Published in final edited form as:

Curr Protoc Bioinformatics. 2019 June ; 66(1): e70. doi:10.1002/cpbi.70.

Discovering transcriptional regulatory elements from run-on and sequencing data using the web-based dREG gateway

Tinyi Chu¹, Zhong Wang¹, Shao-Pei Chou¹, and Charles G. Danko^{1,*}

¹Baker Institute for Animal Health, Cornell University, Hungerford Hill Rd, Ithaca, NY 14853.

Abstract

Transcription is an effective mark that can be used to identify the location of active enhancers and promoters, collectively known as transcriptional regulatory elements (TREs). We have recently introduced dREG, a tool for the identification of TREs using run-on and sequencing (RO-seq) assays, including global run-on and sequencing (GRO-seq), precision run-on and sequencing (PRO-seq), and chromatin run-on and sequencing (ChRO-seq). In this protocol, we present step-by-step instructions for running dREG on an arbitrary run-on and sequencing dataset. Users provide dREG with bigWig files representing the location of RNA polymerase in a cell or tissue sample of interest. dREG returns genomic regions that are predicted to be active TREs. Finally we demonstrate the use of dREG regions in discovering transcription factors controlling response to a stimulus and predict their target genes. Together, this protocol provides detailed instructions for running dREG on arbitrary run-on and sequencing data.

Keywords

gene regulation; enhancers; PRO-seq; GRO-seq; ChRO-seq

INTRODUCTION

DNA sequence control regions, such as promoters, enhancers, and insulators, collectively known as transcriptional regulatory elements (TREs), are critical components of the genetic regulatory programs of all organisms. TREs remain challenging to identify using existing molecular and computational tools. Histone modification ChIP-seq has a poor resolution for open chromatin regions that comprise the TRE core (Core et al. 2014; Scruggs et al. 2015; Chen et al. 2016). Nuclease accessibility, such as DNase-I-seq and ATAC-seq, mark a multiple kinds of genomic regions, such as binding sites for the insulator protein CTCF or inactive regulatory elements, without the capacity to distinguish between them (Danko et al. 2015; Xi et al. 2007). Each of these tools is also limited by a high background, which

* Contact information: Charles G. Danko, Ph.D., Baker Institute for Animal Health, Cornell University, Hungerford Hill Rd., Ithaca, NY 14853, Phone: (607) 256-5620, dankoc@gmail.com.

INTERNET RESOURCES (optional)

dREG web server: <http://dreg.dnasequence.org>

dREG source code: <https://github.com/Danko-Lab/dREG>

tfTarget source code: <https://github.com/Danko-Lab/tfTarget>

Danko lab run-on and sequencing alignment pipeline: https://github.com/Danko-Lab/proseq_2.0.

Convert BAM files to bigWigs compatible with dREG: <https://github.com/Danko-Lab/RunOnBamToBigWig>

prevents the detection of weakly TREs. Sequencing techniques targeting nascent RNA species, including run-on and sequencing (RO-seq) assays (Core, Waterfall, and Lis 2008; Kwak et al. 2013; Chu et al. 2018), provide significantly higher sensitivity in detecting short-lived enhancer RNAs (eRNAs) which are indicative of the regulatory activity of TREs. In addition, RO-seq detects various RNA species in addition to eRNAs, enabling simultaneous measurements of transcription at protein-coding/non-coding genes in addition to the discovery of active regulatory elements. However, using RO-seq signals to distinguish bona fide TREs locations from other genomic loci, such as gene bodies and polyadenylation site, requires efficient computational tools. The detection of transcriptional regulatory elements by GRO-seq, PRO-seq, and ChRO-seq data (dREG) is a method that can be used to identify the location of TREs (Danko et al. 2015; Z. Wang et al. 2018).

In this article we provide a detailed protocol for using dREG to identify TREs in any RO-seq experiment. We provide two separate ways to run dREG: First, we demonstrate the use of the dREG web server, available at (<http://dreg.dnasequence.org>). Second, we provide a detailed account of the steps that are required to download and run dREG in a user's own computer system. Finally, we provide an example of how the output of dREG can be used to map the location of transcription factors and predict which genes are targets. In summary, this protocol allows researchers to discover the location of active TREs by running dREG on RO-seq data collected in their own lab.

STRATEGIC PLANNING

The input to dREG consists of mapped reads from a GRO-seq, PRO-seq, or ChRO-seq experiment (henceforth referred to as RO-seq) (Core, Waterfall, and Lis 2008; Kwak et al. 2013; Chu et al. 2018). The quality and quantity of the experimental data are major factors in determining how sensitive dREG will be in detecting TREs. We have found that dREG has a reasonable statistical power for discovering TREs with as few as ~40M uniquely mappable reads, and saturates detection of TREs in well-studied ENCODE cell lines with >80M reads (Z. Wang et al. 2018). To increase the number of reads available for TRE discovery, we typically merge biological replicates to improve our statistical power prior to running dREG. To further improve data quality, our lab makes extensive use of unique molecular identifiers (UMIs) in RNA adapters during library prep, which allow us to identify and remove any PCR duplicates (Mahat et al. 2016; Fu et al. 2014). Typical duplication rates vary due to a variety of factors, including the quality of the input sample, the amount of starting material, and the number of cycles of PCR amplification. These experimental controls must be considered carefully while planning a RO-seq experiment.

Once investigators have experimental data in hand, the next step is to produce two bigWig files (Mills 2003), which represent the position of RNA polymerase on the positive and negative strands. The sequence alignment and processing steps to make the input bigWig files is another major factor influencing the success of dREG. Users can create bigWig files from their own alignment pipeline that are compatible with dREG. However, dREG makes several assumptions about data processing that are critical for success. Critical elements of a bioinformatics pipeline will include:

- **Include a copy of the Pol I transcription unit in the reference genome.** RO-seq data resolves the location of all four RNA polymerases found in Metazoan cells (Pol I, II, III, and Mt) (Core, Waterfall, and Lis 2008; Hah et al. 2011; Kwak et al. 2013; Blumberg et al. 2017). DNA encoding the Pol I transcription unit is highly repetitive, and is not included in most mammalian reference genomes. Nevertheless, the Pol I transcription unit is a substantial source of reads in a typical RO-seq experiment (10–30%). Many of these reads will align spuriously to retrotransposed and non-functional copies of the Pol I transcription unit, which can create mapping artifacts (Core, Waterfall, and Lis 2008). To solve this issue, we include a single copy of the repeating DNA that encodes the Pol I transcription unit in the reference genome used to map reads. We use GenBank ID# [U13369.1](#). Including a copy of this transcription unit provides an alternative place for Pol I reads to map, preventing reads from accumulating in Pol I repeats.
- **Trim 3' adapters, but leave the fragments.** Much of the signal for dREG comes from paused RNA polymerase. RNA polymerase pauses 30–60 bp downstream of the transcription start site (Kwak et al. 2013). Due to this short RNA fragment length, paused reads in most RO-seq libraries will sequence a substantial amount of adapter. This leads to poor mapping rates in full-length reads. Therefore, it is crucial to remove contaminating 3' adapters so that paused fragments will map to the reference genome properly.
- **Representing RNA polymerase location using a single base.** RO-seq measures the location of the RNA polymerase active site, in many cases at nearly single nucleotide resolution. Therefore, it is logical to represent the coordinate of RNA polymerase using the genomic position that best represents the polymerase location, rather than representing the entire read. dREG assumes that each read is represented in the bigWig file by a single base. We have noted poor performance when reads are extended. It is critical that users pass in bigWig files that represent RNA polymerase using a single nucleotide.
- **Data represents unnormalized raw counts.** dREG assumes that data represents the number of individual sequence tags that are located at each genomic position. For this reason, it is critical that input data is not normalized. The dREG server checks to ensure that input data is expressed as integers, and will return an error if this is not the case.

As an alternative to developing their own pipeline, users are also able to use our bioinformatic pipeline for aligning RO-seq data. Our pipeline produces bigWig files that are compatible with dREG, and can be found at the following URL: https://github.com/Danko-Lab/proseq_2.0. Our RO-seq pipeline takes single-end or pair-ended sequencing reads (fastq format) as input. The pipeline automates routine pre-processing and alignment steps, including pre-processing reads to remove the adapter sequences and trim based on base quality, and deduplicate the reads if UMI barcodes are used. Sequencing reads are mapped to a reference genome using BWA. Aligned BAM files are converted into bigWig format in which each read is represented by a single base.

To run our pipeline users must first download the pipeline files and install dependencies indicated in the README.md. In addition, users need to provide a path to a BWA index file and the path to the chromInfo file for the genome of choice. After running this pipeline, users should have processed data files in the specified output directory.

Finally, we also provide a tool that converts mapped reads from a BAM file into bigWig files that are compatible with bigWig. This tool is available here: <https://github.com/Danko-Lab/RunOnBamToBigWig>

We have found that visualizing aligned data in a genome browser prior (e.g., IGV or UCSC) to downstream analysis is a useful way to catch any data quality or alignment issues. Users are directed to the **Troubleshooting** section for additional information and examples.

BASIC PROTOCOL 1: Finding TREs in RO-seq data using the dREG web server.

dREG identifies active TREs based on a pre-trained Support Vector Regression (SVR) model, which can be used to do TRE discovery and peak-calling on RO-seq data. In general, running dREG by the web server executes the following steps.

- 1) Identify informative genomic positions. Loci that are low in RO-seq reads are pre-filtered and excluded from running peak calling. We select loci for analysis that meet either of the following heuristics: 1) contain more than 3 reads in a 100 bp interval on either strand, or 2) more than 1 read in 1 kbp interval on both strands. We refer to positions meeting these criteria as “informative positions”.
- 2) Predicting dREG scores. We used SVR to score 50 bp intervals along the genome, using a pre-trained SVR model. The RO-seq profile of each informative position was described using a 360-dimensional feature vector. This feature vector integrates the RO-seq counts using sliding windows at 5 different scales, and transformed using logistic normalization to better represent their shapes. Extracting the feature vector was done on CPUs, and can be distributed on multiple CPU cores. dREG runs the actual prediction on the GPU, leveraging the power of parallelized computing, and hence greatly improves the efficiency of computing.
- 3) Calling dREG peaks. We stitch regions of high dREG scores into candidate peaks, and then estimate the probability that these peaks are drawn from the negative set of sites. The final predictions for genomic regions that contain transcription start sites are corrected using the false discovery rate correction for multiple testing and reported to the user.

Necessary Resources

- 1) Javascript and Cookie-enabled browsers. Currently 3 browsers are recommended: Firefox, Google Chrome, and Safari.
- 2) Sample data:

bigWig files compatible for running dREG can be downloaded from Gene Expression Omnibus (GEO). Links to the example bigWig files are listed in Table 1. For simplicity, we rename each files by removing the GSM/GSE ID, such that GSM2265095_H1-U_plus.bw becomes H1-U_plus.bw, etc. The bigWig files can also be downloaded from <ftp://cbsuftp.tc.cornell.edu/danko/hub/protocol.files/bigWigs.raw>.

1) Map reads to the reference genome and confirm the appropriate format and data quality (Mahat et al. 2016). dREG makes several assumptions about how RNA polymerase is represented in the input bigWig files that substantially affect the results (see Strategic Planning section). In particular: (1) The location of each read must be represented by a single base that denotes as accurately as possible the location of the RNA polymerase active site, and (2) Data must be unnormalized raw read counts. Users who have not worked with RO-seq data before can use our alignment pipeline, which is compatible with GRO-seq, PRO-seq, and ChRO-seq data that is both single or paired-end (URLs are provided in the INTERNET RESOURCES section). For the purpose of demonstration, we provide an example using human primary T cells with/without PMA and ionomycin treatment (PI). The sample files are listed above and can be downloaded as dREG-ready bigWigs files from the GEO database using either ftp/http protocol.

2) To increase the sensitivity of dREG, users may merge the bigWigs of biological replicates under each experimental condition, i.e. PI treated and untreated, using the mergeBigWigs.bsh. The script is shown below. The merged bigWig files can also be downloaded from <ftp://cbsuftp.tc.cornell.edu/danko/hub/protocol.files/bigWigs.merged>.

```
$ mergeBigWigs.bsh -c chromInfo.hg19 H-U_plus.bw H1-U_plus.bw H2-U_plus.bw
H4-U_plus.bw

$ mergeBigWigs.bsh -c chromInfo.hg19 H-U_minus.bw H1-U_minus.bw H2-
U_minus.bw H4-U_minus.bw

$ mergeBigWigs.bsh -c chromInfo.hg19 H-PI_plus.bw H1-PI_plus.bw H2-
PI_plus.bw H4-PI_plus.bw

$ mergeBigWigs.bsh -c chromInfo.hg19 H-PI_minus.bw H1-PI_minus.bw H2-
PI_minus.bw H4-PI_minus.bw
```

The chromInfo.hg19 used by the mergeBigWigs.bsh is a text file that specifies the chromosome size, and can be downloaded and generated from <http://hgdownload.cse.ucsc.edu/goldenpath/hg19/database/>. As an example of how to perform dREG analysis, we will utilize one pair of merged bigWig files for the controls (H-U_plus.bw and H-U_minus.bw) from above. However, in an actual experimental set-up, we would also perform dREG analysis on the experimental condition (H-PI_plus.bw and H-PI_minus.bw) and compare the control and experimental data. The output of dREG of these two pairs of files can be downloaded from <ftp://cbsuftp.tc.cornell.edu/danko/hub/protocol.files/dREG.output>. To avoid wasting unnecessary computing resources on running the examples, users are advised to directly download the results for the examples from the above ftp link, or use their own data of interest.

3) Navigate to the dREG Science Gateway. The dREG Science Gateway can be accessed at <http://dREG.dnasequence.org/>.

4) Register for an account. The dREG gateway requires users create a new account. Users may register for an account at the homepage of dREG gateway. The dREG gateway will send an email containing the link to activate the account. Please check the spam email folder in case the registration email is blocked. Under rare circumstances, the activation email can be quarantined by institutional email accounts, which usually are not delivered to the email box, and hence, cannot be found in any email folders, including inbox or spam. If the emails from dREG gateway are found to be undeliverable, please contact your administrator or use another email account, such as Gmail, for registration propose.

Once the registration is completed, users may sign in to the account and use the following steps to run a dREG analysis.

5) Once logged in, a dashboard will show up to the user. Click the “Start dREG” icon to create a new dREG analysis.

6) The next window (Figure 2) requests information about the dREG experimental design, including the name of the experiment, the project name “Default Project”, and other metadata.

7) Upload the bigWig files representing the location of RNA polymerase on the plus and minus strand (Figure 3), in which case we use H-U_plus.bw and H-U_minus.bw. Users also need to specify the prefix to the names of the output files, which will be used to label the files that are delivered to the user in the output. Once bigWig files are uploaded onto the server, please click “Save and launch”.

Note: Upon being launched, dREG gateway will submit the computing task to computers in the XSEDE cluster using the Apache Airavata server (Pamidighantam et al. 2016). These processes include i) transferring user-submitted files to storage space of the GPU node, ii) submitting a bash script which specifies the runs of dREG to the GPU node, iii) the execution of bash script is queued, and an notification email will be sent back to Apache Airavata server once the script is executed, iv) once the dREG run is complete, another notification email will be sent to the Apache Airavata server, v) the Apache Airavata server returns the result of dREG run to the user’s web storage and notifies the user through user’s email.

8) After launching dREG, users will be directed to a summary table of the current task, shown in Figure 4, which lists the cluster address, the status of queue, the input files, and the creation time of the task. If the computing task is returned quickly, it usually means the dREG run was interrupted by errors.

There are two possibilities for these errors: the use of bigWig input files that do not meet the requirement, such as use of normalized values or the mapping of whole reads instead of only the end of the reads to the bigWig, or insufficient computing resource of the server (see Troubleshooting). Click “Open” under the “Storage Directory”, and access the log files to

identify any errors. If the project runs normally, users may close the webpage. Each run usually takes 4–12 hours, depending on the queue and the execution time on the GPU node.

9). Users will be notified by e-mail when the dREG run is complete. Once dREG is complete, users may log in the dREG gateway and click the “Browse Experiments” icon on the user dashboard to access the dREG results. Users will be directed to “Experiment Summary”, where the results of the dREG run will be available for download onto a local machine (Figure 5). Users will want to download the “Full Results” file for most applications (see **Guidelines for Understanding Results**). Results can also be visualized using the WashU Epigenome Genome browser.

10) To visualize the results in the WashU Epigenome Browser (Zhou et al. 2011), choose / type in the reference genome build in University of California Santa Cruz (UCSC) version numbering that was used to create the bigWig files (e.g., hg38 or hg19 for human, or mm10 for mouse) and then click “Switch to genome browser”. The browser will open a new tab that will lead users to the WashU epigenome browser webpage. Both the input bigWig files and results of dREG will be visualized in separate tracks, as shown in Figure 6.

11) Once dREG is complete, results will be stored on the server for a period of 30 days.

ALTERNATE PROTOCOL: Running a local copy of dREG

Many applications may require downloading and running dREG locally. Here we provide a detailed protocol for running a local copy of dREG.

Necessary Resources

Estimates of hardware resources are based on a deeply sequenced (~40–400 M mapped reads) PRO-seq for Human Genome Reference GRCh37d5

Hardware

- A Linux computer with at least 128 GB of RAM
- 8 CPU cores
- GPU with 12 GB memory (supports CUDA 6.5 or above)
- Disk storage of 1TB
- Run-time, 4–12 hrs

Software

- (1) GIT (<https://git-scm.com/download/linux>)
- (2) BEDOPS (<http://bedops.readthedocs.org/en/latest/index.html>)
- (3) Boost Library (<https://www.boost.org/users/download/>)
- (4) CUDA 6.5 or above (<https://developer.nvidia.com/cuda-toolkit>)
- (4) R software with the following package:

a) dREG and its dependencies (<https://github.com/Danko-Lab/dREG>)

bigWig ($\geq 0.2-9$), data.table, e1071, mvtnorm, parallel, rmutil, randomForest, snowfall.

See *Support Protocol* for installation instructions

b) Rgtsvm and its dependencies (<https://github.com/Danko-Lab/Rgtsvm>)

bit64, snow, SparseM, Matrix

See *Support Protocol* for installation instructions

Files

dREG SVR model used for peak calling, it can be downloaded from <ftp://cbsuftp.tc.cornell.edu/danko/hub/dreg.models>

As of this writing, the most recent model is named asvm.gdm.6.6M.20170828.rdata.

- 1) Map reads to the reference genome and confirm the appropriate format and data quality.
- 2) Run the main dREG application. The main dREG pipeline scores 50 bp intervals along the genome for similarity to a TRE, and generates a BED file with narrow peaks, peak scores, probability, and peak center positions. We provide a bash script which allows users to automatically execute all of the stages of this pipeline. The script is under the dREG directory, and can be configured and run as follows:

1. Set an environment variable for the path to the RData file containing the pre-trained SVM

```
export dREG_MODEL=/your/path/asvm.gdm.6.6M.20170828.rdata
```

2. Run dREG by executing the main bash script: run_dREG.bsh. First define and assign variables required for running the run_dREG.bsh

```
#-- PRO-seq data (plus strand).
```

```
# Read counts (unnormalized) formatted as a bigWig file.
```

```
PLUS_STRAND_BW=H-U_plus.bw
```

```
#-- PRO-seq data (minus strand).
```

```
# Read counts (unnormalized) formatted as a bigWig file.
```

```
MINUS_STRAND_BW=H-U_minus.bw
```

```
#-- The prefix of the output file.
```

```
OUT_PREFIX=H-U
```

```
# CPU cores can be used for feature extraction and peak identification.
```

```
# [optional, default=1]
```

```
CPU_CORES=16
```

```
# GPU id when multiple GPU cards are available. The first ID is 0.
```



```
# [optional, default=NA]
```

```
GPU_ID=0
```

Build the run_dREG.bsh command (this example uses parameters defined above):

```
$ bash run_dREG.bsh\
    $PLUS_STRAND_BW\
    $MINUS_STRAND_BW\
    $OUT_PREFIX\
    $dREG_MODEL\
    $CPU_CORES\
    $GPU_ID
```

Note: The actual time for running run_dREG.bsh depends on the number of informative positions, the number of broad peaks generated from these informative positions, and the speed of the computer on which dREG is run. Due to the large size of the new dREG model, large amounts of intermediate data are generated when running dREG. Users are advised to make sure that they have sufficient amount of free memory, otherwise the dREG process may be killed by the system.

3) Once dREG exits, it should add 5 main files under the current working directory. These files are described in detail under **Guidelines for Understanding Results**.

SUPPORT PROTOCOL 1: INSTALLATION OF dREG AND DEPENDENCIES

dREG has been packaged to minimize the complexity of installation. The examples below use the version available at the time of publication. Please see the repositories for up-to-date instructions.

In the following examples, please modify /your/cuda/home and /your/boost/home to appropriate locations. Also, please use the same path to dREG and use this path in all of these steps.

Necessary Resources

Linux-based system with Web access

dREG and Rgtsvm Installation

1. Install R, CUDA, and Boost libraries. Please discuss this with your local systems administrator if you are unsure how to proceed. Make sure you know the path to both the CUDA home and BOOST home directories.

2. Install Rgtsvm package for GPU

```
$ export YOUR_CUDA_HOME=/your/cuda/home
```

```
$ export YOUR_BOOST_HOME=/your/boost/home  
$ git clone https://github.com/Danko-Lab/Rgtsvm.git  
$ cd Rgtsvm  
$ make R_dependencies  
$ R CMD INSTALL --configure-args="--with-cuda-home=  
$YOUR_CUDA_HOME --with-boost-home=$YOUR_BOOST_HOME"  
Rgtsvm
```

3. Install dREG package for R

```
$ git clone https://github.com/Danko-Lab/dREG.git  
$ cd dREG  
$ make R_dependencies  
$ make dreg
```

4. Add the dREG directory to the path environment variable

```
export PATH=/your/dreg/path:$PATH
```

BASIC PROTOCOL 2: Using dREG to identify transcription factors and their downstream target genes

Transcription factors (TFs) are proteins that affect the abundance of RNA polymerase on genes by binding to specific DNA sequence elements in TREs which can be identified using dREG. RO-seq assays measure RNA polymerase at both regulatory elements and annotated genes. This information can be used to identify specific groups of TREs regulated by each TF, and predict a set of putative target genes responding to each TF. This information results in predictions for a partial regulatory network connecting TFs to the set of bound TREs, and the potential target genes associated with each binding event (TF-TRE-target gene).

An important task in many biological applications is to identify changes in TF binding between two conditions (e.g. treatment vs. control). Other applications require connecting changes in TF recruitment to the activity of downstream target genes. We have recently developed a strategy to solve both of these problems, and implemented our solution in an R package called tfTarget (Chu et al. 2017). This protocol describes how to use tfTarget to identify the TF-TRE-target gene networks that control differences between groups of samples.

Necessary Resources

Recommend requirements:

- A Linux computer with 128 GB of RAM
- CPU 16 cores
- Data storage of 2TB

Run-time, 1 hr

Minimum requirements:

A Linux computer with 16 GB of RAM

1 core

Data storage of 200 GB

Run-time, 5 hr

Input files:

- 1) dREG narrow peaks of two conditions. An example can be downloaded from <ftp://cbsuftp.tc.cornell.edu/danko/hub/protocol.files/dREG.output>.
- 2) bigWig files for RO-seq data of two conditions, with at least two replicates for each condition. An example can be downloaded from <ftp://cbsuftp.tc.cornell.edu/danko/hub/protocol.files/bigWigs.raw>.
- 3) A gene annotation file, of the same genome assembly as the bigWig files.
- 4) The tfs.rdata file containing the TF motif database (required for non-human species).
- 5) The 2bit file representing the genome of interest. See 3.4 for details.

Install the tfTarget package and dependencies

1. Install R from <https://www.r-project.org>, and dependent packages including rphast, rtfs_db, cluster, DESeq2, gplots. The rtfs_db package can be installed from https://github.com/Danko-Lab/rtfs_db. All other packages are available directly installed from CRAN.

2. Install tfTarget package for R

```
$ git clone https://github.com/Danko-Lab/tfTarget.git
```

```
$ cd tfTarget
```

```
$ R CMD INSTALL tfTarget
```

Note: tfTarget works by 1) identifying differentially transcribed genes and TREs, 2) scanning the differentially transcribed TREs and assigning TF motifs to each of them, and 3) tabulating the TF, TRE and genes nearby with the information about differential transcription. This package requires genomic intervals specifying the regions of genes, i.e. the gene annotation file, and TREs, i.e. the dREG regions, and the additional information about the TF motif database, stored in a rdata file.

Prepare input files

3. Prepare a BED file specifying genomic intervals of TREs (using dREG). The genomic intervals of TREs are in bed3 format. Only the first three columns will be used. Use “cat” instead of “zcat” if the input dREG files are unzipped.

```
$ zcat H-PI.dREG.peak.score.bed.gz H-U.dREG.peak.score.bed.gz \
| LC_COLLATE=C sort -k1,1 -k2,2n \
| bedtools merge -i stdin > merged.dREG.bed
```

Note: Some thought must be put into how to handle separate dREG intervals from multiple separate conditions. We will typically merge dREG regions across different biological conditions and use these BED regions for downstream analysis (Danko et al. 2018; Chu et al. 2017).

4. Prepare the gene annotation files. The gene annotation file should be in bed6 format, i.e. strand specific. This can be prepared from GENCODE or Refseq gtf files. We recommend specifying gene ID and gene name as 4th and 5th columns of the annotation file, which will show up in the output for identification. GENCODE files can be downloaded from <https://www.genecodegenes.org/releases/current.html>. The script below gives an example of downloading the gene annotation gtf file and then converting it to bed6 format. The output file is also available to download at <ftp://cbsuftp.tc.cornell.edu/danko/hub/protocol.files/tfTarget.files/encode.v19.annotation.bed>.

```
$ wget
    ftp://ftp.ebi.ac.uk/pub/databases/genocode/Gencode_human/release_19/
    gencode.v19.annotation.gtf.gz

$ zcat gencode.v19.annotation.gtf.gz \
| awk 'OFS="\t" {if ($3=="gene") {print $1,$4-1,$5,$10,$18,$7}}' \
| tr -d '";' > gencode.v19.annotation.bed
```

5. Generate the database of motifs (required only for non-homo sapiens species). The tfTarget package uses motifs predicted in the Cis-BP database (Weirauch et al. 2014), and computes locations using RTFBSDB (Zhong Wang, Martins, and Danko 2016). For Homo sapiens, the database of motifs is self-contained in tfTarget package, and will be used by default. For others species, users may use the following command to generate the species.tfs.rdata, which contains the curated transcription factor motifs database for the species of interests. The look-up table for species name can be found from the “species” column (the 1st column) of <http://cisbp.cabr.utoronto.ca/summary.php?by=1&orderby=Species>

```
$ R --vanilla --slave --args Mus_musculus < get.tfs.R
```

6. Download the reference genome in 2bit format. Reference genome can be found at <http://hgdownload.cse.ucsc.edu/downloads.html>. For the example, we download the reference genome for hg19.

```
$ wget http://hgdownload.cse.ucsc.edu/goldenPath/hg19/bigZips/hg19.2bit
```

Running tfTarget package

7. tfTarget can be run using the following command “bash run_tfTarget.bsh [...]”, with [...] specifying the parameters used to run tfTarget. In our case, we specify the following parameters. Users may need to change to their own directory correspondingly.

```
$ dreg_path=merged.dREG.bed
$ gene_path=gencode.v19.annotation.bed
$ bigWig_path=/your/path/
$ twoBit_path=/your/path/hg19.2bit
$ ncores=30
$ prefix=tcell
$ query_files="H1-PI_plus.bw H1-PI_minus.bw H2-PI_plus.bw H2-PI_minus.bw H4-PI_plus.bw H4-PI_minus.bw"
$ control_files="H1-U_plus.bw H1-U_minus.bw H2-U_plus.bw H2-U_minus.bw H4-U_plus.bw H4-U_minus.bw"
$ bash run_tfTarget.bsh \
-TRE.path $dreg_path \
-gene.path $gene_path \
-bigWig.path $bigWig_path \
-2bit.path $twoBit_path \
-query $query_files \
-control $control_files \
-prefix $prefix \
-ncores $ncores
```

Notes: -TRE.path and -gene.path options specify the paths to the bed files of dREG and gene annotations, respectively. -2bit.path specifies the path to the 2bit files. The genomic assembly should be consistent for these three files. -bigWig.path together with -control and -query specify the bigWig files, ordered by plus then minus strand. If -bigWig.path is not present, the current directory will be used as default. -prefix specifies the prefix for all output files. Other parameters are optional, and the details are listed on <https://github.com/Danko-Lab/tfTarget>.

tfTarget will by default run the complete workflow. tfTarget will identify differentially regulated TF-TRE-gene combinations between the two conditions. Alternatively, users may run only subsets of modules. Use the tag “-deseq” (without argument) to only run DEseq2 on TREs and genes. Use the tag “-rtfbsdb” to only run

DESeq2 and then rtfsdb to identify TF motifs enriched in differentially regulated TREs.

Interpreting the results from tfTarget

A complete tfTarget run will output several .pdf files and .txt files. The TFs enriched in differentially regulated TREs are shown in 2D dot plots grouped in two pdfs (Figure 7), up.motif.pdf and down.motif.pdf. The *p* values of enrichment/depletion of motifs, calculated by two-sided Fisher's exact test, are represented by the radius of the circle, and enrichment (red) or depletion (blue) are represented by the rainbow color scale.

Methods relying on the use of each TFs' position weighted matrix are limited in the ability to distinguish between paralogous TFs that share similar DNA binding specificities. To account for that when interpreting the enrichment results, tfTarget generates two additional heatmaps in pdf format that show the relation among motifs by clustering them into distinct groups based on their position in differentially regulated TREs. Note that the ordering of the motifs are consistent between 2D plots and heatmaps. The example of heatmap is shown in Figure 8.

The detailed statistics of tfTarget output are provided in three txt files. The ".TRE.deseq.txt" and the ".gene.deseq.txt" file lists DESeq2 statistics for each TRE and gene. Rows with all NA value are genes excluded from DESeq2 runs due to short gene length (≤ 1 Kb).

The ".TF.TRE.gene.txt" file tabulates the relation between TFs, TREs and target genes. The results are subjected to the restriction by distance between TREs and the transcriptional start site of target gene (specified by "-dist" tag, default=50kb), the nth closest gene to the TRE (specified by "-closest.N" tag, default=2), and the *p* values for genes that showed same direction of log2foldchange as its regulator TRE (specified by "-pval.gene" tag, default=0.05). If needed, the latter two tags can be switched off by specifying "-closest.N off" or "-pval.gene off" to output a more inclusive list of potential target genes.

GUIDELINES FOR UNDERSTANDING RESULTS

The results obtained from dREG contain the following files compressed using zip format, dREG scores at each informative position (BED format), significant dREG peaks with full information, significant dREG peaks with score only, significant dREG peaks with probability only and raw peaks. Users may either download all files as a whole or individual files separately. Raw data and results will be stored in the web storage space for up to 1 month, and outdated data will be cleaned periodically. Users are advised to download their results in time.

Running dREG will generate 5 main files under the current directory, as follows:

File name	Information
\$OUT_PREFIX.dREG.infp.bed.gz	BEDGRAPH file, includes all informative sites and dREG scores.

File name	Information
\$OUT_PREFIX.dREG.peak.full.bed.gz	BED file, reports all statistically significant peaks under the FDR correction (p-value < 0.05) with information about the peak position, max score, p-value (corrected using the Benjamini and Hochberg (Benjamini and Hochberg 1995) false discovery rate), and peak center.
\$OUT_PREFIX.dREG.peak.score.bed.gz	BED file, Significant peaks with dREG score using FDR correction (p-value < 0.05), it is partial of full information.
\$OUT_PREFIX.dREG.peak.prob.bed.gz	BED file, Significant peaks with probability using FDR correction (p-value < 0.05), it is partial of full information.
5. \$OUT_PREFIX.raw.peak.bed.gz	BED file, All raw peaks without p-value correction and any filters. This file is only available in the storage directory.

The peak calling file obtained from the web server outputs additional information for each dREG peak in the file .dREG.peak.full.bed.gz. This information includes the maximum dREG score, the probability of containing the transcription start site (TSS), the position of the peak center. An example file shows as follows:

\$ zcat H-U.dREG.peak.full.bed.gz | head -

chromosome	start	end	maximum dREG score	the probability of containing the TSS	peak center postion
chr1	565610	565820	0.48195	0	565730
chr1	567400	567760	0.89918	0.00000	567590
chr1	569770	570140	0.59807	0.00042	569960
chr1	713850	714390	1.03942	0	714210
chr1	714410	714780	0.42673	0.00319	714580
chr1	718370	718720	0.30784	0.02331	718600
chr1	723510	723830	0.33311	0	723690
chr1	762570	762800	0.52137	0.01045	762740
chr1	762820	763230	0.65556	0.00034	762970
chr1	776390	776730	0.28367	0.03507	776590

COMMENTARY

Background Information

DNA sequence control regions, such as promoters, enhancers, and insulators, collectively known as transcriptional regulatory elements (TREs), are critical components of the genetic regulatory programs of all organisms. TREs regulate gene expression by facilitating (or inhibiting) chromatin decompaction, transcription initiation, and the release of RNA polymerase II (Pol II) into productive elongation (Fuda, Ardehali, and Lis 2009). In addition to well-characterized roles in cellular development, dysfunction in TREs also plays pivotal roles in a myriad of different disease states (Shlyueva, Stampfel, and Stark 2014; Long, Prescott, and Wysocka 2016). The comprehensive identification of TREs has therefore emerged as a primary challenge in genomic research.

Active TREs recruit RNA polymerase and initiate a local and highly characteristic pattern of transcription initiation (Kim et al. 2010; De Santa et al. 2010; Core et al. 2014; Scruggs et al. 2015). Transcription initiation is a highly specific signal that can be useful for identifying active TREs in a cell type-specific manner (Melgar, Collins, and Sethupathy 2011; Core et al. 2014; Danko et al. 2015; Andersson, Gebhard, et al. 2014; Azofeifa and Dowell 2016). Although first characterized in mammals, initiation appears to mark enhancers in other Metazoan organisms (Henriques et al. 2018; Mikhaylichenko et al. 2018; Rennie et al. 2018). However, the majority of initiation events give rise to highly unstable RNA species that are rapidly degraded by the nuclear exosome complex (Preker et al. 2008; Andersson, Refsing Andersen, et al. 2014). For this reason, methods that measure the production of nascent RNAs on chromatin, such as precision run-on and sequencing (PRO-seq) and related run-on assays, are particularly sensitive experimental methods to detect these transient enhancer-associated RNAs because they measure primary transcription before unstable RNAs are degraded by the exosome (Core et al. 2014).

We have recently introduced a novel computational method called the detection of regulatory elements using GRO-seq, PRO-seq, or ChROseq (dREG) to identify TREs de novo using PRO-seq, GRO-seq, or ChRO-seq data (Danko et al. 2015; Z. Wang et al. 2018). Most recently, we have developed a web-based portal using XSEDE servers to run dREG (Z. Wang et al. 2018; Zhong Wang et al. 2018). Here we provide a detailed step-by-step tutorial into how to use both the dREG web server and the downloaded dREG software. Finally, we close by providing insights into the downstream applications of these methods for discovering transcription factors responsible for a variety of biological processes.

Critical Parameters

The quality and quantity of the experimental data are major factors in determining how sensitive dREG will be in detecting TREs. We have found that dREG has a reasonable statistical power for discovering TREs with as few as ~40M uniquely mappable reads, and saturates detection of TREs in well-studied ENCODE cell lines with >75M reads (Z. Wang et al. 2018). To increase the number of reads available for TRE discovery, we typically merge biological replicates to improve our statistical power prior to running dREG.

To further improve data quality, our lab makes extensive use of unique molecular identifiers (UMIs) in RNA adapters during library prep, which allow us to identify and remove any PCR duplicates (Mahat et al. 2016; Fu et al. 2014). Typical duplication rates vary due to a variety of factors, including the quality of the input sample, the amount of starting material, and the number of cycles of PCR amplification. These experimental parameters must be considered carefully while planning a RO-seq experiment.

Troubleshooting

The most common problems associated with running dREG can be identified by a careful examination of the input bigWig files using a genome browser (e.g., IGV, WashU, or UCSC). A genome-browser view that shows high-quality PRO-seq data is depicted in Figure 9. Note that the direction of transcription resolved by PRO-seq is largely consistent with gene annotations, and gene bodies tend to have a uniform coverage of reads without

excessively large gaps. Notes on identifying several common problems that are likely to be faced by users are listed below:

- **Poor quality RO-seq data.** Poor quality RO-seq data is characterized by high numbers of reads at only a handful of genomic locations (Figure 9). Unfortunately, this problem requires re-making new data. Troubleshooting tips for the experimental data are covered elsewhere (Mahat et al. 2016). Users are also encouraged to start with more input material and make use of UMIs in their sequencing adapters, which can help to clean up data that has been amplified for too many cycles (at the expense of sequencing depth).
- **Extending reads.** The location of RNA polymerase in RO-seq data is naturally represented by a single nucleotide position. dREG assumes that bigWig files will represent RNA polymerase in this manner. The solution to this problem is to remake bigWig files while representing the data using only a single position.
- **Using normalized counts in bigWig files.** dREG assumes that input data will consist of integers (i.e., 0, 1, 2, ...), and will return an error if it finds this is not the case. The solution to this problem is to remake bigWig files with raw counts.
- **Failure to reverse the strand.** Many (but not all) RO-seq protocols sequence from the reverse complement of the tagged RNA, and as a result reads must be reversed prior to downstream analysis. Reversed data is shown in Figure 9. Note that most of the reads aligning within annotated genes is reversed relative to the annotation, and the divergent transcription and pause peak appear on the end (rather than the beginning) of each transcription unit. At the time of this writing, dREG does not detect this issue automatically. The solution to this problem is to remake bigWig files reversing the strand.

ACKNOWLEDGEMENT

We thank XSEDE allocation numbers TG-BIO160048, TG-MCB180027 and TG-MCB160061 for providing computational resources required in the dREG gateway, including Web server, Apache service, Airavata service, Web storage and GPU nodes. Work in this publication was supported by an NHGRI (National Human Genome Research Institute) grant R01-HG009309 to CGD. The content is solely the responsibility of the authors and does not necessarily represent the official views of the US National Institutes of Health.

LITERATURE CITED

- Robin Andersson, Gebhard Claudia, Miguel-Escalada Irene, Hoof Ilka, Bornholdt Jette, Boyd Mette, Chen Yun, et al. 2014 “An Atlas of Active Enhancers across Human Cell Types and Tissues.” *Nature* 507 (7493): 455–61. [PubMed: 24670763]
- Robin Andersson, Andersen Peter Refsing, Valen Eivind, Core Leighton J., Bornholdt Jette, Boyd Mette, Jensen Torben Heick, and Sandelin Albin. 2014 “Nuclear Stability and Transcriptional Directionality Separate Functionally Distinct RNA Species.” *Nature Communications* 5 (11): 5336.
- Azofeifa Joseph G., and Dowell Robin D.. 2016 “A Generative Model for the Behavior of RNA Polymerase.” *Bioinformatics*, 9 10.1093/bioinformatics/btw599.
- Benjamini Yoav, and Hochberg Yosef. 1995 “Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing.” *Journal of the Royal Statistical Society. Series B, Statistical Methodology* 57 (1): 289–300.

- Blumberg Amit, Rice Edward J., Kundaje Anshul, Danko Charles G., and Mishmar Dan. 2017 “Initiation of mtDNA Transcription Is Followed by Pausing, and Diverge across Human Cell Types and during Evolution.” *Genome Research*, 1 10.1101/gr.209924.116.
- Chen Yun, Pai Athma A., Herudek Jan, Lubas Michal, Meola Nicola, Järvelin Aino I., Andersson Robin, et al. 2016 “Principles for RNA Metabolism and Alternative Transcription Initiation within Closely Spaced Promoters.” *Nature Genetics* 48 (9): 984–94. [PubMed: 27455346]
- Chu Tinyi, Rice Edward J., Booth Gregory T., Salamanca Hans H., Wang Zhong, Core Leighton J., Longo Sharon L., et al. 2017 “Chromatin Run-on Reveals Nascent RNAs That Differentiate Normal and Malignant Brain Tissue.” *bioRxiv*. 10.1101/185991.
- Chu Tinyi, Rice Edward J., Booth Gregory T., Salamanca H. Hans, Wang Zhong, Core Leighton J., Longo Sharon L., et al. 2018 “Chromatin Run-on and Sequencing Maps the Transcriptional Regulatory Landscape of Glioblastoma Multiforme.” *Nature Genetics*, 10 10.1038/s41588-018-0244-3.
- Core Leighton J., Martins André L., Danko Charles G., Waters Colin T., Siepel Adam, and Lis John T.. 2014 “Analysis of Nascent RNA Identifies a Unified Architecture of Initiation Regions at Mammalian Promoters and Enhancers.” *Nature Genetics* 46 (12): 1311–20. [PubMed: 25383968]
- Core Leighton J., Waterfall Joshua J., and Lis John T.. 2008 “Nascent RNA Sequencing Reveals Widespread Pausing and Divergent Initiation at Human Promoters.” *Science* 322 (5909): 1845–48. [PubMed: 19056941]
- Danko Charles G., Choate Lauren A., Marks Brooke A., Rice Edward J., Wang Zhong, Chu Tinyi, Martins Andre L., et al. 2018 “Dynamic Evolution of Regulatory Element Ensembles in Primate CD4+ T Cells.” *Nature Ecology & Evolution*, 1 10.1038/s41559-017-0447-5.
- Danko Charles G., Hyland Stephanie L., Core Leighton J., Martins Andre L., Waters Colin T., Won Lee Hyung, Cheung Vivian G., Kraus W. Lee, Lis John T., and Siepel Adam. 2015 “Identification of Active Transcriptional Regulatory Elements from GRO-Seq Data.” *Nature Methods* 12 (5): 433–38. [PubMed: 25799441]
- Santa De, Francesca Iros Barozzi, Mietton Flore, Ghisletti Serena, Polletti Sara, Betsabeh Khoramian Tusi Heiko Muller, Ragoussis Jiannis, Wei Chia-Lin, and Natoli Gioacchino. 2010 “A Large Fraction of Extragenic RNA Pol II Transcription Sites Overlap Enhancers.” *PLoS Biology* 8 (5): e1000384. [PubMed: 20485488]
- Fuda Nicholas J., Ardehali M. Behfar, and Lis John T.. 2009 “Defining Mechanisms That Regulate RNA Polymerase II Transcription in Vivo.” *Nature* 461 (7261): 186–92. [PubMed: 19741698]
- Fu Glenn K., Xu Weihong, Wilhelmy Julie, Mindrinos Michael N., Davis Ronald W., Xiao Wenzhong, and Fodor Stephen P. A.. 2014 “Molecular Indexing Enables Quantitative Targeted RNA Sequencing and Reveals Poor Efficiencies in Standard Library Preparations.” *Proceedings of the National Academy of Sciences of the United States of America* 111 (5): 1891–96. [PubMed: 24449890]
- Hah Nasun, Danko Charles G., Core Leighton, Waterfall Joshua J., Siepel Adam, Lis John T., and Kraus W. Lee. 2011 “A Rapid, Extensive, and Transient Transcriptional Response to Estrogen Signaling in Breast Cancer Cells.” *Cell* 145 (4): 622–34. [PubMed: 21549415]
- Henriques Telmo, Scruggs Benjamin S., Inouye Michiko O., Muse Ginger W., Williams Lucy H., Burkholder Adam B., Lavender Christopher A., Fargo David C., and Adelman Karen. 2018 “Widespread Transcriptional Pausing and Elongation Control at Enhancers.” *Genes & Development*, 1 10.1101/gad.309351.117.
- Kim Tae-Kyung, Hemberg Martin, Gray Jesse M., Costa Allen M., Bear Daniel M., Wu Jing, Harmin David A., et al. 2010 “Widespread Transcription at Neuronal Activity-Regulated Enhancers.” *Nature* 465 (7295): 182–87. [PubMed: 20393465]
- Kwak Hojoong, Fuda Nicholas J., Core Leighton J., and Lis John T.. 2013 “Precise Maps of RNA Polymerase Reveal How Promoters Direct Initiation and Pausing.” *Science* 339 (6122): 950–53. [PubMed: 23430654]
- Long Hannah K., Prescott Sara L., and Wysocka Joanna. 2016 “Ever-Changing Landscapes: Transcriptional Enhancers in Development and Evolution.” *Cell* 167 (5): 1170–87. [PubMed: 27863239]

- Mahat Dig Bijay, Kwak Hojoong, Booth Gregory T., Jonkers Iris H., Danko Charles G., Patel Ravi K., Waters Colin T., Munson Katie, Core Leighton J., and Lis John T.. 2016 “Base-Pair-Resolution Genome-Wide Mapping of Active RNA Polymerases Using Precision Nuclear Run-on (PRO-Seq).” *Nature Protocols* 11 (8): 1455–76. [PubMed: 27442863]
- Melgar Michael F., Collins Francis S., and Sethupathy Praveen. 2011 “Discovery of Active Enhancers through Bidirectional Expression of Short Transcripts.” *Genome Biology* 12 (11): R113. [PubMed: 22082242]
- Mikhaylichenko Olga, Bondarenko Vladyslav, Harnett Dermot, Schor Ignacio E., Males Matilda, Viales Rebecca R., and Furlong Eileen E. M.. 2018 “The Degree of Enhancer or Promoter Activity Is Reflected by the Levels and Directionality of eRNA Transcription.” *Genes & Development*, 1 10.1101/gad.308619.117.
- Mills Lauren. 2003 “Common File Formats.” *Current Protocols in Bioinformatics / Editorial Board*, Andreas D. Baxeavanis ... [et Al.] 00 (1): A.1B.1–A.1B.18.
- Pamidighantam Sudhakar, Nakandala Supun, Abeysinghe Eroma, Wimalasena Chathuri, Shameera Rathnayaka Yodage Suresh Marru, and Pierce Marlon. 2016 “Community Science Exemplars in Seagrid Science Gateway: Apache Airavata Based Implementation of Advanced Infrastructure.” *Procedia Computer Science* 80: 1927–39.
- Preker Pascal, Nielsen Jesper, Kammler Susanne, Søren Lykke-Andersen Marianne S. Christensen, Mapendano Christophe K., Schierup Mikkel H., and Jensen Torben Heick. 2008 “RNA Exosome Depletion Reveals Transcription Upstream of Active Human Promoters.” *Science* 322 (5909): 1851–54. [PubMed: 19056938]
- Rennie Sarah, Dalby Maria, Marta Lloret-Llinares Stylianos Bakoulis, Vaagensø Christian Dalager, Jensen Torben Heick, and Andersson Robin. 2018 “Transcription Start Site Analysis Reveals Widespread Divergent Transcription in *D. Melanogaster* and Core Promoter-Encoded Enhancer Activities.” *Nucleic Acids Research* 46 (11): 5455–69. [PubMed: 29659982]
- Scruggs Benjamin S., Gilchrist Daniel A., Nechaev Sergei, Muse Ginger W., Burkholder Adam, Fargo David C., and Adelman Karen. 2015 “Bidirectional Transcription Arises from Two Distinct Hubs of Transcription Factor Binding and Active Chromatin.” *Molecular Cell* 58 (6): 1101–12. [PubMed: 26028540]
- Shlyueva Daria, Stampfel Gerald, and Stark Alexander. 2014 “Transcriptional Enhancers: From Properties to Genome-Wide Predictions.” *Nature Reviews. Genetics* 15 (4): 272–86.
- Wang Z, Chu T, Choate LA, and Danko CG. 2018 “Identification of Regulatory Elements from Nascent Transcription Using dREG.” *bioRxiv*. <https://www.biorxiv.org/content/early/2018/05/14/321539.abstract>.
- Wang Zhong, Christie Marcus A., Abeysinghe Eroma, Chu Tinyi, Marru Suresh, Pierce Marlon, and Danko Charles G.. 2018 “Building a Science Gateway For Processing and Modeling Sequencing Data Via Apache Airavata” In *Proceedings of the Practice and Experience on Advanced Research Computing*, 39:1–39:7. PEARC ‘18. New York, NY, USA: ACM.
- Wang Zhong, Martins André L., and Danko Charles G.. 2016 “RTFBSDB: An Integrated Framework for Transcription Factor Binding Site Analysis.” *Bioinformatics*, 6 10.1093/bioinformatics/btw338.
- Weirauch Matthew T., Yang Ally, Albu Mihai, Cote Atina G., Alejandro Montenegro-Montero Philipp Drewe, Najafabadi Hamed S., et al. 2014 “Determination and Inference of Eukaryotic Transcription Factor Sequence Specificity.” *Cell* 158 (6): 1431–43. [PubMed: 25215497]
- Xi Hualin, Shulha Hennady P., Lin Jane M., Vales Teresa R., Fu Yutao, Bodine David M., McKay Ronald D. G., et al. 2007 “Identification and Characterization of Cell Type-Specific and Ubiquitous Chromatin Regulatory Structures in the Human Genome.” *PLoS Genetics* 3 (8): e136. [PubMed: 17708682]
- Zhou Xin, Maricque Brett, Xie Mingchao, Li Daofeng, Sundaram Vasavi, Martin Eric A., Koebe Brian C., et al. 2011 “The Human Epigenome Browser at Washington University.” *Nature Methods* 8 (12): 989–90. [PubMed: 22127213]

Significance Statement

DNA sequences in promoter and enhancer regions control complex gene expression programs. Despite the availability of complete reference genomes in many organisms, regulatory DNA sequences remain challenging to identify. Here we demonstrate the use of a sensitive machine learning tool, dREG, that detects the location of enhancers and promoters using maps of nascent transcription derived from an experimental run-on and sequencing assay. We demonstrate how sites discovered using dREG can identify which genes may be regulated by specific transcription factors.

Figure 1. The Dashboard page.

This page is the entry point for dREG peak calling. Select “Start dREG” to launch a new computation experiment for a new PRO-seq data set.

The screenshot shows the dREG website interface. At the top, there is a navigation bar with the dREG logo (Discriminative Regulatory Element Detection) and links for Home, About, Documents, and FAQ. Below this is a dark blue header with a 'Project' dropdown, 'Start dREG' button, 'Saved dREG runs' link, 'Storage' link, a notification bell, 'Dashboard' link, and 'dregtest' dropdown. The main content area is titled 'Create a new dREG experiment'. It contains four input fields: 'Experiment Name' (required), 'Experiment Description' (optional), 'Project' (required), and 'Application'. The 'Project' dropdown is set to 'Default Project' and the 'Application' dropdown is set to 'dREG peak calling'. At the bottom of the form are 'Continue' and 'Reset values' buttons.

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

Figure 2. The webpage to launch a new dREG experiment.

Users need to select the dREG peak calling application and input metadata about the experiment. The experiment name can be used to identify the experiment in shared projects.

ProjectStart dREGSaved dREG runsStorageDashboarddregtest

Application configuration

Application Inputs

BigWig(+strand)

view file

Browse...H-U_plus.bw

GRO-seq, PRO-seq, or ChRO-seq sequencing data file in bigWig

BigWig(-strand)

view file

Browse...No file selected.

GRO-seq, PRO-seq, or ChRO-seq sequencing data file in bigWig

Prefix for output file

H-U

Compute Resource *

comet.sdsc.xsede.org

Notifications

☐ Do you want to receive email notifications for status changes in the experiment?

SaveSave and launchStart over

Figure 3. The Data Upload page.
This page is the second step of starting new dREG experiment.

Figure 4. The Experiment Summary page.
This table page lists the basic experimental information, jobs status, and allows the user to download or visualize data once complete.

Figure 5. The Output File list.
The drop-down list shows main 4 results can be downloaded using the download link. In the web storage page, additional files are available for download. To run downstream analysis of differential regulation, download the results to local directory.

Figure 6. The Genome Browser page.
The four genome browser tracks show mapped PRO-seq reads in plus strand, mapped reads in minus strand, dREG scores for each informative position, and the location of significant peak region (FDR<0.05). Use the zoom buttons at the top line to view dREG peaks near your locus of interest.

Figure 7. Motifs enriched in TREs up-regulated (left) and down-regulated (right) in PMA and ionomycin treatment, ordered by motif clusters.

Figure 8. Heatmap shows clusters of TF motifs enriched in TREs up-regulated (upper) and down-regulated (lower) in PMA and ionomycin treatment.

Figure 9. Genome browser shows high quality PRO-seq data (top), poor quality data (center), and data that was mapped to the reverse strand (bottom).

Table 1.

The GEO links to the example files used in the protocol.

Gene Expression Omnibus ID	Sample name	Link	File names
GSM2265095	Human 1 - CD4+ T-cells Untreated	https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSM2265095	GSM2265095_H1-U_plus.bw GSM2265095_H1-U_minus.bw
GSM2265096	Human 1 - CD4+ T-cells PMA +Ionomycin	https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSM2265096	GSM2265096_H1-PI_plus.bw GSM2265096_H1-PI_minus.bw
GSM2265098	Human 2, draw 2 - CD4+ T-cells Untreated	https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSM2265098	GSM2265098_H2-U_plus.bw GSM2265098_H2-U_minus.bw
GSM2265097 and GSM2265099	Human 2, (merged from draw 1 and 2) - CD4+ T- cells PMA +Ionomycin	https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE85337	GSE85337_H2-PI_plus.bw GSE85337_H2-PI_minus.bw
GSM3021718	Human 4 - CD4+ T-cells Untreated	https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSM3021718	GSM3021718_H4-U_plus.bw GSM3021718_H4-U_minus.bw
GSM3021719	Human 4 - CD4+ T-cells PMA +Ionomycin	https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSM3021719	GSM3021719_H4-PI_plus.bw GSM3021719_H4-PI_minus.bw