Caherins associated genes, are affected by the environment and change during development. Changes in gene expression and their targets effect the epigenetics of cells through development and reprogramming, yet the magnitude of their effects and when they are most influential is still unclear[29, 40, 15]. The chromatin state either helps activate or suppress transcription factors and create the epigenetic landscape for cell identify and morphology. By changing histone modifications, the chromatin state changes, from compact to open or vice versa and in turn regulates DNA transcription. Research on intermediate states and how to potentially skip those states or guide epigenetics remains ongoing and data intensive. Histones can be either methylated or acetylated, depending on where the modification occurs determines recruitment of transcriptional machinery or repression a model example seen in Figure 4. The histone modifications H3K4me1 (activation), H3K4me2 (activation), H3K4me3 (activation), H3K27me3 (repression), H3K27ac (activation), H3K4me36 (activation) were utilized in our analysis. Each modification was also analyzed in the following conditions: from primary BJ foreskin

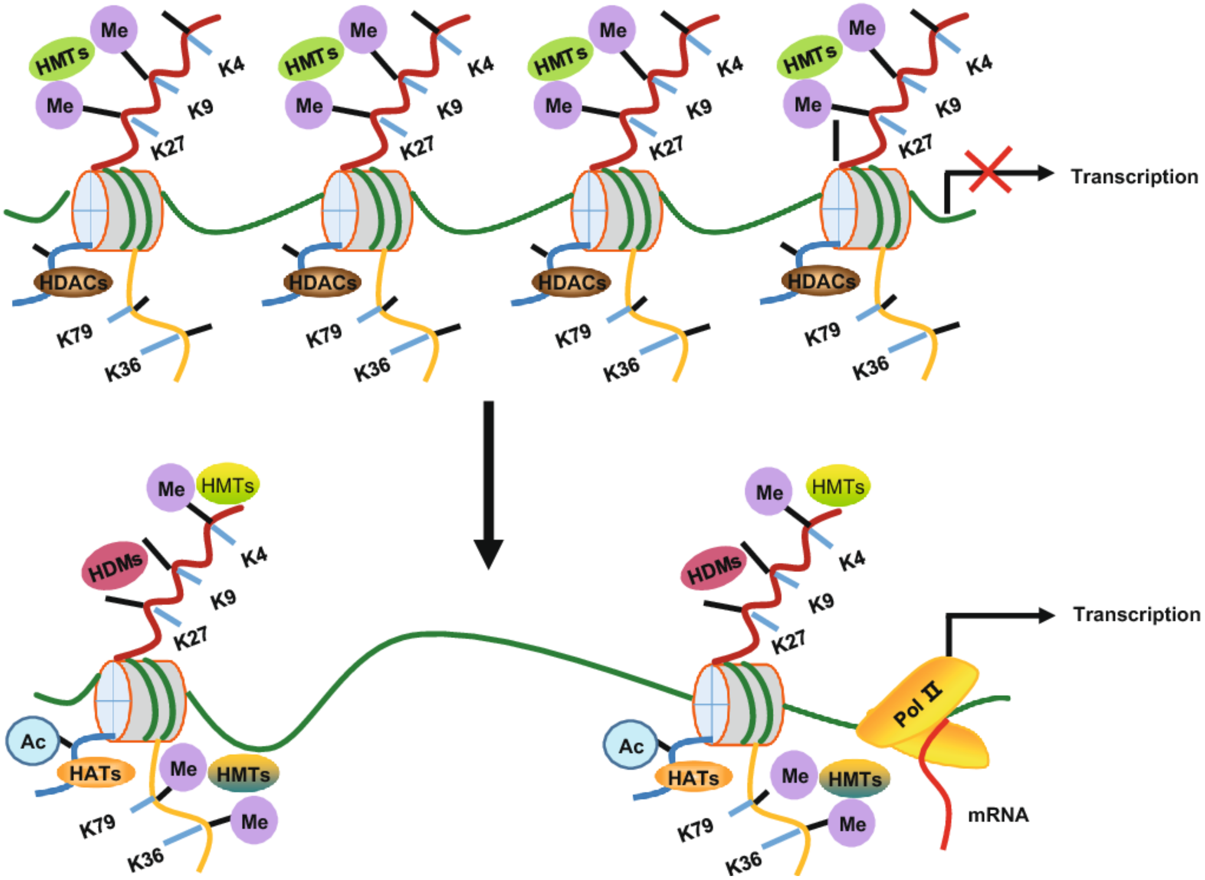


Figure 4: Histone modification model example. Tri-methylated H3K27 and H3K9 are repressive histone markers and create a compact chromatin. Tri-methylated H3K4, H3K36 and H3K79 are active histone markers and open DNA for transcription. HDMs are histone demethylases. HMTs are histone methyltransferases and HATs are histone acetylases. HDACS are histone deacetylases. Top diagram: no transcription occurs because of the methylation of repressors. Bottom diagram: transcription occurs because of the opposite, chromatin are loosened and transcriptional machinery is recruited[30].

fibroblast hiF-T (day-0), day-5, day-10, day-24 with DOX, day-24 without DOX and hIPSC[4]. Day-10 data for H3k27me3 and H3K36me3 was not available. Figure 5 further illustrates transcription factors interacting with a strand of DNA with certain histone modifications and changing the state of the chromatin and DNA "openness."

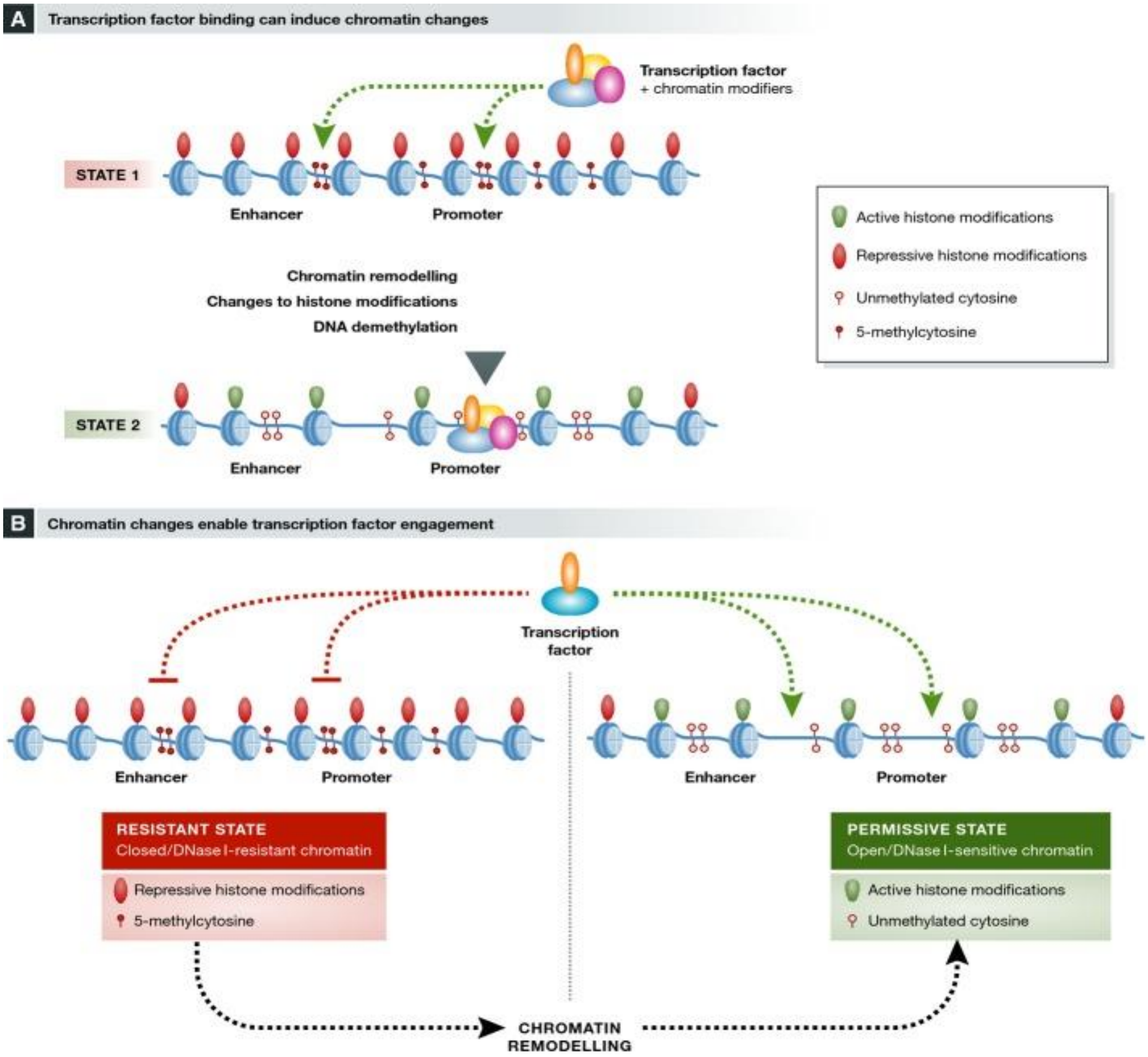

A Transcription factor binding can induce chromatin changes
Transcription factor
+ chromatin modifiers
STATE 1
Enhancer
Promoter
Chromatin remodelling
Changes to histone modifications
DNA demethylation
STATE 2
Enhancer
Promoter
Active histone modifications
Repressive histone modifications
Unmethylated cytosine
5-methylcytosine
B Chromatin changes enable transcription factor engagement
Transcription factor
Enhancer
Promoter
Enhancer
Promoter
RESISTANT STATE
Closed/DNase I-resistant chromatin
Repressive histone modifications
5-methylcytosine
PERMISSIVE STATE
Open/DNase I-sensitive chromatin
Active histone modifications
Unmethylated cytosine
CHROMATIN
REMODELLING

Figure 5: Transcription factor effects on histone modifications and relationship with chromatin configurations. A) Transcription factors are introduced, they bind to the DNA and change the chromatin environment as well as recruit histone modifiers (HDMS, HMTS, HATS, HDACS). B) A homeostatic specific cell identity of closed DNA state on the left and an open environment on the right after transcription factors are introduced and initiating reprogramming[25].

ChIP-seq techniques have consistently proven to provide insight into identifying the effects of histone modifications and epigenetics of target DNA sites encoding proteins and transcription factors. By analyzing and interpreting ChIP-seq data, we can begin to understand the chromatin environment and the activated or repressed state of the DNA. ChIP-seq data gives information on the presence and quantity of a specific histone modification. Essentially, DNA, associated proteins and chromatin are cross linked, each section are sheared into ~500 basepairs, then those cross-linked fragments of the specific histone proteins are selectively immunoprecipitated with specific antibodies and then those fragments are sequenced to create ChIP-seq data[33]. ChIP-seq data contains sequences from regions specifically or indirectly bound to the antibody or histone protein. We can also determine the proximity of histone modification enrichment and location of transcription factor binding based on the number of reads relative to the center of a known transcription start site. In order to finely tune the data from ChIP-seq and find the most accurate information through the noise we perform peak calling. A peak in ChIP-seq data is defined as a location where multiple reads of the DNA fragments line up and indicate a higher and more reliable indication the histone modification exists. Through analysis of this information, the epigenetic state of a cell can be mapped and understood. Understanding the chromatin environment allows us to determine if a particular gene set helps or hinders passing through a barrier of cellular reprogramming. The process to study epigenetic mechanisms and chromatin interactions contains massive amounts of data as well as a pipeline of comprehensive analytics.

From the Cacchiarelli experiment, utilizing only their ChIP-seq data, the initial raw FASTQ data for 34 files (each histone modification and each time condition) totaled 46GB of data. To put this in perspective, downloading all the files from off the online database at a very high rate of 50Mb/s takes approximately 2 hours. As discussed in the Methods section, the raw files undergo alignment to the human genome, compressed for quicker computations, remove duplicates from Polymerase chain reaction (PCR) for quality, conversion to a format with readable and computational information ChIP-seq data for analysis and heatmap generation. Each step of the process differs in time, on average each step takes 5-10 minutes depending on the original size due to the number of computations required. Approximately 280 GBs of information was produced for analysis from the original 46GBs. The result of high throughput ChIP-seq and inclusive analysis of the derived information brings to light new discoveries of linkages, directionality, and downstream effects which control the epigenetic and developmental process leading to a greater understanding of cellular gene expression and causation[14, 35]. There are a multitude of diverse software program techniques and tools to analyze data, selected based on the users desired result and goal[26].

ChIP-seq heatmap generation softwares available have problems, they are very specific to the creators' input files, programming language and desired outputs and cannot be readily customized. All ChIP-seq heatmap generation software such as Deeptools, NGSPLOT, ChAsE, HOMER, Genomation, SeqPlots, EaSeq, additional R Packages and more take an input (usually a peak file), utilize their internal functions and output a heatmap sorted based only on the number of reads with given parameters across all TSS[36]. With only the ability to sort ChIP-seq data a certain way and only over all TSS, limits the ability to innovatively visualize the data differently and discover new relationships. Our goal is to create a unique toolchain and analysis of ChIP-seq data for heatmap visualization that can be thoroughly customized based on the user. To visualize enrichment of genomic signals over specific target regions as well as sort other ChIP-seq heatmaps based on another ChIP-seq sort we created a pipeline that can be customized and cross-examined between different histone modifications and conditions. The unique code and process allows for the user to define which ChIP-seq data they wish to sort another ChIP-seq data by to visualize how they compare at the same location in a different condition. Additionally, we create a new plot that compares the ChIP-seq peak values temporally during reprogramming, finds the overlapping genes and plots those genes of significance. We will utilize data from the Cacchiarelli lab to reanalyze and develop new ways to visualize, investigate and understand the ChIP-seq data. In addition, we will discuss the specific chosen analytical pipeline necessary to achieve heatmap visualizations and plots as well as the ability to re-use and customize the process to meet the intent of the user.

METHODS

Software and computing

Many different tools for any step in the data pipeline process exist as well as diverse ways to select certain functions of that tool. Determining the most useful software depends upon the user's desired output or goal as well as comfortability and expertise in the language or program. Due to the massive amounts of data generated through distinctive histone modifications and time points, we utilized the UCI HPC Cluster in order to store files and high-powered computing to analyze the data. As stated earlier, the amount of data and the number of computations in order to align genomic data take approximately 5-10 minutes. From the raw FASTQ file to create one heatmap the process takes 9 steps, as seen in Figure 6, and approximately 1 hour and 30 minutes. To access the HPC Cluster, the Windows SSH terminal MobaXterm allowed operability from any computer with Cisco AnyConnect. HPC Cluster conveniently has installed software with executables, functions and libraries that can be loaded through modules. Each HPC can have special nodes or versions as well as a specific order of loading different modules for proper version control

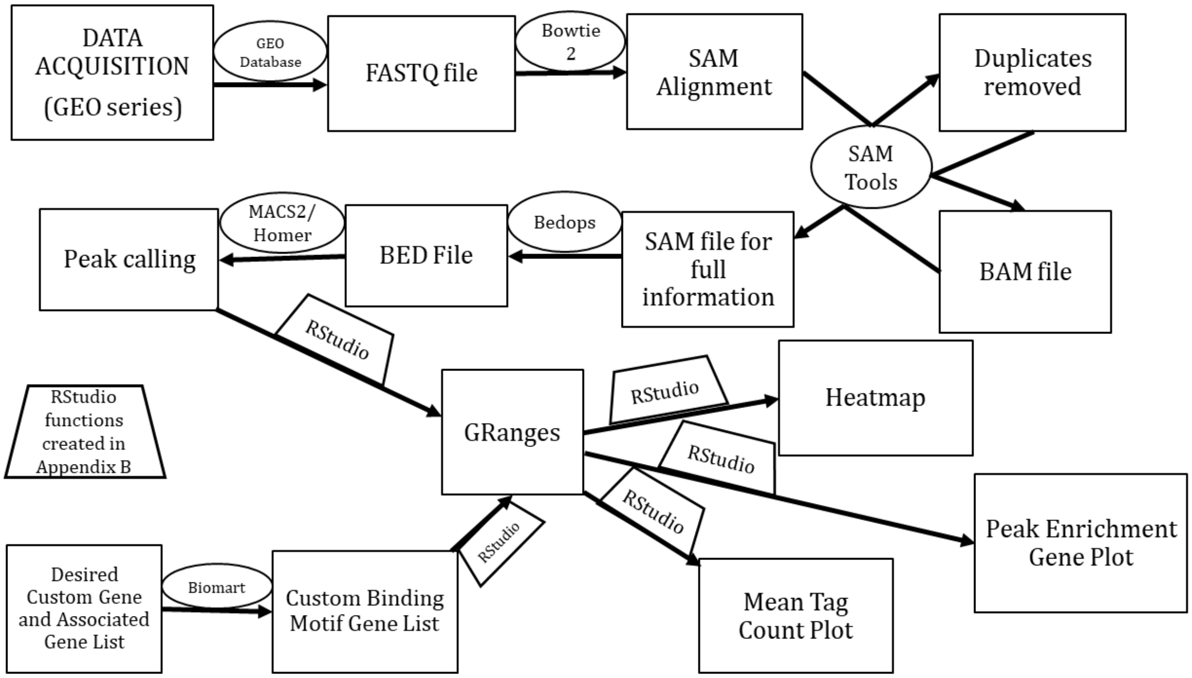

DATA ACQUISITION (GEO series)
GEO Database
FASTQ file
Bowtie 2
SAM Alignment
Duplicates removed
SAM Tools
BAM file
SAM file for full information
Bedops
BED File
MACS2/ Homer
Peak calling
RStudio
RStudio functions created in Appendix B
GRanges
RStudio
Heatmap
RStudio
Peak Enrichment Gene Plot
RStudio
Mean Tag Count Plot
RStudio
Desired Custom Gene and Associated Gene List
Biomart
Custom Binding Motif Gene List

Figure 6: Data pipeline toolchain. Beginning in the upper left and ending in the lower right. Each box is the file, each circle is the software/program used to convert between one file and the next. The trapezoids are code and corresponding functions, found in Appendix B for more detail, created to generate the outputs we desired.

and functionality. In our case for RStudio, we accessed qrsh (the interactive node), then loaded the module rstudio, followed by the version, R/3.5.1 and finally the initialized the application with the command rstudio[31]. Bismark software can also be used for analysis and generation of epigenetic data[18].

Data Acquisition

First, in order to derive information from an experiment, ChIP-seq data must be generated by sending ChIP data for sequencing from the Illumina platform, receiving a raw sequencing data file in FASTQ format. Additionally, and more common for bioinformatic analysis, the data associated with a ChIP experiment can be available through a GEO accession number. In our case we utilized GSE62777 (GEO Series) and supplementary ChIP-seq data, GSE71033, from the Cacchiarelli lab to analyze and visualize epigenetic modifications[4]. The raw data may also be located through accessing a Sequence Read

Archive (SRA) associated with the GEO accession repository. Within the SRA are the SRX Sequence Read Experiments (SRX) and finally, what we want, the Sequence Read Runs (SRR) for the experiments. All the SRR ChIP-seq files were downloaded into a working directory within MobaXterm and the HPC Cluster. SRR file numbers were matched to the correlating ChIP-seq condition, e.g., GSM1825750 (GEO Sample) with SRR for hiF-T_P15_H3K4me3_Rep1 to conserve data information with condition. Six histone modification conditions, H3K4me1, H3K4me2, H3K4me3, H3K27me3, H3K27ac and H3K36me3 and six time points, hiF-T, 5-day, 10-day, 24-day with DOX, 24-day without DOX and hIPSC, were gathered from the Cacchiarelli data repository. The 10-day time point for H3K27me3 and H3K36me3 were not available on the repository and not analyzed.

Data Processing and Conversion

For appropriate and accurate analysis of FASTQ file information the ChIP-seq reads are aligned with the human genome, we utilized the hg19 genome release for all analysis for continuity and accuracy. The Bowtie 2 tool aligns and sequences the very noisy and

Table 1: Sample lines of code to convert raw data alignment to the genome in SAM format

```
#!/bin/bash
# fastq to sam
module load bowtie2
bowtie2 -p 4 -x /pub/rtwest/hg19 -U /pub/rtwest/IAD2/SRR2106001.fastq > hiFT-
T_P15_H3K27ace_Rep1.sam
bowtie2 -p 4 -x /pub/rtwest/hg19 -U /pub/rtwest/IAD2/SRR2106002.fastq > hiFT-
T_P15_H3K4me1_Rep1.sam
bowtie2 -p 4 -x /pub/rtwest/hg19 -U /pub/rtwest/IAD2/SRR2106003.fastq > hiFT-
T_P15_H3K4me2_Rep1.sam
bowtie2 -p 4 -x /pub/rtwest/hg19 -U /pub/rtwest/IAD2/SRR2106004.fastq > hiFT-
T_P15_H3K4me2_Rep2.sam
bowtie2 -p 4 -x /pub/rtwest/hg19 -U /pub/rtwest/IAD2/SRR2106005.fastq > hiFT-
T_P15_H3K4me3_Rep1.sam
bowtie2 -p 4 -x /pub/rtwest/hg19 -U /pub/rtwest/IAD2/SRR2106006.fastq > hiFT-
T_P15_H3K27me3_Rep1.sam
bowtie2 -p 4 -x /pub/rtwest/hg19 -U /pub/rtwest/IAD2/SRR2106007.fastq > hiFT-
T_P15_H3K36me3_Rep1.sam
bowtie2 -p 4 -x /pub/rtwest/hg19 -U /pub/rtwest/IAD2/SRR2106008.fastq > hiFT-
T_P15_INP_Rep1.sam
bowtie2 -p 4 -x /pub/rtwest/hg19 -U /pub/rtwest/IAD2/SRR2106009.fastq > hIPSC-
T_P10_H3K27ace_Rep1.sam
bowtie2 -p 4 -x /pub/rtwest/hg19 -U /pub/rtwest/IAD2/SRR21060010.fastq > hIPSC-
T_P10_H3K27me3_Rep1.sam
```

unorganized data to the genome through 50-100 characters comparisons, the memory footprint is about 3.2 GBs of RAM[19]. The output of Bowtie 2 is Sequence Alignment Map (SAM) formatted files and stores the sequenced data with eleven mandatory fields crucial for alignment[19]. Once the files are aligned and in SAM format, we sorted the sequencing data, removed duplicates and finally converted to a BAM (Binary Alignment Map) file through SAM tools. We pipelined each SAM file through the following code, creating directories for each modification condition for consistency, ease of access and functionality for follow on analysis and visualization.

Table 2: Sorting SAM format files, removing duplicates from PCR amplification, converting to BAM format, finding peak data and converting to a readable BED file.

```
#!/bin/bash
#module load samtools
#module load homer
#module load bedops
mkdir -p /pub/rtwest/IAD2/ChIP-SeqSamples/_${file}/
cd /pub/rtwest/IAD2/ChIP-SeqSamples/_${file}/
filedir=/pub/rtwest/IAD2/ChIP-SeqSamples/_${file}/
samtools view -bS /pub/rtwest/IAD2/ChIP-SeqSamples/${file}.sam > ${filedir}${file}.bam
 samtools sort ${filedir}${file}.bam -o ${filedir}${file}_Sorted.bam

 samtools view -b -F 0x400 ${filedir}${file}_Sorted.bam >
 ${filedir}${file}_Sorted_PCRDupesRemoved.bam
 samtools index ${filedir}${file}_Sorted_PCRDupesRemoved.bam
 samtools view -h ${filedir}${file}_Sorted_PCRDupesRemoved.bam >
 ${filedir}${file}_Sorted_PCRDupesRemoved.sam

 #makeTagDirectory and finding peaks for TF, Histone and TSS
 makeTagDirectory $filedir ${filedir}${file}_Sorted_PCRDupesRemoved.sam

 findPeaks ${filedir} -style factor -o ${filedir}/${file}_peaks.txt

 findPeaks ${filedir} -style histone -o ${filedir}/${file}_regions.txt

 findPeaks ${filedir} -style tss -o ${filedir}/${file}_tss.txt

 #annotatePeaks
 annotatePeaks.pl ${file}_peaks.txt hg19 > ${file}_AnnotatedPeaks.txt
 annotatePeaks.pl ${file}_regions.txt hg19 -size 6000 -hist 10 -ghist -d ${filedir} >
 ${file}_heatmapMatrix.txt

 #convert annotated peak files to bed
 pos2bed.pl ${file}_AnnotatedPeaks.txt > ${file}_peak.bed
 pos2bed.pl ${file}_heatmapMatrix.txt > ${file}_heatmap.bed
 pos2bed.pl ${file}_regions.txt > ${file}_regions.bed

 #convert to bed
  sam2bed < ${filedir}${file}_Sorted_PCRDupesRemoved.sam > ${filedir}${file}.bed
```

BAM files are compressed versions of SAM, require less memory to store and can be modified and converted faster, saving processing time for each file. PCR duplicates are removed, the file is aligned and sorted by mapping location and converted BAM back to SAM for follow on conversion to a BED file. BED files were converted utilizing Bedops and have the rudimentary required structure of chrom, chromStart and chromEnd features in the data, these three data points are the basis for DNA structure. Additionally, we also included all the information from the SAM file, name of the file, strand, score, sequence, etc. and all the necessary information to create a heatmap of the ChIP-seq data. Heatmaps allow an easy way for visualization and examination of large-scale snapshots of certain genomic distributions to understand biological activity and state.

Peak Calling

The extreme amount of data that ChIP-seq generates produces noise, making the precise area of histone modification obscured. Next-generation sequencing (NGS) "produces an unprecedented amount of data. Raw data and images are on the order of terabytes per machine run."[28] In order to more accurately predict the binding sites of histone modifications, calling peaks of the read count data within the bed file further permits an accurate location. Both Homer and MACS2 (Model-based Analysis for ChIP-seq, used in pipeline) are customizable software modules that read BED file information. Certain parameters within the code, respective to either software, produce an output of peak data across the input file of binding-site sequences as seen in Table 3. Homer has multiple programs and one, findPeaks, performs peak calling that helps identify regions with more sequencing reads in defined parameter regions to output a new file with less noise[12]. MACS2 enhances the spatial resolution of potential binding sites for histones by merging the sequence antibody tag position and orientation of the strand[42]. Essentially, the software models each alignment fragment, compares forward and reverse strands against the genome; if there are enriched regions, the software calculates a p-value using a Poisson distribution to capture a local bias and identifies a “peak”, allowing a robust identification. Specifically, for our commands, “callpeak” to use MACS2 call peaks from alignment results, “-t” defines the filename/treatment, “-f” format of the file, “-g” mappable genome size, “-n” output file name, “--broad” the parameter to search for the peak. We used broad as the MACS2 algorithm utilizes nearby highly enriched regions to determine a legitimate peak or not which is better suited for histone modifications because their binding site regions can be much wider. MACS2 outputs seven files of different peak information that can be used in further downstream analysis and descriptions can be found on their github[42]. We utilize the .xls peak file and broadpeak BED file, each containing chromosome name, start position of peak, end position of peak, length of peak region, absolute peak summit position, pileup height at peak summit, -log10(pvalue) for the peak summit and a fold enrichment value against random Poisson distribution. The broadpeak BED file also has an integer score, fold change, and -log10qvalue. These two files are used for further downstream analysis.

Table 3: MACS2 Peak enrichment generation of excel files for follow on analysis and plotting

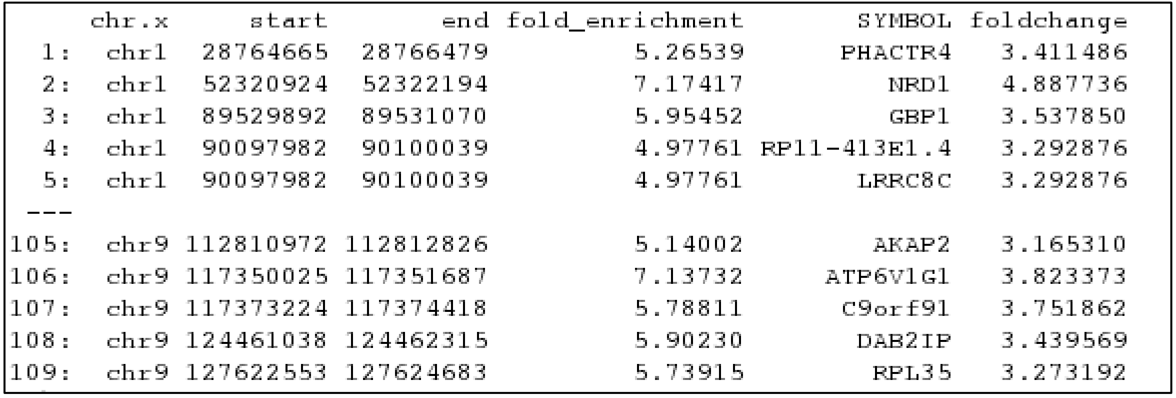
```
#!/bin/bash
#module load samtools
#module load macs2/2.0.10
cd /pub/rtwest/IAD2/ChIP-SeqSamples/_${file}/

filedir=/pub/rtwest/IAD2/ChIP-SeqSamples/_${file}/

macs2 callpeak -t ${file}_Sorted.bam -f BAM -g hs -n ${file} --broad

chmod +x ${file}_peaks.xls

#removing blank space and comments in header
sed -i '/#/d' ${file}_peaks.xls
sed -i '/^$/d' ${file}_peaks.xls
```

Computational and Downstream Analysis of Data

For most of the analysis, the workflow depends on RStudio version 3.5.1 and various packages from Bioconductor. Before visualization of the data, the BED file needs to be converted into a Genomic Ranges (GRanges) file with information relative to the genome. Understanding the GenomicRanges package from Bioconductor remains vital to comprehend and manipulate the data to access and utilize accurate data. During conversion to GRanges, the sequences were sorted from Chromosome 1 to Chromosome Y for consistency. With the aim to create a Heatmap, we employed RStudio software to convert BED files and generated a matrix of the sequence data. We binned each BED file and vectorized, then compared the information to the overlaps within the desired region of interest. Transcriptional Start Sites TSS are typically used for analysis to see which parts of the genome are silenced or activated based on the histone modification. We generated a matrix centered plus or minus 3000 base pairs distance around the TSS of binding sites similar to ChIPseeker[9]. Additionally, we compared the GRanges of each modification and condition to select known binding motif regions of the TEAD1, SRF, YAP/TAZ transcription factor gene targets, the Adhesome genes and Late Embryogenesis genes. To do this, we created a custom transcription database from the list of transcripts for each set. The YAP/TAZ and TEAD1 list came from Dupont, et al. data, Adhesome and SRF list came from Medjkane et. al, the Late Embryogenesis came from Cacchiarelli et. al Supplemental Data[4,8,21]. The list of genes, we used at the Ensemble gene, was uploaded to the mart http://uswest.ensembl.org/biomart/martview, filtered for strand, start, end, chr, tx name, tx count (score), and tx type then outputted to an .xls format. Specific details to follow for the custom transcript database generation can be found in Appendix B, Custom Promoter Generator.

Finally, an analysis of the peak fold change enrichment data changing over time for each modification was conducted in order to determine which sets of promoters were most affected by histone modification during reprogramming. Only peaks that are present from one condition to the next are saved. We compared each modification to the next sequential time point modification through conversion to GRanges and utilization of both dplyr and fuzzyjoin to compare overlapping sequences with peak data[32]. Dplyr within the tidyverse allows memory efficiencies, grammar simplicities and eases data manipulation due to dataframe difficulties within RStudio[11]. Fuzzyjoin takes data manipulation another step by comparing two dataframes and selecting certain matching, or within certain parameters, columns or rows and joining the two dataframes together. The parameter aspect is crucial as the identified peaks of two broad regions (histone binding) in two timepoint conditions that code for a gene can be a hundred DNA base pairs different, yet still be the same gene that a histone will bind. This overlap is where the histone modification and epigenetic state are the same across time points. The overlaps that we kept for visualization were those DNA ranges with the desired peak fold change threshold (defined by the user) as well as sensitivity for the number of sequences considered to be an overlap. Once the GRanges were combined, they were converted to NCBI format in order to perform a database call and select for the desired gene ID names. This can also be manually done through UCSC Genome Browser, however time intensive. After the gene names are associated with the desired GRanges overlaps, they are then added back to the original GRanges with corresponding peak fold change and enrichment. An example of list of output from one comparison result before plotting depicted in Table 4.

Table 4: Overlapping peak data between day-5 and day-10 in H3K4me3 histone modification. Each column is the combination of joining the peak data that had a fold enrichment above 4 and the specific gene represented in the start and end sequence with the fold change from the previous (day-5) to the sequential time point condition (day-10)



```
     chr.x     start       end fold_enrichment        SYMBOL foldchange
  1:  chr1  28764665  28766479         5.26539       PHACTR4   3.411486
  2:  chr1  52320924  52322194         7.17417          NRD1   4.887736
  3:  chr1  89529892  89531070         5.95452          GBP1   3.537850
  4:  chr1  90097982  90100039         4.97761 RP11-413E1.4   3.292876
  5:  chr1  90097982  90100039         4.97761        LRRC8C   3.292876
 ---
105:  chr9 112810972 112812826         5.14002         AKAP2   3.165310
106:  chr9 117350025 117351687         7.13732      ATP6V1G1   3.823373
107:  chr9 117373224 117374418         5.78811       C9orf91   3.751862
108:  chr9 124461038 124462315         5.90230        DAB2IP   3.439569
109:  chr9 127622553 127624683         5.73915         RPL35   3.273192
```

Data Visualization

To best view the massive amounts of information ChIP-Seq data provides, heatmaps are typically chosen as the most ideal way to view the data across the genome. Heatmaps take up to millions of data points, correlates and plots the data as defined on a graph. Mean tag counts across the centered promoters of interest were also created to analyze the frequency of coverage near the TSS. For each condition, a control hiF-T dataset was sorted by number of read counts and then the order of the next heatmap time point conditions (5, 10, 24 and hIPSC) were based on the hiF-T order for the given histone modification. For comparative purposes, the DNA sequence from start to end and ordered by read count were generated for each time point and each histone modification.

As previously described, the modifications with overlapping sequences of the sequential time point and their corresponding gene IDs were plotted. The genes that had the desired peak fold enrichment change, defined by the user, were compared to the next time point and then generated a list for each histone modification from hiF-T to hIPSC. We captured a fold change greater than 3 or less than 0.1 between conditions for significance. We set sensitivity for overlaps to 1000 base pairs, the sequence must have 1000 base pairs matching within and on either side of the GRanges of interest to obtain an accurate overlap of promoter binding. A greater number creates more sensitivity, while a lower number allows for more overlaps. For one histone modification, H3K27me3, we set 5 as the overlap sensitivity in order to capture any overlap, anything higher did not find any GRange overlaps in the hiF-T to 5-day and 24-day plus and minus conditions. The comparison perspective between timepoints, start to end, can be switched manually to compare end to start, as well as across any time point and across histone modifications; the comparison does not have to be sequential. For this visualization and analysis, the plots were conducted sequentially, and we manually updated the code in each histone modification for consistency and analysis. Utilizing ggplot2, the list of conditions were combined and the genes with the highest change from the previous condition were highlighted and we plotted those gene names for visualization and further correlation. Integrative Genomics Viewer (IGV) can also be utilized for additional visualizations and comparisons.

RESULTS

Heatmaps of active chromatin modification depict most ChIP-seq reads

To visualize and compare histone modifications and epigenetic state across time point/conditions during reprogramming we generated heatmaps centered +/- 3000 base pairs around all TSS. The profiled histone modifications showed predicted spatial density distributions over the TSS. The H3K4 modifications associated with activation displayed interesting results. H3K4me3 heatmap, associated with transcriptional initiation, had the highest centered average over the TSS as seen in Figure 7. The H3K4me1, associated with *cis*-regulation and open chromatin were lowest and mean tag count lowered over the center of TSS, while the H3K4me2 displayed similarities to the tri-methylation in APPENDIX A: Supplemental Figures Figure Set 1 and Figure Set 2. H3K27ac, also associated with open chromatin and activation, in Figure Set 4 of Supplemental Figures had similar characteristics as the H3K4 modifications, however, there were lower counts around the

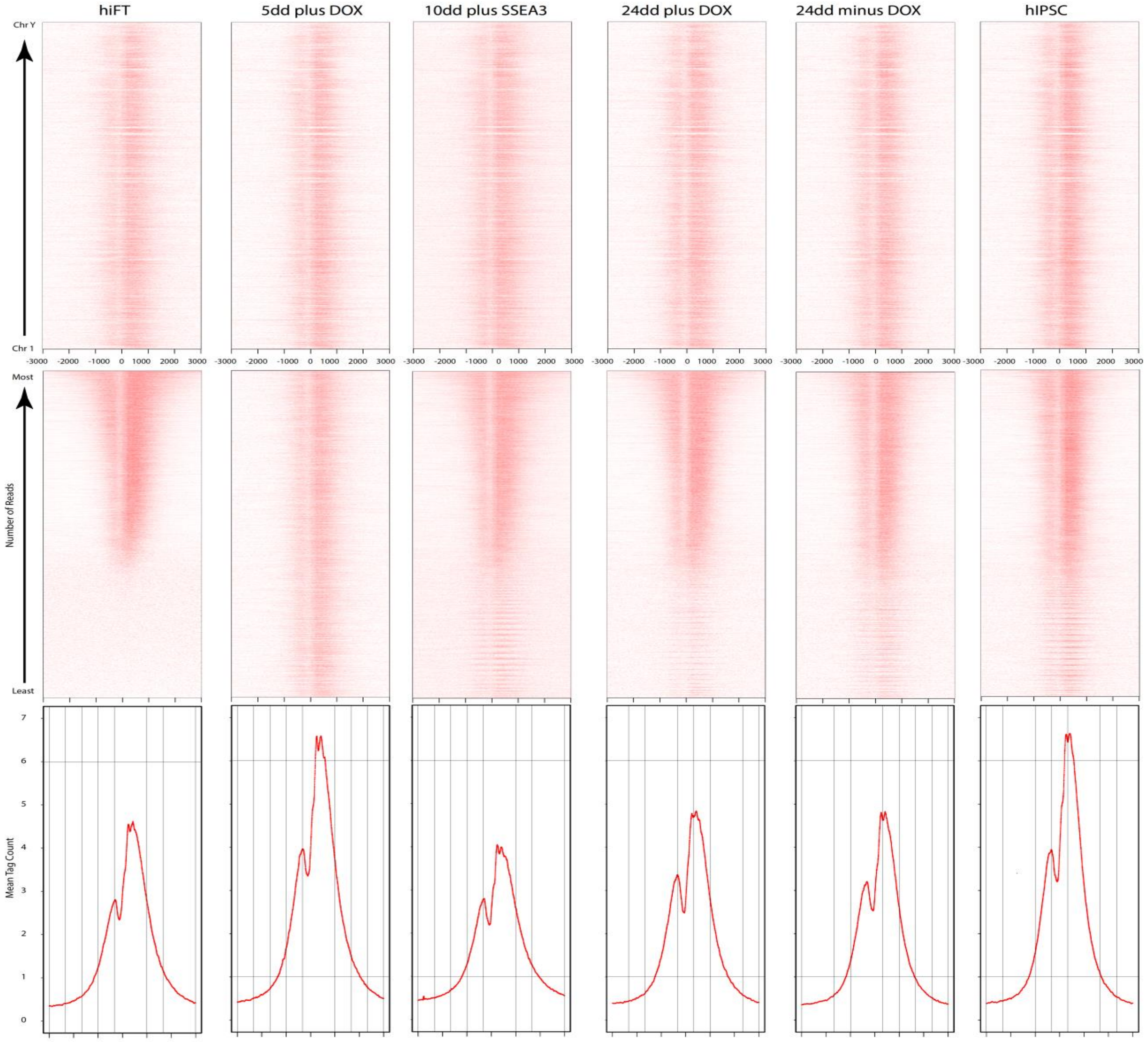


Figure 7: H3K4me3 histone modification heatmap centered +/- 3000 basepairs from the Transcription Start Sites of all genes on each of the X-axis. From left to right are the conditions and time points for the depicted histone modification, the first six are ordered based on chromosome. The middle six are sorted based upon the initial hiF-T sorted from least reads to the most. The bottom six are the mean tag count +/- 3000 base pairs from the TSS for the corresponding condition. The 5dd condition is most like the hIPSC, the one with the highest mean tag count and most of the tags are on the downstream portion of the TSS.

center of the TSS. How a cell defines its cell state and morphology is largely dependent on the epigenetics of the cell. The epigenetics are largely dependent on the histone

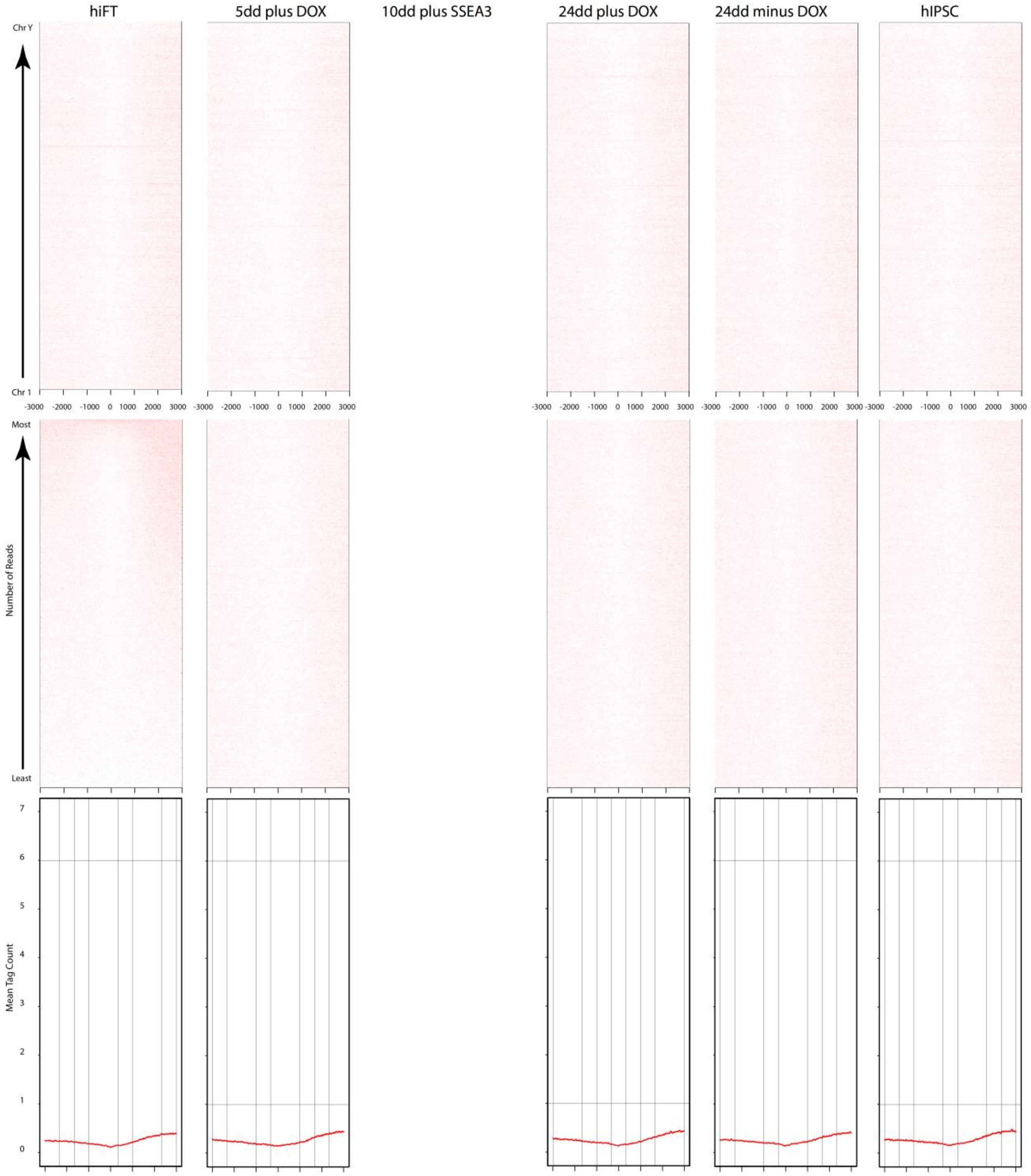


Figure 8: H3K36me3 histone modification heatmap centered +/- 3000 basepairs from the Transcription Start Sites of all genes on each of the X-axis. Day-10 ChIP-seq data was not available from Cacchiarelli, so 3rd column is missing. Same layout as Figure 7, conditions from left to right, sorted based on chromosome on the top and sorted based on low to high reads of the initial hiF-T in the center row. Very low mean tag count and sparse distribution of read counts.

modifications of the chromatin which are dependent on the transcription factors currently expressed within the cell, an interdependent process. The histone modifications that activate and poise the DNA for replication are more prominent during reprogramming. The repression H3K27me3, in Figure set 5 of Appendix A, had consistency through each time point, slowly raising the mean tag count over the TSS, and the highest repression in the hIPSC. The transcriptional elongation modification, H3K36me3, had the lowest overall distribution as well as mean tag count over the TSS centers, as seen in Figure 8, hardly any histone change occurred during the time course. More elongation means more DNA to RNA conversion, the less the elongation the more silenced the genome; with low H3K36me3 binding, reprogramming has more of an active epigenetic state.

Custom Motif Binding List similar mean tag counts

We picked genes and associated genes of TEAD1, YAP/TAZ, SRF, Adhesome and late embryogenesis genes because we know they play some role in reacting to a cells environment as well as determining cell morphology. For each custom motif binding and related gene list, each had a similar mean tag read count across each time point and across each modification. A wider distribution of hits for ChIP-seq data also exists when sorted by number of reads as compared to sequentially over the genome. Visually, each heatmap for the custom motif binding lists, whether sorted by read count or by location in the genome had similar spread-out tag read counts. For the Adhesome list of genes, the Mean tag count seems to be consistent at about 0.3 throughout reprogramming, the 10 day does have a spike for H3K27me3 and H3K4me1. Two sections or bands of the H3K27me3 do appear to increase in read count as reprogramming progresses. The SRF promoters heatmaps are consistent throughout each time point condition. H3K27me3 again has a band that becomes more prominent throughout reprogramming. TEAD1 gene and its associates had relatively very similar heatmaps and mean read tag counts. The 10-day condition also appeared to have spacious histone tags throughout with consistent mean tag counts.

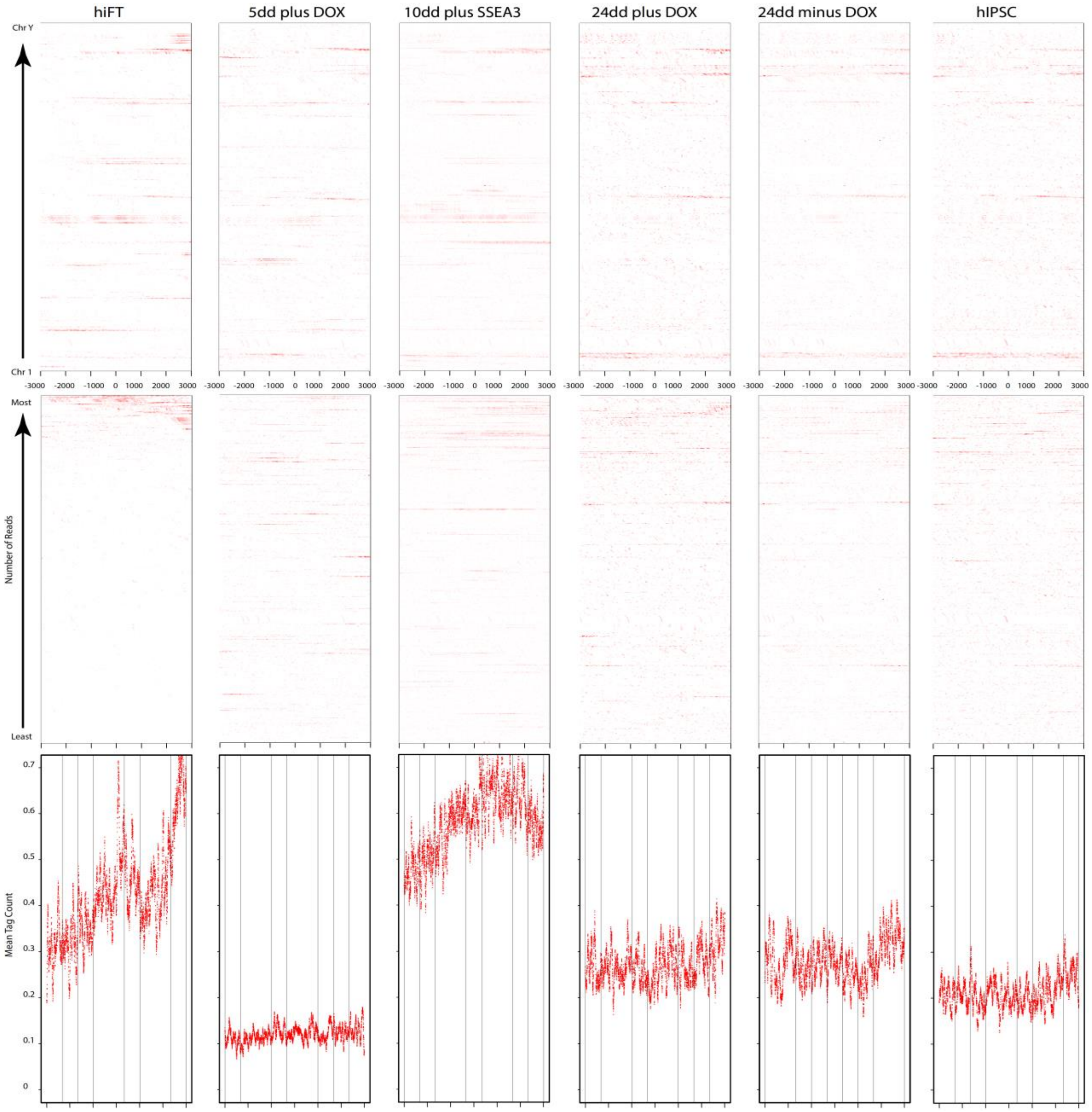


Figure 9: H3K27ac histone modification heatmap with YAP/TAZ transcription regulators and their targets centered +/- 3000 basepairs from the Transcription Start Sites of those promoters each of the X-axis.. Same layout as Figure 7, conditions from left to right, sorted based on chromosome on the top and sorted based on low to high reads of the initial hiF-T in the center row. The bottom six have a different y-axis range to account for less promoters reads. 5dd condition is again most similar to the hIPSC, with low tag counts and similar clustering.

YAP/TAZ insignificance during reprogramming

To identify if YAP/TAZ genes and their related downstream genes had a significant change in their epigenetic state during reprogramming we compared the histone modifications for each condition and time point. The YAP/TAZ had comparatively similar mean tag counts in the course of reprogramming. For H3K27me3, the mean tag count increased slightly and the heatmap for hIPSC clustered in certain areas, while none of the other conditions had any clustering. As seen in Figure 9, H3K27ac had the most drastic change across the heatmaps and mean read tag counts. The 5-day condition heatmap for each separate histone modification indicated it to be most similar to the hIPSC. Overall, YAP/TAZ genes and downstream regulators had consistent mean tag count across each histone modification seen in Figure 10. Consistent mean tag count signifies that YAP/TAZ genes do not play a substantial role in the process of reprogramming.

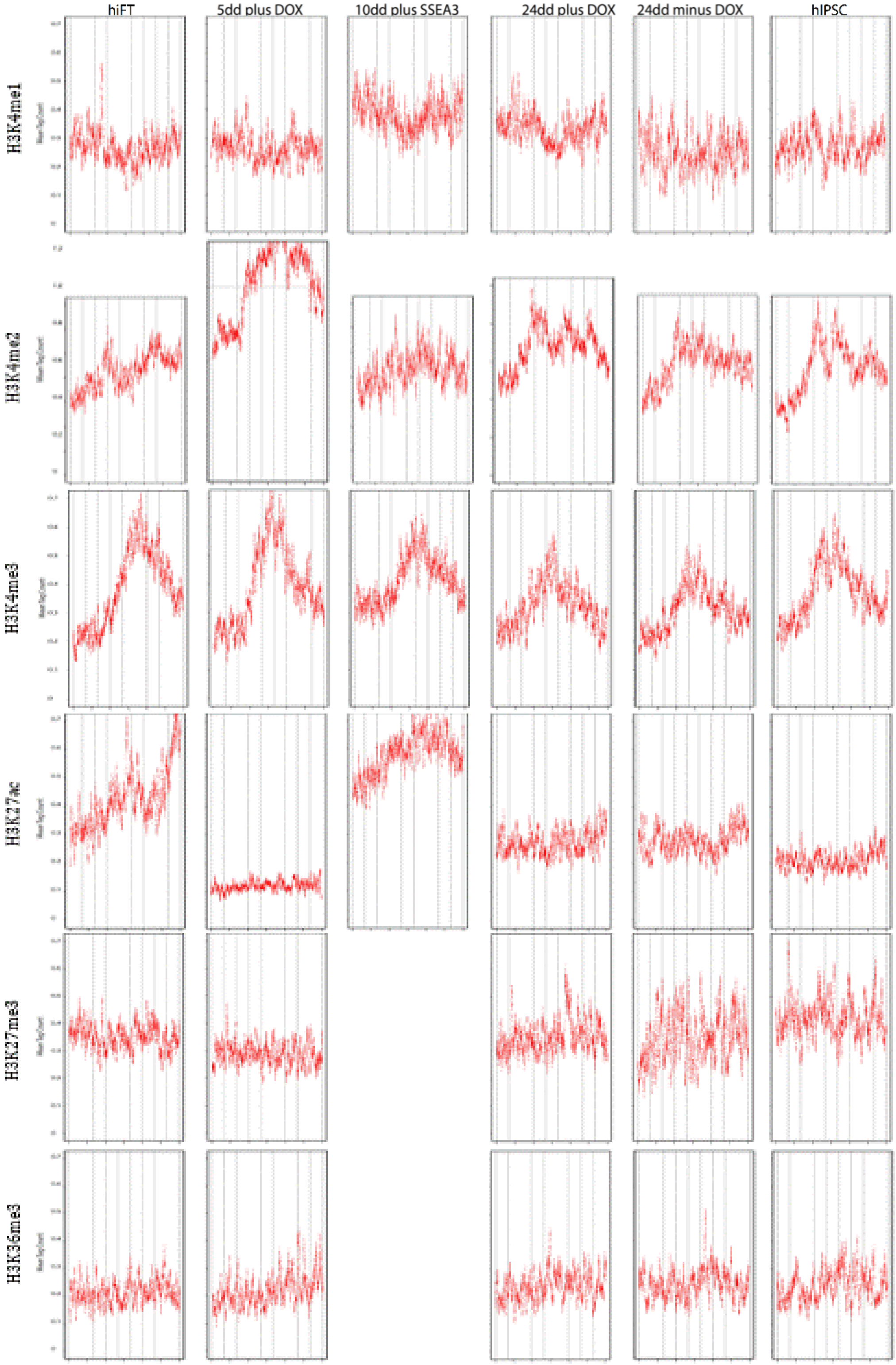


Figure 10: YAP/TAZ mean ChIP tag counts of all histone modifications. Custom gene list generation followed by mean tag count plot. Each plot is centered +/- 3000 basepairs around the TSS of YAP/TAZ genes and their downstream effectors on each of the X-axis. All plots y-axis goes to 0.7, besides H3K4me2 (row 2), the top is 1.2 read mean tag count, only condition with a higher increase of mean tag count due to merging of data replicates. Column 3 is missing in row 5 and 6 because the data was not available from Cacchiarelli.

Late Embryogenesis gene epigenetic state similar to hIPSCs

In order to analyze a potential barrier in reprogramming we identified a set of genes provided by Cacchiarelli's lab and produced heatmaps on for each histone modification. A transient wave of gene expression at day-10 in Cacchiarelli's experiments are genes

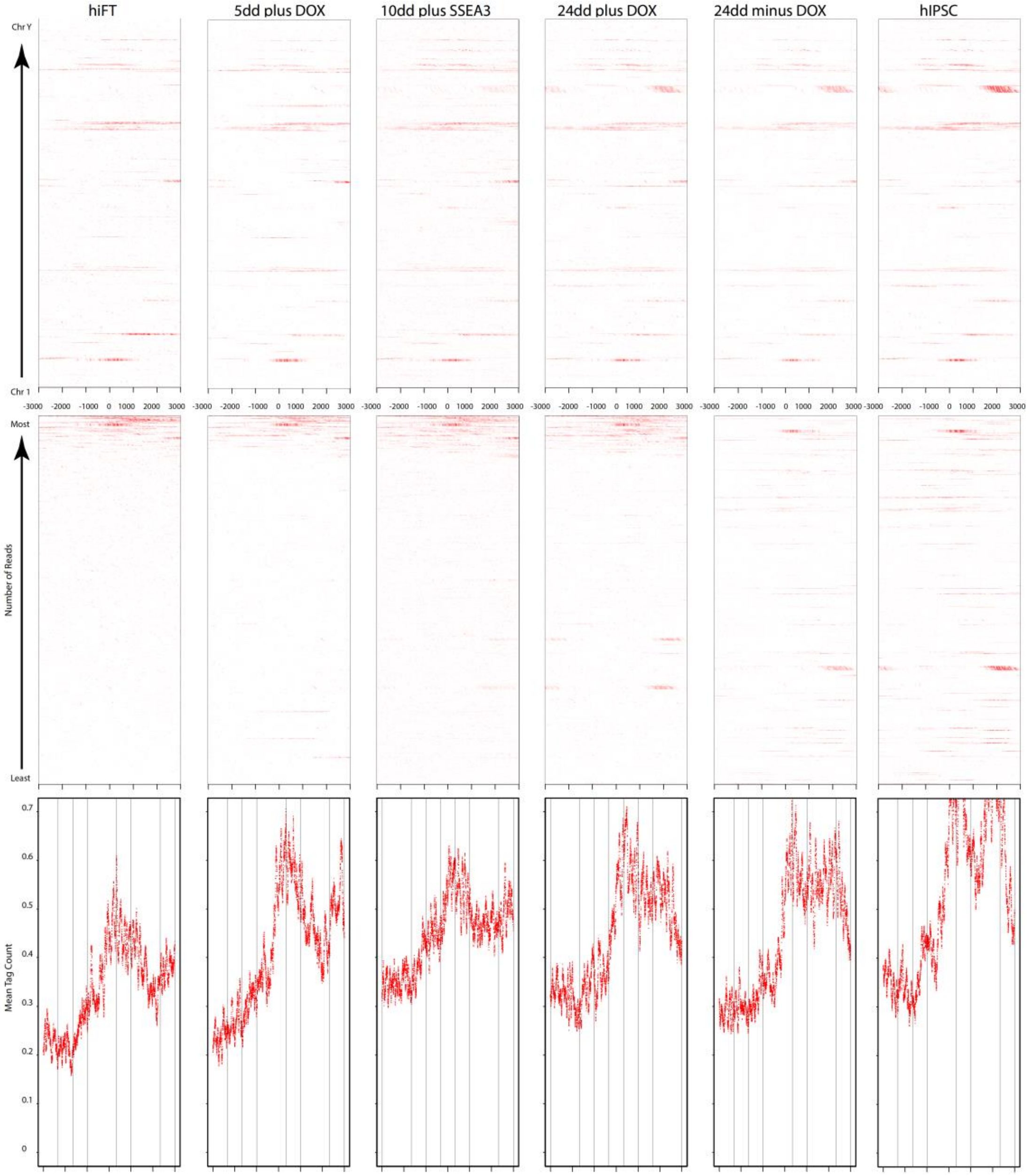


Figure 11: H3K4me3 histone modification of Cacchiarelli's late embryogenesis genes. Heatmap with those genes start transcription start sites centered +/- 3000 basepairs of those gene binding motifs on each of the X-axis. Same layout as Figure 7, conditions from left to right, sorted based on chromosome on the top and sorted based on low to high reads in the center row.

typically expressed during late embryogenesis and body patterning[4]. We utilized the custom set of genes provided by their supplemental data to observe the expression for each set of histone modifications. The clustering over the TSS of the custom set of genes stayed

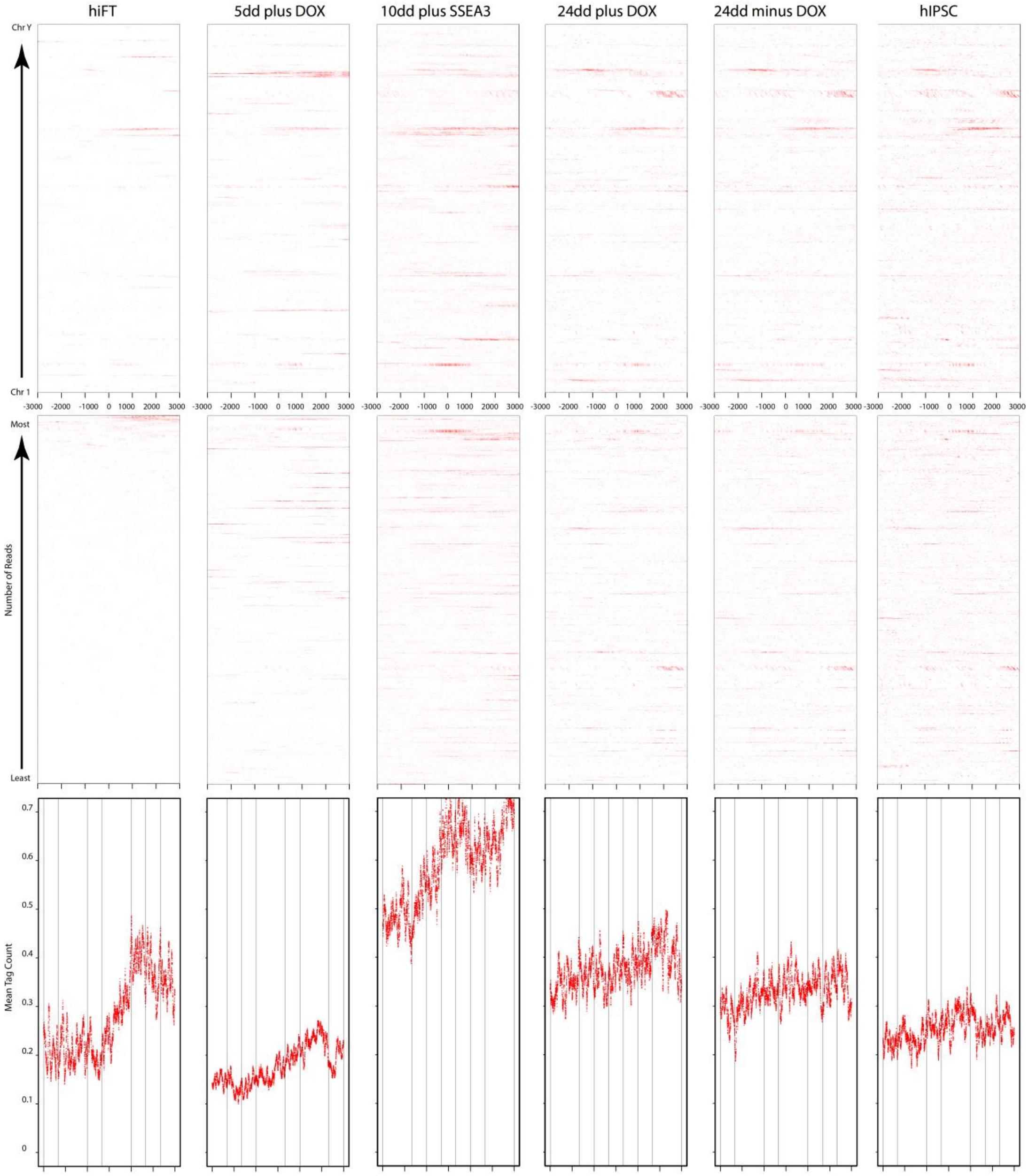


Figure 12: H3K27ac histone modification of Cacchiarelli's late embryogenesis genes. Heatmap with those genes start transcription start sites centered +/- 3000 basepairs of those gene binding motifs on each of the X-axis. Same layout as Figure 7, conditions from left to right, sorted based on chromosome on the top and sorted based on low to high reads in the center row.

relatively similar over each histone modification. H3K4me3 comparatively had some late increase in enrichment reads over the transcription sites with the largest increase at day-24 and hIPSC, also downstream of the center point of the TSS. The cluster of genes in the center row (sorted by hiF-T) one-third up the heatmap of day-24 increase in number of counts and are very prominent in the hIPSC. Without DOX and hIPSC conditions had the most similar heatmaps and thus their epigenetic state similarities as well. For each heatmap set for Cacchiarelli's Late Embryogenesis Genes depicted in the Supplemental Appendix, the sorted by reads versus not sorted were interestingly analogous, specifically the 24-day condition and the hIPSC condition. Clustering of H3K27ac histone modifications reads in Figure 12 were downstream of the genome in hiF-T and then shifted to an even distribution in hIPSC. The lowest read tag count precedes an abrupt increase in read tag counts in day-10 of the gene set. Areas and locations of the chromatin are most alike in day-24 conditions and the hIPSC, conversely hiF-T, day-5 and day-10 had very little similarities.

Significant genes with large peak enrichment changes

In order to visualize and draw tangible results from the changes between each heatmap, the peak enrichment plots provided additional insight. Utilizing the robust peak enrichment data provided by MACS2 to identify a drastic change from one condition to the next will further narrow the scope of genomic activity. That drastic change represents a histone modification binding site that either became more methylated, which then activated or repressed the DNA depending on the histone. By comparing the overlaps and defining the gene associated with the genome sequence we can identify a gene that has its activity changed during reprogramming. We can then create future experiments that knockdown or amplify the gene and see if the result improves reprogramming efficiency.

The hiF-T conditions across each histone modification had a significant change in peak enrichment fold change compared to the day-5. As known, each histone modification affects a multitude of varying sets of genes across the time points depending on the cell state and function. H3K4me2, H3K4me3, and H3K427ac, associated with activation, have significant activity on day-10 of reprogramming. Different comparisons can be utilized, for purposes of our analysis we compared the data across time points to see how the peak data changed over time to see which genes are affected the most. ChIP data can also be compared between individual time points, for example, hiF-T could be directly compared to hIPSC to see which ChIP peaks changed the most. Different histone modifications can also be compared, a day-5 H3Kme3 can be compared to an H3K27ac day-5. Our interests were the temporal changes during the process of reprogramming. Figure 13 illustrates the peak enrichment that occurs from one timepoint to the next. If peak enrichment fold exceeded 4, the overlap between the two conditions was identified and the gene name was generated on the plot for visualization. Multiple genes can populate because the DNA may code for multiple transcription factors and their sequences could overlap between the two conditions.

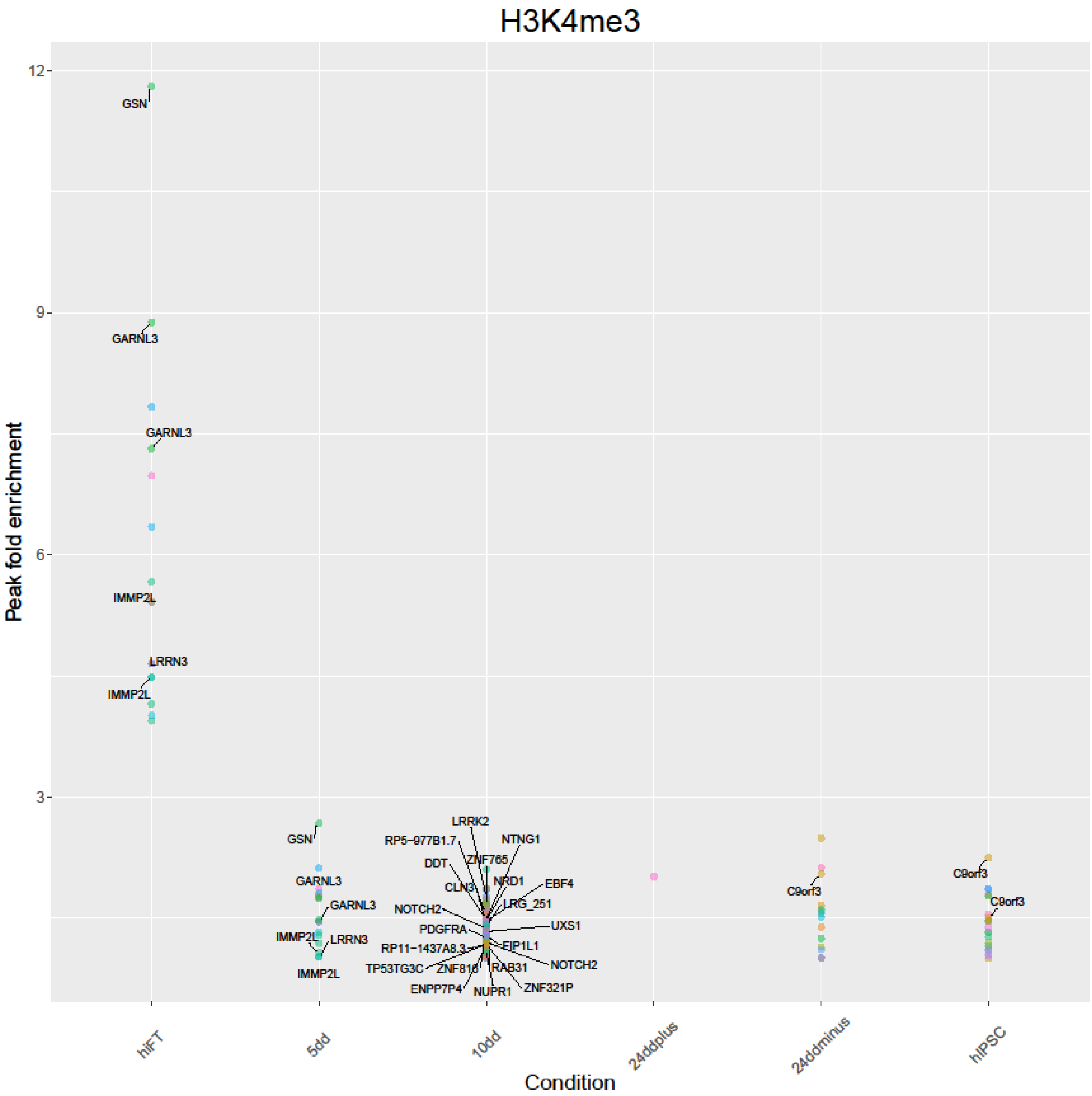


Figure 13: Scatter plot created from RStudios ggplot2 of the processed and compared peak data across conditions for H3K4me3. Peak fold enrichment on the y-axis from peak calling for each condition on the x-axis. Each gene listed has a fold change greater than 4, hiF-T has a high peak enrichment that changes drastically once reprogramming begins. Fewer changes occur towards the end of reprogramming.

H3K27me3 had hardly any significant overlaps or changes throughout reprogramming, however, the gene PCSK5 has a high enrichment for hiF-T at day-0. Lower but significant enrichment at day-5 and finally a low enrichment for the hIPSC condition depicted in Figure 14. Several additional genes, CTSV, NTRK2, CACNA1C (LRG_334) all appear for the hIPSC condition, their repression could be needed in order to maintain stem cell pluripotency.

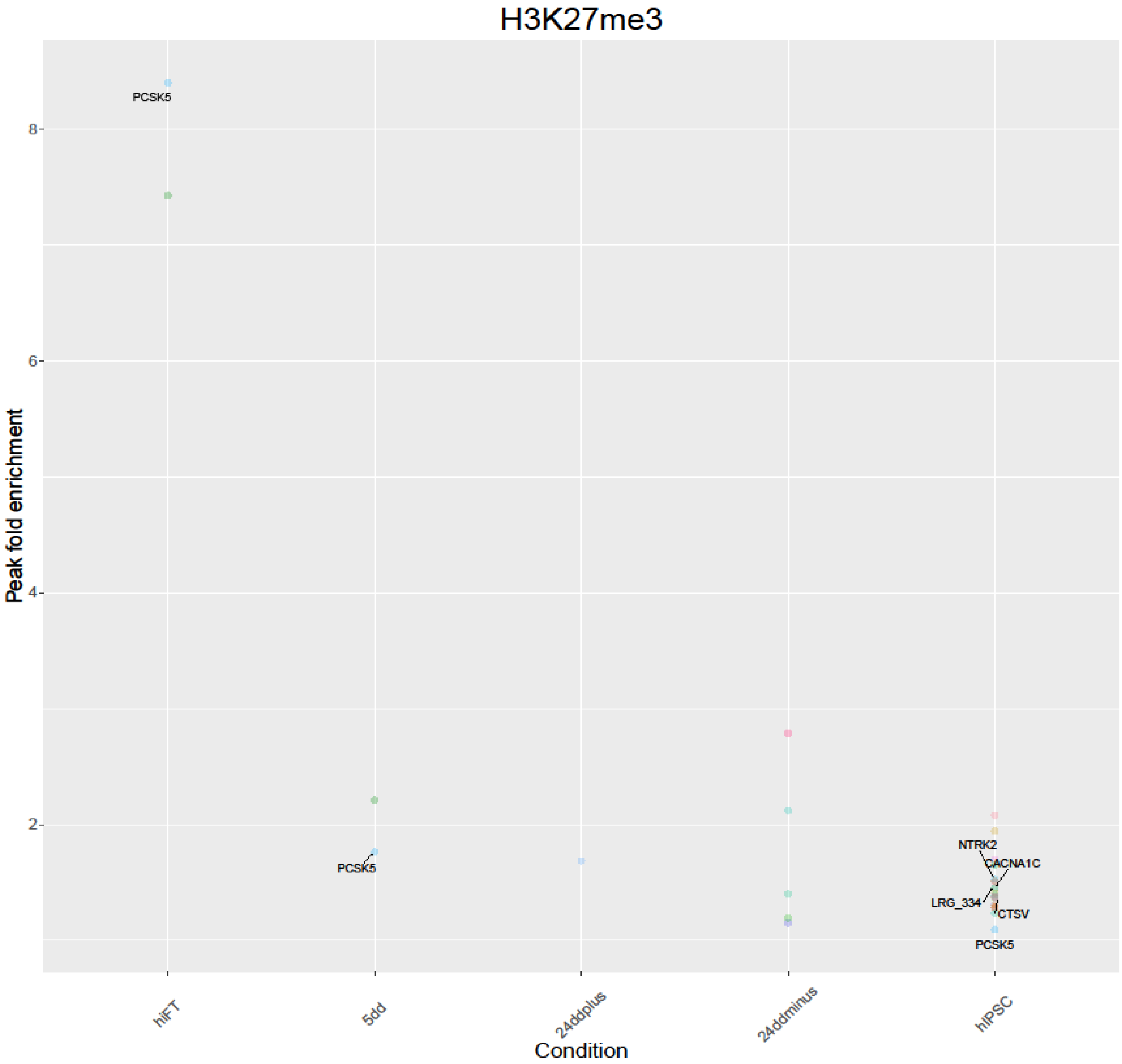


Figure 14: Scatter plot created from RStudios ggplot2 of the processed and compared peak data across conditions. Peak fold enrichment on the y-axis from peak calling for each condition on the x-axis. Each gene listed has a fold change greater than 4. hiF-T has a high peak enrichment that changes drastically once reprogramming begins. Fewer changes occur towards the end of reprogramming.

Throughout each histone modification for hIPSC, whether an activator or repressor, we observed the least amount of peak enrichment. By removal of DOX from day-24 per Cacchiarelli's experiments, an increase in peak enrichment across each modification occurs.

Across each histone modification heatmap figure sets, whether a custom set of genes or original ChIP-seq data, at day-5 a definitive change occurred. Additionally, some of the day-5 conditions location of chromatin read data and mean tag count reflected similarly to the hIPSC conditions. A few of the gene sets had low read counts and peak data across each histone modification as they most likely do not play a role in reprogramming. A myriad of diverse approaches to analyze and interpret the information exist and the ChIP-Seq data engendered by Cacchiarelli is one of many labs that generate such data. Their data does provide advantageous insight into the phenomena of reprogramming. Through visualization, correlation and comparison conclusions can be drawn based on the changes between heatmaps and conditions to determine the mechanics of epigenetic control of DNA through histone methylation over time during reprogramming.

DISCUSSION

The dawn of the digital age opened multiple doors for industries, over the years Bioengineering and Bioinformatics benefit through the ability to analyze and develop mass amounts of data. However, with so much data, only certain aspects can be derived based on the desire of the specific researcher's purpose, leaving a majority of that data unanalyzed and discoveries left unknown and hidden. ChIP-seq data has proven to help bring understanding one step closer to breaking through the unknown and establishing a technique to comprehend epigenetics. Through analysis a significance difference, as expected, in the ChIP-Seq data from the hiF-T as compared to the 5-day time point for each histone modification exists due to the introduction of OKMS reprogramming factors. By introducing OKMS factors, the DNAs immediate epigenetic state changes drastically to account for the introduction of transcription factors. Histones undergo modifications unnatural to the current cell morphology, thus a change in ChIP-seq data. Interestingly, the 5-day timepoint in some cases such as H3K4me3, looked epigenetically closer to the hIPSCs heatmaps and mean tag counts than the 24-day time points. We predict this could be caused by the influx of hIPSC-like factors that initiate reprogramming before housekeeping genes attempt to regulate and keep the cell in their current state. The reprogramming factors cause an unnatural disturbance causing the cell to attempt to imitate an epigenetic state as if it was a stem cell. By examining each customized promoter and transcription factor heatmap, we saw some sections overlap or cluster similarly with another heatmap during reprogramming and this relation exposed some key players, yet to be identified, existing throughout reprogramming. Further analysis to identify specific genes of those key players could potentially be identified by the peak enrichment analysis gene plot and experimental testing with knockdowns of those genes. Similarly, some areas have no enrichment data in earlier time points; however, an increase of tag counts in certain areas appear at day-24 and in hIPSCs or *vice-versa*, indicating a high amount of tag counts that decrease. Depending on the histone modification, the epigenetic state either silences or enhances through time. These sections of DNA clearly have an importance in the generation of stem cells, epigenetic control and reprogramming. Whether those genes have a positive or negative influence, or if they are downstream affects from other genes at play requires further analysis. The influences genes have upon one another can also help comprehend mechanisms that occur during reprogramming.

Discovering which genes are enriched during reprogramming based on ChIP-seq peak data helps to understand the role a gene plays during the epigenetic process. During analysis, for the H3K27me3 modification, no overlaps were observed between the hiF-T condition and day 5 condition as well as between the DOX plus or minus day 24 conditions. Between those days and conditions, the heatmaps also displayed low hits over the TSS and slowly increased to some general peak for the hIPSCs. In order to observe any gene change, the sensitivity in our code needed to be decreased in order to see any overlap of the conditions. Two reasons could create the significant differences. One, there may be no significant change in ChIP-seq peak data, the conditions could have a very similar epigenetic landscape for the H3K27me3, and the modification does not change significantly during reprogramming. Conversely, the Genomic Ranges that do change may have a significant change, just not in the same range as the previous condition. PCKS5 was one gene of significance, between the hiF-T and day-5, which encodes proteases, is widely expressed as one of the amino acids that cleaves substrates as well as relates to the nerve growth factor process[37]. The decrease in peak data could account for the unnecessary or reduced need for downstream post-translational proteins or complex network of neurons in stem cells. Similar instances across each modification were observed in the comparison analysis, one or more genes either had an increase or decrease in peak enrichment; these genes or sequences could be modified to enhance reprogramming and understand epigenetics.

Interestingly, each custom promoter heatmap maintained a consistent decrease, some very slight, mean read peak tag count over time, however for the H3K27me3 modification, there was a slight increase. Repression of genes that determine cell fate and morphology could become more prominent to keep the stem cells from differentiating. Perhaps heightening the H3K27me3 modification numbers within the cells could increase the speed and efficiency of generating hIPSCs. Furthermore, over each custom promoter heatmap and mean read tag count, the 5-day reflected ChIP data most like the desired hIPSC. The 5-day condition could be the initial barrier of homeostasis balance between repression and activation before further reprogramming factors push the cell to change their epigenetic states closer to induced stem cells. And once they reach the induced pluripotency state, their homeostasis reaches another balance. As stated earlier, the hIPSC displayed a down-regulated peak enrichment across all histone modifications, indicating an epigenetically more silent cell state as a stem cell. Uncovering those specific genes that drive the balance could potentially skip the time points in between and go directly to the epigenetic state of the hIPSC.

The Late Embryogenesis gene set provided by the Cacchiarelli Harvard research group displays some interesting results. The increase in the H3K27ac read tag counts further promotes the existence of an epigenetic barrier at day-10. Once the stem cell epigenetic cell state reaches day-10, the increase of the H3K27ac activator indicates an increase of late embryogenesis transcription factor production, followed by a sharp decrease as the barrier is passed and finally decreases to a low average in hIPSC. Remarkably, the H3K4me3 histone modification shows a slow increase in mean tag count. As an activator, there are some genes in late embryogenesis that play a role in stem cell epigenetics. HOX genes, that play a role, most likely help dictate the epigenetics of the stem cell reprogramming. Because of the variability and significance of the gene list a future experiment enhancing or knocking down Late Embyogenesis genes could show some interesting results.

As seen in the H3K4me2 5-day condition, using multiple replicates and merging each file may provide better insight to the number, frequency and accuracy of ChIP data. This was the only data provided that had a merged replicate which also had a relatively high increase in tag read count data, providing more information to utilize for analysis across the custom promoter ranges and more genes of interest that had a peak enrichment flux at day-5. In addition, because of the abundancy of methods to pre-process BAM/SAM and BED files into peak data, certain techniques may overlook aspects of raw data that may be desired. Differing software for peak analysis techniques should be conducted, compared and then combined for an average for follow on heatmap generation and plotting. The complexity of experiments, data generation, analysis and interpretation of the results create variability, so an average or consistency would augment results.

Actively comparing histone modifications and conditions, gene expression and peak data can reveal distinct genes that may significantly affect reprogramming. Knockdowns or enhancements of those genes can potentially speed up the process of reprogramming as well as create a better understanding of epigenetics. Further studies can be generated with other GEO accession repositories as well as additional histone modifications. Moreover, diverse custom ranges of promoters can be created and then utilized to discover the changes in their expression with desired modifications over time. As a result of our analyses and visualizations, we hope to further promote and advance gene identity through reprogramming. As a result of our coding technique to sort heatmaps by one another, a day-0 hiF-T in our case, the process can help researchers compare the broad epigenetic state in a new way between two different conditions. In addition, the novelty of generating a heatmap by building a list on the selection of a certain set of genes, associated transcription factors and binding motifs instead of all TSS will help narrow the scope of understanding the cells behavior epigenetically. Furthermore, with the ability of our code to identify specific genes which have a significant peak enrichment from one condition to next, can help focus which genes to target next in the experimental process.

Modern computing does open doors; R Studio is a GNU project, or free software, and as an open science platform it can be used by anyone. Because our code examined another lab's data for a general analysis, the code can be re-used for other experiments as well. And by way of identifying specific genes, experiments that knockdown or amplify expression can be developed to expedite and reduce cost of stem cell production. In addition, the specific analysis techniques we created can identify unique mechanisms that could potentially help cure genetic diseases by way of silencing or activating the DNA during development. The amount of data and the many ways for interpretation from one raw file of ChIP-seq remains overwhelming. ChIP-seq data platforms and techniques in development are still in infancy and quickly advancing, providing useful tools for analysis, but because of the complexity that exists in epigenetics, high power computing and visualizations are just beginning to tap into their potential.

REFERENCES

1. Aboyoun, P., Lawrence, M., & Pagès, H. (2019, April 16). Package 'IRanges'. https://bioconductor.riken.jp/packages/3.8/bioc/manuals/IRanges/man/IRanges.pdf

2. Belt, H., Koponen, J., Kekarainen, T., Puttonen, K., Mäkinen, P., Niskanen, H., . . . Ylä-Herttuala, S. (2018, March 14). Temporal Dynamics of Gene Expression During Endothelial Cell Differentiation From Human iPS Cells: A Comparison Study of Signaling Factors and Small Molecules. https://www.frontiersin.org/articles/10.3389/fcvm.2018.00016/full

3. Boué, Stéphanie, Ida Paramonov, María José Barrero, and Juan Carlos Izpisúa Belmonte. "Analysis of Human and Mouse Reprogramming of Somatic Cells to Induced Pluripotent Stem Cells. What Is in the Plate?" *PLOS ONE*. Public Library of Science, 17 Sept. 2010. https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0012664

4. Cacchiarelli, D., Trapnell, C., Ziller, M., Soumillon, M., Cesana, M., Karnik, R., . . . Mikkelsen, T. (2015, July 16). Integrative Analyses of Human Reprogramming Reveal Dynamic Nature of Induced Pluripotency. https://doi.org/10.1016/j.cell.2015.06.016

5. Cho, H. J., Lee, C. S., Kwon, Y. W., Paek, J. S., Lee, S. H., Hur, J., ... Kim, H. S. (2010). Induction of pluripotent stem cells from adult somatic cells by protein-based reprogramming without genetic manipulation. *Blood*. https://doi.org/10.1182/blood-2010-02-269589

6. C.H. Waddington, *The Strategy of the Genes*, George Allen & Unwin, 1957

7. David, L., & Polo, J. M. (2014). Phases of reprogramming. Stem Cell Research. https://doi.org/10.1016/j.scr.2014.03.007

8. Dupont, S., Morsut, L., Aragona, M., Enzo, E., Giulitti, S., Cordenonsi, M., ... Piccolo, S. (2011). Role of YAP/TAZ in mechanotransduction. *Nature*. https://doi.org/10.1038/nature10137

9. Guangchuang Yu, Li-Gen Wang, Qing-Yu He. ChIPseeker: an R/Bioconductor package for ChIP peak annotation, comparison and visualization. Bioinformatics 2015, 31(14):2382-2383

10. Hansen, K. (2016, May 23). GenomicRanges - Rle. https://kasperdanielhansen.github.io/genbioconductor/html/GenomicRanges_Rle.html

11. Hadley Wickham, Romain François, Lionel Henry and Kirill Müller (2018). dplyr: A Grammar of Data Manipulation. R package version 0.7.6. https://CRAN.R-project.org/package=dplyr

12. Heinz S, Benner C, Spann N, Bertolino E et al. Simple Combinations of Lineage-Determining Transcription Factors Prime cis-Regulatory Elements Required for Macrophage and B Cell Identities. Mol Cell 2010 May 28;38(4):576-589. PMID: 20513432

13. The Human Genome Project FAQ | NHGRI. (2018, November 12). https://www.genome.gov/human-genome-project/Completion-FAQ

14. Huihuang, Yan, Shulan, Tian , Slager, Susan, L. Slager and Zhifu, Sun. (2016, June 20). ChIP-seq in studying epigenetic mechanisms of disease and promoting precision medicine: progresses and future directions. https://www.futuremedicine.com/doi/full/10.2217/epi-2016-0053?url_ver=Z39.88-2003&rfr_id=ori%3Arid%3Acrossref.org&rfr_dat=cr_pub%3Dpubmed&

15. Ikeda, T., Hikichi, T., Miura, H., Shibata, H., Mitsunaga, K., Yamada, Y., . . . Masui, S. (2018, April 11). Srf destabilizes cellular identity by suppressing cell-type-specific gene expression programs. https://doi.org/10.1038/s41467-018-03748-1

16. Jiang, Mandy Shan. (2018, October 24). Hacking peak calling analysis. UCI Bioinformatics Support Group https://drive.google.com/file/d/1ijC5lseXCASbJSYQCXhZrTqDxitDQdOi/view

17. Khilji, S., Hamed, M., Chen, J., & Li, Q. (2018, July 30). Loci-specific histone acetylation profiles associated with transcriptional coactivator p300 during early myoblast differentiation. https://doi.org/10.1080/15592294.2018.1489659

18. Krueger, F., & Andrews, S. R. (2011). Bismark: A flexible aligner and methylation caller for Bisulfite-Seq applications. *Bioinformatics*. https://doi.org/10.1093/bioinformatics/btr167

19. Langmead B, Salzberg S. Fast gapped-read alignment with Bowtie 2. Nature Methods. 2012, 9:357-359.

20. Lawrence, M. Huber, W., Pagès, H., Aboyoun, P., Carlson, M., Gentleman, R., Morgan, M. T., and Carey, J.V.. Software for computing and annotating genomic ranges. . (2018, January). PLOS Computational Biology, 4(3), 2013. GenomicRanges HOWTOs. http://bioconductor.org/packages/devel/bioc/vignettes/GenomicRanges/inst/doc/GenomicRangesHOWTOs.pdf

21. Medjkane, S., Perez-Sanchez, C., Gaggioli, C., Sahai, E., & Treisman, R. (2009). Myocardin-related transcription factors and SRF are required for cytoskeletal dynamics and experimental metastasis. Nature Cell Biology. https://doi.org/10.1038/ncb1833

22. Mikkelsen, T., Hanna, J., Zhang, X., Ku, M., Wernig, M., Schorderet, P., . . . Meissner, A. (2008, May 28). Dissecting direct reprogramming through integrative genomic analysis. https://www.nature.com/articles/nature07056

23. Mikkelsen, T. S., Xu, Z., Zhang, X., Wang, L., Gimble, J. M., Lander, E. S., & Rosen, E. D. (2010). Comparative epigenomic analysis of murine and human adipogenesis. Cell. https://doi.org/10.1016/j.cell.2010.09.006

24. Muñoz, Mindy & Craske, Madeleine & Severino, Patricia & Lima, Thais & Labhart, Paul & Chammas, Roger & Velasco, Irineu & Machado, Marcel & Egan, Brian & Nakaya, Helder & Pinheiro da Silva, Fabiano. (2016). Antimicrobial peptide LL-37 participates in the transcriptional regulation of melanoma cells. Journal of Cancer. 7. 2341-2345.

10.7150/jca.16947.https://www.researchgate.net/figure/A-Average-density-plot-of-tag-distributions-across-peak-regions-Note-the-strong_fig1_311769342

25. Nashun, B., Hill, P. W., & Hajkova, P. (2015). Reprogramming of cell fate: epigenetic memory and the erasure of memories past. *The EMBO Journal*. https://doi.org/10.15252/embj.201490649

26. Ou, J., Yu, J., & Zhu, L. (2017, July 28). Integrated analysis and visualization of ChIP-seq data using ChIPpeakAnno, GeneNetworkBuilder and TrackViewer. https://www.bioconductor.org/help/course-materials/2017/BioC2017/Day2/Workshops/ChIP-seq/doc/workflow.html

27. Park, J., Son, Y., Lee, N., Lee, K., Lee, D., Song, J., . . . Min, J. (2018, June 14). DSG2 Is a Functional Cell Surface Marker for Identification and Isolation of Human Pluripotent Stem Cells. https://doi.org/10.1016/j.stemcr.2018.05.009

28. Park, P. J. (2009). ChIP-Seq: advantages and challenges of a maturing technology. Nature Reviews. Genetics, 10(10), 669. https://doi.org/10.1038/NRG2641

29. Piccolo, S., Dupont, S., & Cordenonsi, M. (2014, October 01). The Biology of YAP/TAZ: Hippo Signaling and Beyond. https://www.physiology.org/doi/full/10.1152/physrev.00005.2014

30. Qin, H., Zhao, A., Zhang, C., & Fu, X. (2016). Epigenetic Control of Reprogramming and Transdifferentiation by Histone Modifications. *Stem Cell Reviews and Reports*. https://doi.org/10.1007/s12015-016-9682-4

31. RStudio Team (2015). RStudio: Integrated Development for R. RStudio, Inc., Boston, MA URL http://www.rstudio.com/

32. Robinson, D. (2016). Fuzzyjoin: Join data frames on inexact matching. https://github.com/dgrtwo/fuzzyjoin

33. Schmidt, D., Wilson, M. D., Spyrou, C., Brown, G. D., Hadfield, J., & Odom, D. T. (2009). ChIP-seq: Using high-throughput sequencing to discover protein–DNA interactions. *Methods*, *48*(3), 240–248. https://doi.org/10.1016/J.YMETH.2009.03.001

34. Shen, L., Shao, N., Liu, X. and Nestler, E. (2014) ngs.plot: Quick mining and visualization of next-generation sequencing data by integrating genomic databases, BMC Genomics, 15, 284.

35. Shin, H., Liu, T., Duan, X., Zhang, Y., & Liu, X. (2013, March 1). Computational methodology for ChIP-seq analysis. https://doi.org/10.1007/s40484-013-0006-2

36. Sinji, C. (2016, October 16). Visualizations of ChIP-Seq data using Heatmaps. https://www.biostars.org/p/180314/

37. Stelzer G, Rosen R, Plaschkes I, Zimmerman S, Twik M, Fishilevich S, Iny Stein T, Nudel R, Lieder I, Mazor Y, Kaplan S, Dahary D, Warshawsky D, Guan - Golan Y, Kohn A, Rappaport N, Safran M, and Lancet D. The GeneCards Suite: From Gene Data Mining to Disease Genome Sequence Analysis. PCSK5. (2014). https://www.genecards.org/cgi-bin/carddisp.pl?gene=PCSK5

38. Tang, D. (2015, April 6) BED to GRanges https://davetang.org/muse/2015/02/04/bed-granges/

39. Tidball, A. M., Neely, M. D., Chamberlin, R., Aboud, A. A., Kumar, K. K., Han, B., … Bowman, A. B. (2016). Genomic instability associated with p53 knockdown in the generation of Huntington's disease human induced pluripotent stem cells. *PLoS ONE*. https://doi.org/10.1371/journal.pone.0150372

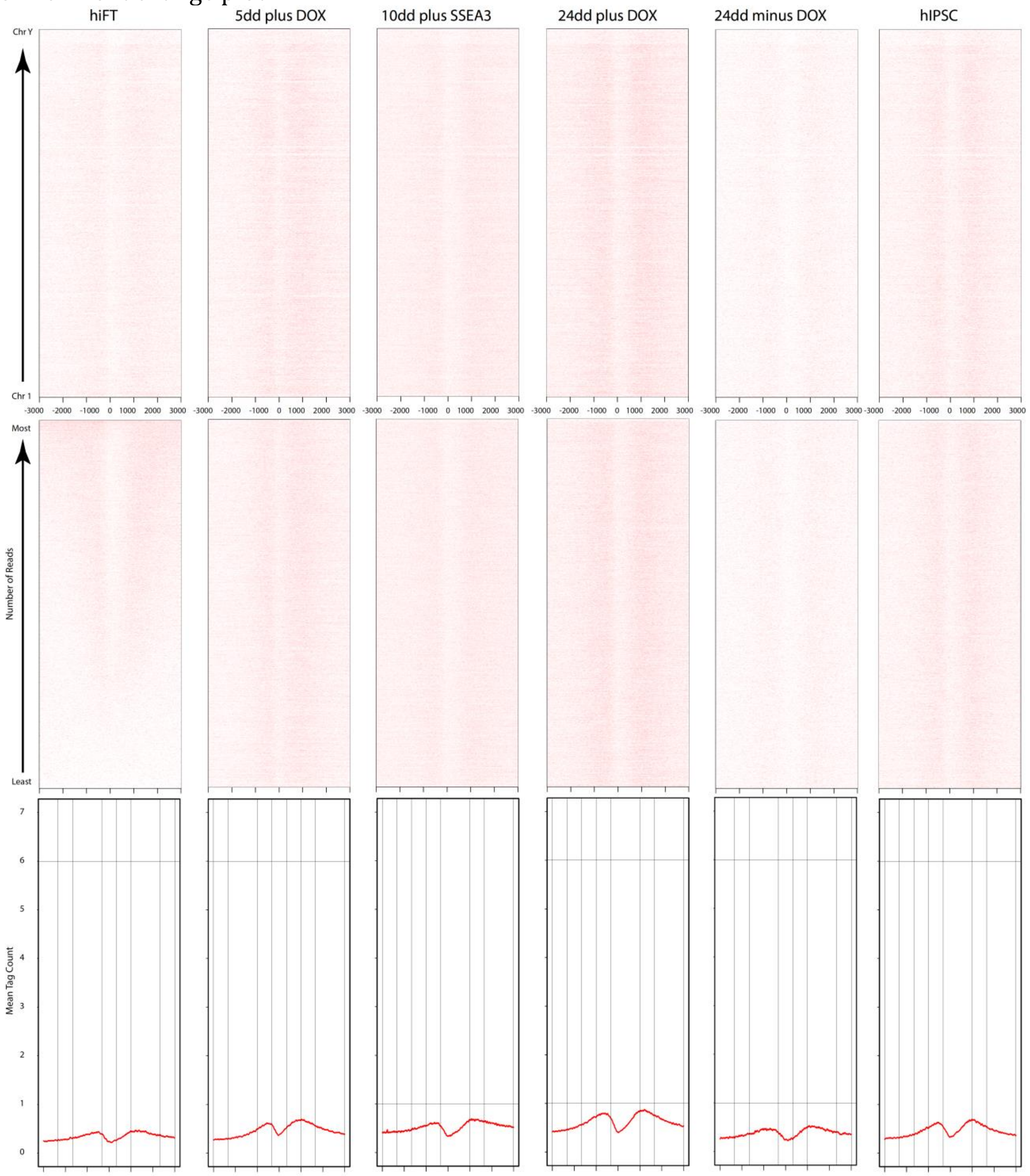
40. Vining, K., & Mooney, D. (2017, December). Mechanical forces direct stem cell behaviour in development and regeneration. https://doi.org/10.1038/nrm.2017.108

41. Wang, Y., Zhao, C., Hou, Z., Yang, Y., Bi, Y., Wang, H., . . . Gao, S. (2018, January 30). Unique molecular events during reprogramming of human somatic cells to induced pluripotent stem cells (iPSCs) at naïve state. https://doi.org/10.7554/eLife.29518.001

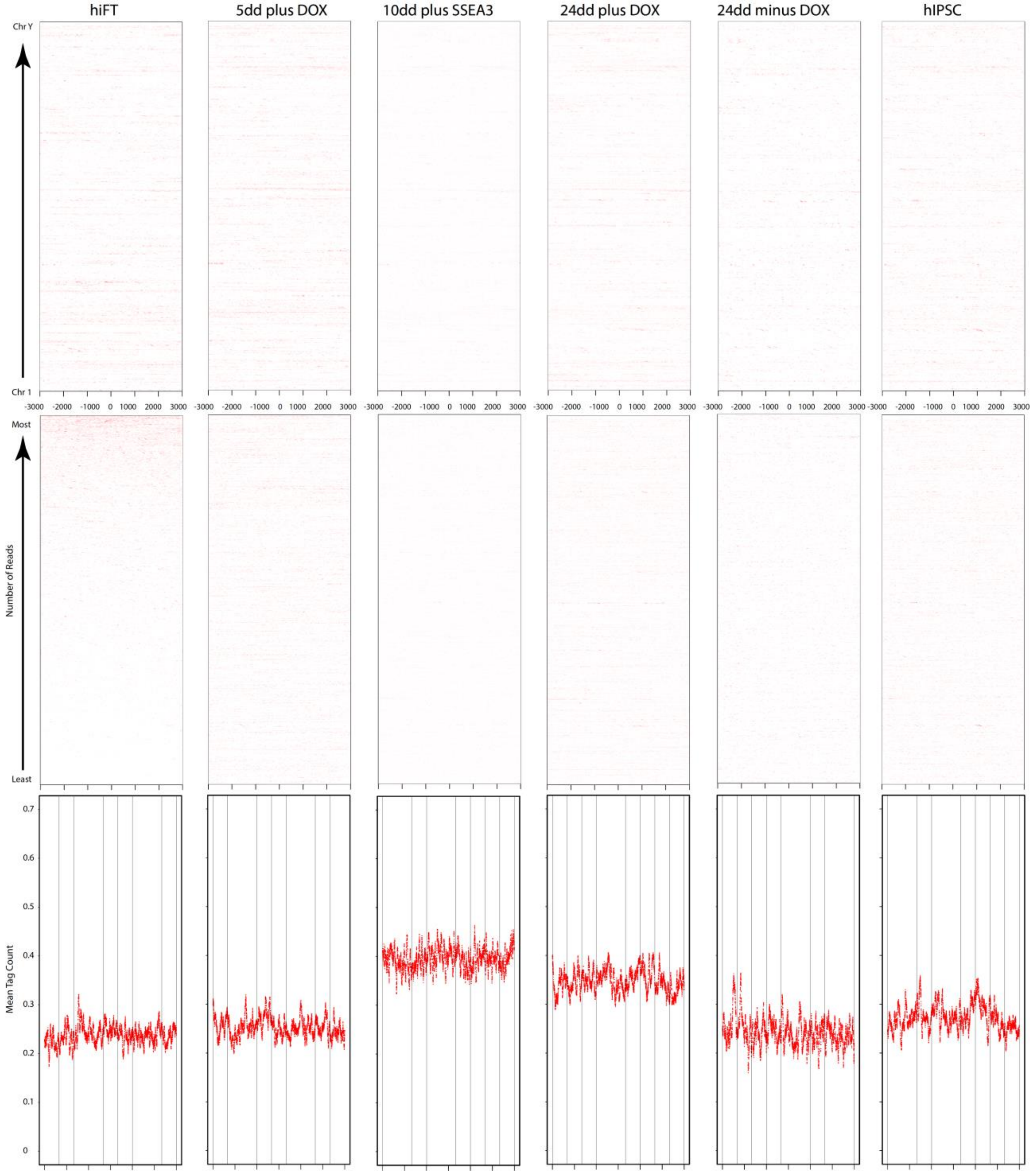
42. Zhang Y, Liu T, Meyer CA, Eeckhoute J, Johnson DS, Bernstein BE, Nusbaum C, Myers RM, Brown M, Li W, Liu XS. (2008) Model-based Analysis of ChIP-Seq (MACS), Genome Biology, 2008;9(9):R137.

APPENDIX A: Supplemental Figures

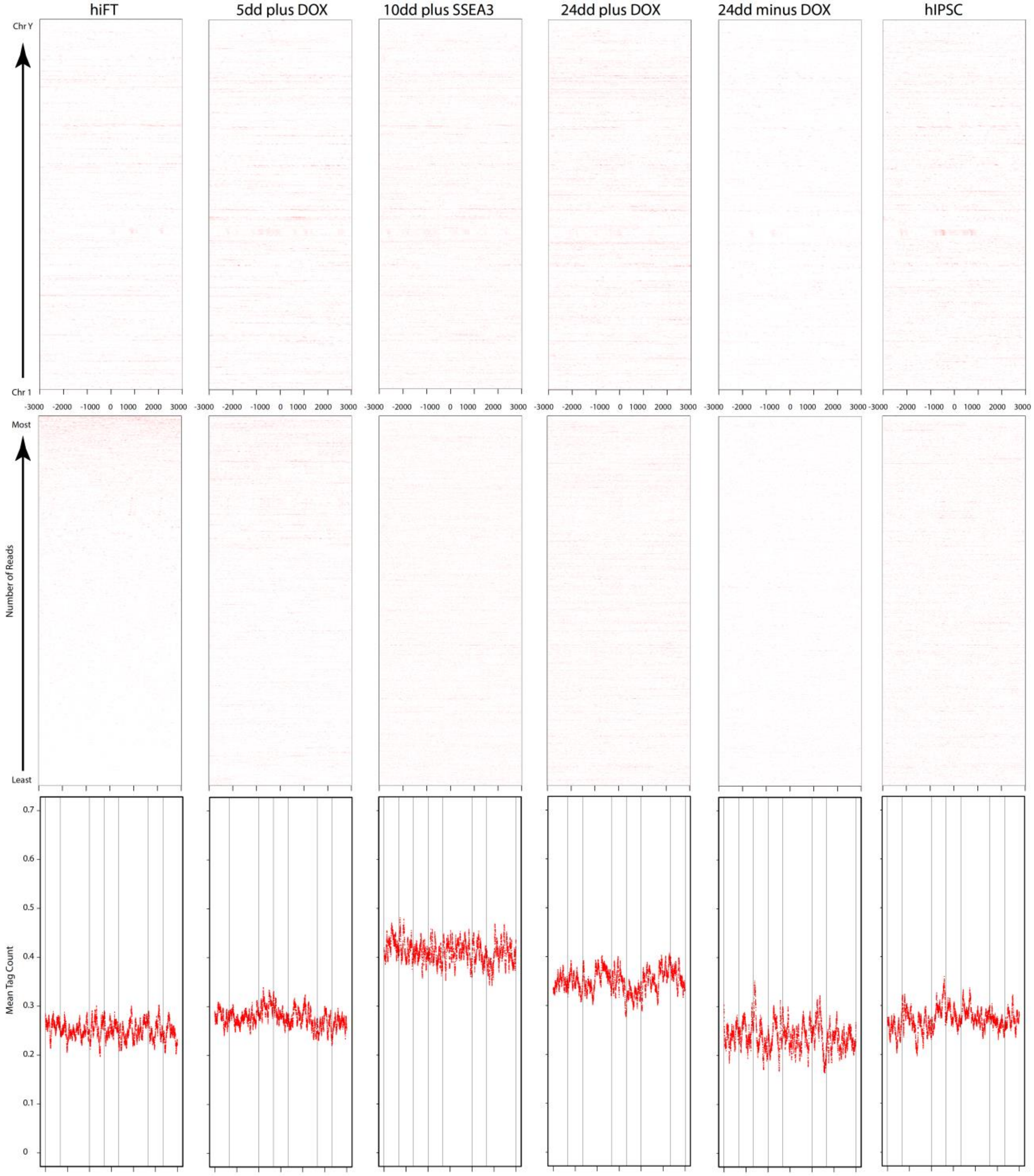
Figure Set 1: H3K4me1 histone modification, custom gene set analyses and peak enrichment change plot



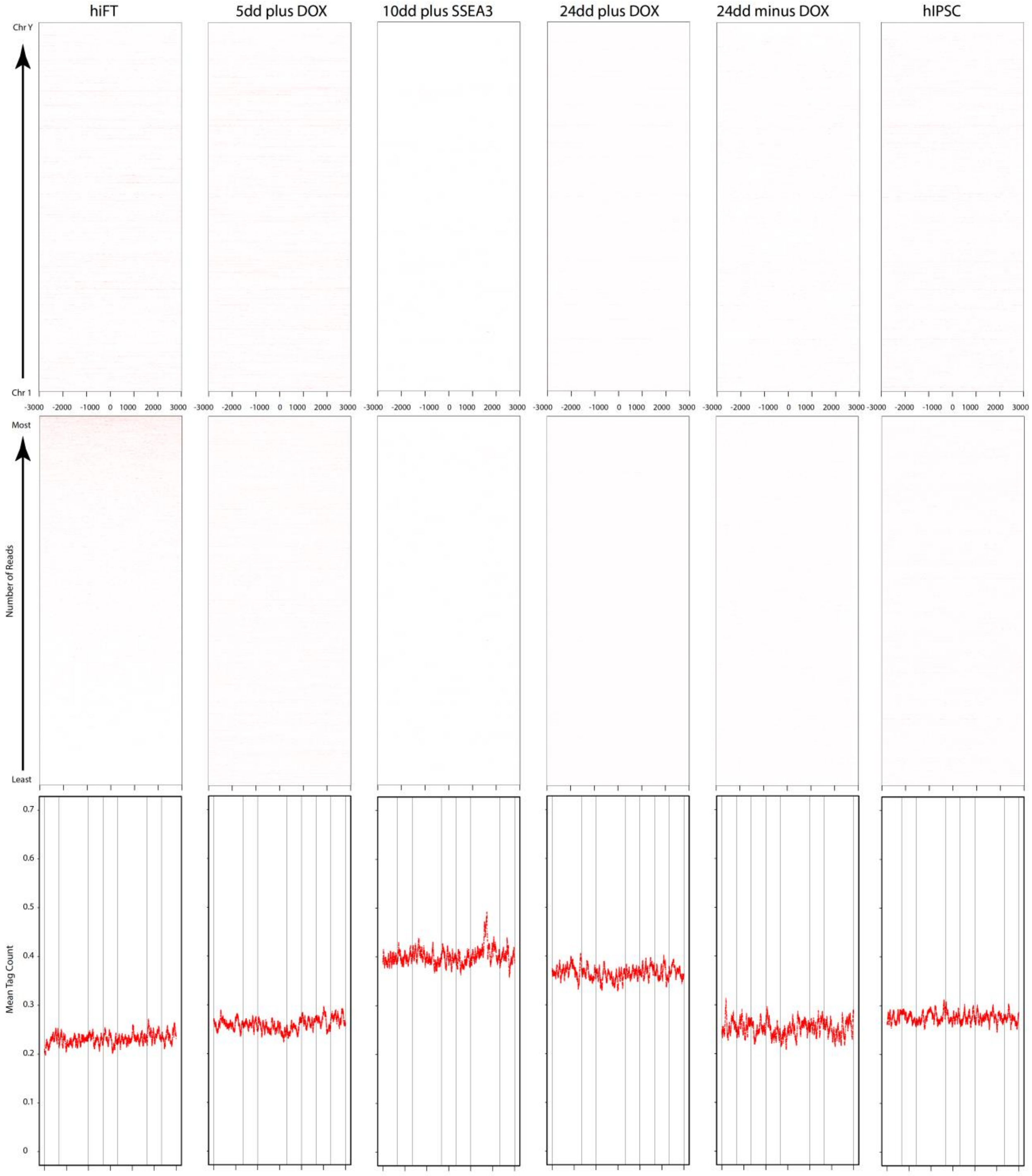
Adhesome


hiFT
5dd plus DOX
10dd plus SSEA3
24dd plus DOX
24dd minus DOX
hIPSC
Chr Y
Chr 1
-3000 -2000 -1000 0 1000 2000 3000
Most
Number of Reads
Least
Mean Tag Count
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0

SRF

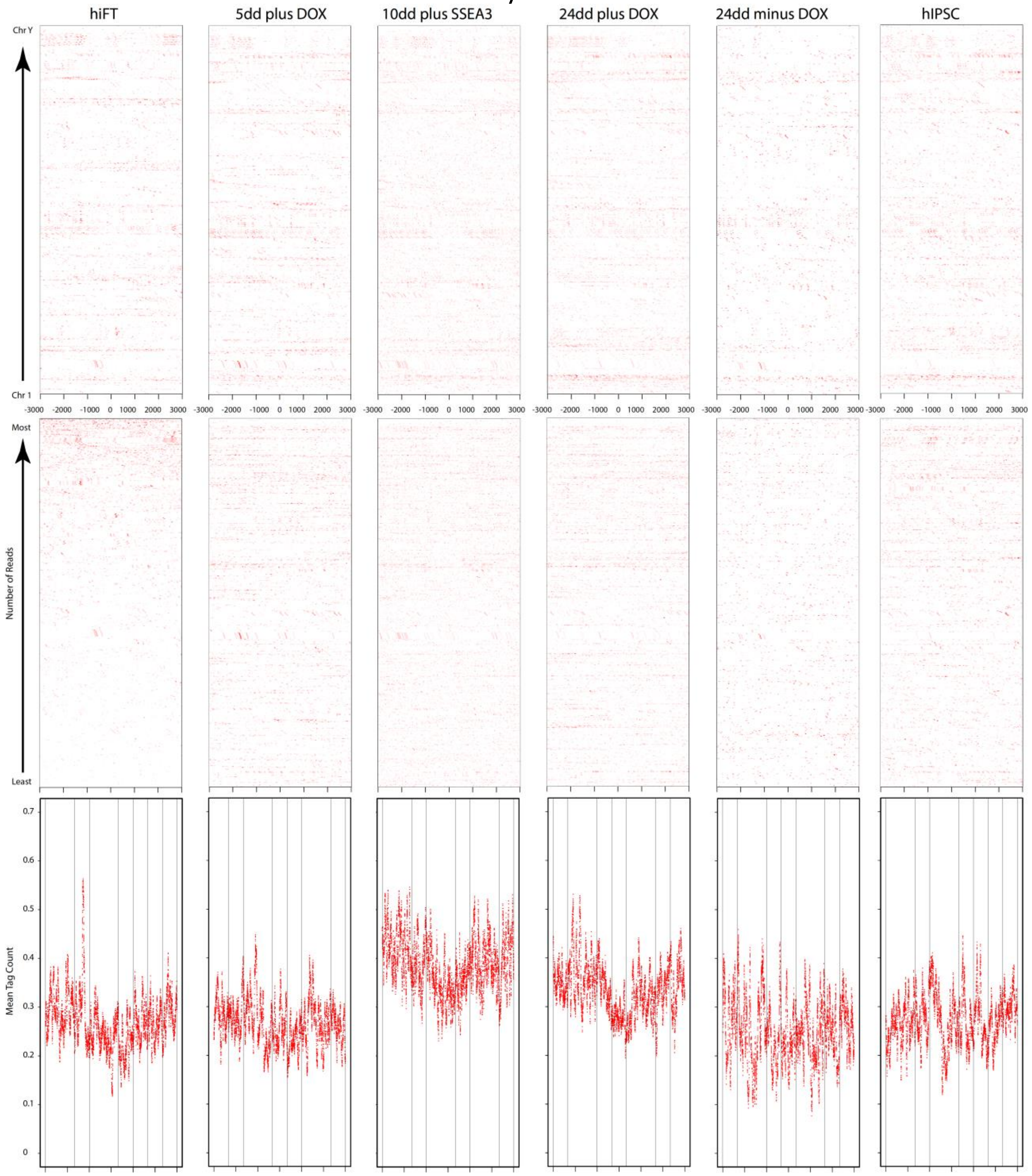

hiFT
5dd plus DOX
10dd plus SSEA3
24dd plus DOX
24dd minus DOX
hIPSC
Chr Y
Chr 1
-3000 -2000 -1000 0 1000 2000 3000
Most
Number of Reads
Least
Mean Tag Count
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0

TEAD1

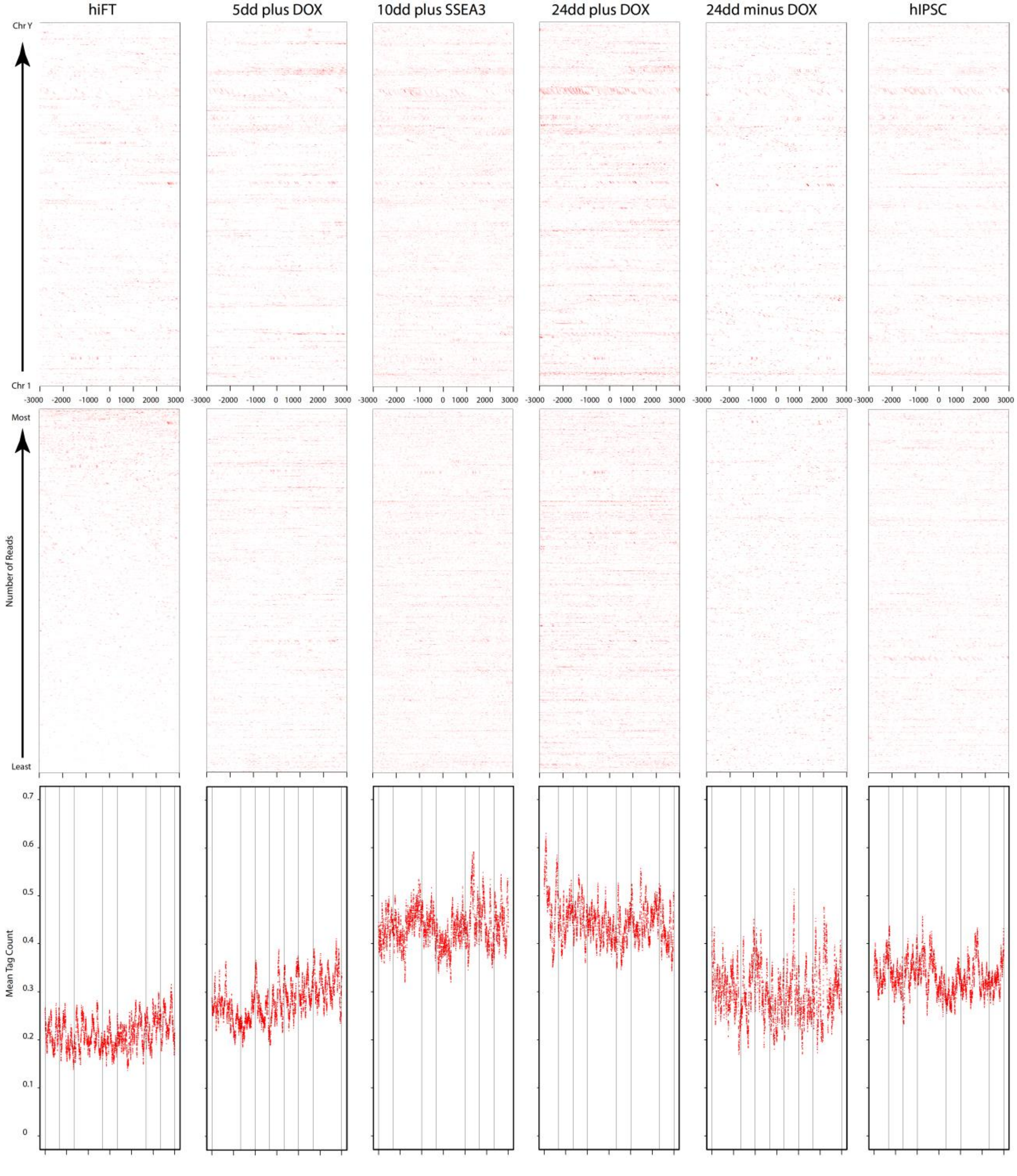


hiFT
5dd plus DOX
10dd plus SSEA3
24dd plus DOX
24dd minus DOX
hIPSC
Chr Y
Chr 1
-3000
-2000
-1000
0
1000
2000
3000
Most
Number of Reads
Least
Mean Tag Count
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0

YAP/TAZ


hiFT
5dd plus DOX
10dd plus SSEA3
24dd plus DOX
24dd minus DOX
hIPSC
Chr Y
Chr 1
-3000 -2000 -1000 0 1000 2000 3000
Most
Number of Reads
Least
Mean Tag Count
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0

Late Embryogenesis



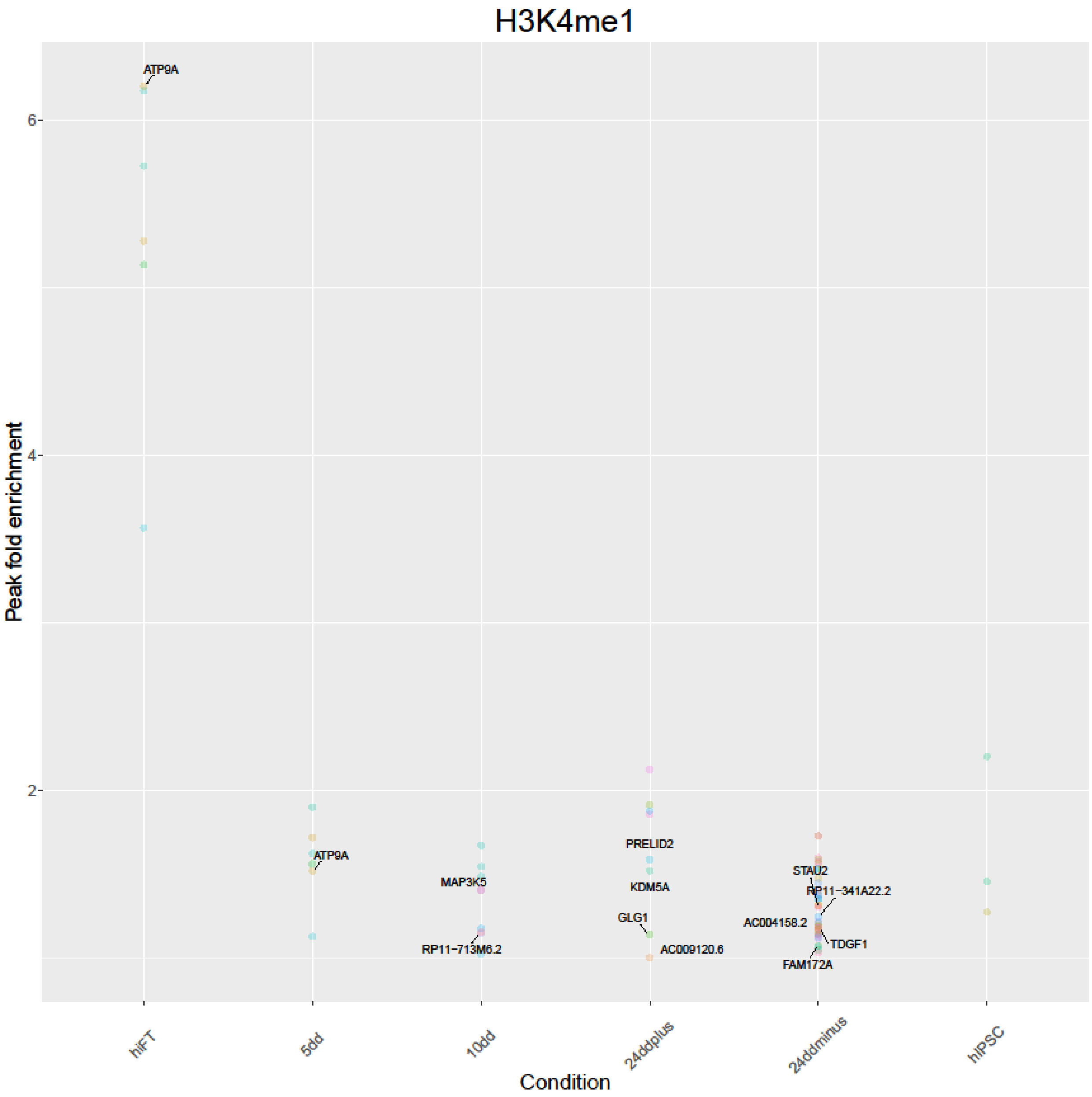
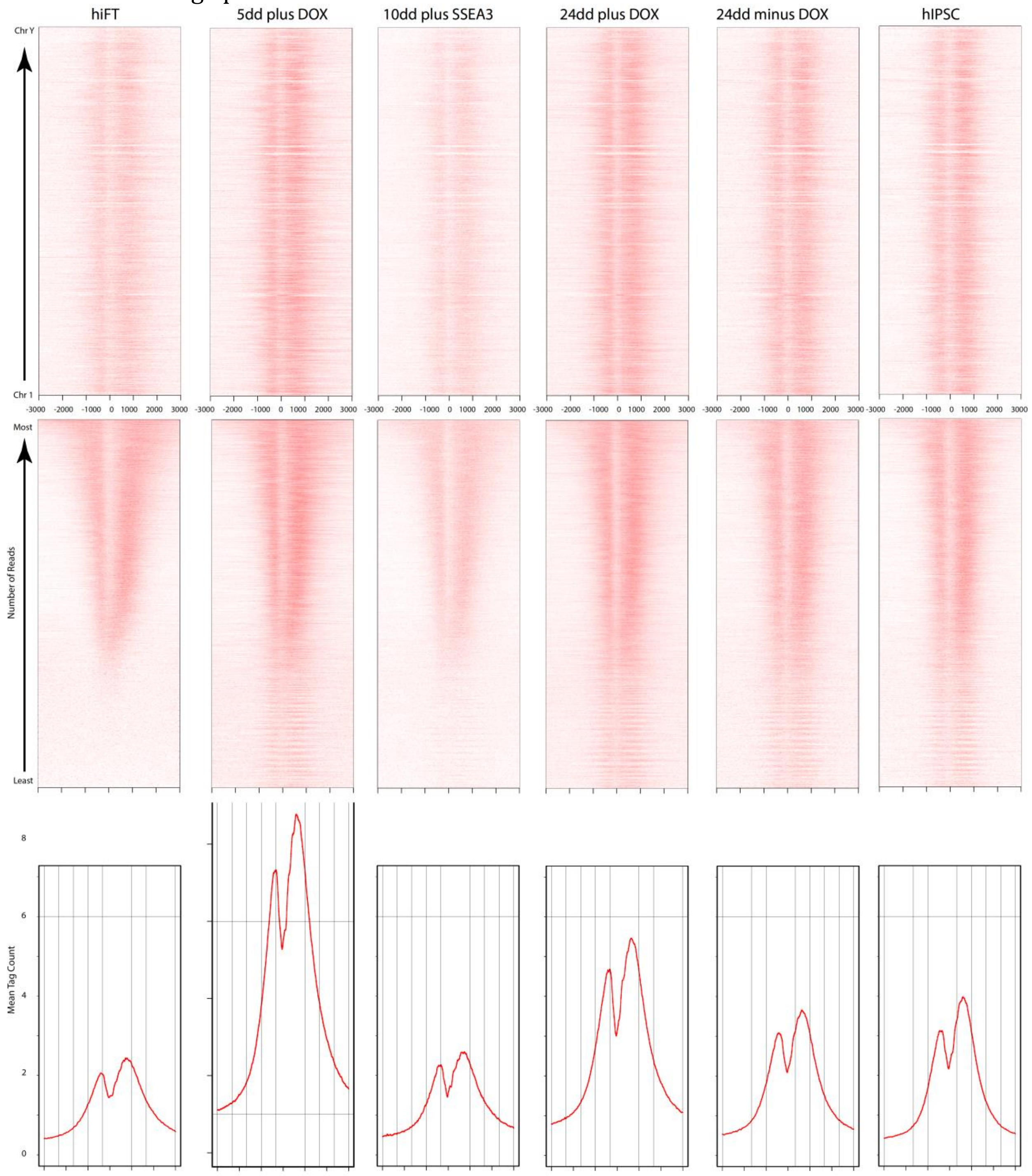
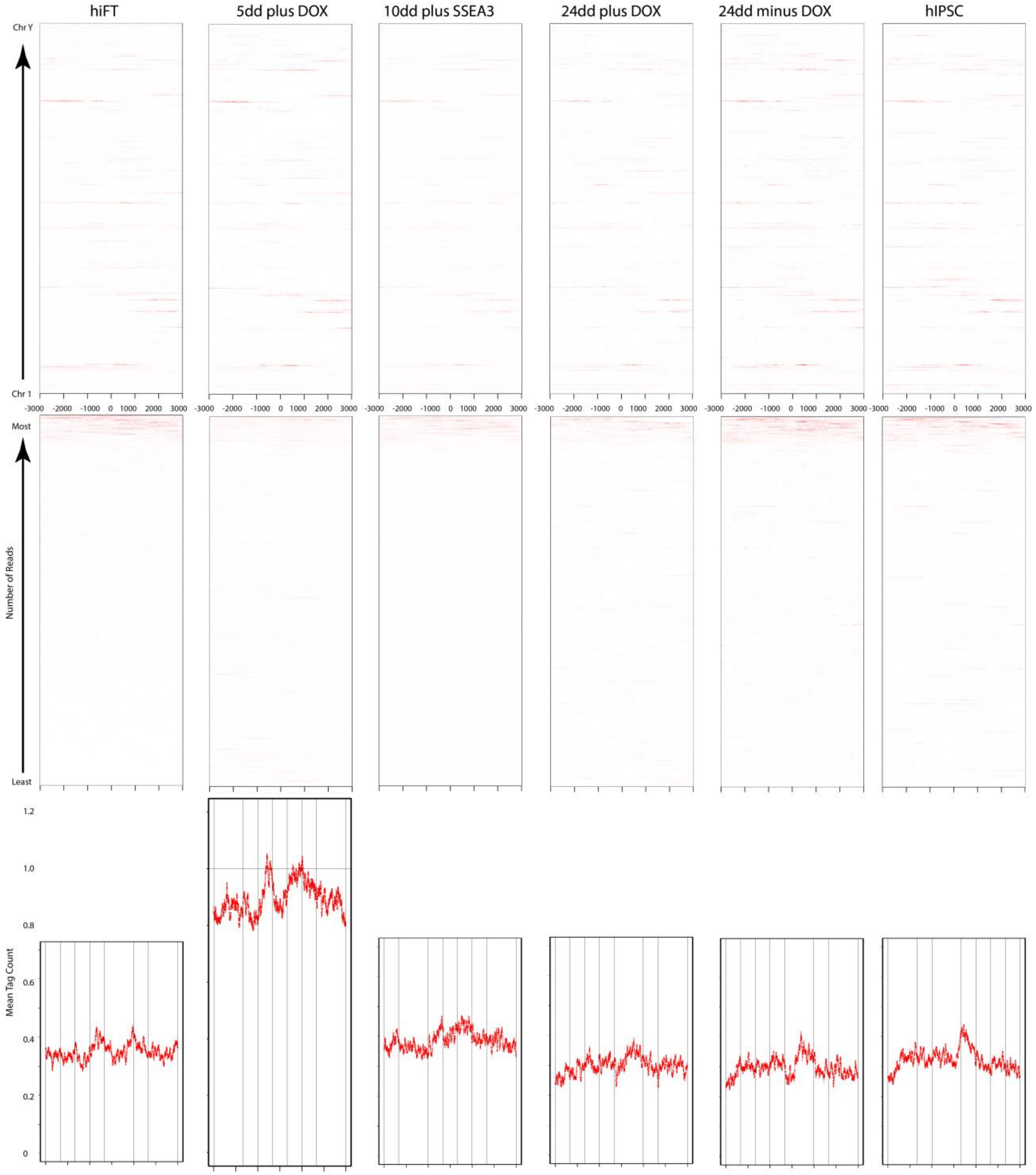

H3K4me1
Peak fold enrichment
6
4
2
ATP9A
ATP9A
MAP3K5
RP11−713M6.2
PRELID2
KDM5A
GLG1
AC009120.6
STAU2
RP11−341A22.2
AC004158.2
TDGF1
FAM172A
hiFT
5dd
10dd
24ddplus
24ddminus
hiPSC
Condition

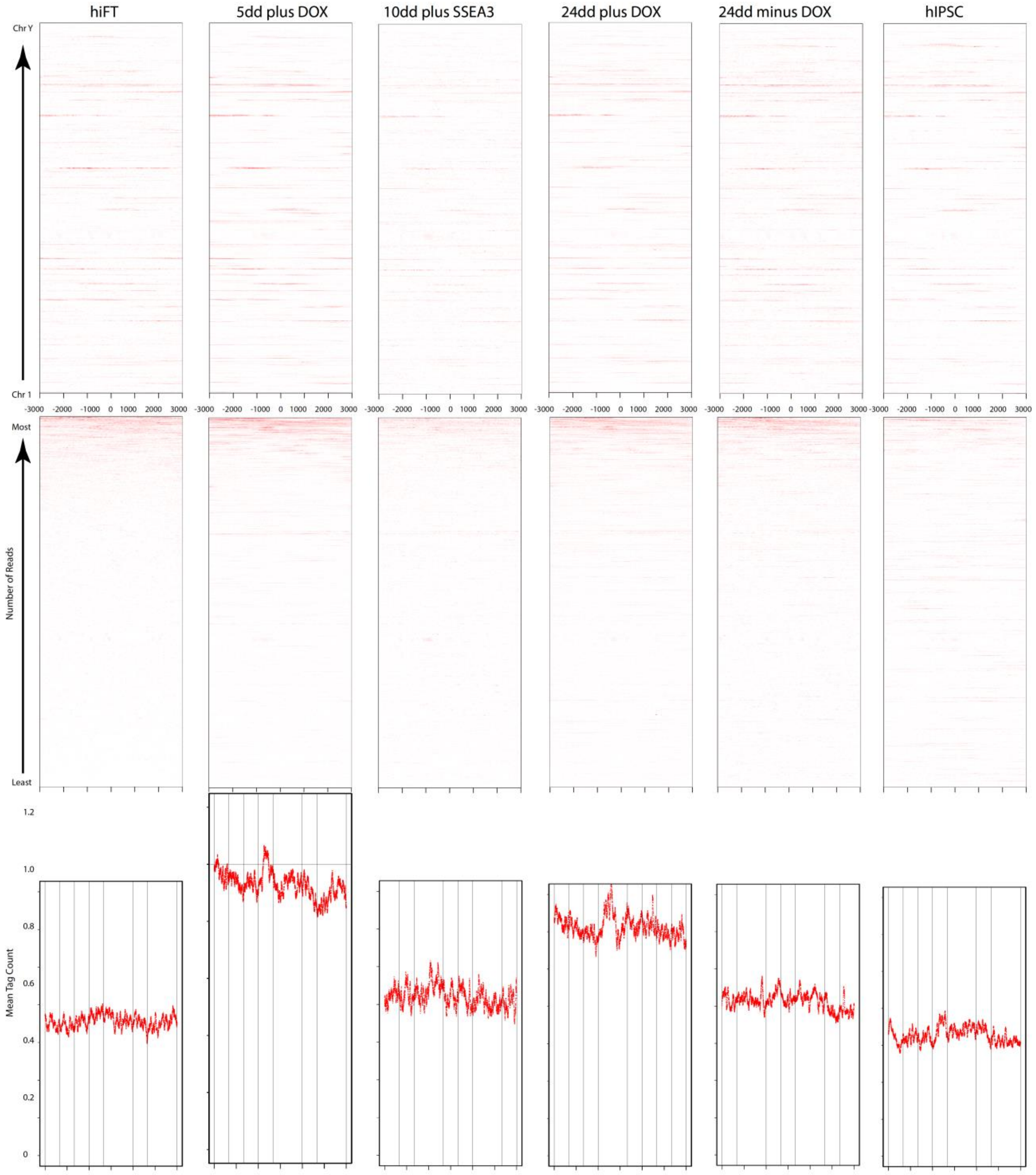
Figure Set 2: H3K4me2 histone modification, custom gene set analyses and peak enrichment change plot



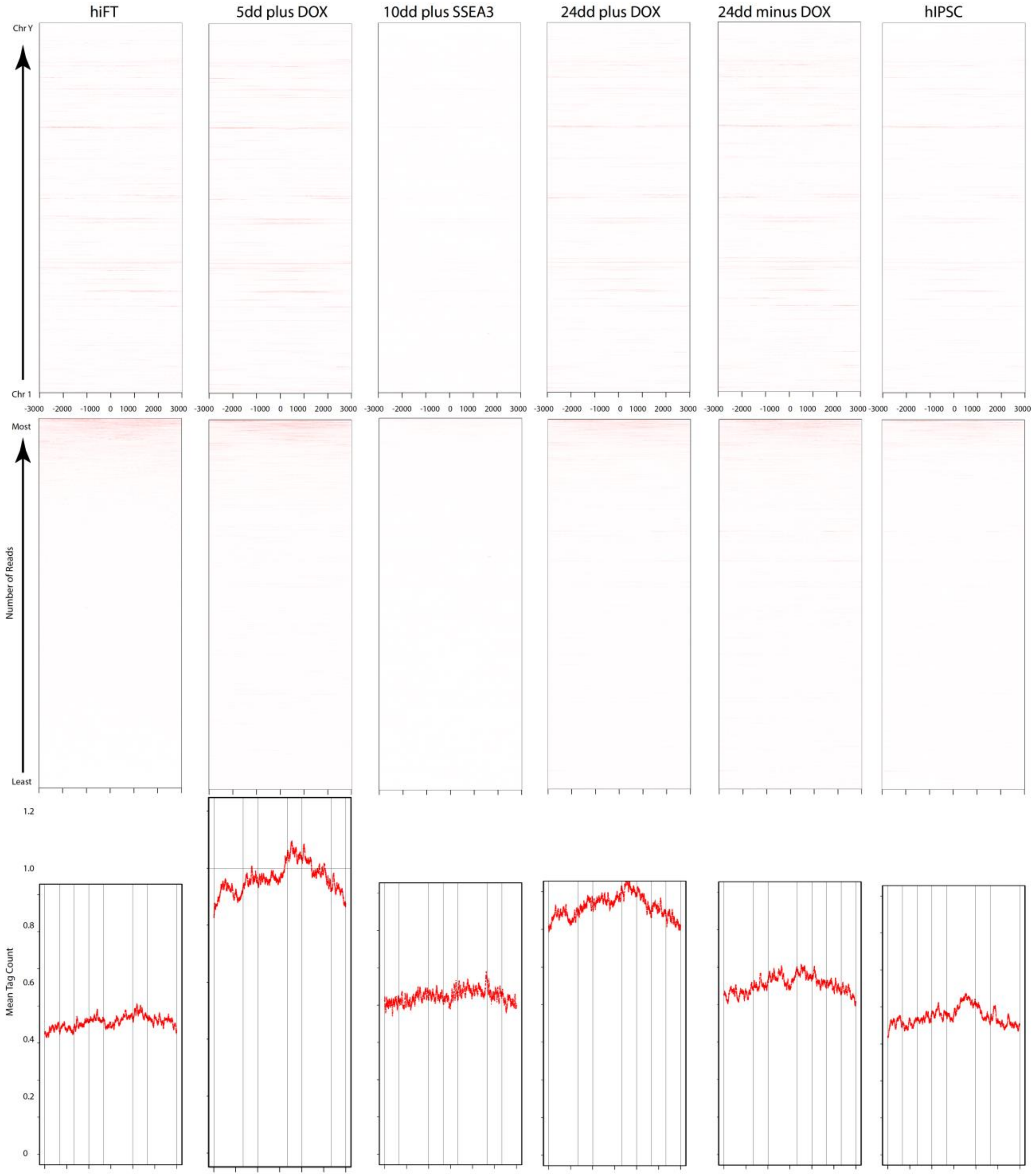
Adhesome


hiFT
5dd plus DOX
10dd plus SSEA3
24dd plus DOX
24dd minus DOX
hIPSC
Chr Y
Chr 1
-3000
-2000
-1000
0
1000
2000
3000
Most
Number of Reads
Least
Mean Tag Count
1.2
1.0
0.8
0.6
0.4
0.2
0

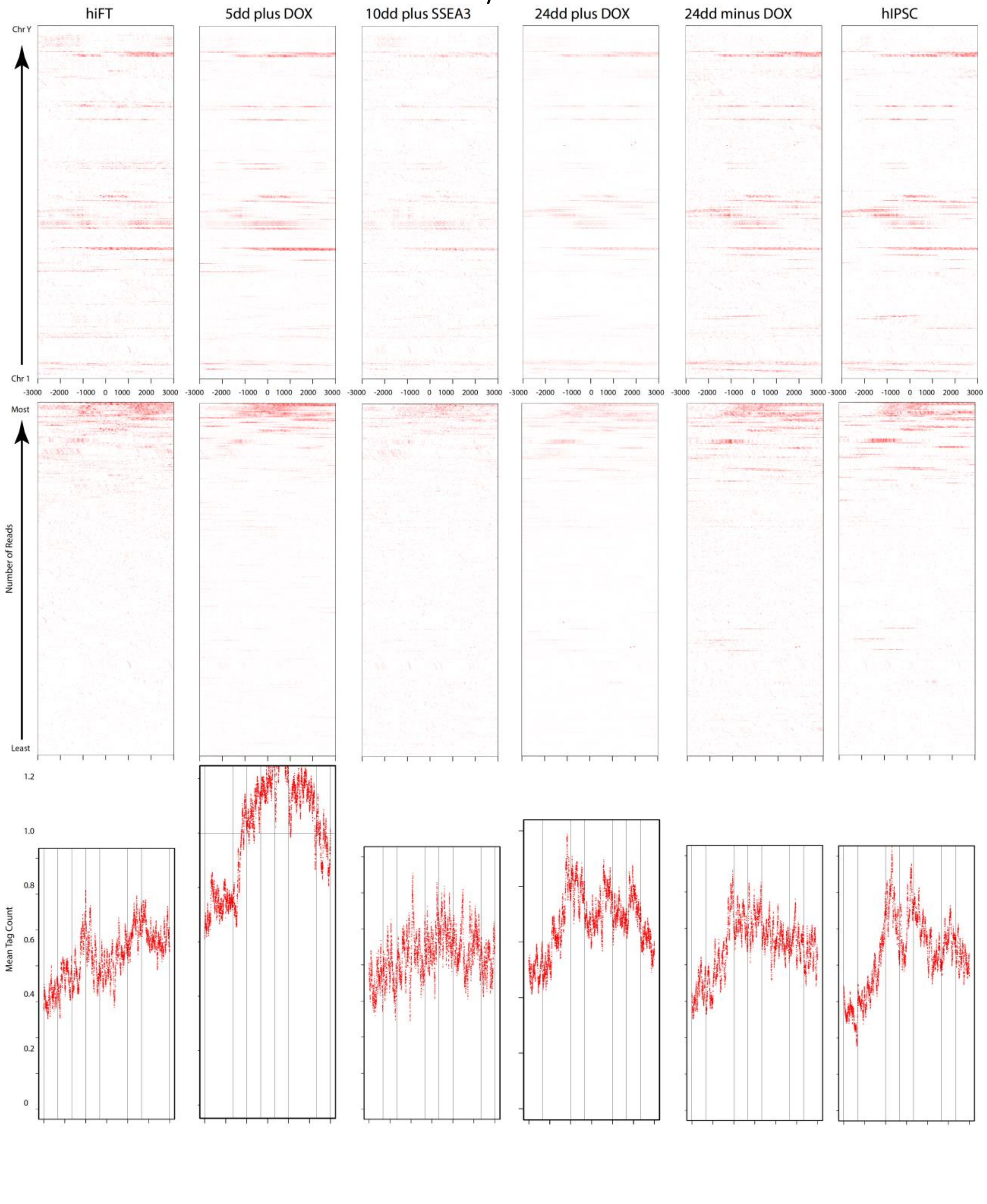
SRF


hiFT
5dd plus DOX
10dd plus SSEA3
24dd plus DOX
24dd minus DOX
hIPSC
Chr Y
Chr 1
-3000 -2000 -1000 0 1000 2000 3000
Most
Number of Reads
Least
1.2
1.0
0.8
0.6
0.4
0.2
0
Mean Tag Count

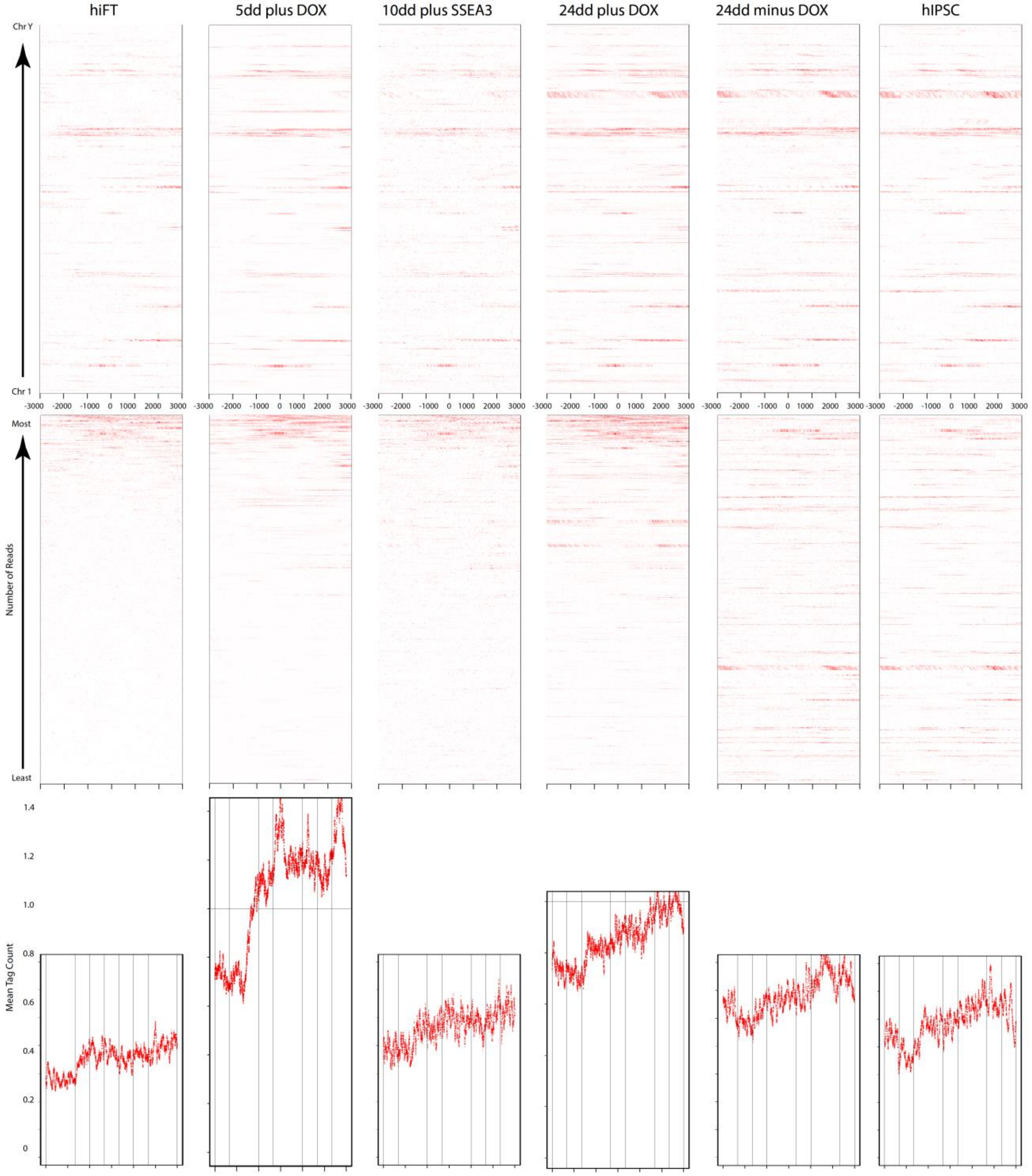
TEAD1


hiFT
5dd plus DOX
10dd plus SSEA3
24dd plus DOX
24dd minus DOX
hIPSC
Chr Y
Chr 1
-3000 -2000 -1000 0 1000 2000 3000
Most
Number of Reads
Least
1.2
1.0
0.8
0.6
0.4
0.2
0
Mean Tag Count

YAP/TAZ


hiFT
5dd plus DOX
10dd plus SSEA3
24dd plus DOX
24dd minus DOX
hIPSC
Chr Y
Chr 1
-3000 -2000 -1000 0 1000 2000 3000
Most
Number of Reads
Least
Mean Tag Count
1.2
1.0
0.8
0.6
0.4
0.2
0

Late Embryogenesis



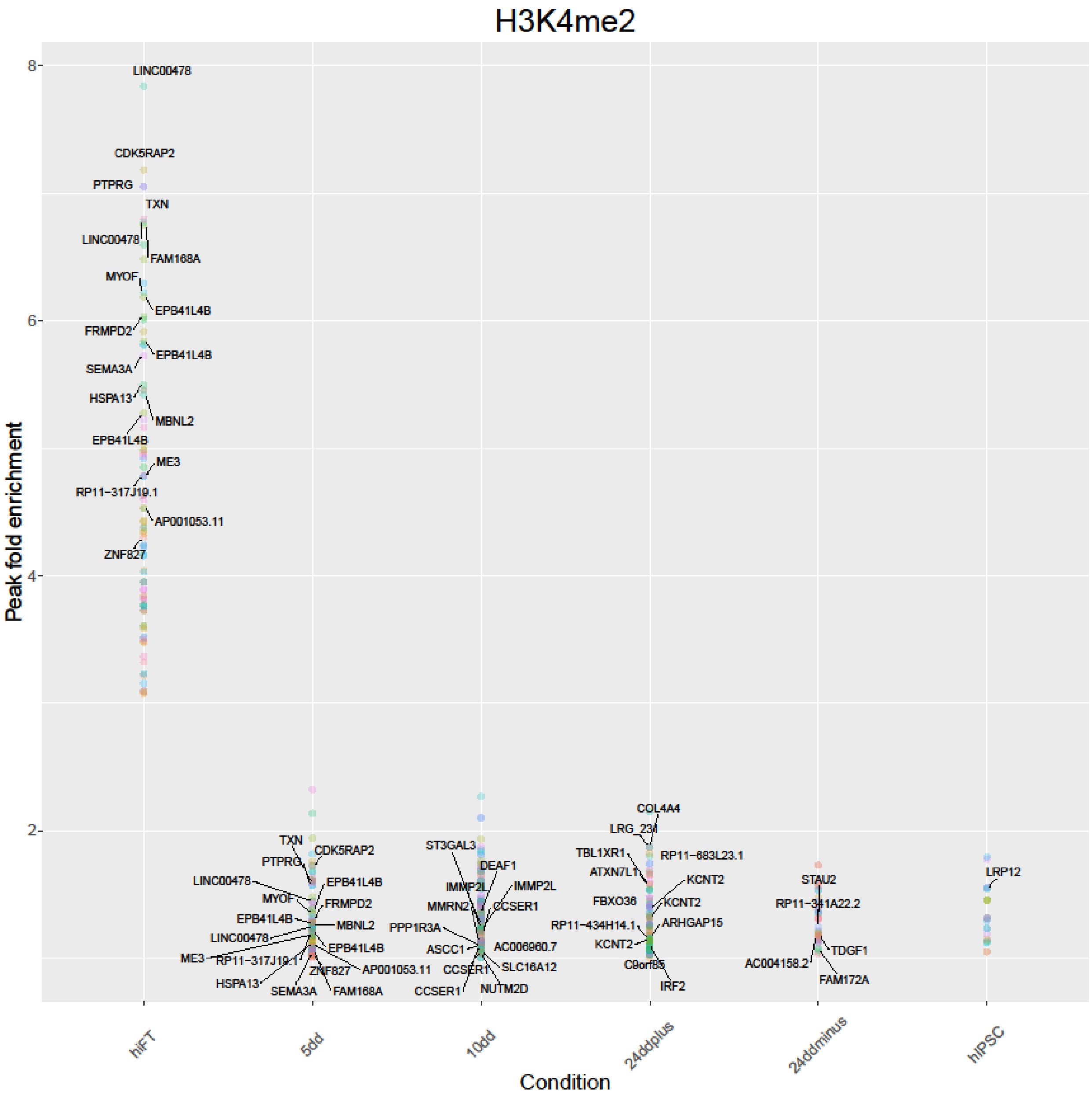
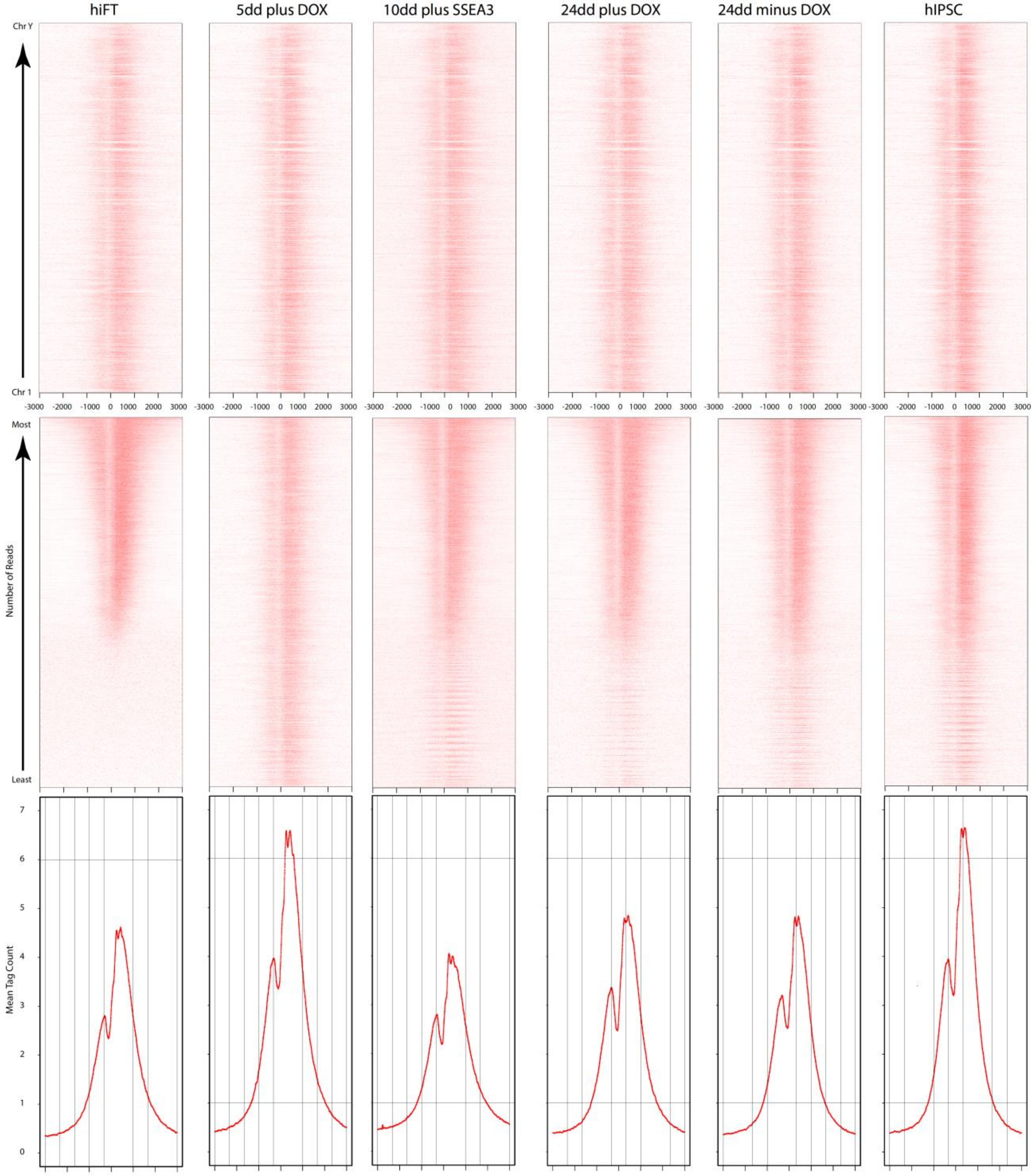
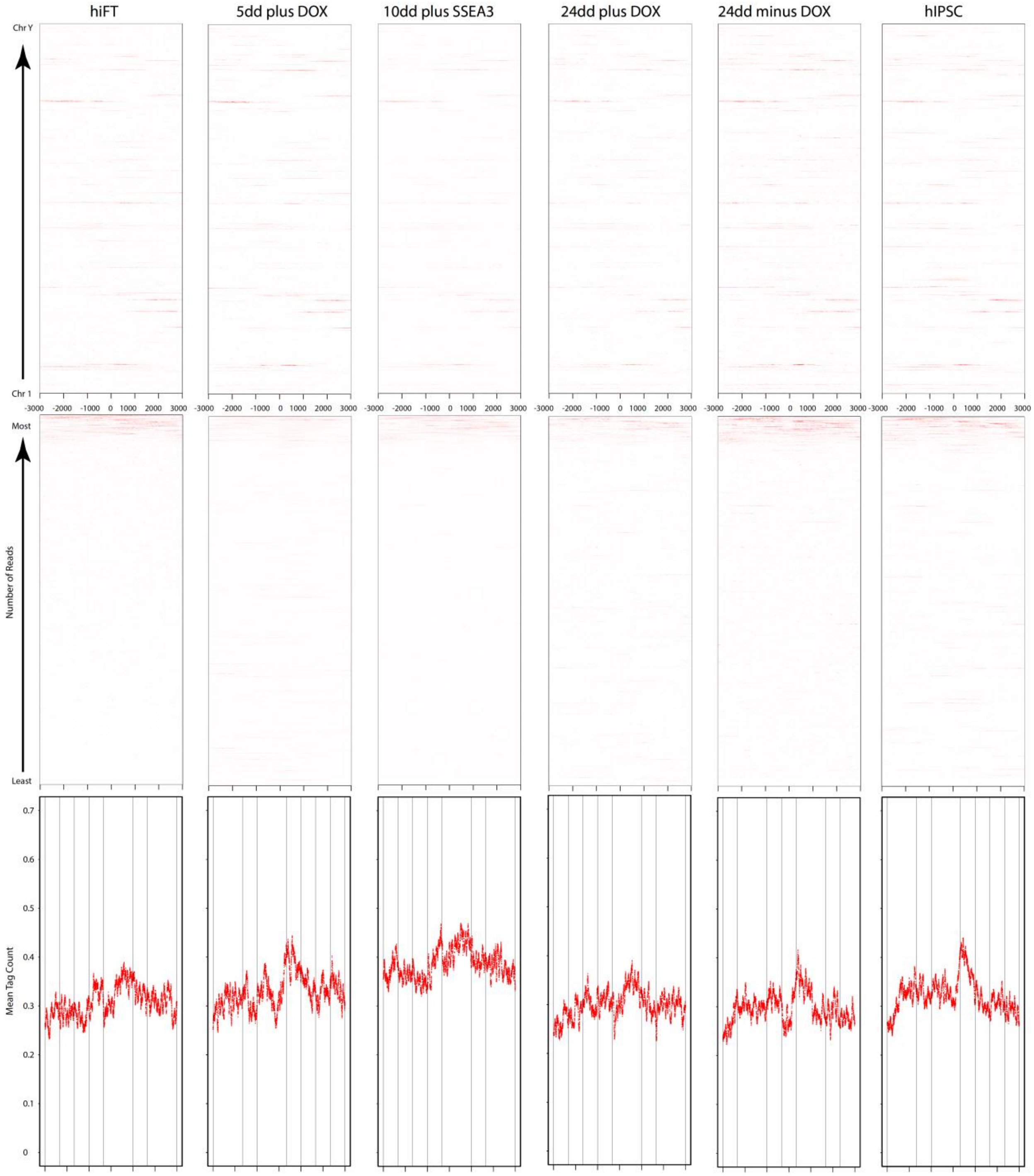

H3K4me2
Peak fold enrichment
8
6
4
2
LINC00478
CDK5RAP2
PTPRG
TXN
LINC00478
FAM168A
MYOF
EPB41L4B
FRMPD2
EPB41L4B
SEMA3A
HSPA13
MBNL2
EPB41L4B
ME3
RP11−317J19.1
AP001053.11
ZNF827
TXN
CDK5RAP2
PTPRG
LINC00478
EPB41L4B
MYOF
FRMPD2
EPB41L4B
MBNL2
LINC00478
EPB41L4B
ME3
RP11−317J19.1
ZNF827
AP001053.11
HSPA13
SEMA3A
FAM168A
ST3GAL3
DEAF1
IMMP2L
IMMP2L
MMRN2
CCSER1
PPP1R3A
ASCC1
AC006960.7
CCSER1
SLC16A12
CCSER1
NUTM2D
COL4A4
LRG_231
TBL1XR1
RP11−683L23.1
ATXN7L1
KCNT2
FBXO38
KCNT2
RP11−434H14.1
ARHGAP15
KCNT2
C9orf85
IRF2
STAU2
RP11−341A22.2
TDGF1
AC004158.2
FAM172A
LRP12
hiFT
5dd
10dd
24ddplus
24ddminus
hIPSC
Condition

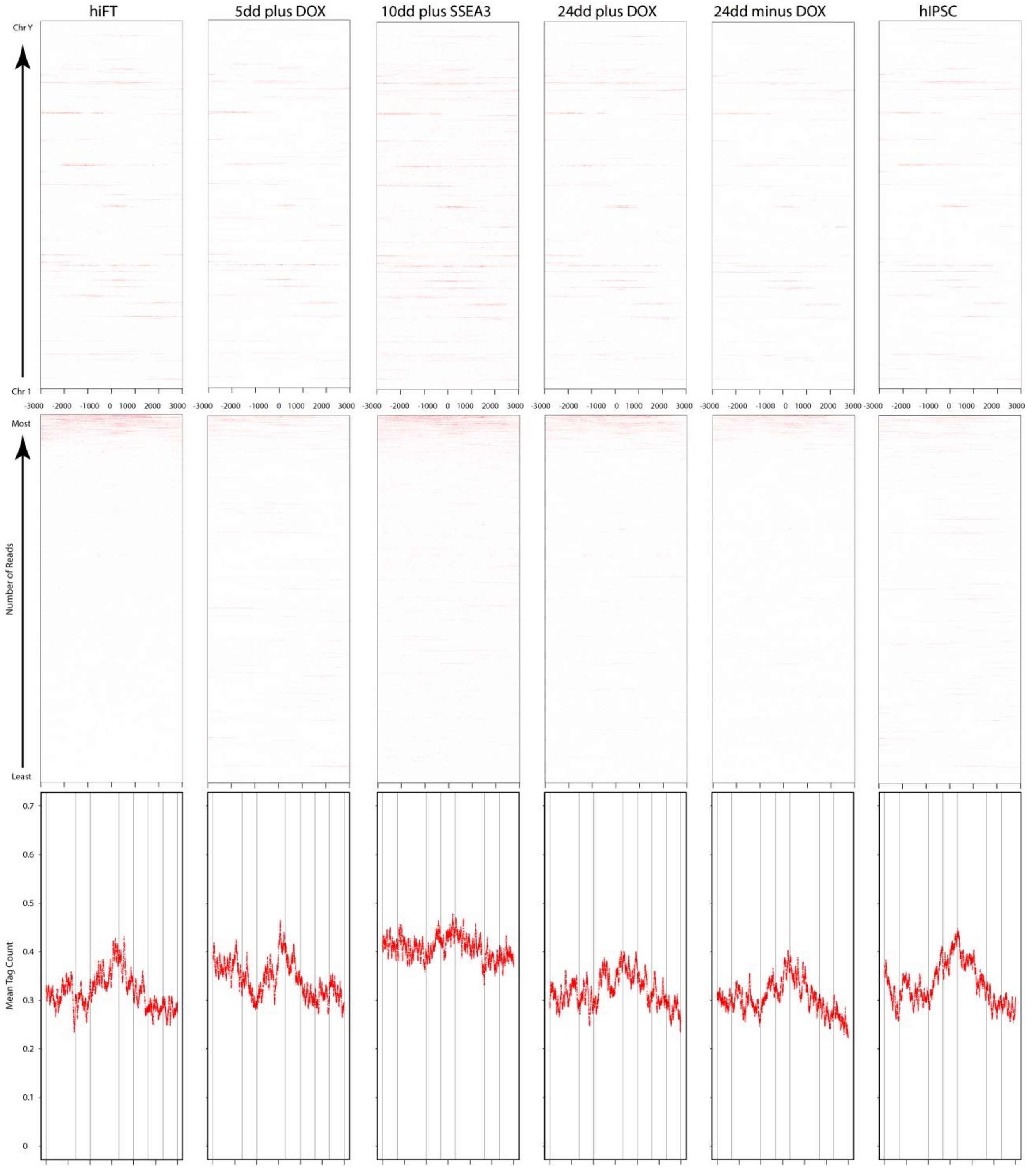
Figure Set 3: H3K4me3 histone modification, custom gene set analyses and peak enrichment change plot



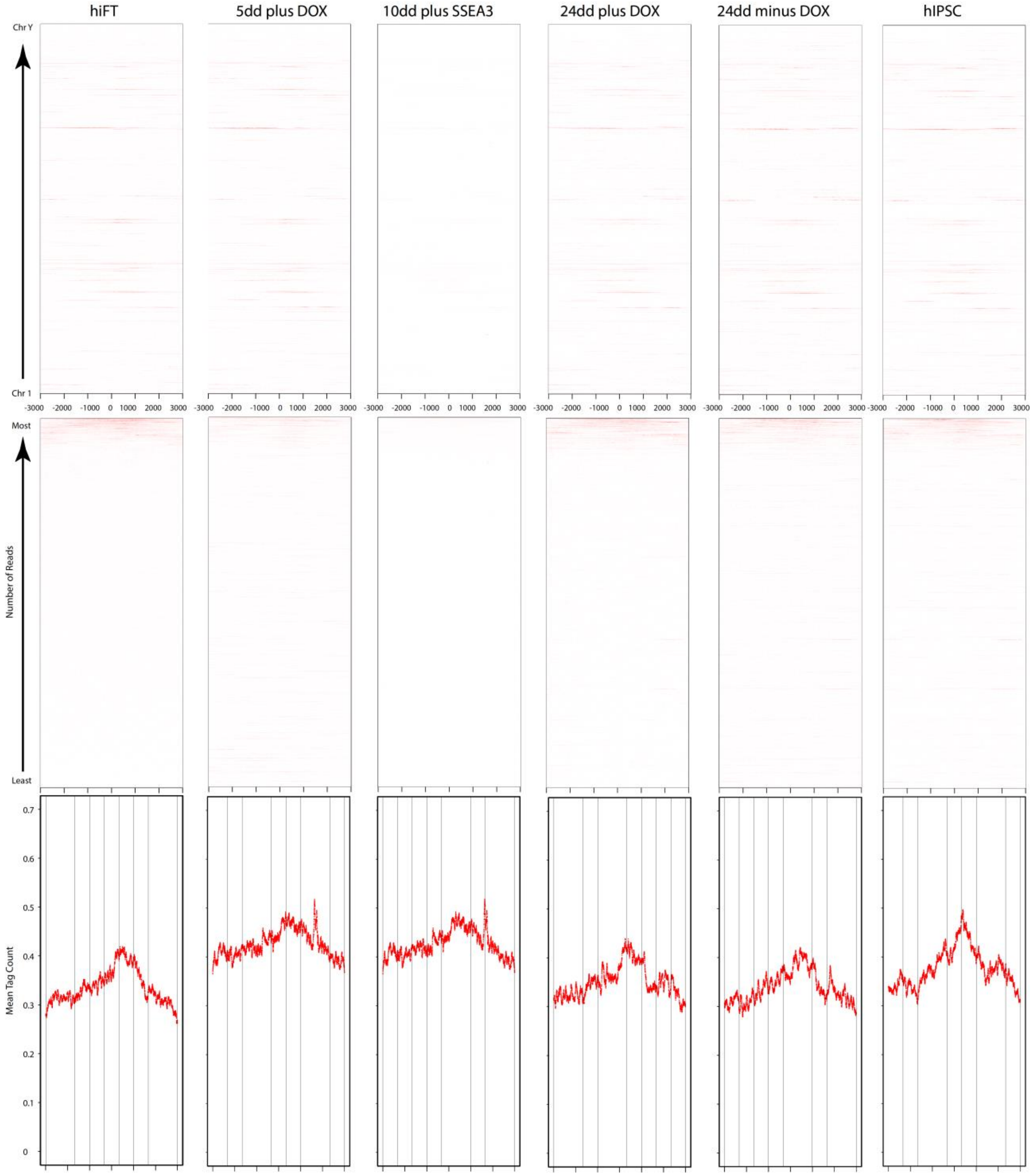
Adhesome


hiFT
5dd plus DOX
10dd plus SSEA3
24dd plus DOX
24dd minus DOX
hIPSC
Chr Y
Chr 1
-3000
-2000
-1000
0
1000
2000
3000
Most
Number of Reads
Least
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Mean Tag Count

SRF


hiFT
5dd plus DOX
10dd plus SSEA3
24dd plus DOX
24dd minus DOX
hIPSC
Chr Y
Chr 1
-3000 -2000 -1000 0 1000 2000 3000
Most
Number of Reads
Least
Mean Tag Count
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0

TEAD1


hiFT
5dd plus DOX
10dd plus SSEA3
24dd plus DOX
24dd minus DOX
hIPSC
Chr Y
Chr 1
-3000
-2000
-1000
0
1000
2000
3000
Most
Number of Reads
Least
Mean Tag Count
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0

YAP/TAZ

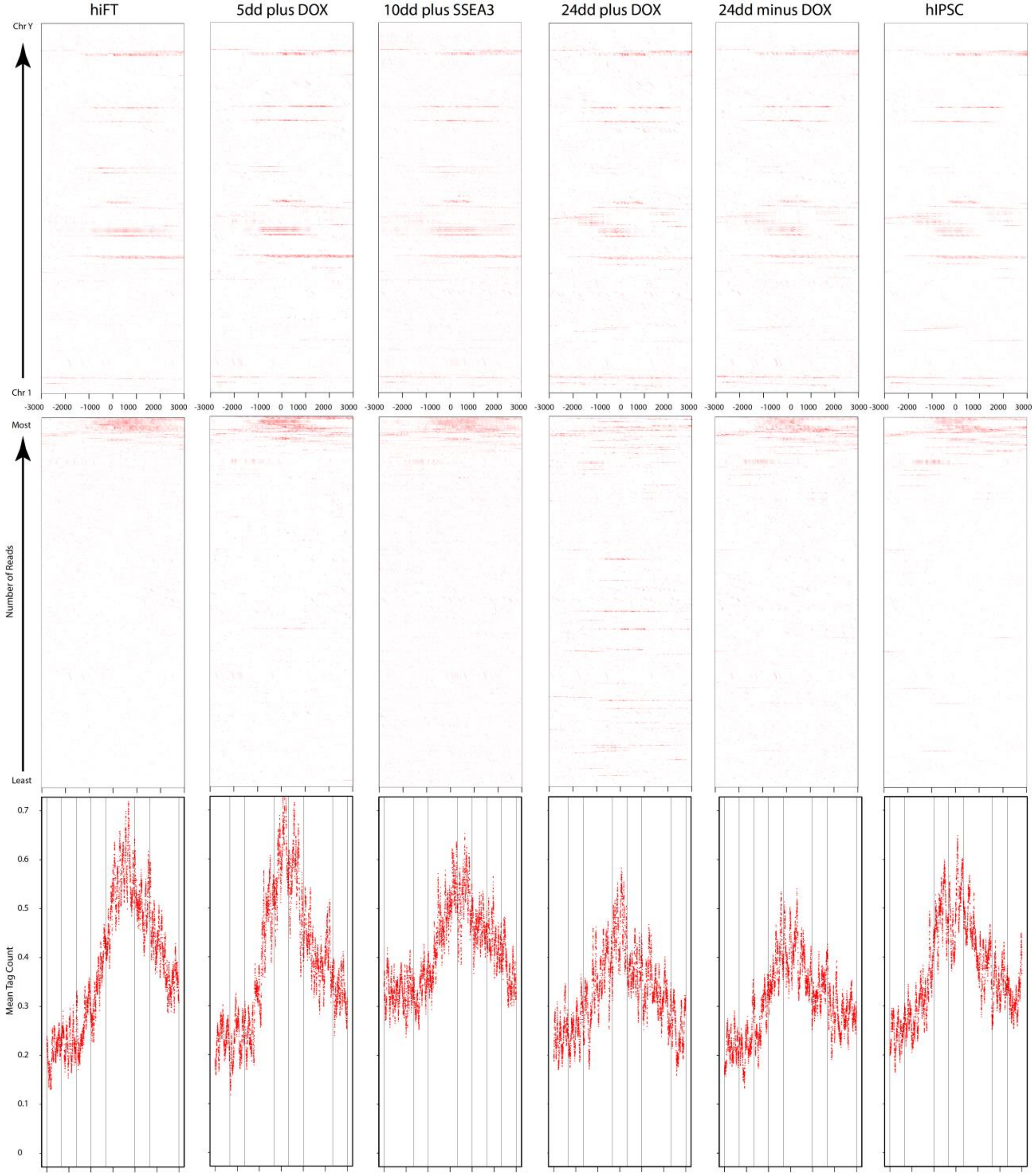
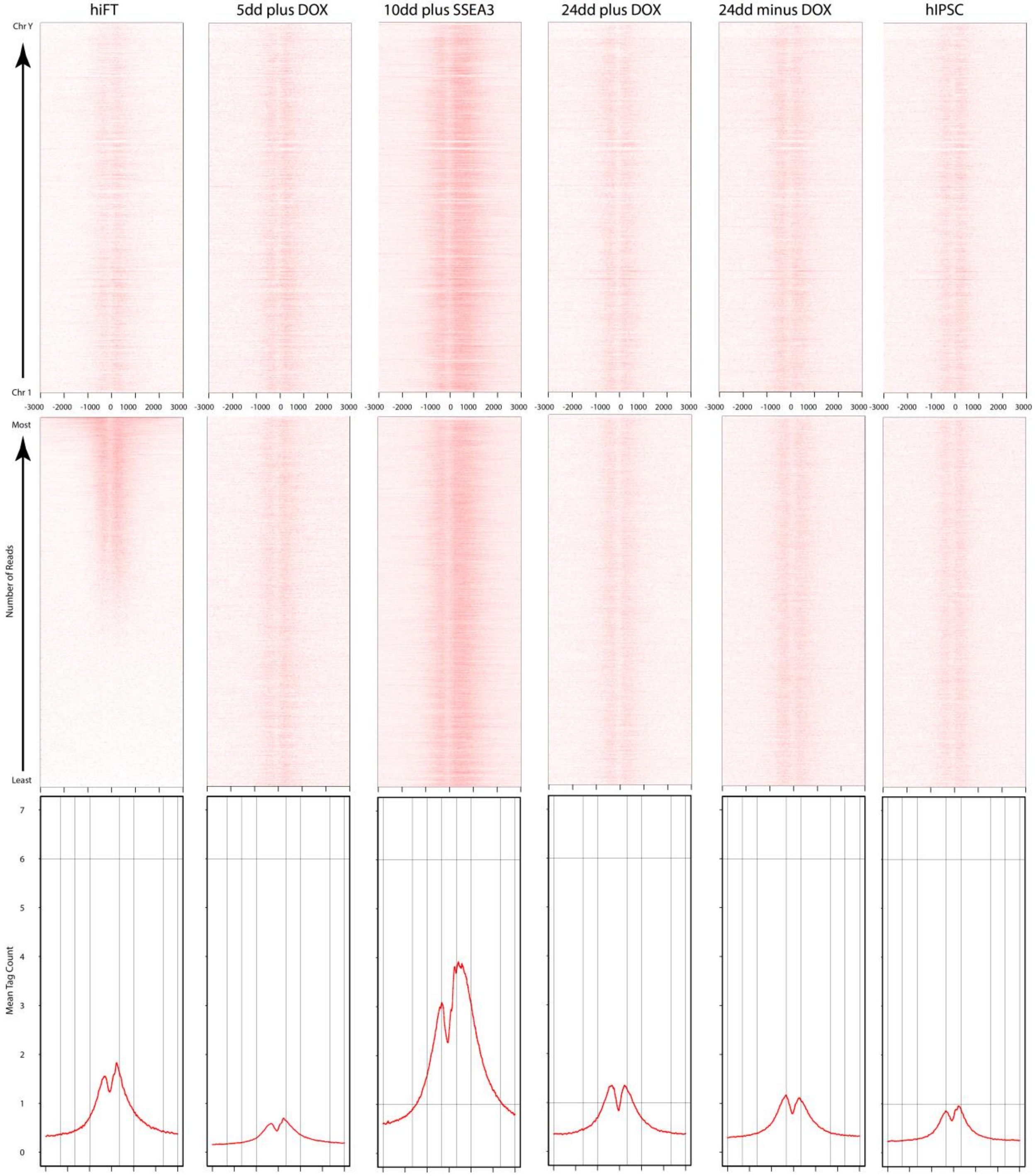
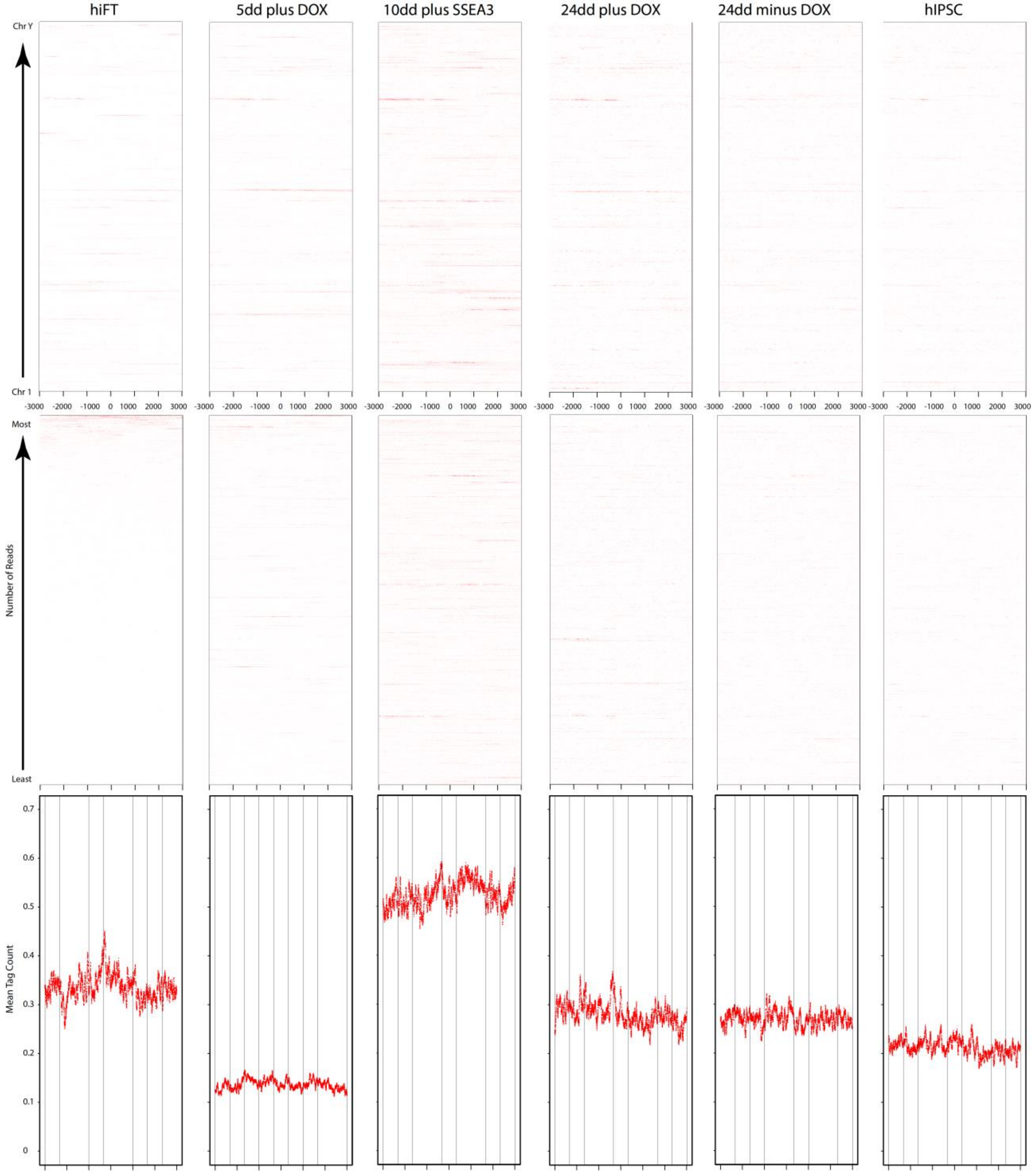

hiFT
5dd plus DOX
10dd plus SSEA3
24dd plus DOX
24dd minus DOX
hIPSC
Chr Y
Chr 1
-3000
-2000
-1000
0
1000
2000
3000
Most
Number of Reads
Least
Mean Tag Count
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0

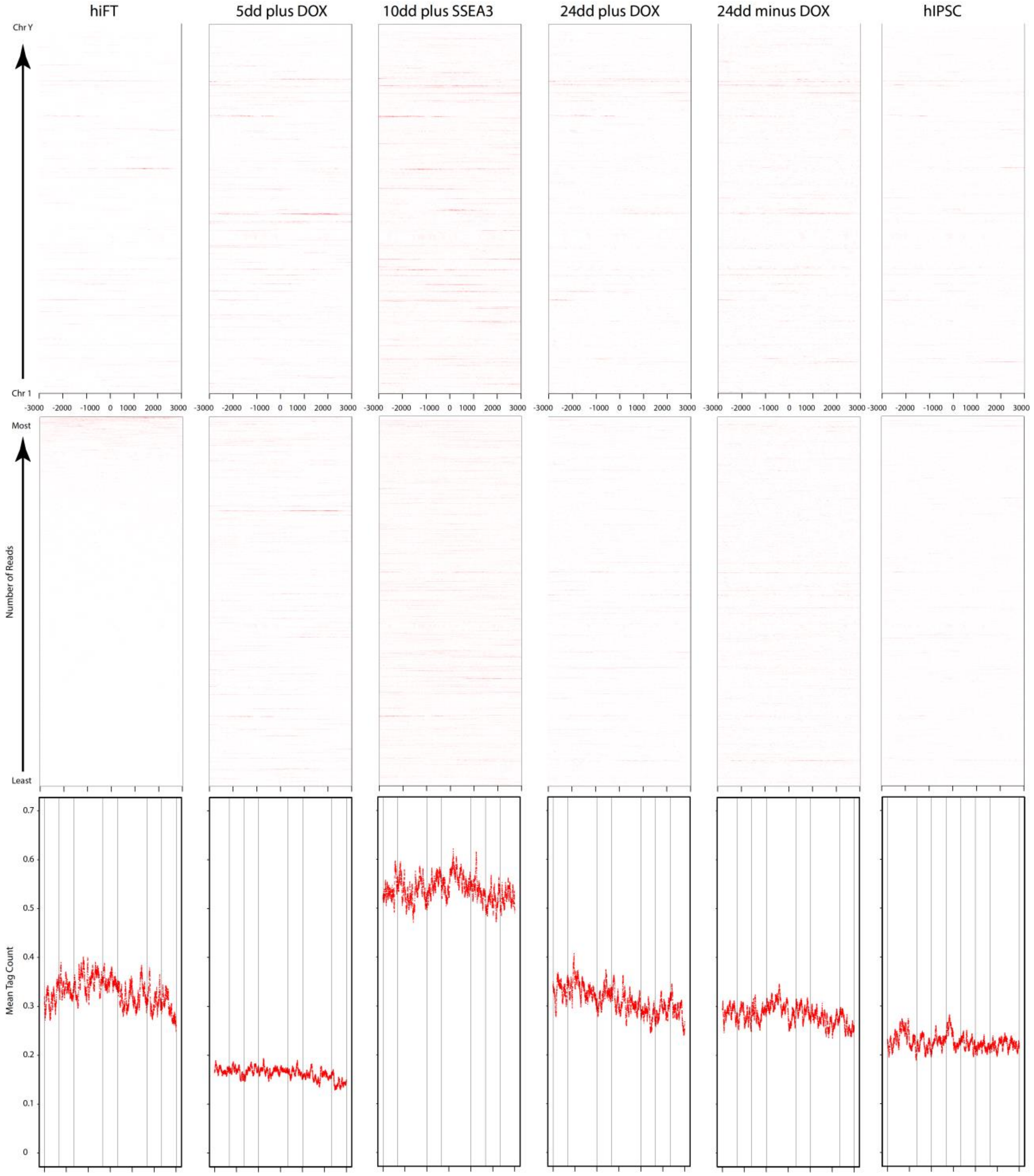
Figure Set 4: H3K27ac histone modification, custom gene set analyses and peak enrichment change plot



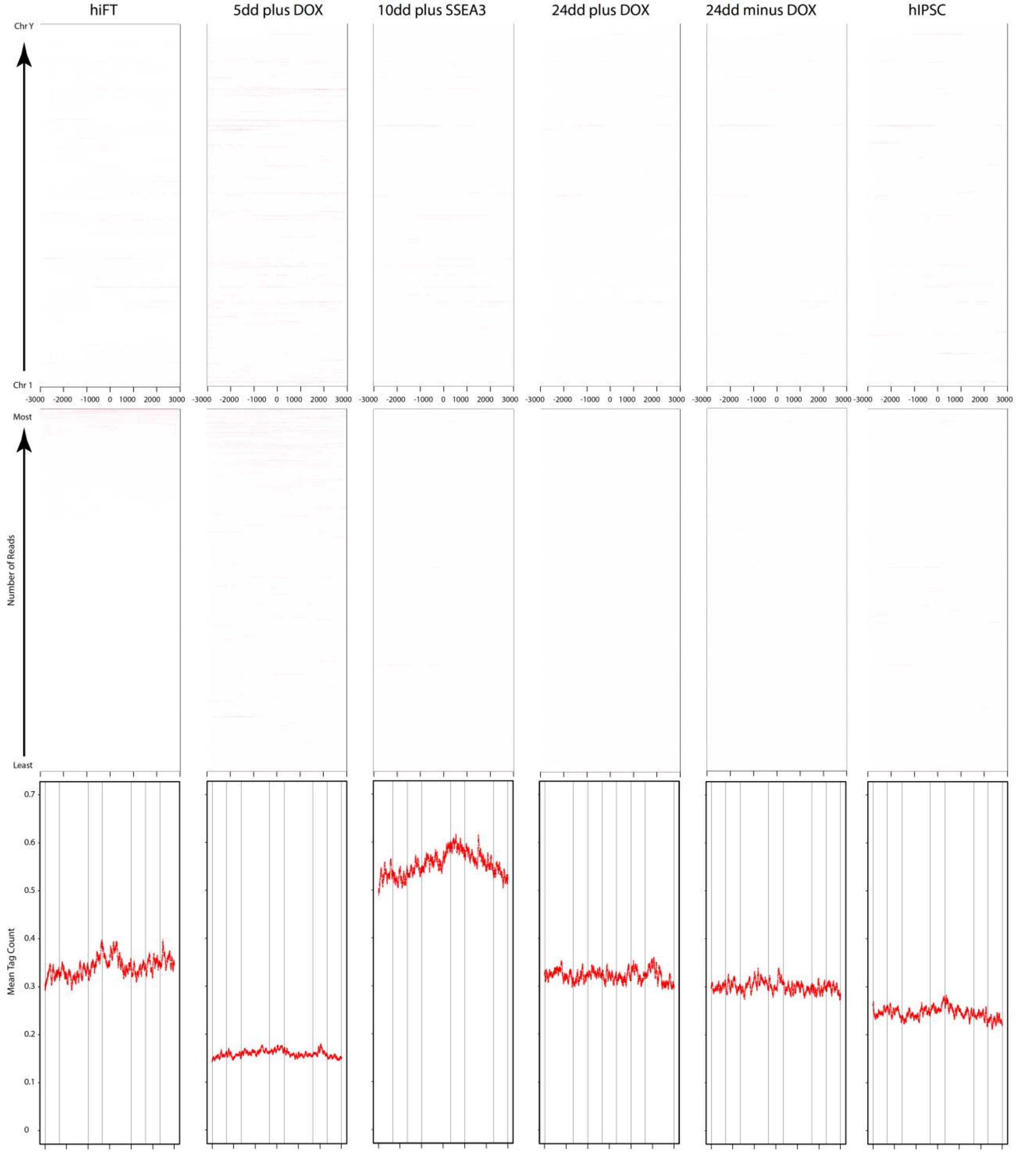
Adhesome


hiFT
5dd plus DOX
10dd plus SSEA3
24dd plus DOX
24dd minus DOX
hIPSC
Chr Y
Chr 1
-3000 -2000 -1000 0 1000 2000 3000
Most
Number of Reads
Least
Mean Tag Count
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0

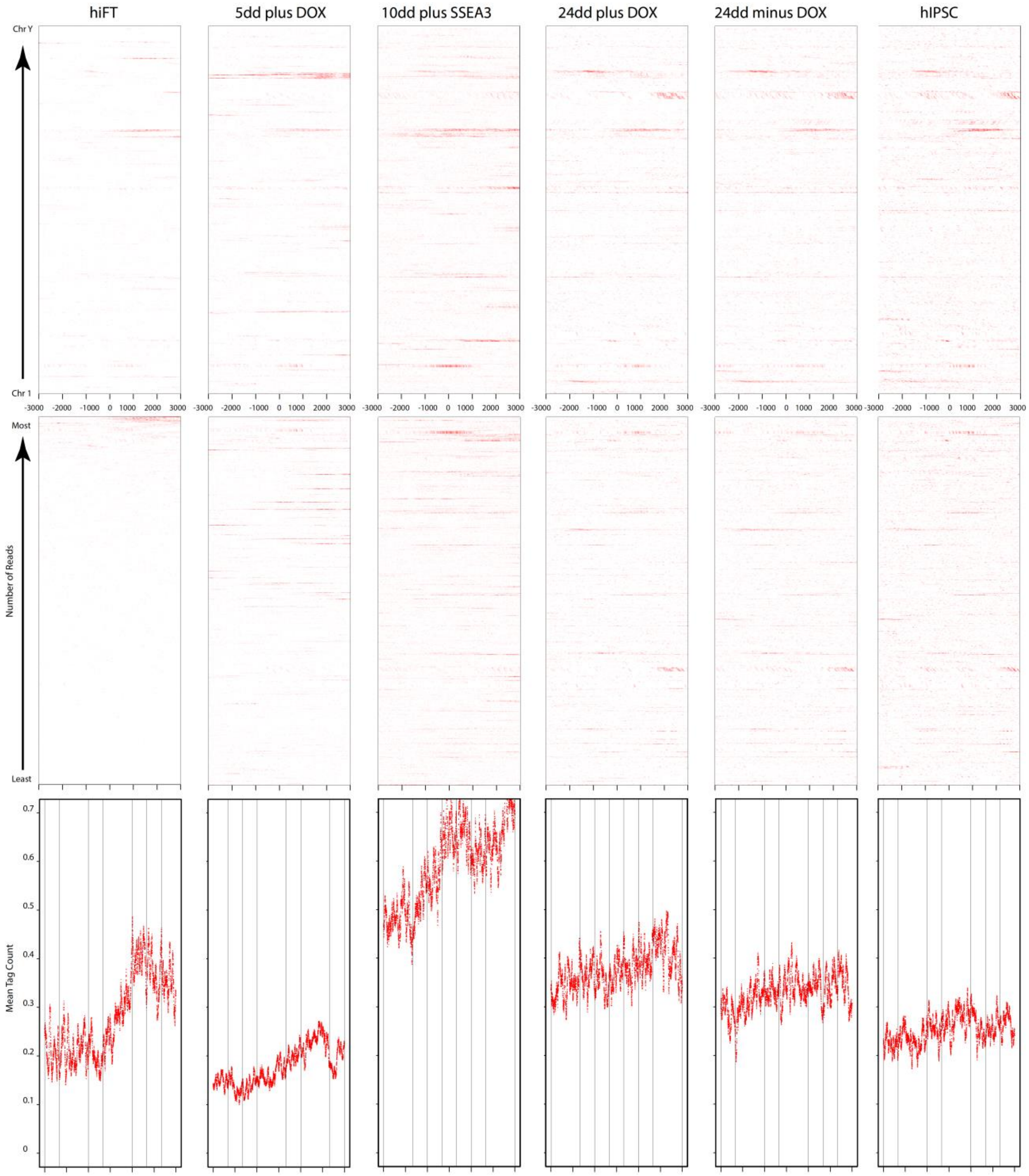
SRF


hiFT
5dd plus DOX
10dd plus SSEA3
24dd plus DOX
24dd minus DOX
hIPSC
Chr Y
Chr 1
-3000
-2000
-1000
0
1000
2000
3000
Most
Number of Reads
Least
Mean Tag Count
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0

TEAD1


hiFT
5dd plus DOX
10dd plus SSEA3
24dd plus DOX
24dd minus DOX
hIPSC
Chr Y
Chr 1
-3000 -2000 -1000 0 1000 2000 3000
Most
Number of Reads
Least
Mean Tag Count
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0

Late Embryogenesis



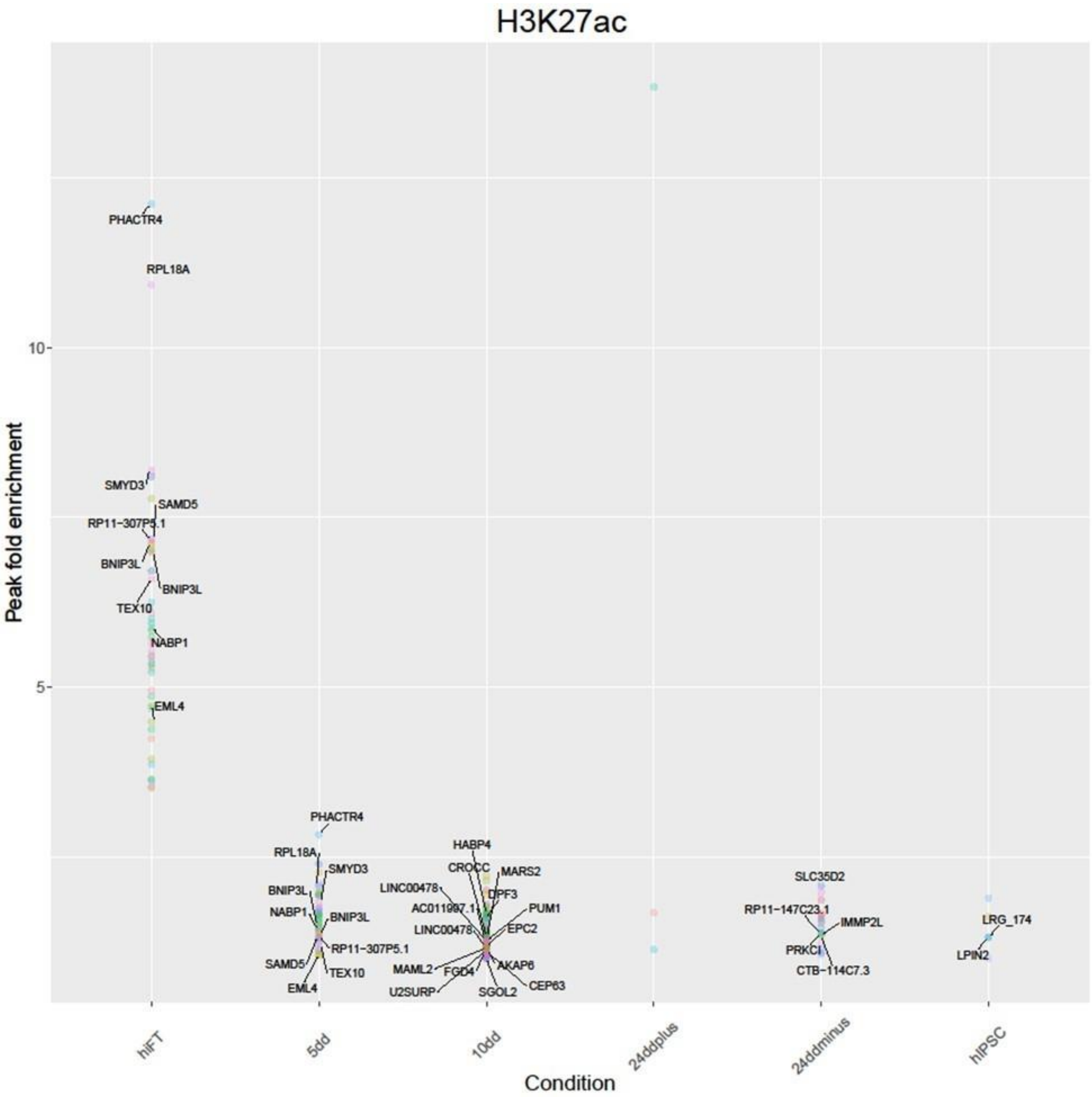
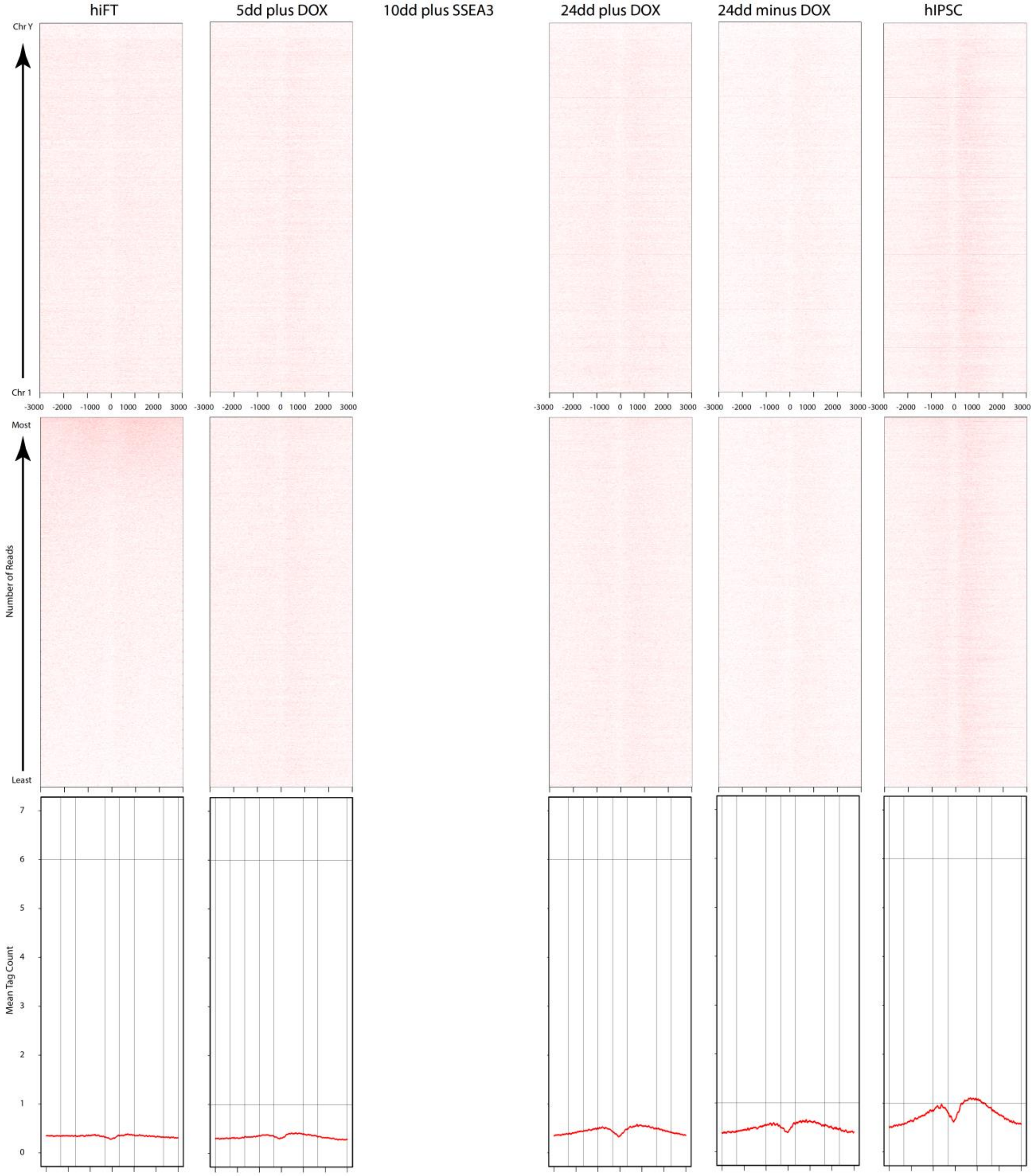
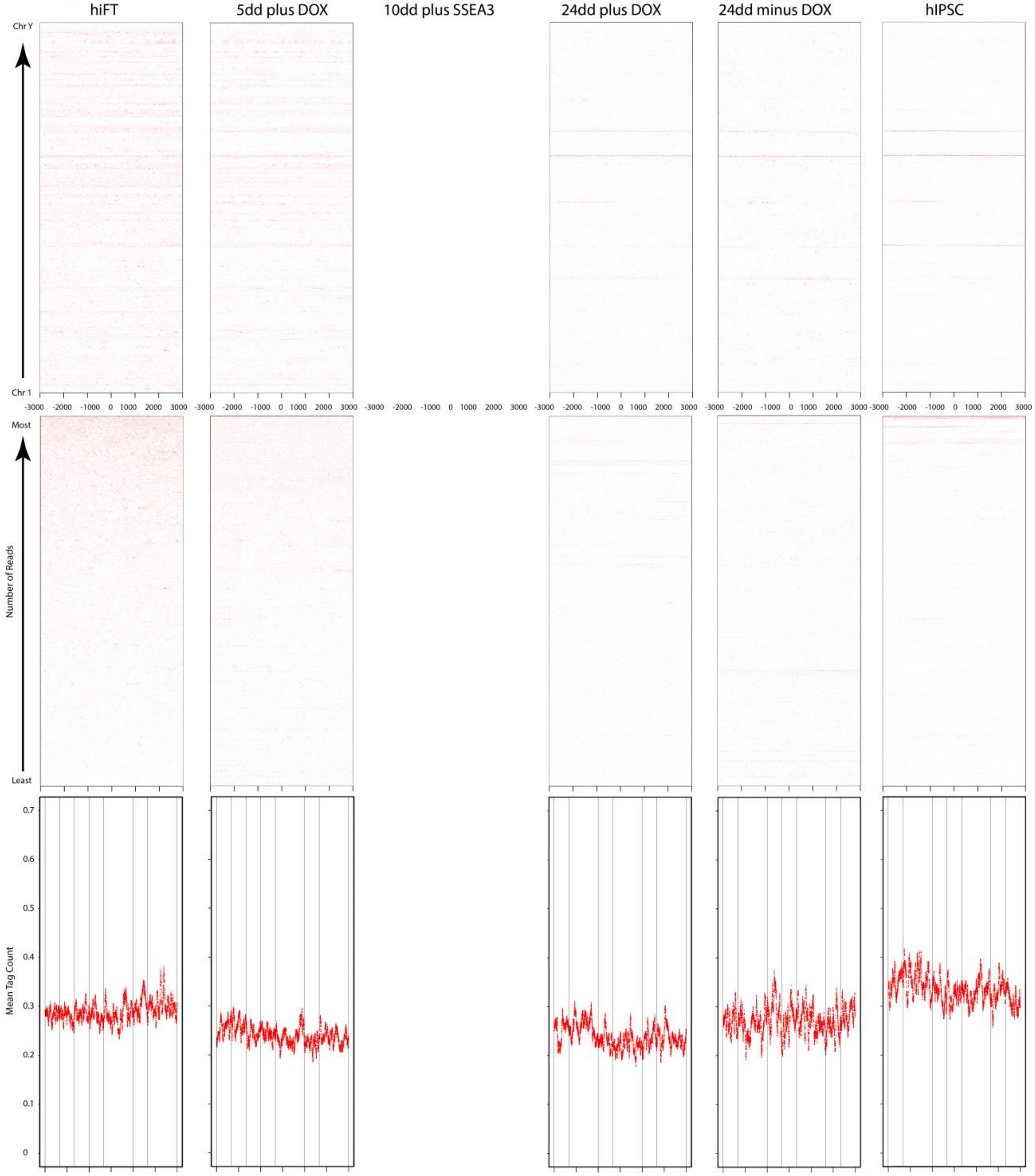

H3K27ac
Peak fold enrichment
10
5
Condition
hiFT
5dd
10dd
24ddplus
24ddminus
hIPSC
PHACTR4
RPL18A
SMYD3
SAMD5
RP11-307P5.1
BNIP3L
BNIP3L
TEX10
NABP1
EML4
PHACTR4
RPL18A
SMYD3
BNIP3L
NABP1
BNIP3L
RP11-307P5.1
SAMD5
TEX10
EML4
HABP4
CROCC
MARS2
LINC00478
DPF3
AC011997.1
PUM1
LINC00478
EPC2
MAML2
FGD4
AKAP6
CEP63
U2SURP
SGOL2
SLC35D2
RP11-147C23.1
IMMP2L
PRKCI
CTB-114C7.3
LRG_174
LPIN2

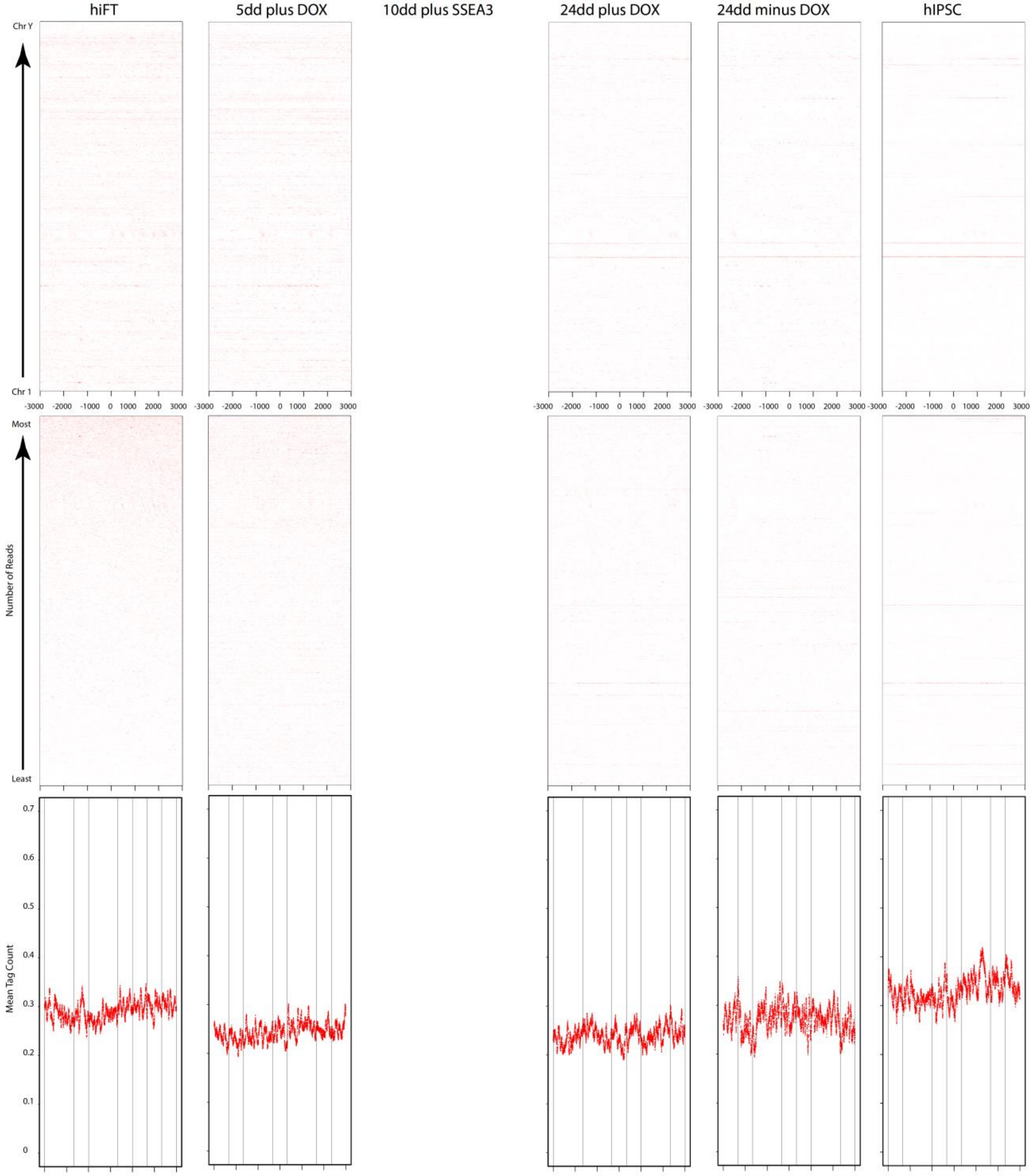
Figure Set 5: H3K27me3 histone modification, custom gene set analyses and peak enrichment change plot



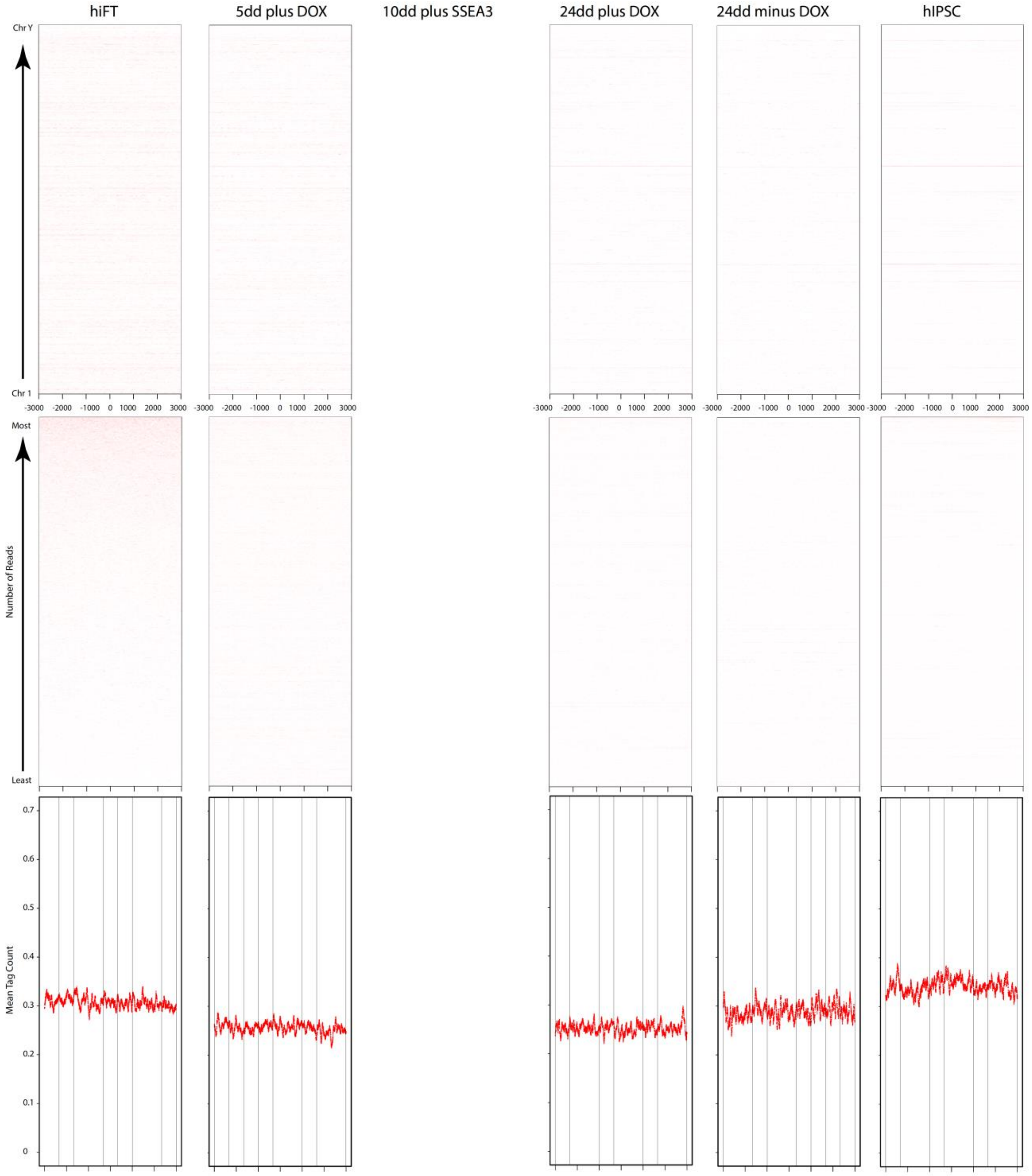
Adhesome


hiFT
5dd plus DOX
10dd plus SSEA3
24dd plus DOX
24dd minus DOX
hIPSC
Chr Y
Chr 1
-3000 -2000 -1000 0 1000 2000 3000
Most
Number of Reads
Least
Mean Tag Count
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0

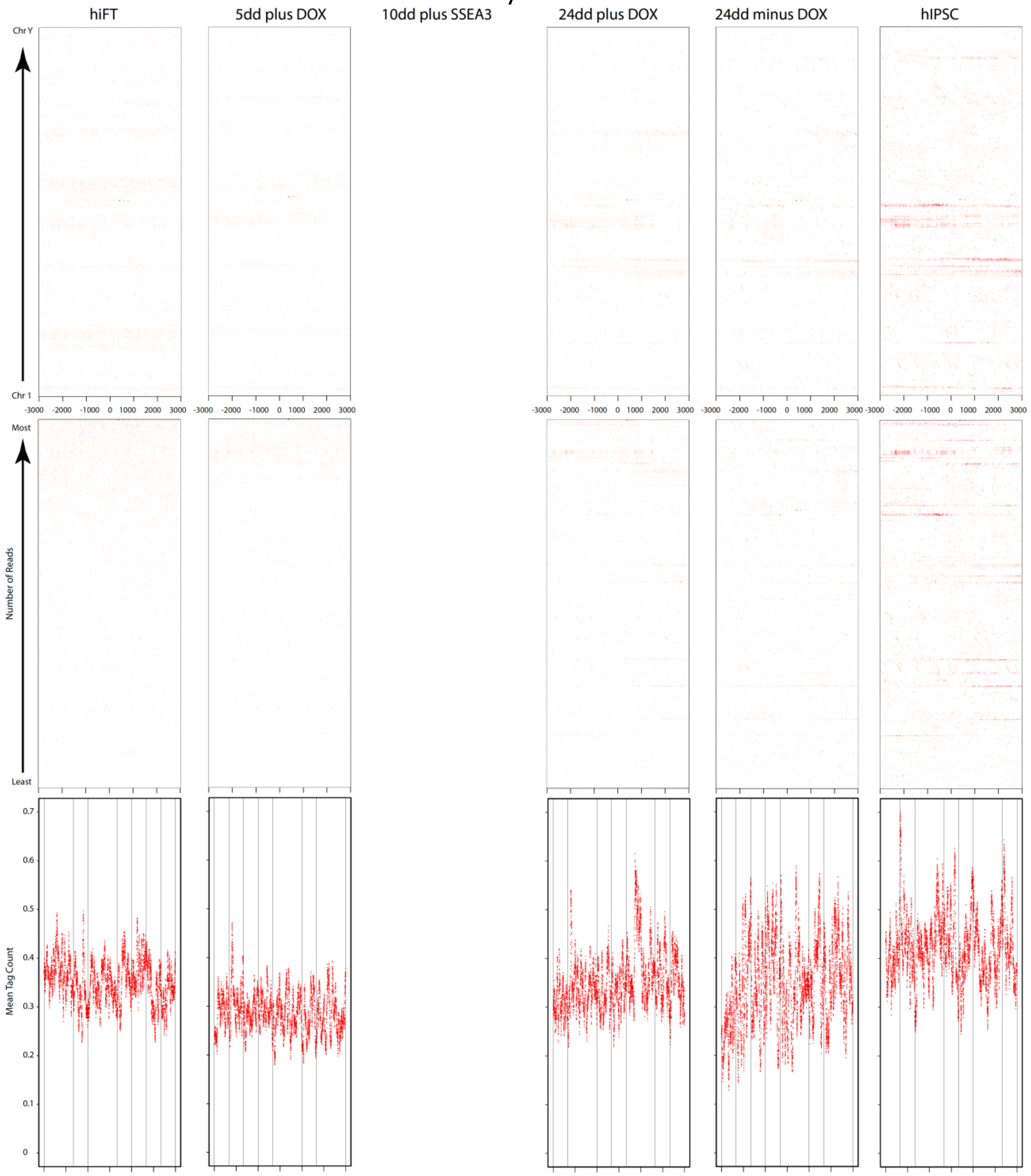
SRF


hiFT
5dd plus DOX
10dd plus SSEA3
24dd plus DOX
24dd minus DOX
hIPSC
Chr Y
Chr 1
-3000 -2000 -1000 0 1000 2000 3000
Most
Number of Reads
Least
Mean Tag Count
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0

TEAD1


hiFT
5dd plus DOX
10dd plus SSEA3
24dd plus DOX
24dd minus DOX
hIPSC
Chr Y
Chr 1
-3000 -2000 -1000 0 1000 2000 3000
Most
Number of Reads
Least
Mean Tag Count
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0

YAP/TAZ


hiFT
5dd plus DOX
10dd plus SSEA3
24dd plus DOX
24dd minus DOX
hIPSC
Chr Y
Chr 1
-3000
-2000
-1000
0
1000
2000
3000
Most
Number of Reads
Least
Mean Tag Count
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0

Late Embryogenesis

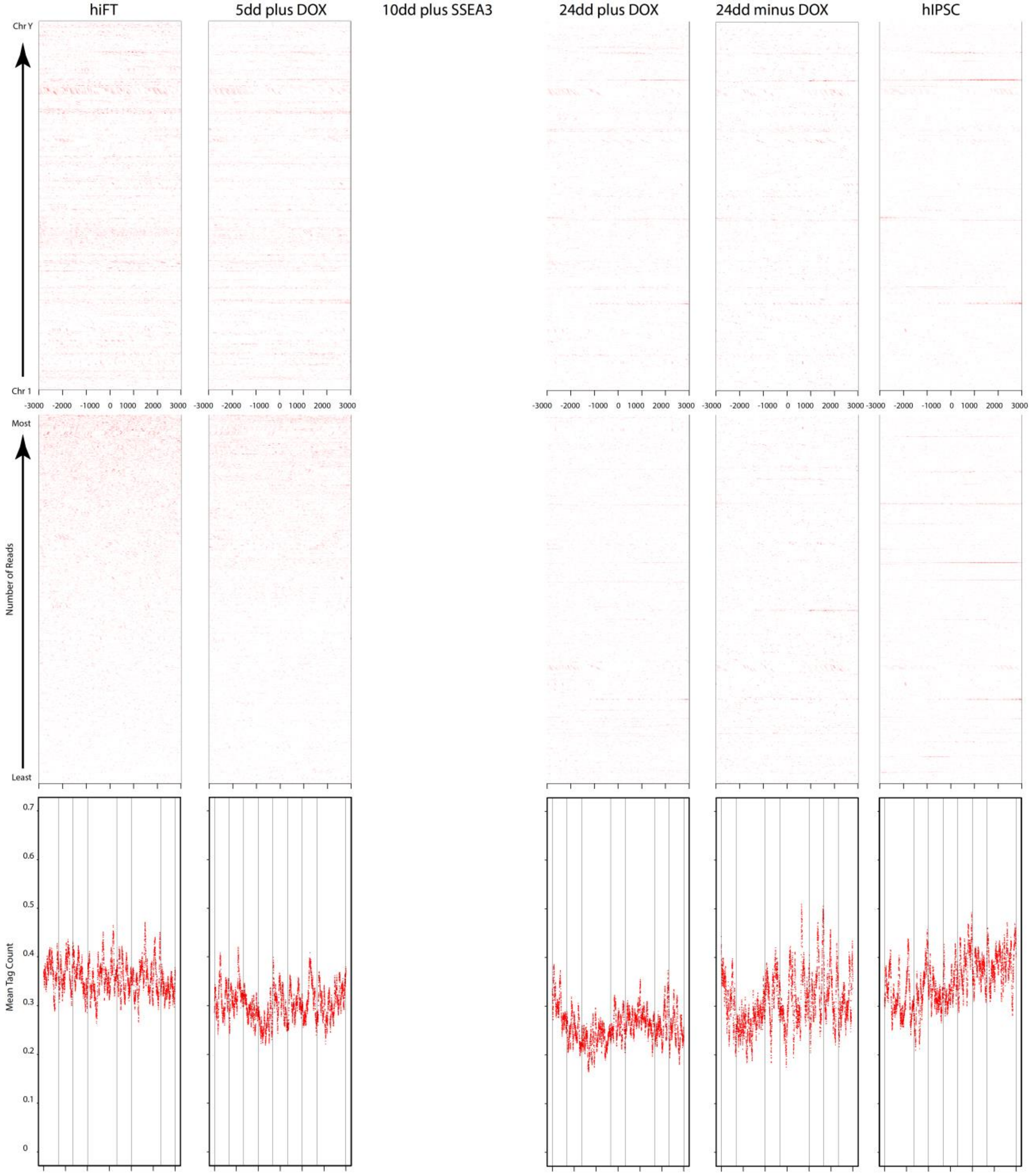
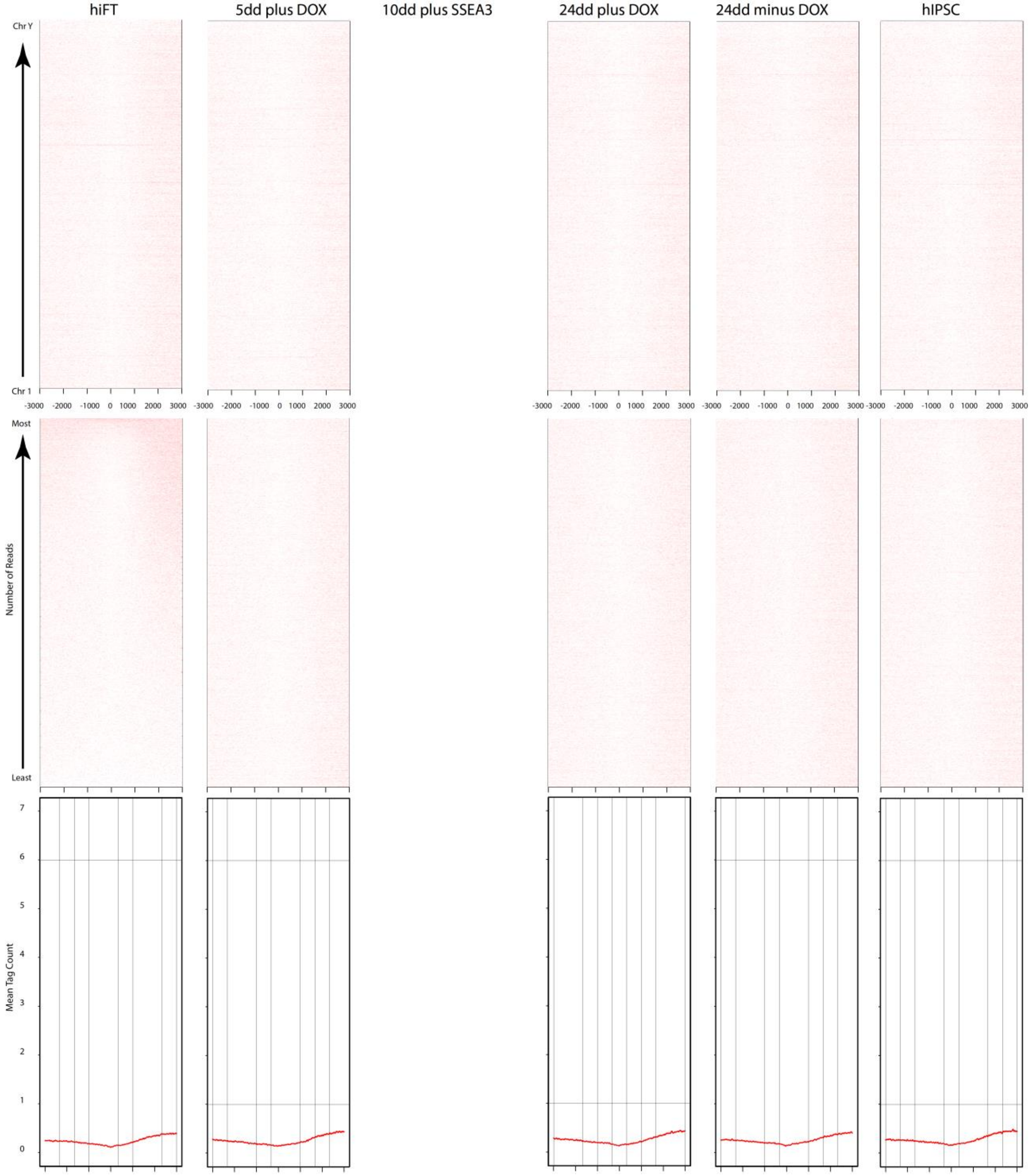
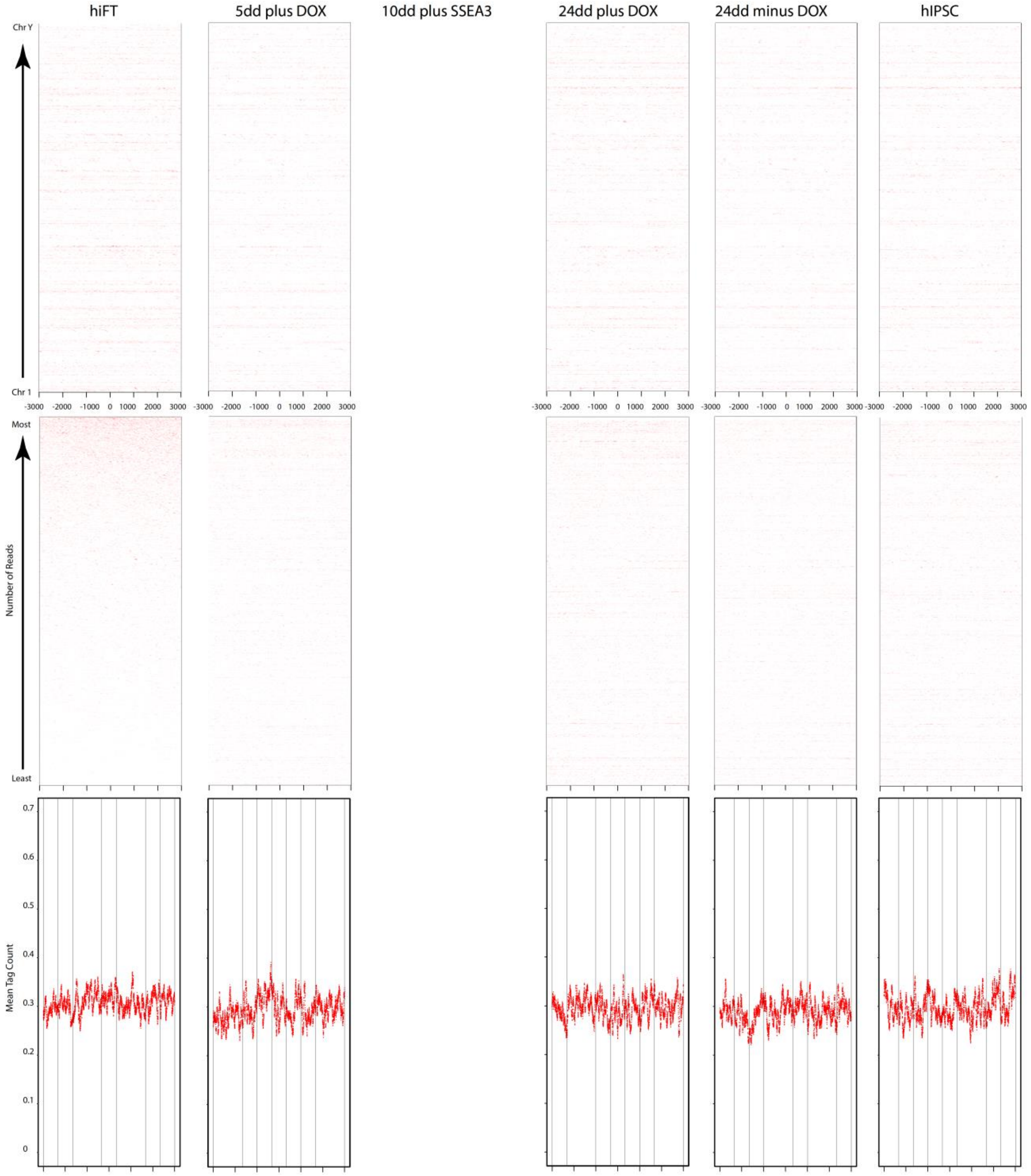


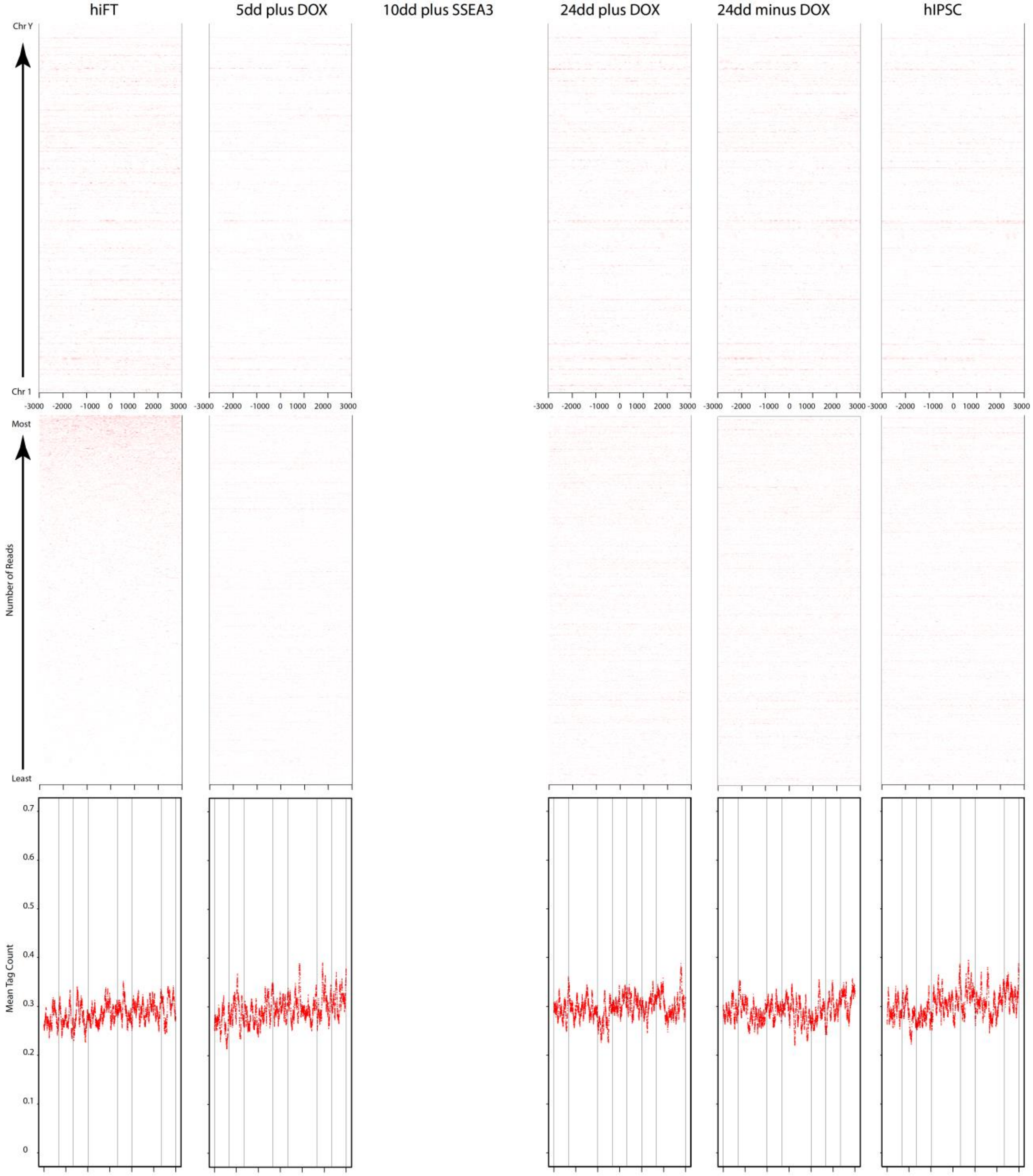
Figure Set 6: H3K36me3 histone modification, custom gene set analyses and peak enrichment change plot



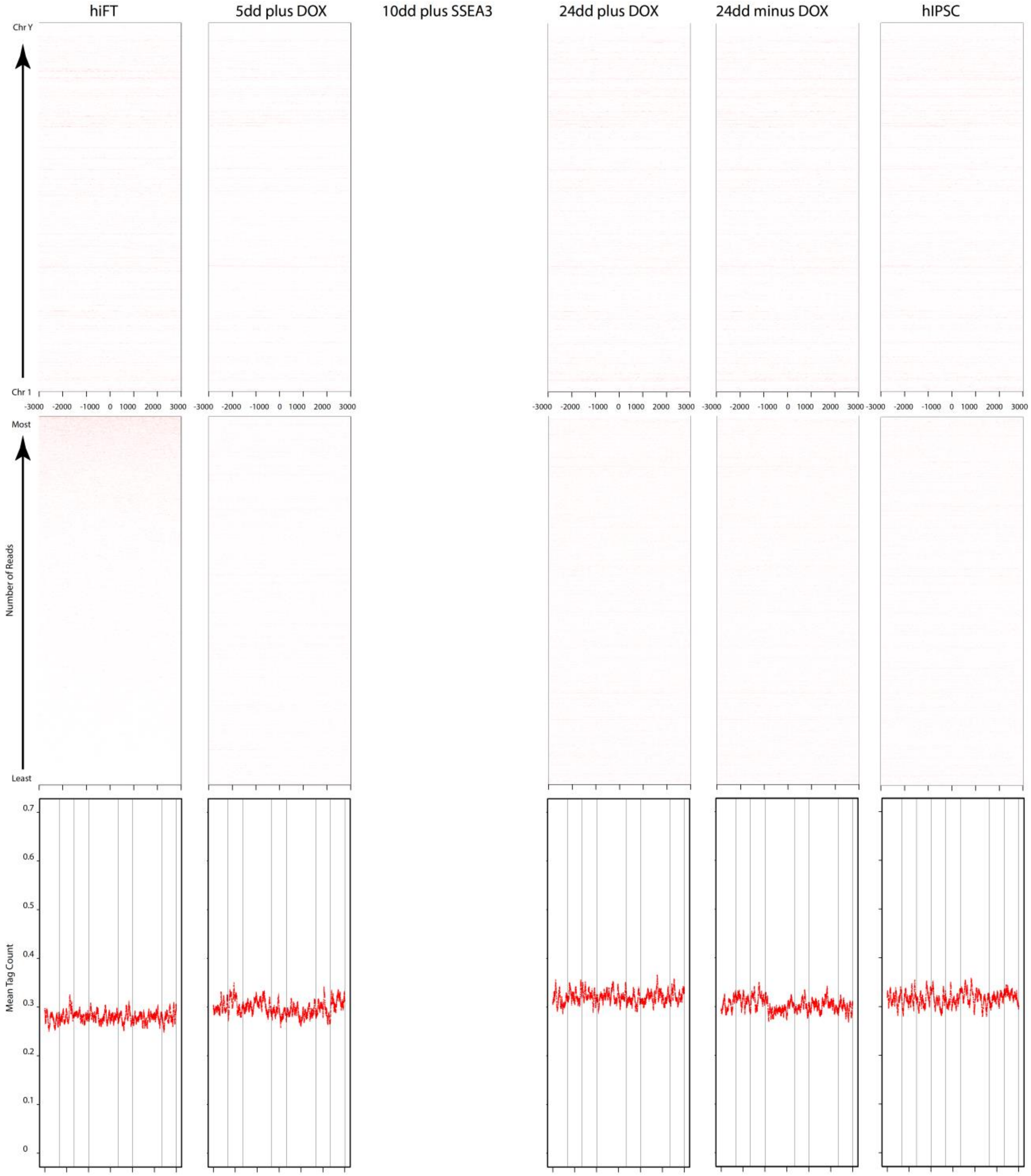
Adhesome


hiFT
5dd plus DOX
10dd plus SSEA3
24dd plus DOX
24dd minus DOX
hIPSC
Chr Y
Chr 1
-3000 -2000 -1000 0 1000 2000 3000
Most
Number of Reads
Least
Mean Tag Count
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0

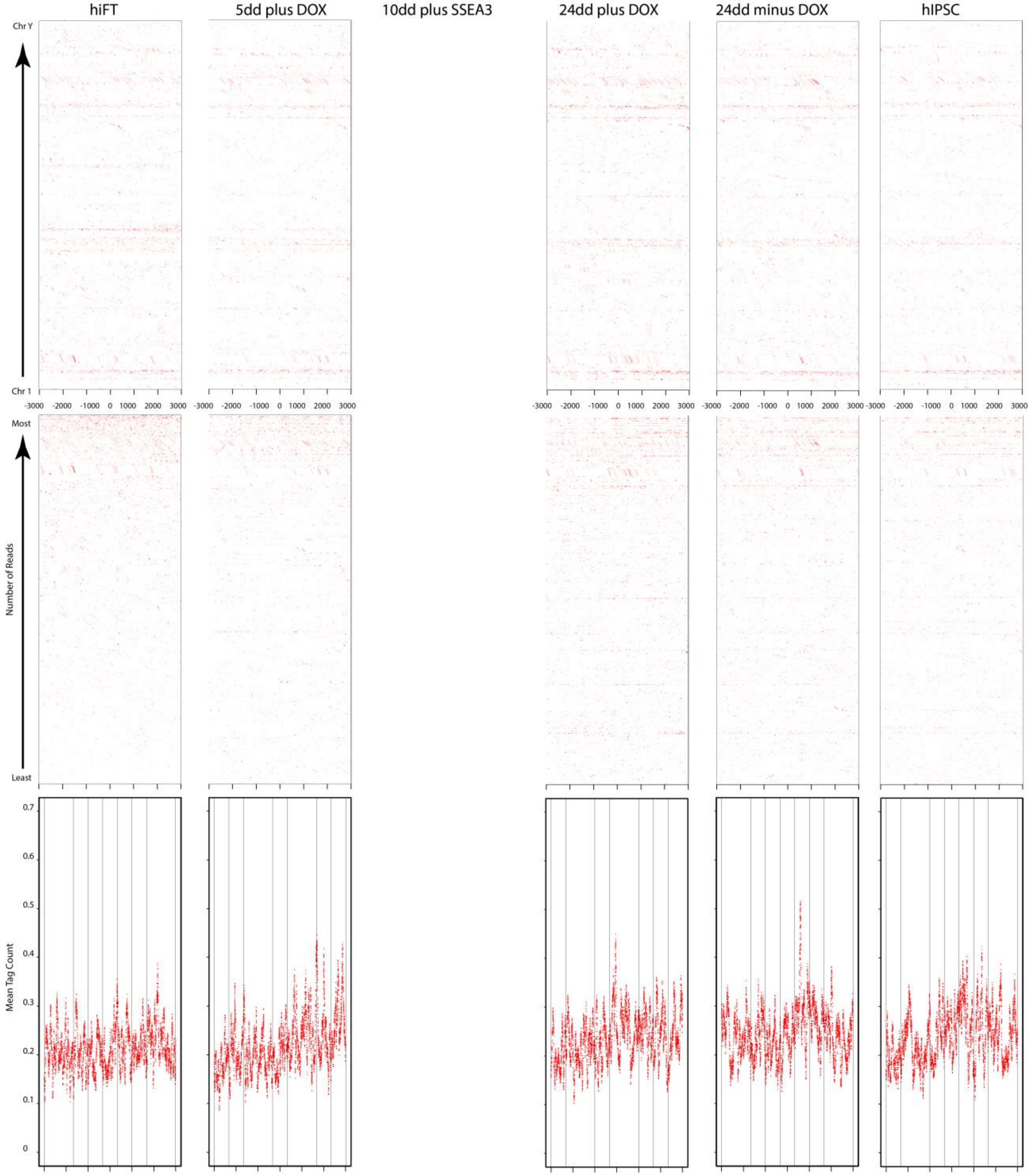
SRF


hiFT
5dd plus DOX
10dd plus SSEA3
24dd plus DOX
24dd minus DOX
hIPSC
Chr Y
Chr 1
-3000 -2000 -1000 0 1000 2000 3000
Most
Least
Number of Reads
Mean Tag Count
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0

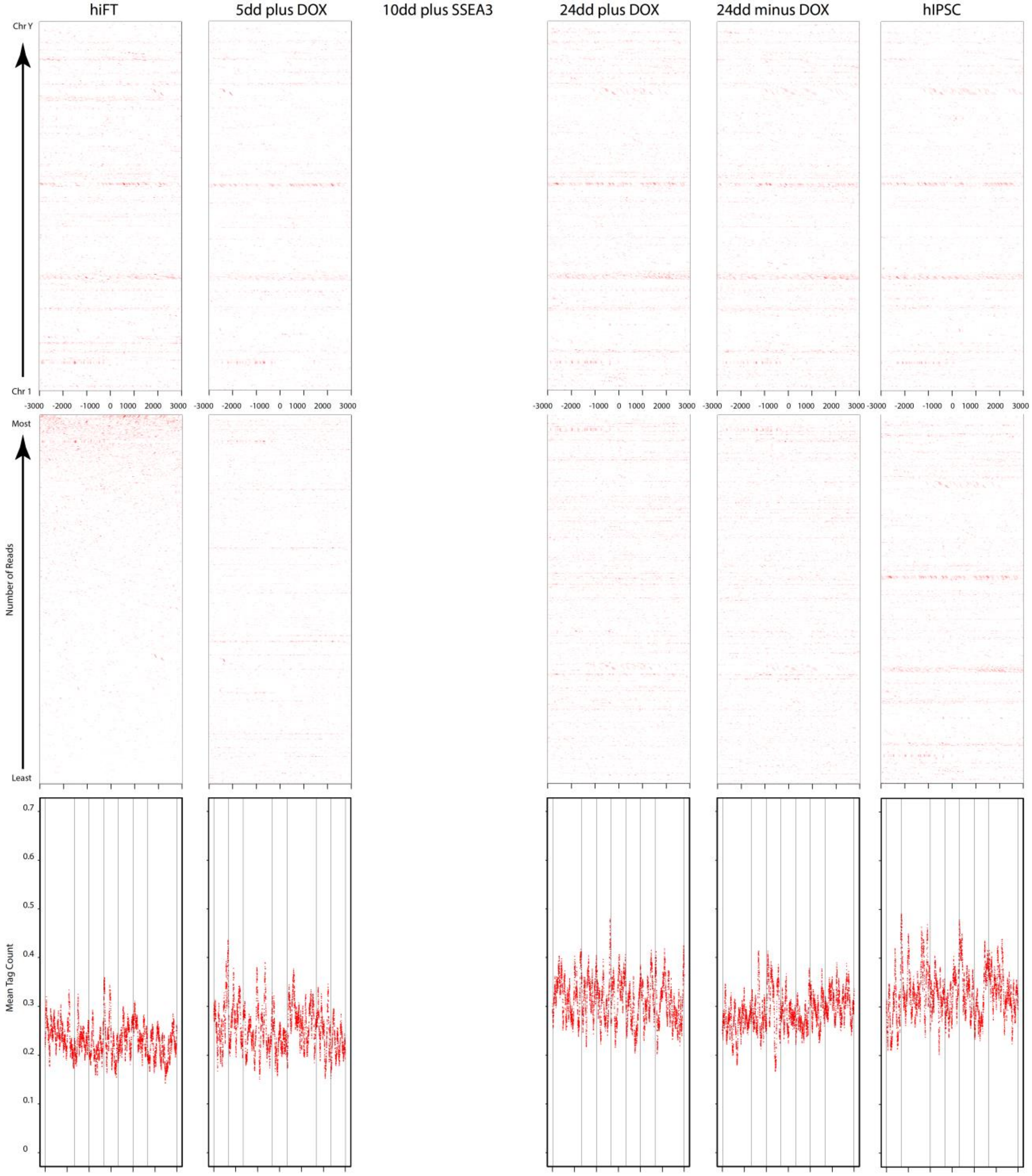
TEAD1

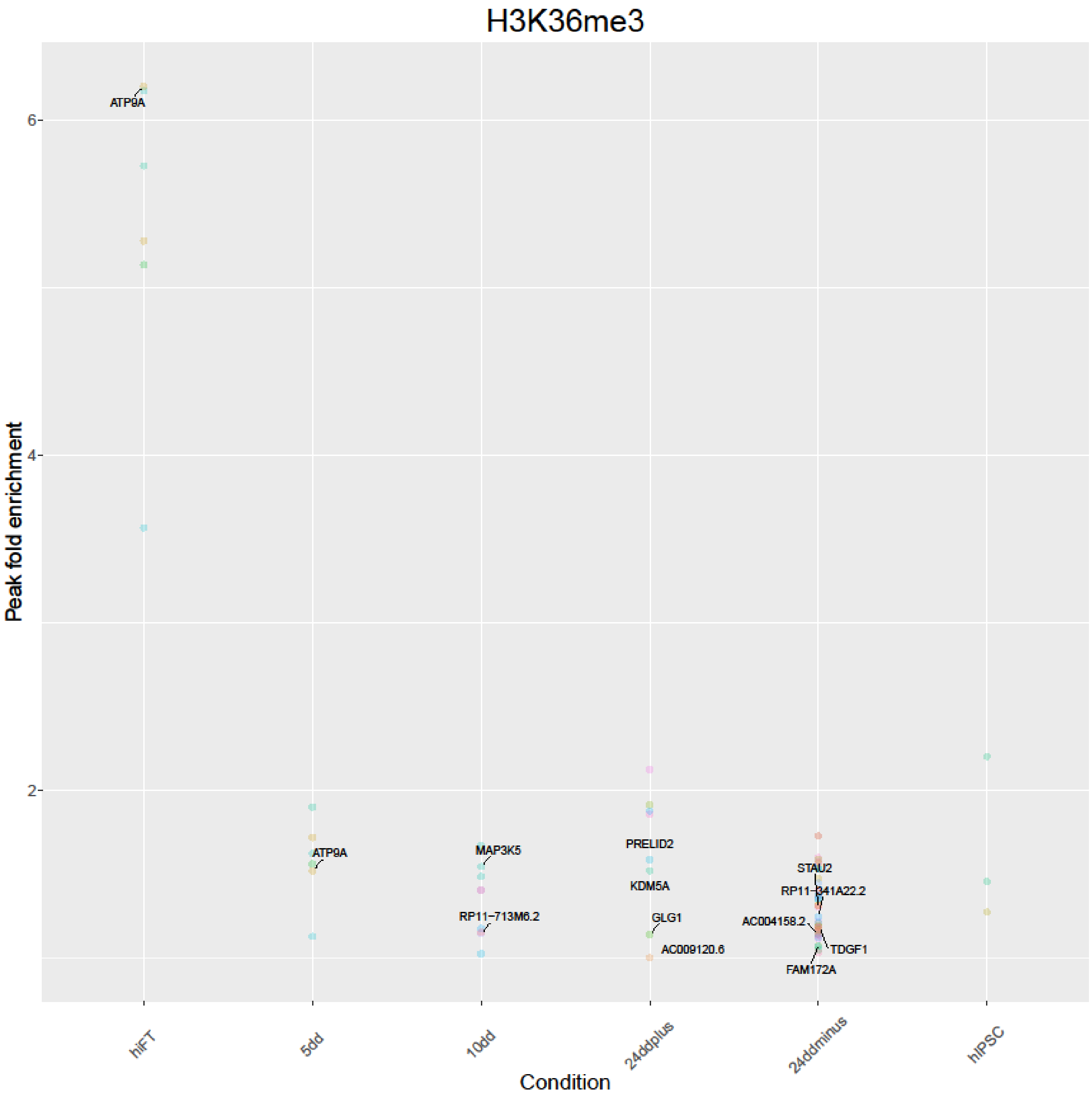

hiFT
5dd plus DOX
10dd plus SSEA3
24dd plus DOX
24dd minus DOX
hIPSC
Chr Y
Chr 1
-3000 -2000 -1000 0 1000 2000 3000
Most
Number of Reads
Least
Mean Tag Count
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0

YAP/TAZ


hiFT
5dd plus DOX
10dd plus SSEA3
24dd plus DOX
24dd minus DOX
hIPSC
Chr Y
Chr 1
-3000 -2000 -1000 0 1000 2000 3000
Most
Number of Reads
Least
Mean Tag Count
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0

Late Embryogenesis




H3K36me3
Peak fold enrichment
6
4
2
ATP9A
ATP9A
MAP3K5
RP11−713M6.2
PRELID2
KDM5A
GLG1
AC009120.6
STAU2
RP11−341A22.2
AC004158.2
TDGF1
FAM172A
hiFT
5dd
10dd
24ddplus
24ddminus
hIPSC
Condition

APPENDIX B: Code

Pre-processing raw files code

```
#!/bin/bash
# fastq to bam
module load bowtie2
bowtie2 -p 4 -x /pub/rtwest/hg19 -U /pub/rtwest/IAD2/SRR2106001.fastq > hiFT-
T_P15_H3K27ace_Rep1.sam
bowtie2 -p 4 -x /pub/rtwest/hg19 -U /pub/rtwest/IAD2/SRR2106002.fastq > hiFT-
T_P15_H3K4me1_Rep1.sam
bowtie2 -p 4 -x /pub/rtwest/hg19 -U /pub/rtwest/IAD2/SRR2106003.fastq > hiFT-
T_P15_H3K4me2_Rep1.sam
bowtie2 -p 4 -x /pub/rtwest/hg19 -U /pub/rtwest/IAD2/SRR2106004.fastq > hiFT-
T_P15_H3K4me2_Rep2.sam
bowtie2 -p 4 -x /pub/rtwest/hg19 -U /pub/rtwest/IAD2/SRR2106005.fastq > hiFT-
T_P15_H3K4me3_Rep1.sam
bowtie2 -p 4 -x /pub/rtwest/hg19 -U /pub/rtwest/IAD2/SRR2106006.fastq > hiFT-
T_P15_H3K27me3_Rep1.sam
bowtie2 -p 4 -x /pub/rtwest/hg19 -U /pub/rtwest/IAD2/SRR2106007.fastq > hiFT-
T_P15_H3K36me3_Rep1.sam
bowtie2 -p 4 -x /pub/rtwest/hg19 -U /pub/rtwest/IAD2/SRR2106008.fastq > hiFT-
T_P15_INP_Rep1.sam
bowtie2 -p 4 -x /pub/rtwest/hg19 -U /pub/rtwest/IAD2/SRR2106009.fastq > hIPSC-
T_P10_H3K27ace_Rep1.sam
bowtie2 -p 4 -x /pub/rtwest/hg19 -U /pub/rtwest/IAD2/SRR21060010.fastq > hIPSC-
T_P10_H3K27me3_Rep1.sam


#!/bin/bash
#module load samtools
#module load homer
#module load bedops
mkdir -p /pub/rtwest/IAD2/ChIP-SeqSamples/_${file}/
cd /pub/rtwest/IAD2/ChIP-SeqSamples/_${file}/
filedir=/pub/rtwest/IAD2/ChIP-SeqSamples/_${file}/
samtools view -bS /pub/rtwest/IAD2/ChIP-SeqSamples/${file}.sam > ${filedir}${file}.bam
samtools sort ${filedir}${file}.bam -o ${filedir}${file}_Sorted.bam
samtools view -b -F 0x400 ${filedir}${file}_Sorted.bam >
${filedir}${file}_Sorted_PCRDupesRemoved.bam
samtools index ${filedir}${file}_Sorted_PCRDupesRemoved.bam
samtools view -h ${filedir}${file}_Sorted_PCRDupesRemoved.bam >
${filedir}${file}_Sorted_PCRDupesRemoved.sam

#makeTagDirectory and finding peaks for TF, Histone and TSS
makeTagDirectory $filedir ${filedir}${file}_Sorted_PCRDupesRemoved.sam

findPeaks ${filedir} -style factor -o ${filedir}/${file}_peaks.txt

findPeaks ${filedir} -style histone -o ${filedir}/${file}_regions.txt

findPeaks ${filedir} -style tss -o ${filedir}/${file}_tss.txt

#annotatePeaks
annotatePeaks.pl ${file}_peaks.txt hg19 > ${file}_AnnotatedPeaks.txt
annotatePeaks.pl ${file}_regions.txt hg19 -size 6000 -hist 10 -ghist -d ${filedir} >
${file}_heatmapMatrix.txt

#convert annotated peak files to bed
pos2bed.pl ${file}_AnnotatedPeaks.txt > ${file}_peak.bed
pos2bed.pl ${file}_heatmapMatrix.txt > ${file}_heatmap.bed
pos2bed.pl ${file}_regions.txt > ${file}_regions.bed

#convert to bed
sam2bed < ${filedir}${file}_Sorted_PCRDupesRemoved.sam > ${filedir}${file}.bed



#!/bin/bash
#module load samtools
#module load macs2/2.0.10
```

```
cd /pub/rtwest/IAD2/ChIP-SeqSamples/_${file}/

filedir=/pub/rtwest/IAD2/ChIP-SeqSamples/_${file}/

macs2 callpeak -t ${file}_Sorted.bam -f BAM -g hs -n ${file} --broad

chmod +x ${file}_peaks.xls

#removing blank space and comments in header
sed -i '/#/d' ${file}_peaks.xls
sed -i '/^$/d' ${file}_peaks.xls
```

HeatEnv – RStudios environment to load packages and create fundamental background information for running GRanges conversions, Matrix Generation, Heatmap Generation and Overlapping Plots

```
LoadHeat <- function(){
  HeatEnv <<- new.env(parent=globalenv())
  HeatEnv <- get("HeatEnv", envir=.GlobalEnv)

  if (!requireNamespace("BiocManager"))
    install.packages("BiocManager")
  if (!requireNamespace(c("XML", "RCurl", "GenomicFeatures", "stringi",
"GenomicRanges", "mgcv", "TxDb.Hsapiens.UCSC.hg19.knownGene", "data.table")))
    BiocManager::install(c("XML", "RCurl", "GenomicFeatures", "stringi",
"GenomicRanges", "mgcv", "TxDb.Hsapiens.UCSC.hg19.knownGene", "data.table",
suppressUpdates=TRUE))
  require(GenomicFeatures)
  require(gtools)
  require(TxDb.Hsapiens.UCSC.hg19.knownGene)
  require(data.table)
  require(GenomicRanges)
  require(GenomicAlignments)
  require(tidyverse)
  require(dplyr)
  require(fuzzyjoin)
  require(ggplot2)
  require(ggrepel)
  require(scales)
  require(EnsDb.Hsapiens.v75)
  require(org.Hs.eg.db)

  upstream <- 3000
  assign("upstream", 3000, envir= HeatEnv)
  downstream <- 3000
  assign("downstream", 3000, envir= HeatEnv)

  if (exists("upstream", envir=HeatEnv, inherits=FALSE) &&
      exists("downstream", envir=HeatEnv, inherits=FALSE) ) {
    ups <- HeatEnv[["upstream"]]
    downs <- HeatEnv[["downstream"]]
  }
  if (exists("txdb", envir=HeatEnv, inherits=FALSE)){
    txdb <- HeatEnv[["txdb"]]
  }
  if (ups == upstream && downs == downstream &&
       exists("tsses", envir=HeatEnv, inherits=FALSE) ){
  tsses <- HeatEnv[["tsses"]]
  }

  txdb <<- TxDb.Hsapiens.UCSC.hg19.knownGene
  ##the standard Transcription Start Sites of the genome
  Genes <- genes(txdb)
  Genes
  ## get start position based on strand
  tss <- ifelse(strand(Genes) == "+", start(Genes), end(Genes))
  bins <- GRanges(seqnames=seqnames(Genes),
                  ranges=IRanges(tss-HeatEnv[["downstream"]],
tss+HeatEnv[["upstream"]]),
                  strand=strand(Genes))
  tsses <- unique(bins)
  tsses <- sortSeqlevels(tsses)
  tsses <- sort(tsses)
  assign("tsses", tsses, envir=HeatEnv)
  assign("tsses", tsses, envir=.GlobalEnv)
  HeatEnv$tsses=tsses
  return(tsses)
}

LoadHeat()

Change.Promoter.Window <- function(txdb=NULL,
                  upstream=3000,
                  downstream=3000) {

  HeatEnv <- get("HeatEnv", envir=.GlobalEnv)

  if ( exists("upstream", envir=HeatEnv, inherits=FALSE) &&
       exists("downstream", envir=HeatEnv, inherits=FALSE) ) {
    ups <- HeatEnv[["upstream"]]
    downs <- HeatEnv[["downstream"]]
    if (ups == upstream && downs == downstream &&
        exists("tsses", envir=HeatEnv, inherits=FALSE) ){
      tsses <- HeatEnv[["tsses"]]
    }
  }
  Genes <- genes(txdb)
  tss <- ifelse(strand(Genes) == "+", start(Genes), end(Genes))
  bins <- GRanges(seqnames=seqnames(Genes),
                  ranges=IRanges(tss-HeatEnv[["downstream"]],
tss+HeatEnv[["upstream"]]),
                  strand=strand(Genes))
  tsses <- unique(bins)

  assign("tsses", tsses, envir=HeatEnv)
  return(tsses)
}
```

to_granges

```
to_granges <- function(file){
  if(is.data.frame(file) == T){
    if(length(file) > 6){
      file <- file[,-c(7:length(file))]
    }
    names <- c(“chr”,”start”,”end”,”id”,”score”,”strand”)
    mixedsort([order(nchar(file), file)])
    names(file) <- names[1:length(names(file))]

    if(“strand” %in% colnames(file)){
      file$strand <- gsub(pattern="-+", replacement = “*”, x = file$strand)
    }

    if(length(file)==3){
      granges <- with(file, GRanges(chr, IRanges(start, end)))
    } else if (length(file)==4)
       {
      granges <- with(file, GRanges(chr, IRanges(start, end), id=id))
    } else if (length(file)==5)
       {
      granges <- with(file, GRanges(chr, IRanges(start, end), id=id, score=score))
    } else if (length(file)==6)
       {
      granges <- with(file, GRanges(chr, IRanges(start, end), id=id, score=score,
strand=strand))
    }

return(granges)

  } else {

  if(length(file) > 5){
  file <- data.frame(fread(file, select = c(1,2,3,4,5), fill=T))
  }
  file <- data.frame(fread(file, select = c(1,2,3,4,5,6), fill=T))

  if(length(file) > 6){
    file <- file[,-c(7:length(file))]
  }

  names <- c(“chr”,”start”,”end”,”id”,”score”,”strand”)
  mixedsort([order(nchar(file), file)])
  names(file) <- names[1:length(names(file))]

  if(“strand” %in% colnames(file)){
    file$strand <- gsub(pattern="-+", replacement = “*”, x = file$strand)
  }

  if(length(file)==3){
    granges <- with(file, GRanges(chr, IRanges(start, end)))
  } else if (length(file)==4)
       {
    granges <- with(file, GRanges(chr, IRanges(start, end), id=id))
  } else if (length(file)==5)
       {
    granges <- with(file, GRanges(chr, IRanges(start, end), id=id, score=score))
  } else if (length(file)==6)
       {
    granges <- with(file, GRanges(chr, IRanges(start, end), id=id, score=score,
strand=strand))
  }

return(granges)
}

}
```

Matrix Generator

```
#' Generate Matrix for plotting
#'
#' @param Grangefile, A GRanges object with at a minimum the chr, start and end.
Typically the output of bed_to_granges
#' @param tsses, A GRanges object of the start and end of desired segment of DNA
(promoters) to view overlap of ChIP-Seq file. Standard output of the HeatEnv (and
corresponding input to this function) are the TSSes of the genome
#' @return Matrix, matrix from rows Chromosome 1 to Y of ChIP-Seq data frequency
overlapping with inputted promoters
#' @examples overlap.matrix <- OverlapMatrix(Grange_file, tsses)
#'
OverlapMatrix <- function(Grangefile, tsses) {
  HeatEnv <- get("HeatEnv", envir=.GlobalEnv)

  ##number of reads that cover each position in the genome and then defining bins
  coverage <- sortSeqlevels(coverage(Grangefile))
  length <- elementNROWS(coverage)
  width <- GRanges(seqnames=names(length),
                  IRanges(start=rep(1, length(length)),
                  end=length))

  ##setting bin for matrix of the defined Transcription Start Sites
  tsses <- subsetByOverlaps(tsses, width, type="within", ignore.strand=TRUE)
  tsses <- tsses[order(seqnames(tsses))]

  ##finding the coverage overlap on each TSS and sorting by chromosome, then combine
overlaps into Rle (Run length encoded) ranges that overlap
  chroms <- intersect(names(coverage), unique(as.character(seqnames(tsses))))
  chroms <- mixedsort(chroms[order(nchar(chroms), chroms)])
  overlaps <- Views(coverage[chroms], as(tsses, "IntegerRangesList")[chroms])

  ##vectorizing the list of overlaps with the tss bins, and then putting each vector
into a matrix by row
  overlap.list <- lapply(overlaps, function(x) t(viewApply(x, as.vector)))
  overlap.matrix <- do.call("rbind", overlap.list)
  ##index of TSS and reorganize as a list of intersecting chromosomes ranges of the
TSS and create a vector through c()
  chroms.list <- split(1:length(tsses), as.factor(seqnames(tsses)))
  intersect <- do.call("c", chroms.list)

  ##creating the list of start and stop of the IRanges in each chromosome from the TSS
and ordering the matrix based on position of chromosome for consitency
  rownames(overlap.matrix) <- intersect
  overlap.matrix <- overlap.matrix[order(intersect),]

  ##ensuring correct orentation of the ChIP-seq data within the matrix and flipping
minus strand
  minus.str <- which(as.character(strand(tsses)) == "-")
  overlap.matrix[minus.str,] <- overlap.matrix[minus.str, ncol(overlap.matrix):1]

  ##create the matrix through summing up each overlap of peak data within TSS window
  overlap.matrix <- overlap.matrix[rowSums(overlap.matrix)!=0,]

  ## assign("overlap.matrix", overlap.matrix, envir=ChIPseekerEnv)
  return(overlap.matrix)
}
```

Custom Promoter Window

```
#' Generate custom promoter window for heatmap generation
#'
#' @param promoterlist, an .xls, .xlsx, .bed or .csv list of genes/promoters that want
to be utilized for comparison
#' @return custom.granges, A GRanges object of the start and end of desired segment of
DNA (promoters) to view overlap of ChIP-Seq file
#' @examples CustomTXDB("yaptaz_promoters.xlsx")

CustomTXDB <- function(promoterlist = "")
{ require(GenomicRanges)
  HeatEnv$promoterlist <- promoterlist
  if(grepl("*.xlsx",promoterlist) || grepl("*.xls",promoterlist)){
  require(readxl)
  custompromoters <- read_excel(promoterlist)
  custompromoters <- custompromoters[, order(names(custompromoters))]
    if(length(custompromoters) > 7 ){
      custompromoters <- custompromoters[,-c(8:length(custompromoters))]
      setnames(custompromoters, c("chr", "strand", "score", "end", "id", "start",
"type"))
      custom_order <-
custompromoters[order(custompromoters$chr,custompromoters$start), c(1,2,3,4,5,6,7)]
    }
    if(length(custompromoters)== 7){
      setnames(custompromoters, c("chr", "strand", "score", "end", "id", "start",
"type"))
       custom_order <-
custompromoters[order(custompromoters$chr,custompromoters$start), c(1,2,3,4,5,6,7)]
    }
      if(length(custompromoters)== 6){
        setnames(custompromoters, c("chr", "strand", "score", "end", "id", "start"))
        custom_order <-
custompromoters[order(custompromoters$chr,custompromoters$start), c(1,2,3,4,5,6)]
    }
    custom_order <- custom_order[custom_order$chr %in% c((1:22),"X","Y"),]
    custom_order$strand <- gsub("-1", "-", custom_order$strand)
    custom_order$strand <- gsub("1", "+", custom_order$strand)
    custom_order$chr <- paste("chr",custom_order$chr,sep="")

    custom.granges <- makeGRangesFromDataFrame(custom_order, keep.extra.columns =
TRUE)
    custom.granges <- sort(custom.granges)
  }
    else if(grepl("*.bed",promoterlist)){
    custom.granges <- unique(to_granges(promoterlist))
    custom.granges <- sortSeqlevels(custom.granges)
    custom.granges <- sort(custom.granges)
  }
    else if(grepl("*.tsv",promoterlist)){
```

```
    custompromoters <- read.table(file = promoterlist, sep = '\t', header = TRUE)
    custom.granges <- makeGRangesFromDataFrame(custompromoters)
  }
  assign("custom.granges", custom.granges, envir=HeatEnv)
  HeatEnv$custom.granges <- custom.granges
  return(custom.granges)
}

##load list to http://uswest.ensembl.org/biomart/martview
##**important** select Ensemble genes 86, Human genes(GRCh38.p12) filters->GENE and
##paste genes, then Attributes and select for chr/scaffold name, strand, tx start,
##tx end,tx name,tx count(score),tx type. Please limit to these options for conversion
purposes
##click results and output, remove hyperlinks and save accordingly
```

Targeted Heatmap ChIP-Seq

```
#' Core function to generate Heatmap and Mean Tag Count of Chip-Seq data
#'
#' @param Modification, A string of Histone Modification file name (can be a GRange
file as well)
#' @param control, A string of Histone Modification file name of control, should match
the Histone Modification type
#' @param xlab,, A string, can add custom x label
#' @param ylab A string, can add custom y label, Mean Tag Count will automatically be
generated for Mean Tag Count graph
#' @param orderByReads, if TRUE a control String Modification must be inputed and will
order the Heatmap the same order as the control number of reads
#' @param withCustom, if TRUE will only create a heatmap for the defined TSS/promoters
#' @param set your working directory
#' @return 2 PDFs a Heatmap of the Modification, sorted from Chromosome 1 to
Chromosome Y or by number of reads from low to high based on the control or startpoint
condition of the modification as well as a PDF of the Mean Tag Count over the desired
TSS/promoter window or the same as previously stated over a customized promoter window
#' @examples
#' GenHeat("hIPSC-T_P10_H3K4me1_Rep1_peaks.xls", "hiFT-T_P15_H3K4me3_Rep1_peaks.xls",
xlab="TSS", ylab="Genome location", T, F)
#' GenHeat("5dd_DOX_plus_H3K4me2_merged", "hiFT-T_P15_H3K4me2_Rep1", xlab="TSS",
ylab="Genome location", F, F)
#' GenHeat("5dd_DOX_plus_H3K4me1_Rep1", "hiFT-T_P15_H3K4me1_Rep1", xlab="TSS",
ylab="Genome location", F, T)

GenHeat <- function(Modification, control, xlab="", ylab="",orderByReads = FALSE,
withCustom = FALSE, WorkDir = "/dfs3/pub/rtwest/IAD2/ChIP-SeqSamples/"){
  ##my subfolders where the .bed files are stored
  setwd(WorkDir)

  ##identifying file type for the desired modification and converting to GRange object
and creating heatmap matrix, automatically defaults to .bed if no file extension is
specified
  if(isS4(Modification)){
      overlap.matrix <<- OverlapMatrix(Modification, tsses)
      Modification <- toString(Modification, width = 12)
  }else{
    if(isFALSE(exists("Grange_file")) || isTRUE(Modification!=HeatEnv$Modification) ||
isFALSE(exists("Modification"))){
      if(isTRUE(grepl(".bed", Modification))){
      Grange_file <<- to_granges(paste("_",str_remove(Modification,
".bed"),"/",Modification,sep=""))
      }else if(isTRUE(grepl(".xls", Modification))){
      Grange_file <<- to_granges(paste("_",str_remove(Modification,
"_peaks.xls"),"/",Modification,sep=""))
      }else{
      file <- paste("_",Modification,"/",Modification,".bed", sep="")
      Grange_file <<- to_granges(file)
  }
  assign("Modification", Modification, envir=HeatEnv)
  overlap.matrix <<- OverlapMatrix(Grange_file, tsses)
  }
  }
  ##defining control, defaults to .bed if no file extension is specified
  if(isTRUE(orderByReads) && (isTRUE(control!=HeatEnv$control) ||
isFALSE(exists("overlap.matrix.hift")) || isFALSE(exists("controlfile")))){
    if(isTRUE(grepl(".bed", control))){
      controlfile <<- to_granges(paste("_",str_remove(control,
".bed"),"/",control,sep=""))
    }else if(isTRUE(grepl(".xls", control))){
      controlfile <<- to_granges(paste("_",str_remove(control,
"_peaks.xls"),"/",control,sep=""))
    }else{
      control_file <- paste("_",control,"/",control,".bed", sep="")
      controlfile <<- to_granges(control_file)
    }
  overlap.matrix.hift <<- OverlapMatrix(controlfile, tsses)
  assign("overlap.matrix.hift", overlap.matrix.hift, envir=HeatEnv)
  assign("control", control, envir=HeatEnv)
  }
  ##control matrix needed to sort other matrices
  if(isTRUE(orderByReads) && isFALSE(exists("controlfile"))){
    message("Control file for ordering is not defined. Provide desired control (e.g.
hiFT) file.
    To define initial control heatmap, Modification and controlfile should be inputted
and the same histone modification")
  }
  ##core function to genearte heatmap PDF and sort as required
  Heatmap <- function(overlap.matrix, xlim=c(-3000, 3000), xlab="", ylab="",
title=Modification, orderByReads = F, withCustom=F) {
    cols <- colorRampPalette(c("white","red"))(100)
    overlap.matrix <- t(apply(overlap.matrix, 1, function(x) x/max(x)))
    nrow.hift <<- nrow(get("overlap.matrix.hift", envir=HeatEnv))
    nrow.matrix <- nrow(overlap.matrix)

    ##creating a new matrix based on custom TSS/promoter windows defined by user
    if (isTRUE(withCustom)) {
      if(isTRUE(exists("custom.granges",HeatEnv))==F){
        message("Custom TxDb is not defined. Run Custom_TxDb with desired file")
      } else {
        cgr <- HeatEnv$custom.granges
        tss_custom <- ifelse(strand(cgr) == "+", start(cgr), end(cgr))
        bins_custom <- GRanges(seqnames=seqnames(cgr),
                               ranges=IRanges(tss_custom-3000, tss_custom+3000),
                               strand=strand(cgr))
        promoter_custom <- unique(bins_custom)
        overlap.matrix.custom <<- OverlapMatrix(Grange_file, promoter_custom)

        ##option to order the heatmap based on control
        if (isTRUE(orderByReads)){
          overlap.matrix.hift.custom <<- OverlapMatrix(controlfile, promoter_custom)
          nrow.matrix.custom <- nrow(overlap.matrix.custom)
          nrow.hift.custom <- nrow(overlap.matrix.hift.custom)
          if(nrow.matrix.custom > nrow.hift.custom) {
            for(k in 1:(nrow.matrix.custom-nrow.hift.custom))
            { l <- c(k)
              overlap.matrix.hift.custom <- rbind(overlap.matrix.hift.custom, l)
            }
          }
          else if(nrow.matrix.custom < nrow.hift.custom){
            order(rowSums(overlap.matrix.hift.custom))
            overlap.matrix.hift.custom <- overlap.matrix.hift.custom[-
((nrow.matrix.custom+1):nrow.hift.custom), , drop=FALSE]
          }
          j <- order(rowSums(overlap.matrix.hift.custom))
          overlap.matrix.custom <<- overlap.matrix.custom[j,]
        }

        ##creating Mean Tag Count of matrix over the TSS, can change ylim as needed if
topp limit does not capture the highest read count

pdf(paste("MeanTagCount",Modification,if(isTRUE(withCustom))HeatEnv$promoterlist,if(is
TRUE(orderByReads))"sorted",".pdf", sep="_"), width = 4, height=9)
        plot(x=seq(-3000, 3000, length.out=6001),
            y=colMeans(overlap.matrix.custom),
            ty="b", pch=18,
             ylim = c(0,0.7),
             cex = .3,
             col="red",
             ylab="Mean tag count",
             xlab="Distance from TSS (bp)",
             main=Modification)
        abline(h=seq(1,100,by=5), v=seq(-3000, 3000, length.out=10), lwd=0.2,
col="black")
        box(col="black", lwd=2)
        dev.off()
      }
    } else if (isTRUE(orderByReads)){
      if(nrow.matrix > nrow.hift) {
        for(k in 1:(nrow.matrix-nrow.hift))
        { l <- c(k)
        overlap.matrix.hift <- rbind(overlap.matrix.hift, l)
        }
      }
      else if(nrow.matrix < nrow.hift){
        order(rowSums(overlap.matrix.hift))
        overlap.matrix.hift <- overlap.matrix.hift[-((nrow.matrix+1):nrow.hift), ,
drop=FALSE]
      }
        j <- order(rowSums(overlap.matrix.hift))
        overlap.matrix <- overlap.matrix[j,]
      }

    if (is.null(xlim)) {
      xlim <- 1:ncol(overlap.matrix)
    } else if (length(xlim) == 2) {
      xlim <- seq(xlim[1], xlim[2])
    }
    #Heatmap pdf
    if (isTRUE(withCustom)){
    pdf(paste(HeatEnv$promoterlist, Modification, if(isTRUE(orderByReads))"sorted",
".pdf", sep="_"), width = 4, height=9)
    image(x=xlim,
          y=1:nrow(overlap.matrix.custom),
          z=t(overlap.matrix.custom),
          useRaster=TRUE,
          col=cols,
          yaxt="n",
          ylab="",
          xlab=xlab,
          main=title)
    dev.off()
    }
    ##Heatmap pdf
    if (isTRUE(withCustom)==F){
    pdf(paste(Modification,if(isTRUE(orderByReads))"sorted",".pdf", sep="_"), width =
4, height=9)
    image(x=xlim,
          y=1:nrow(overlap.matrix),
          z=t(overlap.matrix),
          useRaster=TRUE,
          col=cols,
          yaxt="n",
          ylab="",
          xlab=xlab,
          main=title)
    dev.off()
    }
    if(isTRUE(nrow(overlap.matrix.hift) != nrow.hift)){
      on.exit(overlap.matrix.hift <- OverlapMatrix(controlfile, tsses))
    }
    if(isTRUE(withCustom) && isTRUE(orderByReads)){
      if(isTRUE(nrow(overlap.matrix.hift.custom) != nrow.hift.custom)){
      on.exit(overlap.matrix.hift.custom <- OverlapMatrix(controlfile,
promoter_custom))
    }}
  }
```

```
    ##creating Mean Tag Count of matrix over the TSS, can change ylim as needed if
topp limit does not capture the highest read count
    pdf(paste("MeanTagCount",Modification,if(isTRUE(orderByReads))"sorted",".pdf",
sep="_"), width = 4, height=9)
    plot(x=seq(-3000, 3000, length.out=6001),
         y=colMeans(overlap.matrix),
         ty="b", pch=18,
         ylim = c(0,7),
         cex = .3,
         col="red",
         ylab="Mean tag count",
         xlab="Distance from TSS (bp)",
         main=Modification)
    abline(h=seq(1,100,by=5), v=seq(-3000, 3000, length.out=10), lwd=0.2, col="black")
    box(col="black", lwd=2)
    dev.off()
  Heatmap(overlap.matrix, xlim=c(-3000, 3000), xlab=xlab, ylab=ylab,
title=Modification, orderByReads=orderByReads, withCustom=withCustom)
}
```

Examples of generating multiple heatmaps

```
GenHeat("hiFT-T_P15_H3K4me1_Rep1", "hiFT-T_P15_H3K4me1_Rep1", xlab="TSS", ylab="Genome
location", F, F)
GenHeat("hiFT-T_P15_H3K4me1_Rep1", "hiFT-T_P15_H3K4me1_Rep1", xlab="TSS", ylab="Genome
location", F, T)
GenHeat("hiFT-T_P15_H3K4me1_Rep1", "hiFT-T_P15_H3K4me1_Rep1", xlab="TSS", ylab="Genome
location", T, F)
GenHeat("hiFT-T_P15_H3K4me1_Rep1", "hiFT-T_P15_H3K4me1_Rep1", xlab="TSS", ylab="Genome
location", T, T)
GenHeat("5dd_DOX_plus_H3K4me2_merged", "hiFT-T_P15_H3K4me2_Rep1", xlab="TSS",
ylab="Genome location", F, F)
GenHeat("5dd_DOX_plus_H3K4me2_merged", "hiFT-T_P15_H3K4me2_merged", xlab="TSS",
ylab="Genome location", F, T)
GenHeat("5dd_DOX_plus_H3K4me1_Rep1", "hiFT-T_P15_H3K4me1_Rep1", xlab="TSS",
ylab="Genome location", T, F)
GenHeat("5dd_DOX_plus_H3K4me2_merged", "hiFT-T_P15_H3K4me2_Rep1", xlab="TSS",
ylab="Genome location", T, T)
GenHeat("10dd_DOX_plus_SSEA3_pos_H3K4me1_Rep1", "hiFT-T_P15_H3K4me1_Rep1", xlab="TSS",
ylab="Genome location", F, F)
GenHeat("10dd_DOX_plus_SSEA3_pos_H3K4me1_Rep1", "hiFT-T_P15_H3K4me1_Rep1", xlab="TSS",
ylab="Genome location", F, T)
GenHeat("10dd_DOX_plus_SSEA3_pos_H3K4me1_Rep1", "hiFT-T_P15_H3K4me1_Rep1", xlab="TSS",
ylab="Genome location", T, F)
GenHeat("10dd_DOX_plus_SSEA3_pos_H3K4me1_Rep1", "hiFT-T_P15_H3K4me1_Rep1", xlab="TSS",
ylab="Genome location", T, T)
GenHeat("24dd_TRA_pos_DOX_plus_H3K4me1_Rep1", "hiFT-T_P15_H3K4me1_Rep1", xlab="TSS",
ylab="Genome location", F, F)
GenHeat("24dd_TRA_pos_DOX_plus_H3K4me1_Rep1", "hiFT-T_P15_H3K4me1_Rep1", xlab="TSS",
ylab="Genome location", F, T)
GenHeat("24dd_TRA_pos_DOX_plus_H3K4me1_Rep1", "hiFT-T_P15_H3K4me1_Rep1", xlab="TSS",
ylab="Genome location", T, F)
GenHeat("24dd_TRA_pos_DOX_plus_H3K4me2_Rep2", "hiFT-T_P15_H3K4me2_Rep1", xlab="TSS",
ylab="Genome location", T, T)
GenHeat("24dd_TRA_pos_DOX_minus_H3K4me1_Rep1", "hiFT-T_P15_H3K4me1_Rep1", xlab="TSS",
ylab="Genome location", F, F)
GenHeat("24dd_TRA_pos_DOX_minus_H3K4me1_Rep1", "hiFT-T_P15_H3K4me1_Rep1", xlab="TSS",
ylab="Genome location", F, T)
GenHeat("24dd_TRA_pos_DOX_minus_H3K4me1_Rep1", "hiFT-T_P15_H3K4me1_Rep1", xlab="TSS",
ylab="Genome location", T, F)
GenHeat("24dd_TRA_pos_DOX_minus_H3K4me1_Rep1", "hiFT-T_P15_H3K4me1_Rep1", xlab="TSS",
ylab="Genome location", T, T)
GenHeat("hIPSC-T_P10_H3K4me1_Rep1", "hiFT-T_P15_H3K4me1_Rep1", xlab="TSS",
ylab="Genome location", F, F)
GenHeat("hIPSC-T_P10_H3K4me2_merged", "hiFT-T_P15_H3K4me1_Rep1", xlab="TSS",
ylab="Genome location", F, T)
GenHeat("hIPSC-T_P10_H3K4me1_Rep1", "hiFT-T_P15_H3K4me1_Rep1", xlab="TSS",
ylab="Genome location", T, F)
GenHeat("hIPSC-T_P10_H3K4me1_Rep1", "hiFT-T_P15_H3K4me1_Rep1", xlab="TSS",
ylab="Genome location", T, T)
CustomTXDB("Adhesome_promoters.xlsx")
CustomTXDB("yaptaz_promoters.xlsx")
CustomTXDB("Tead1_promoters1.xlsx")
CustomTXDB("Srf_promoters1.xlsx")
```

Overlap Plot Generator

```
ensdb <- EnsDb.Hsapiens.v75
tranx <- transcripts(ensdb)

##defining peak files desired for comparison
df_hiFT <- fread("_hiFT-T_P15_H3K4me3_Rep1/hiFT-T_P15_H3K4me3_Rep1_peaks.xls")
df_5dd <- fread("_5dd_DOX_plus_H3K4me3_Rep1/5dd_DOX_plus_H3K4me3_Rep1_peaks.xls")
df_10dd <-
fread("_10dd_DOX_plus_SSEA3_pos_H3K4me3_Rep1/10dd_DOX_plus_SSEA3_pos_H3K4me3_Rep1_peak
s.xls")
df_24plus <-
fread("_24dd_TRA_pos_DOX_plus_H3K4me3_Rep1/24dd_TRA_pos_DOX_plus_H3K4me3_Rep1_peaks.xl
s")
df_24minus <-
fread("_24dd_TRA_pos_DOX_minus_H3K4me3_Rep1/24dd_TRA_pos_DOX_minus_H3K4me3_Rep1_peaks.
xls")
df_hIPSC <- fread("_hIPSC-T_P10_H3K4me3_Rep1/hIPSC-T_P10_H3K4me3_Rep1_peaks.xls")

df_hiFT <- dplyr::select(df_hiFT, chr, start, end, fold_enrichment)
df_5dd <- dplyr::select(df_5dd, chr, start, end, fold_enrichment)
df_10dd <- dplyr::select(df_10dd, chr, start, end, fold_enrichment)
df_24plus <- dplyr::select(df_24plus, chr, start, end, fold_enrichment)
df_24minus <- dplyr::select(df_24minus, chr, start, end, fold_enrichment)
df_hIPSC <- dplyr::select(df_hIPSC, chr, start, end, fold_enrichment)

create_comparison <- function(df_1, df_2, foldincrease = 1.5, folddecrease = 0.2,
sensitivity = 1000, return_previous_compare = F){
combined <- genome_inner_join(df_1, df_2, by = c("chr","start","end"))

combined <- combined %>%
  mutate(foldchange = c(.$fold_enrichment.x/.$fold_enrichment.y))%>%
  dplyr::filter((foldchange > 3) | (foldchange < 0.1)) %>%
  mutate(chr = chr.x, start = round((start.x+start.y)/2), end =
round((end.x+end.y)/2))%>%
  group_by(chr) %>%
  dplyr::select(chr, start, end, foldchange)

convert_gr <- with(combined, GRanges(chr, IRanges(start, end)))

## renaming styles.
ncbi_format <- mapSeqlevels(seqlevels(convert_gr),"NCBI")
convert_gr_ncbi <- renameSeqlevels(convert_gr, ncbi_format)

combined_overlaps <- subsetByOverlaps(tranx, convert_gr_ncbi, minoverlap =
sensitivity)
my_tx_keys <- combined_overlaps$tx_id
clist <- c("ENTREZID", "SYMBOL", "TXID","TXNAME","TXSEQSTART","TXSEQEND")
matching_enst <- select(ensdb, keys=my_tx_keys, columns = clist, keytype = "TXID")

df_overlaps <- as.data.frame(combined_overlaps)
overlapsofall <- inner_join(df_overlaps, matching_enst, by = c("tx_id" = "TXID"))
##colnames(overlapsofall)[1] <- "chr"
overlapsofall$seqnames <- sub("^", "chr", overlapsofall$seqname)
df_overlapsofall <- as.data.frame(overlapsofall)
df_combined <-as.data.frame(combined)

interests <- genome_inner_join(df_combined, df_overlapsofall, by = c("chr"="seqnames",
"start","end"))
interests <- dplyr::select(interests, chr, start.x, end.x, SYMBOL, foldchange)
interests <- unique(interests)

df1_final <- genome_inner_join(df_1, interests, by = c("chr", "start" =
"start.x","end"="end.x"))
df1_final <- dplyr::select(df1_final, chr.x, start, end, fold_enrichment, SYMBOL,
foldchange)
df2_final <- genome_inner_join(df_2, interests, by = c("chr", "start" =
"start.x","end"="end.x"))
df2_final <- dplyr::select(df2_final, chr.x, start, end, fold_enrichment, SYMBOL,
foldchange)

if(return_previous_compare==T){
  return(df1_final)
} else {
return(df2_final)
}
}
one <- create_comparison(df_hiFT, df_5dd, 3, 0.1, 1000, T)
one
##Output should have the following rows in a dplyr dataframe, if this was not the
output, may need to adjust the parameters in create_comparison

    chr.x     start       end fold_enrichment        SYMBOL foldchange
 1: chr12  46123661  46125111         4.79741         ARID2   3.261836
 2: chr13  48632134  48633543         3.84848          MED4   3.331988
 3: chr17  47090795  47091752         3.81579       IGF2BP1   3.006240
 4: chr19  49713522  49714657         4.90012         TRPM4   3.173098
 5:  chr2  10219323  10221110         3.64453          CYS1   3.057133
 6:  chr2 202147035 202149427         4.09973         CASP8   3.097362
 7:  chr2 202147035 202149427         4.09973        LRG_34   3.097362
 8:  chr2 219081126 219083452         3.41797         ARPC2   3.090640
 9: chr20  36024289  36025655         5.67113           SRC   3.163722
10:  chr3   4344675   4345894         4.78956         SUMF1   3.457667
11:  chr4 113066235 113067545         3.84029       C4orf32   3.828651
12:  chr4 113626211 113628597         3.38979  RP11-148B6.2   3.067544
13:  chr6  31465672  31467127         5.55814          MICB   3.802257
14:  chr7  33391742  33394195         4.50502          BBS9   3.091708
15:  chr8  22856865  22858162         5.70247 RP11-875O11.1   3.066322
16:  chr8  22856865  22858162         5.70247       RHOBTB2   3.066322
17:  chr9  97766420  97767938        10.48509        C9orf3   6.983309
18:  chr9  97810803  97812424         9.05636        C9orf3   4.028361
19:  chr9 108209734 108211249         5.38182         FSD1L   3.033891
20:  chrX 129065131 129067083         3.88369  RP4-537K23.4   3.691054
21:  chrX 129086679 129087449         3.55686  RP4-537K23.4   3.449411

two <- create_comparison(df_5dd, df_hiFT, 3, 0.1, 1000, T)
three <- create_comparison(df_5dd, df_10dd, 3, 0.1, 1000, T)
four <- create_comparison(df_10dd, df_24plus, 3, 0.1, 1000, T)
four.minus <- create_comparison(df_10dd, df_24minus, 3, 0.1)
five <- create_comparison(df_24plus, df_24minus, 3, 0.1, 1000, T)
six <- create_comparison(df_24plus, df_hIPSC, 3, 0.1, 1000, T)
six.minus <- create_comparison(df_24minus, df_hIPSC, 3, 0.1)

three_without10 <- create_comparison(df_5dd, df_24plus, 3, 0.1, 5)
thee_without10minus <- create_comparison(df_5dd, df_24minus, 3, 0.1)

##needs to be adjusted based on what comparison the user wishes to make. Comparisons
are from right to left
list_final <- list(one, two, three, four, five, six)

gene_to_display <- 4
p = ggplot(bind_rows(list_final, .id="df"), aes(df,fold_enrichment)) +
  geom_point(alpha = 0.25, size= 2, aes(color=SYMBOL)) +
  geom_text_repel(
    data = subset(bind_rows(list_final, .id="df"), foldchange > gene_to_display),
    aes(label = SYMBOL),
    size = 3,
    box.padding = unit(0.3, "lines"),
    point.padding = unit(0.3, "lines"),
    segment.size = 0.2
  ) +
  labs(x = "Condition", y = "Peak fold enrichment", title = "H3K4me3") +
  theme(legend.position='none', axis.title.x = element_text(size = 17), axis.text.x =
element_text(angle=45, vjust= 0.4),
      axis.title.y = element_text(size = 17),
      axis.text = element_text(size = 12), plot.title = element_text(size=25,
hjust=0.5)) +
  scale_x_discrete(labels = c("hiFT", "5dd", "10dd", "24ddplus", "24ddminus", 
"hIPSC"))
  ##geom_dotplot(alpha = 0.25, binaxis = "y", stackdir = "center", dotsize = 0.5, 
aes(color=SYMBOL))
p
##ggsave("ggplot/H3K4me3.pdf", p,  width = 12, height = 12, dpi = 300
```