Chapter 13

In Silico Promoter Recognition from deepCAGE Data

Xinyi Yang and Annalisa Marsico

Abstract

The accurate identification of transcription start regions corresponding to the promoters of known genes, novel coding, and noncoding transcripts, as well as enhancer elements, is a crucial step towards a complete understanding of state-specific gene regulatory networks. Recent high-throughput techniques, such as deepCAGE or single-molecule CAGE, have made it possible to identify the genome-wide location, relative expression, and differential usage of transcription start regions across hundreds of different tissues and cell lines. Here, we describe in detail the necessary computational analysis of CAGE data, with focus on two recent in silico methodologies for CAGE peak/profile definition and promoter recognition, namely the Decomposition-based Peak Identification (DPI) and the PROmiRNA software. We apply both methodologies to the challenging task of identifying primary microRNAs transcript (pri-miRNA) start sites and compare the results.

Key words TSS, Promoter, microRNAs, DPI, PROmiRNA

1 Introduction

Gene expression is regulated at many levels, including chromatin packing, transcription initiation, polyadenylation, splicing, mRNA stability, and others. One of the most important regulatory steps is transcription initiation, which is coordinated by the binding of many proteins to gene promoters and enhancers. Combinations of binding sites determine the expression context of a certain gene and its activity in a certain tissue or condition [1, 2].

The annotation of gene promoters, as well as other transcriptionally-active regulatory sequences is essential to understand biological mechanisms underlying context-specific gene regulatory networks. But what is a promoter exactly and how can it be precisely defined? A promoter is not a clearly defined unit and to this question there is no unique answer, although scientists studying gene regulation largely agree nowadays on the fact that a promoter can be defined as the region surrounding the Transcriptional Start Site (TSS) of a gene which contains regulatory elements and Transcription Factor Binding Sites (TFBBs) necessary to initiate gene transcription [1–3].

Ulf Andersson Ørom (ed.), *Enhancer RNAs: Methods and Protocols*, Methods in Molecular Biology, vol. 1468,
DOI 10.1007/978-1-4939-4035-6_13,

The promoter structure of a eukaryotic organism is more complex than a prokaryotic one, with the complexity increasing from single-celled yeast to mammals, and regulatory elements spread over large genomic space [1]. Although eukaryotes have different types of RNA polymerases, RNA polymerase II is responsible for transcription of mRNAs, as well other classes of noncoding RNAs, including some microRNAs and long noncoding RNAs (lncRNAs).

The region of 30–100 nucleotides surrounding the TSS is generally referred to as *core promoter* (Fig. 1) and contains interchangeable sequence elements and general transcription factor binding sites recognized by the preinitiation complex (PIC) which initiates transcription from a loosely conserved Initiator site (Inr). The PIC includes, besides Polymerase II, general initiation factors such as TFIIA, TFIIB, a TATA box binding protein (TBP) which binds specific DNA elements about 25 base pair (bp) upstream of the TSS, and several TBP-associated factors (TAFs). This core promoter may also contain downstream elements like DPE and MTE (in fly), BRE upstream or downstream elements or DCE, downstream core element (in vertebrates) [2, 4].

The region further away (up to 500 bp upstream of the gene TSS) is usually referred to as *proximal promoter* and contains other promoter elements, such as the GC box and/or the CAAT box, as well as more specific TFBS necessary to coordinate transcription in a tissue- and developmental stage-specific manner. TFBSs can also occur in clusters, forming cis-regulatory modules (CRMs) [4, 5].

Distal regulatory elements also influence transcription, including enhancers, active regions which enhance gene transcription, insulators, which mark boundaries between DNA active regions, and silencers, regions which repress gene transcription. These elements are part of the so-called *distal promoter*, which can extend up to several kb from the TSS (upstream and/or downstream) [2, 4]. Finally, in most eukaryotic genomes, chromatin is made of basic units called nucleosomes. A nucleosome is composed of a segment of DNA wrapped around a histone core. Chromatin structure can be tightly wrapped or accessible to proteins: active promoters are usually found in accessible chromatin regions (or nucleosome-free regions) [2].

Earlier experimental methods for promoter identification, such as nuclease protection and primer extension can identify promoters on a gene-by-gene basis and cannot be extended genome wide. Later advances in promoter identification are sequencing methods, such as RACE 5′-tag sequencing of cDNA or mRNA sequences, which rely on reverse transcription, fragmentation, and amplification of cDNAs and alignment to the genome to get information about TSS location [6]. Other high-throughput experimental procedures include hybridization methods, such as oligonucleotide tiling arrays [4].

Back in early 1990s, experimental techniques for promoter identification were costly, labor intensive, time consuming, and not really applicable genome wide. Hence, several in silico methods for

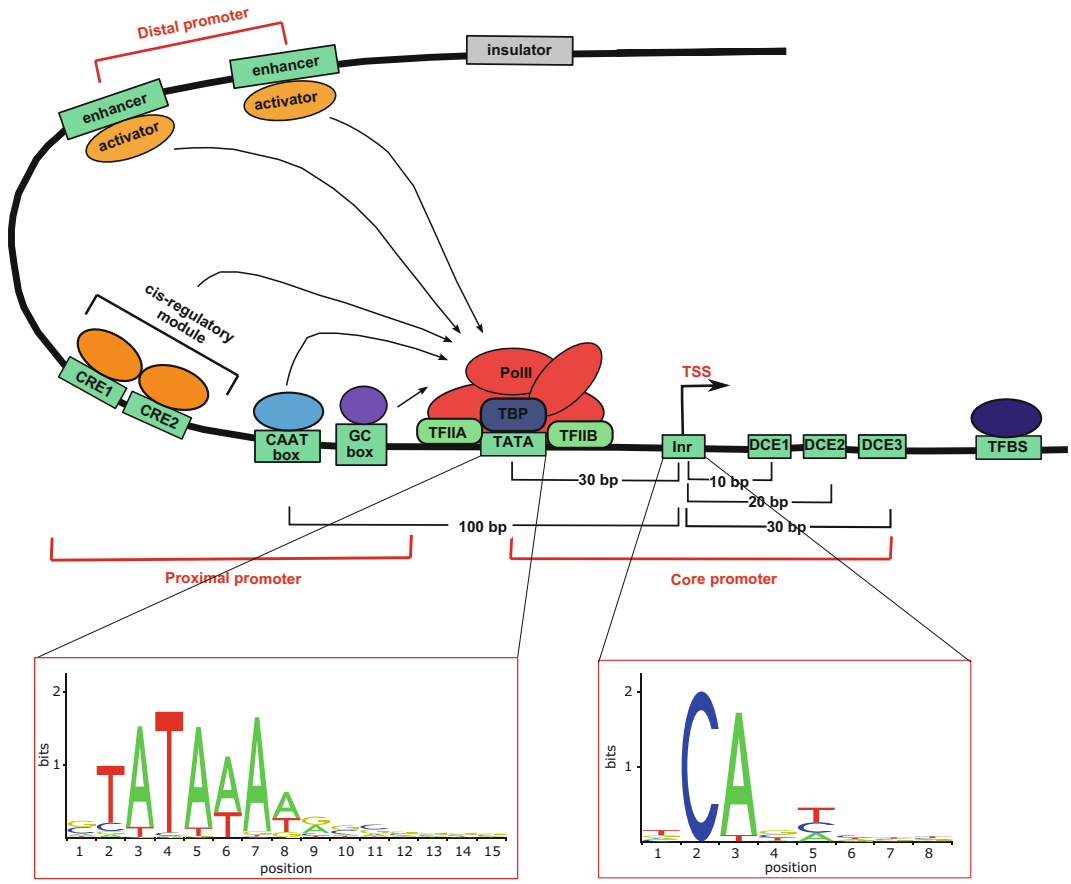

Fig. 1 Summary of regulatory elements and basal transcriptional machinery at a eukaryotic RNA Polymerase II promoter in an open chromatin region. Boundaries between accessible chromatin states are marked by insulators. The region around the Transcription Start Site (TSS) is divided into Core Promoter, Proximal Promoter and Distal Promoter. The Core Promoter contains the regulatory elements necessary to recruit Polymerase II and basal transcription factors (e.g., TFIIA and TFIIB) necessary to activate transcription, as well as the TATA box element (TATA), the Initiator site (Inr) and the downstream core element (DCE). The location of such elements with respect to the TSS is shown here as boxes and their sequence patterns (for the TATA box and Inr only) as logos from the Jaspar database [12]. Some more sequence-specific transcription factors bind to some other sequence elements, such as the CAAT- and the GC box, other Transcription Factor Binding Sites (TFBSs) in the proximal promoter or in enhancer regions. TFBSs can occur in clusters to form cis-regulatory modules (CRMs). Proximal and distal regulatory regions are brought together at TSSs to control the transcription of target genes

promoter predictions were developed to improve the genome annotation when experimental support was not available [7].

The main goal of a promoter recognition algorithm is the computational identification of genomic regions corresponding to 5′ ends of genes in a fast and reliable way, and based on the idea that promoter regions differ in several features (sequence, context, structure) from other genomic features, such as exons, 3′UTRs and intergenic regions [7].

Promoter features can be sequence signals at core-promoter elements and TFBSs, or large-scale features such as CpG islands, k-mer frequency, DNA structure, TFBS density, nucleosome binding, and chromatin modifications. Methods for promoter recognition can be *discriminative*, aiming at finding the optimal classification boundary between promoters and nonpromoters based on some selected features, or *generative* and describe the generative process of the signal. Typical discriminative models use experimentally identified promoter regions or TFBSs from databases as training set for Artificial Neural Networks (ANNs) or support vector machines (SVMs) in order to differentiate promoters from nonpromoter regions. Generative models instead learn signals of promoter elements and/or distance between binding sites from experimentally identified promoters, and apply it to find other regions that score well against the model [7].

Early 1990s' computational methods for promoter prediction combine several sequence patterns (TATA box, Inr, DPE, and BRE motifs) to classify promoter regions versus other genomic sequences [8–10]. Binding specificity is characterized, either by consensus sequences that is, giving the most preferred base at each site position within a binding site, or by Position Weight Matrices (PWMs), which assign a weight to each nucleotide at each position of a putative binding site. New binding sites are scored according to the sum of the scores of the individual positions from the PWM model. Maintained collections of PWMs include TRANSFAC [11] and JASPAR [12]. Methods based on consensus sequences and PWMs might give poor results due to the fact that TFBSs are typically short (5–15 bp long), degenerate and several hits of their consensus/model sequence can be found quite often along the genome just by chance. It became clear during the years that most promoters only have one or a few of the patterns described above, and that some patterns are only found in a small proportion of vertebrate promoters. Therefore it became possible to describe some functional groups of promoters in great detail from TFBS consensus sequences, but the false discovery rate remained high when attempting to detect core promoters genome wide [2, 7].

The late 1990s are signed from advances in algorithms or strategies for pattern finding: promoter prediction methods are not based anymore only on a collection of putative binding sites, but the so-called *context features*, i.e., k-mer content extracted from DNA sequences of promoters, are incorporated in both generative and discriminative models [7]. These algorithms are inspired by linguistic and are based on the rationale that promoter and nonpromoter regions differ in their word content. K-mers may correspond to known biological signals (e.g., TATA box), but they might also correspond to yet unknown promoter signals. PromoterInspector [13] and Promoter2.0 [14] are tools which use k-mers with variable gaps or wildcards to distinguish promoters from nonpromoters. For a comprehensive list refer to [1] and [7].

Since 2001, with the first genome projects and the sequencing efforts of the human genome, people realized that promoter recognition algorithms lack sensitivity and specificity when applied genome wide [7]. The observation that promoter features can be so diverse between different promoter subclasses changed the perspective by which computational algorithms looked at promoter prediction. In particular, CpG islands, regions of vertebrate genomes defined primarily by the lack of methylation at CpG doublets, were observed to be a large-scale signal present in about 70% of the human promoters [2], and gained more and more interest. Also, classifiers that analyze CpG-rich and CpG poor promoters separately achieve better sensitivity and specificity as the two classes seem to have different properties at sequence level [7].

In addition, people started appreciating that TFs recognize DNA-binding regions not only at sequence level, but that the conformation and structure of the DNA play a crucial role in guiding DNA-binding proteins to their sites and also influence promoter activity [7]. Hence, structural features, nucleosome positioning preferences, and others started being included, together with sequence patterns, into promoter prediction algorithms. Among them, the Eponine method, one of the best promoter prediction algorithms still nowadays, applies relevance vector machines to capture the most important sequence signals at promoters, represented by a collection of PWMs and positional constraints between them, together with CG content enrichment [15]. In this method category we find McPromoter [16], ProSOM [17] and ARTS [18] superior among others.

More advanced classifiers are ensemble methods, such as PromoterExplorer [19], CoreBoost [20], MetaProm [21] and EnsemPro [22], which combine results from multiple classifiers on multiple features in order to achieve more robust predictions.

Although it had been suggested for several years that epigenomic features, such as histone acetylation, methylation marks, and nucleosome positioning can provide an extra layer of information beyond DNA sequence features, only after 2001 such signals started to be systematically exploited for correctly locating gene promoters in open chromatin regions. Indeed, although promoters differ in their motif content or GC content, properties such as nucleosome-free regions and epigenetic features around the TSS are quite common to all active promoters [23].

Promoter recognition methods also benefit from the search of evolutionarily related sequences by looking for regions of conservation upstream of annotated genes. However, such methods can only identify homologous promoters when sequence conservation is present, but might miss nonconserved promoters [4].

The aforementioned methods predict promoters using various features but the true promoter usage has to be validated in a context-dependent manner. Recently, thanks to the advent of

next-generation sequencing technologies combined with Chromatin Immunoprecipitation (ChIP-Seq) technology [24], and nascent transcript capturing methods, such as Cap Analysis of Gene Expression coupled to NGS sequencing [25, 26] or Global run on sequencing (GRO-Seq) [27], several promoter recognition methods have moved from being purely predictive approaches based on DNA sequence or structure-related features to be data-driven, i.e., to use the observed genome-wide signals, to unravel mechanisms of transcriptional regulation instead of pure sequence features. For example, the epigenetic mark H3K4me3 and the acetylation of H2 have been identified as a hallmark of active promoters, and computational methods for promoter recognition have begun exploiting this information systematically [28, 29].

Comparative methods, as well as prediction methods based only on "first principles" (DNA sequence and structure) do not identify the conditions where certain promoters are activated. Cap Analysis of gene Expression (CAGE) instead allows high-throughput identification of 5′ ends of capped mRNA in a tissue-specific manner, allowing the localization of the associate core promoters, as well as measuring promoter usage in different states [25, 30].

In this chapter we will focus on the identification of genome-wide signals from the CAGE technology and their importance in promoter recognition. Therefore, in the following we will introduce the CAGE technology and the different FANTOM Consortiums in detail, as well as the algorithms for reliable CAGE peak recognition. Subheading 3 describes in detail the steps of the in silico analysis of CAGE data, focusing on the DPI method for CAGE signal recognition [31], and the PROmiRNA software [32], for miRNA promoter predictions. Subheading 3.7 compares the two methods for the specific task of miRNA promoter recognition.

1.1 The CAGE Techniques and the FANTOM Consortium

Cap Analysis of gene Expression (CAGE) allows the identification of transcriptional starting points genome wide by sequencing 5′ ends from full-length cDNA libraries and mapping back those sequences to the genome, thus determining regions corresponding to active promoters of coding and noncoding transcripts, as well as active enhancers. In detail, in its first version the method uses cap-trapper full-length cDNAs to attach linkers to their 5′-ends. This is followed by cleavage of the first 20 base pairs by class II restriction enzymes, PCR, concatamerization and cloning of the CAGE tags. Sequenced CAGE tags mapped to the genome are then used to identify the TSSs of annotated or novel transcriptional units specific to each tissue, cell or condition, as well as the analysis of differential promoter usage [25]. Compared to RNA-seq or microarray, CAGE allows the separate analysis of multiple promoters linked to the same gene. In fact, most genes have more than one TSS and the regulatory inputs or TFs that determine TSS choice and activity in a particular tissue are diverse.

FANTOM stands for the Functional Annotation Of Mammalian genome and is an international research consortium founded in the year 2000 to assign functional annotations to the full-length complementary DNAs (cDNAs) that were collected during the Mouse Encyclopaedia Project at RIKEN. Research at FANTOM has proceeded in three phases. FANTOM began with the establishment of an annotation pipeline that developed and expanded quickly into more transcriptome and functional analysis.

Only in the second phase, with the FANTOM3, the Consortium started using the CAGE technology to study transcriptional initiation genome wide. FANTOM3, which focused on identifying transcribed components of mammalian cells, improved the estimation of the total number of genes and their alternative transcript isoforms in both human and mouse, and revealed that about 70 % of the genome is transcribed as RNA, confirming the existence of thousands of noncoding RNAs (ncRNAs) [30]. This led us to gain new insights into how transcription initiation works and to revise central dogmas of Molecular Biology, projecting us into an "RNA world," whose functional implications are still partially to be discovered.

More in detail, in the FANTOM3 145 mouse and 41 human libraries are analyzed; CAGE tags of size 20–21 bp are derived from transcripts sequenced in proximity of the cap site. Amplified tag libraries contain between 50,000 and 100,000 tags. Clones are sequenced with Sanger sequencing techniques and their unique mapping positions on the genome identify putative TSSs. Clusters of overlapping tags define promoter strength and shape. Based on these data, Carninci et al. [30] classify tag clusters into different shapes, ranging from single-peak TSSs to broad or bimodal tag distributions, corresponding to different promoter contexts [30]. Given that the data constitute a quantitative profiling of relative promoter usage across tissues and cell types, it is observed that alternative promoter usage is higher than expected, with the majority of protein-coding genes having two or more alternative promoters, especially in brain tissues [33, 34].

In the era of high-throughput sequencing, the FANTOM4 Consortium develops deepCAGE (CAGE followed by deep sequencing of the tags). The CAGE method is adapted to the 454 Life Sciences (Roche) GS20 sequencer and the main difference consists in the fact that cloning is no longer necessary, as after amplification and concatenation the tags can be directly sequenced, generating libraries of up to two million tags [26]. The focus of the FANTOM4 also shifts from the recovery of transcribed elements to the integration of such components into biological networks for functional analysis in specific contexts such as Leukemia or monocyte differentiation [35].

In FANTOM5, the HeliScopeCAGE technique is introduced, an adaptation of CAGE to single molecule sequencing with the revolutionary HeliScope Single Molecule Sequencer measurements [36].

Such technique opens the door to detailed analysis of gene expression levels and rare cell populations, providing the community with a promoter expression atlas where expression profiles are determined at an unprecedented depth and high precision [31]. Unlike earlier sequencers, the Heliscope Sequencer does not employ polymerase chain reaction (PCR) amplification to multiply DNA fragments, a process which can introduce biases into data, instead the reverse-transcribed DNA is sequenced directly, enabling direct, high-precision measurements [36]. The latest CAGE dataset from FANTOM5, includes 573 human and 128 mouse primary cell samples, 152 human post-mortem samples, 271 mouse developmental tissue samples and 250 different cancer cell lines sequenced to a median depth of four million mapped tags per sample [31].

Given that promoter-distal regulatory regions such as enhancers are essential in controlling time- and cell-specific gene regulation, and that they have been shown to be often transcribed by PolII, producing so-called eRNAs, FANTOM5 CAGE data are also used to detect actively transcribed enhancers. Based on the data from hundreds of cell lines and tissues, Andersson et al. identify more than 40,000 enhancer regions, together with their activation levels across human tissues, marked by the presence of bidirectional capped transcripts [37].

1.2 Methods for the Analysis of CAGE Data

Either CAGE data are used to locate active promoters of known genes, or to identify start sites of novel transcripts, or to locate active enhancers, appropriate computational methods are needed to analyze the NGS data and detect transcriptional events above noise. As CAGE tags tend to be clustered, with more or less signal, at active transcription sites, the task of identifying signal-enriched regions is similar to the peak calling step in the analysis of ChIP-seq data. Peak calling methods, such as HOMER [38] can be applied to identify peaks corresponding to initiation events in CAGE data. However CAGE peaks/clusters possess specific features that distinguish them from ChIP-Seq data, so that dedicated methodologies have been developed in the past few years specifically for the analysis of CAGE data. Initial studies of CAGE dataset have employed basic methods for processing mapped CAGE tags and identifying CAGE TSSs [30, 34]. Active promoters have been reconstructed by means of different clustering approaches based either on the proximity of individual TSSs or their density [39]. With the increase of sequencing depth, in order to perform TSS-centered differential expression analysis, normalization approaches, and explicit noise modeling have been introduced [40]. In this chapter we will focus on the Decomposition-based peak identification (DPI) method, especially designed for FANTOM5 CAGE data and methodology of choice for most of the FANTOM5 subsequent analysis (Subheadings 3.1–3.3). As CAGE data can locate both coding and noncoding transcript TSS, we describe the PROmiRNA software, especially

designed for the challenging task of identifying miRNA promoters from either FANTOM4 or FANTOM5 data (Subheadings 3.4–3.6). Although PROmiRNA has several analysis steps in common with other CAGE analysis methodologies (see below), its underlying statistical model is optimized for the de novo detection of lowly expressed TSSs, and this makes it particularly suitable to detect both intergenic and intronic transcription initiation events of transient miRNA primary transcripts, which undergo rapid processing by the Drosha enzyme in the cell nucleus, yielding sparse CAGE tag coverage around true TSSs.

Although not described in this chapter, a relatively new software package which integrates several CAGE analysis workflows is the R/Bioconductor package CAGEr [41]. CAGEr implements various methods for CAGE data processing, it provides several normalization strategies, easy access to published CAGE dataset in several organisms and introduces a novel method for detection of differential TSS usage and promoter shifting in different tissues/contexts.

The main steps of the analysis of CAGE data, common to several CAGE analysis pipelines, can be summarized as follows:

1. *Library preparation and sequencing.* The CAGE technology has evolved during the last 10 years and the different protocols for library preparation and sequencing have been discussed above.
2. *Read mapping.* The first step in the analysis of CAGE data is the mapping of the CAGE tags back to the genome. Depending on the sequencing protocols, different mapping tools and strategies have been employed for this task. In FANTOM3, still based on Sanger sequencing, CAGE tags of 20–21 nt are aligned on the genome using BlastN [42]. Tags mapping on multiple genomic regions are not used for subsequent analysis and only best alignments of at least 18 nt are kept for subsequent analysis [30]. The data from the FANTOM4 are mapped with different tools: for example Valen et al. [34] use BLAST/V alignment programs and only the longest matches without mismatches are selected, whereas matches shorter than 18 nt are discarded and multi-mapping CAGE tags are included according to a computed posterior probability for each mapping location [43]. Balwierz et al. [40] use the same strategy for multi-mapped reads, but CAGE tags are aligned with the Kalign2 alignment tool, which maps tags in multiple passes [44]. Specifically, tags that do not map perfectly to the genome are given as input to a second step, where they are mapped with at most one mismatch or event to a third step, where they are mapped allowing indels. In FANTOM5, sequenced Heliscope reads have different lengths, without associated base quality values and high sequencing error rates (up to 5%) [31]. After removal of reads corresponding to ribosomal RNA, all remaining CAGE reads are mapped to the genome using the probabilistic mapper Delve, which places reads

to single positions in the genome according to a computed probability of being a true match from a Hidden Markov Model [31, 45].

3. Analysis of CAGE tag peaks: the most important step in CAGE data analysis is the identification of regions of significant CAGE tag signal, equivalent to clusters of overlapping tags or highly dense tag regions. Most genes are transcribed in different isoforms that use different TSSs arranged typically in local clusters spanning regions from few to over 100 bps. Depending on the application, different methods deal differently with the question: what defines a Tag Cluster (TC)? On the FANTOM3 data, Carninci et al. grouped individual CAGE tags that had identical sequences into a representative CAGE tag [30]. Representative CAGE tags with the same starting position define a CAGE tag-defined transcriptional start site (CTSS) (*see* **Note 1**). As the focus of the FANTOM3 is to characterize all distinct transcription initiation events, the authors simply cluster CAGE tags whose genomic mapping overlap by at least 1 bp in Tag Clusters (TCs) (*see* **Note 2**). The PROmiRNA methods (extensively described in Subheading 3.4) defines TCs in a similar way, except that it joins together in the same cluster also tags which do not overlap with each other, but are closer than 20 bp from one another. This allows recovering much more sparse tag signal as the one generated by transient microRNA primary transcript TSSs. More sophisticated approaches to define tag clusters include the Paraclu algorithm [39] and the method from Balwierz et al. [40]. The Paraclu algorithm is based on the observation that core promoters do not have a single TSS, but a distribution of initiation sites clustered at multiple scales as a consequence of multiple regulatory processes. The Paraclu algorithm aims at finding these clusters, at multiple scales, among transcription initiation events observed at specific locations in the genome by finding maximal scoring segments with a density of more than d events per nucleotide. Afterwards, an inhomogeneous HMM is learned from dominant TSS (clusters associated to at least five transcription initiation events) to determine sequence preferences of TSSs and apply the trained model to discover new TSSs genome wide. The approach in Balwierz et al. takes into account expression profiles of TSSs across different samples and finds clusters of nearby co-expressed TSSs by using Bayesian hierarchical clustering. More in detail, their goal is to define Transcription Start Clusters (TSCs) of contiguous TSSs such that expression profiles of clustered TSSs are the same among tissues up to measurement noise. In FANTOM5, given the much higher sequencing depth compared to previous studies, a simple clustering procedure such as the one used by Carninci et al. or PROmiRNA, would generate very long clus-

ters. In DPI [31] the authors first group CTSSs (*see* **Note 1**) from different tissues into tag clusters according to the procedure from Carninci et al., and then try to separate distinct transcriptional events inside each cluster by means of Independent Component Analysis (ICA, *see* **Note 3**). All steps of DPI analysis are described in detail in Subheading 3.1.

4. TSS-centered differential expression: to quantify the expression of individual TSSs and enable comparison between samples, raw tag counts have to be normalized. Many studies based on deepCAGE use the number of tag per million (TPM) values, which is the simplest normalized measure also used on the FANTOM5 data and widely used in other high-throughput sequencing tools [46]. One of the more sophisticated approaches [40] is based on the observation that the reverse cumulative distribution of the number of tags per TSS follows a power-law distribution with a very good approximation. Therefore, CAGE tag counts across different samples are transformed to match a common reference power-law distribution. Normalization can be performed at promoter level (cluster tag counts are normalized) or at individual TSS level. For example, in order to take into account substantial differences in the total numbers of read counts, PROmiRNA counts the number of overlapping 5′-ends at each bp position and performs per-position quantile normalization across tissues, inspired by normalization methods for microarray analysis [14, 32]. After applying any of the aforementioned normalization procedures, normalized CAGE tag counts can be used to perform differential expression profiling at single TSS or promoter level.
5. Assignment of tag clusters to genes: unnormalized or normalized CAGE tag clusters (also referred to as CAGE peaks) from a CAGE analysis tool can be used to define transcription start sites of novel transcripts or assign active promoters to known genes. When trying to assign CAGE peaks to known annotation, one needs to define a distance cutoff to assign a peak to the closest gene. In [31] the authors assign a peak to a known transcript if its 5′ end is within 500 bp from the defined peak. Such distance cutoff is arbitrary and depends on the research application. Obviously it can happen that more than one peak is assigned to the same gene, or the same peak is within a certain distance to more than one transcript. Solutions to such situations differ according to the research motivation.

2 Materials

The purpose of this chapter is to give details on practical aspects regarding the computational analysis of CAGE data and usage of the DPI and PROmiRNA software, with emphasis on the CAGE

peak calling step. As the analysis with these two software is based on raw or preprocessed data from the FANTOM4 and/or the FANTOM5 Consortium, in this paragraph we provide some details about data sources and specific data files. The PROmiRNA software was originally designed to recognize miRNA human promoters from FANTOM4 deepCAGE data, but can also be applied to data from the FANTOM5 in both human and mouse.

FANTOM4 raw data, mapped data, as well as tag count files and processed files containing the annotation of the detected promoters can be downloaded at the following link: http://fantom.gsc.riken.jp/4/download/Tables/.

To facilitate data interpretation and integration, as well as navigate through the FANTOM4 dataset, Severin et al. developed the EdgeexpressDB database [47]. Such source not only collects alternative promoters and gene expression patterns across tissues, but also provides a regulatory network view of the data, including regulating factors and microRNAs.

All FANTOM5 data, including visualization and web-based tools, different data access points, CAGE raw data (fasta sequences), mapped CAGE data (bam files), as well as processed data from human and mouse samples, including position and expression of CAGE peaks, are precomputed and available on the following website http://fantom.gsc.riken.jp/5/. In particular, two specialized tools allowing exploring relations between the data, namely ZAMBU, which is useful if one wants to investigate the relationship between CAGE tag distributions and expression profiles [31, 48], and STARR, a semantic tool to explore relationships between promoters, genes, samples, and TFBSs [49].

When interested in sample-specific CAGE peak information, one can download the corresponding CAGE tag starting site file (ctss file) from the aforementioned website, and then use DPI to identify CAGE peaks based on such input file. The GM12878 ctss files used as input to DPI for the example shown in this paper and the GM12878 CAGE bam file used as input for PROmiRNA are downloaded from http://fantom.gsc.riken.jp/5/datafiles/latest/basic/human.cell_line.hCAGE/.

The DPI software (extensively described in Subheading 3.1) is available for download on Github http://github.com/hkawaji/dpi1/.

The PROmiRNA software (extensively described in Subheading 3.4) can be freely downloaded at http://promirna.molgen.mpg.de together with the *external_data.tar.gz* directory.

3 Methods

This section will focus on computational methods for peak calling and promoter identification from CAGE data. First, we will introduce the Decomposition-based Peak Identification (DPI) method,

especially designed for FANTOM5 CAGE data and applicable to both promoter and active enhancer recognition [31]. Second, we will introduce the PROmiRNA software, especially designed for miRNA promoter recognition from both FANTOM4 and FANTOM5 CAGE data [32]. Third, we will show as an example, the results from applying both DPI and PROmiRNA to identify miRNA promoter of expressed miRNAs in the Gm12787 B-lymphoblastoid cell line.

3.1 The DPI Algorithm

The main steps of the DPI algorithm are illustrated in Fig. 2a and described in detail below. The intermediate output files generated from each step are schematically described in Fig. 2b.

1. *Input*. The input to the DPI algorithm is represented by one or more ctss files (*see* **Note 1**) from tissue-specific mapped tags (Fig. 2a, Step 1). These correspond to CAGE profiles at individual biological states. Only ctss supported by two or more CAGE 5′-end reads in a single profile are used by DPI.
2. *Definition of CAGE tag clusters (TCs)*. Cage tags are clustered based on proximity to each other. Input ctss from different tissues are first merged to produce an accumulated CAGE profile (Fig. 2a, Step 2). Selected ctss, supported by no less than two reads are grouped together into the same cluster if they are within 20 bp from each other.
3. *TC Decomposition*. Due to higher sequencing depth compared to previous CAGE dataset, step 2 may produce very long tag clusters, which might contain several transcription start sites. To correct for this, DPI uses Independent Component Analysis (ICA, *see* **Note 3**) on clusters wider than 50 bp (or with a coverage higher than 50 tag counts), in order to decompose the overall signal into distinct TSS signals (Fig. 2a, Step 3). Within each cluster, ICA infers the number of underlying signals which correspond to 95% of the signal variance (and up to a maximum of 5 independent components) and represent individual ctss intensity patterns.
4. *Scaling*. The signal in each inferred independent component is downscaled by 10% of the intensity of its highest ctss. This step is performed in order to avoid detecting "too much signal" in proximity of very active TSSs, where a continuous but modest read coverage is observed (Fig. 2a Step 4).
5. *Smoothing*. At this stage, DPI applies a Gaussian kernel to smooth each independent signal component in each cluster and detect candidate peaks where the signal is higher than the median of each signal component (Fig. 2a Step 5).
6. *Merging*. Inferred peaks are merged if they overlap with each other (Fig. 2a Step 6).

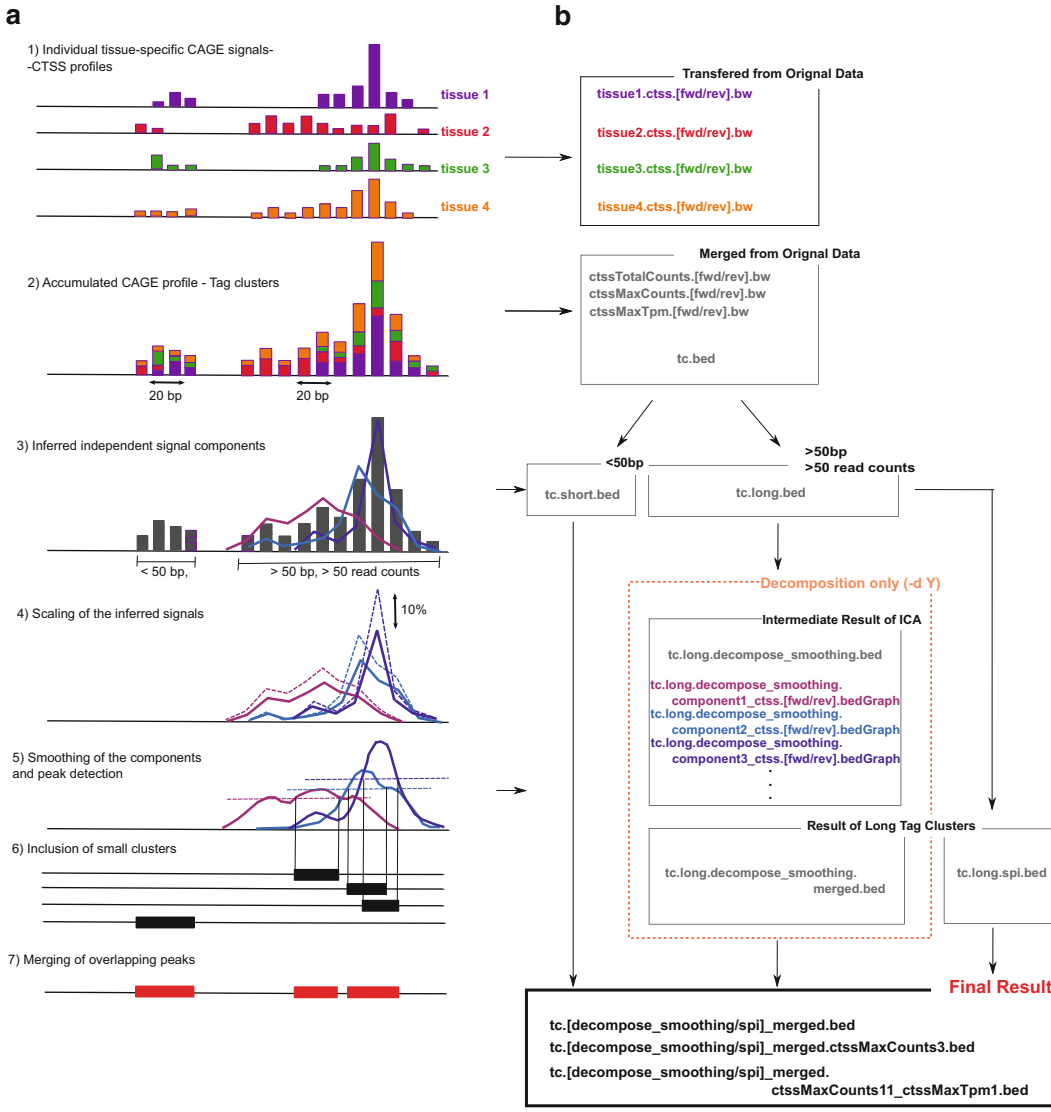

Fig. 2 (**a**) Different steps of the DPI workflow, from parsing of the input ctss files to the final CAGE peaks; (**b**) Intermediate and final output files from the DPI pipeline. Adapted by permission from Macmillan Publishers Ltd: Nature [31], copyright 2014

7. *Output*. Finally, aggregated peak regions are reported together with short clusters (<50 bp) which were not selected in Step 3 for ICA processing (Fig. 2a Step 7). In order to minimize the fraction of peaks mapped to internal exons and enrich for promoter regions, DPI applies a tag threshold to define robust and permissive output peaks, based on the assumption that genuine TSSs have a higher number of 5′ tags starting at the same position than random regions along the transcript. A fold enrichment of at least 2.0 over random regions (equivalently peaks with a single ctss supported by at least 11 reads) defines

the robust cutoff, while the more permissive cutoff corresponds to a fold enrichment of 0.7 (single ctss supported by at least three reads in at least one CAGE profile). Both "robust" peaks and "permissive" peaks are reported by DPI.

Although it is not a part of the DPI pipeline, the detected peaks can be used to quantify tissue-specific expression of transcription start regions. In [31] the authors, after applying the DPI pipeline to the FANTOM5 data, count the number of tags whose 5′ ends start within the boundary of a "robust" peak in that tissue. In order to compare TSS activity between tissues, read counts are transformed into TPM (tag per million) values and normalizing factors are estimated using the relative log expression (RLE) method implemented in the EdgeR R package [50].

3.2 Practical Usage of the DPI Software

DPI runs on the Unix/Linux system with Grid Engine without installation. If Grid Engine is not available, one can still use it without the decomposition step (see below). Before using it, one should insure that the following languages/software are available on the system:

Ruby (https://www.ruby-lang.org)

R (http://cran.r-project.org/) and the R package fastICA (http://cran.r-project.org/web/packages/fastICA/index.html)

Command line bigWig tools (http://hgdownload.cse.ucsc.edu/admin/)

BEDtools (https://github.com/arq5x/bedtools2)

Importantly, one should declare these tools in the system environment.

Download DPI from github using command line:

```
> git clone https://github.com/hkawaji/dpi1.git
```

A packed shell script: *DPI_DIR/dpi1/identify_tss_peaks.sh* is included in the package. One can view detailed package information, parameters, and output explanation by running this script:

```
> DPI_DIR/dpi1/identify_tss_peaks.sh
```

Before peak calling, prepare the following input files:

Chromosome size file in BEDTools (should be automatically provided):

```
BED_DIR/genomes/YOUR_SPECIES.genome
```

ctss files in bed format downloaded from FANTOM http://fantom.gsc.riken.jp/5/datafiles/latest/

After specifying the output folder, simply run:

```
> DPI_DIR/dpi1/identify_tss_peaks.sh -g genome -i CTSS FILE -o OUTPUT_DIR -d Y/N
```

where -d is an optional parameter and is set to "N" by default. When –d is specified to Y, the decomposition step will be performed (see DPI algorithm from Subheading 3.1). In general, DPI takes as input multiple .ctss files: one can simply put all .ctss files in one folder and set the input parameter as:

```
-i'CTSS FOLDER/*.ctss.bed.gz'
```

3.3 The DPI Output

After running DPI as shown above, the output consists of three folders: *outCounts*, *outTpm* and *outPooled*. *outCounts*, and *outTpm* contain bigwig files for each individual input ctss file, with the value being tag counts and tags per million (TPM), respectively. Tags on forward strand (fwd) and reverse strand (rev) are reported separately. The *outPooled* folder contains the following result files for the intermediate steps illustrated in Fig. 2b:

bigwig files correspond to pooled individual ctss files (Fig. 2b Step 2):

ctss.[MaxCounts/MaxTpm/TotalCounts].[fwd/rev].bw

bed files for all/long/short tag clusters (Fig. 2b Step 3):

tc.[-/long/short].bed.gz

If the decomposition parameter –d is specified, bed/bedGraph files from decomposition step will be generated. (Fig. 2b Step 4/5):

```
tc.long.decompose_
smoothing.*.[bed/bedGraph].gz
```

The merged peak files (Fig. 2b Step 6/7):

peaks with robust threshold, i.e., more than 10 ctss tags and no less than 1 TPM (Fig. 2b Step 6/7):

```
tc.[decompose_smoothing/spi]_
merged.[ctssMaxCounts11/ctssMaxCount11_
ctssMaxTpm1].bed.gz
```

peaks with permissive threshold, i.e., more than 2 ctss tags (Fig. 2b Step 6/7):

```
tc.[decompose smoothing/spi]_
merged.ctssMaxCounts3.bed.gz
```

3.4 The PROmiRNA Software

Due to fast Drosha cleavage in the nucleus, miRNA primary transcript TSSs are hard to identify from sparse CAGE tag coverage with conventional methods. The PROmiRNA algorithm combines CAGE tag counts and several promoter sequence properties into a statistical model, in order to identify miRNA promoter at high sensitivity, while distinguishing them form transcriptional noise. The main steps of the PROmiRNA methodology are illustrated in Fig. 3a and described in detail below.

1. *Input*. The input to PROmiRNA is represented by more than one tag alignment file, one for each tissue, in bed format. These are used to build the tissue-specific CAGE tag profiles up to 50 kb upstream of annotated miRNA precursors. Such

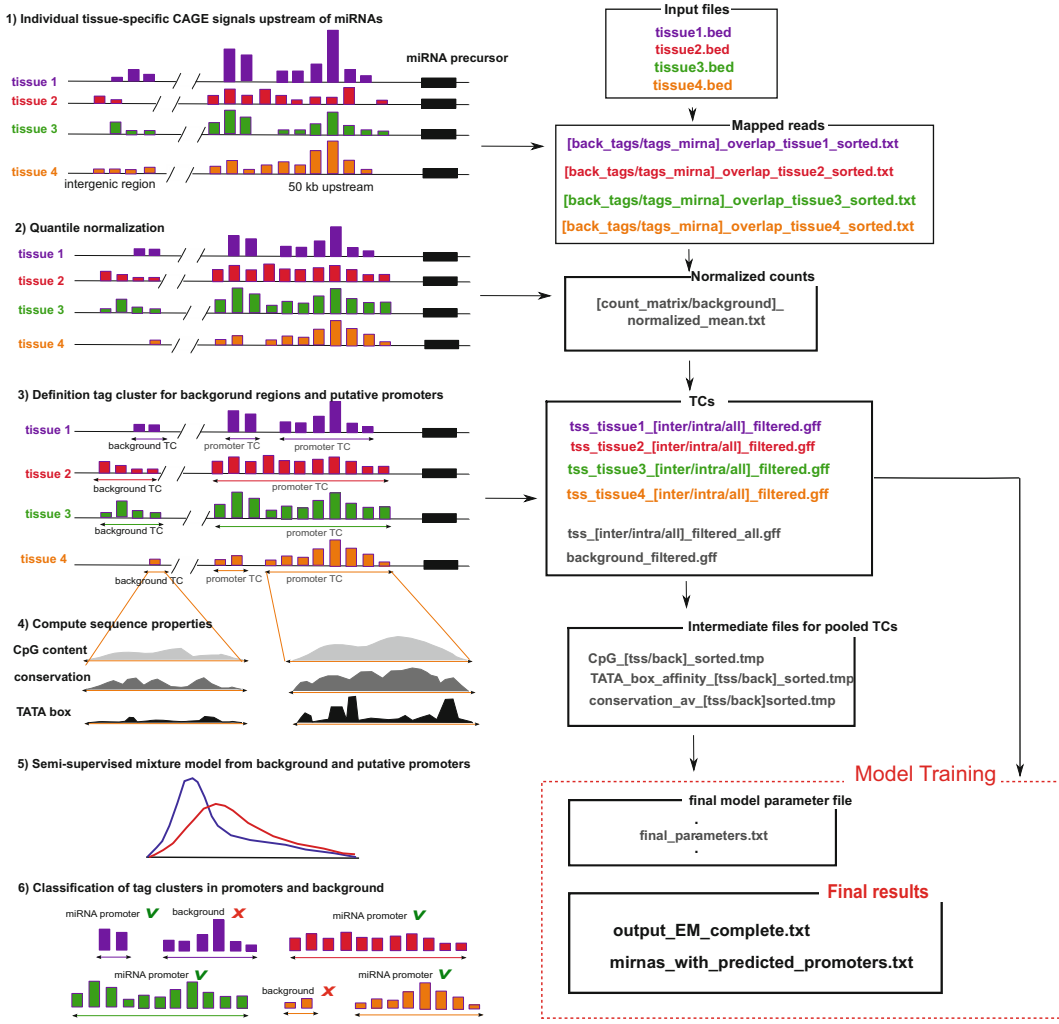

Fig. 3 (**a**) Different steps of the PROmiRNA workflow, from parsing of the input bed file to the final list of CAGE peaks corresponding to miRNA promoters; (**b**) intermediate and final output files from the PROmiRNA pipeline

profiles represent CAGE read coverage (or tag counts) at 1 bp resolution (Fig. 3a, Step 1).

2. *Tag-count normalization.* In order to make tag counts comparable across tissues, row counts at each bp position are quantile-normalized. In detail, position-specific tag counts from each sample are transformed to match a common reference distribution, randomly chosen from the available libraries. Normalized tag counts can be interpreted as expression values at TSS level at 1 bp resolution (Fig. 3a, Step 2).
3. *Cage Tag clusters (TCs)—Identification of putative promoter regions.* CAGE tags are grouped into clusters if the overlap between their genomic coordinates is at least 1 bp (or they are within a distance of 20 bp from each other). Normalized tag

counts inside each cluster are summed up. Tag clusters whose genomic coordinates overlap with the TSSs of other annotated transcripts other than miRNA host genes are filtered out (not shown). Tag clusters located in the 50 kb region upstream of miRNA precursors which do not overlap any known TSS define putative miRNA promoter regions in a tissue-specific manner. Tag clusters located in randomly selected intergenic regions are defined in the same way and are interpreted as non-promoters, therefore assumed to represent background noise (Fig. 3a, Step 3).

4. *Sequence properties of tag clusters.* The statistical model of PROmiRNA computes a prior probability for each TC of being a real promoter, based on the following sequence properties computed in the 1000 bp genomic regions around the center of each defined TC (Fig. 3a, Step 4): Normalized CpG content, computed as described in [32];

   Average PhastCons conservation score across on a 46-way vertebrate alignment downloaded from the UCSC Genome Browser (*see* **Note 5**); Affinity for a TATA box protein, computed by means of the TRAP program (cite) and based on the position-specific scoring matrix (pscm) downloaded from the Jaspar database (http://jaspar.genereg.net/, Jaspar ID: MA01082);

   Genomic proximity score of the defined TC to the miRNA precursor.

5. *PROmiRNA's mixture model.* Pooled TCs from all tissues, together with their normalized tag counts and computed sequence properties, are fed into a semisupervised mixture model which automatically learns, through an EM algorithm, the optimal separation between TCs corresponding to promoters and TCs corresponding to background. TCs from random intergenic regions are interpreted as "exact" negative examples by the model (supervised part), while TCs upstream of miRNAs are nonlabeled examples (unsupervised part) which might either belong to the miRNA promoter class or to the background noise (Fig. 3a, Step 5).

6. *MiRNA promoter assignment.* TCs upstream of miRNA promoters are classified as miRNA promoters, if the computed posterior probability from the model is higher than 0.5, otherwise they are classified as background. The main output of the PROmiRNA software is a list of predicted promoters, for each miRNA gene, together with their genomic coordinate (Fig. 3a, Step 6).

3.5 Practical Usage of the PROmiRNA Software

PROmiRNA runs on every Linux/Unix environment. Before using it, one should make sure that the following languages and tools are available on the system:

python 2.7 (https://www.python.org/download/releases/2.7/).

PROmiRNA does not run with Python 2.6 or Python 3.x

R >=2.12.1 (http://cran.r-project.org)

Perl >= 5.12 (http://perl.org/get.html) and BIO:Graphics perl module (http://search.cpan.org/dist/Bio-Graphics-2.34/)

BEDtools (http://code.google.com/p/bedtools)

cd-hit (http://weizhong-lab.ucsd.edu/cd-hit/download.php)

ANNOTATE 3.04 (http://trap.molgen.mpg.de/download/TRAP/ANNOTATE-3.04.01.tar.gz)

After unzipping and placing the *external_data* directory into the *PROmiRNA* main directory we can have a brief look at the PROmiRNA subdirectories structure. The PROmiRNA folder contains four subdirectories:*PROmiRNA/src*, it contains all necessary code to run PROmiRNA*PROmiRNA/miRBase*, it contains miRNA annotation files downloaded from the miRBase database [51]. *PROmiRNA/external_data*, it is further divided into two subdirectories, *bed_files*, where input files to the software (tag alignments) in bed format should be placed (*see* **Note 4**), and *Phastcons*, where chromosome-wise PhastCons conservation files should be placed (*see* **Note 5**). This directory contains also other data files necessary for the analysis: genome files for the organism under study (a chromosome size file, e.g., *hg19.chrom.sizes* (*see* **Note 6**), a fasta file for the whole genome, e.g., *hg19*.fa and its corresponding index file, e.g., *hg19.fa.fai* (*see* **Note 7**)); the annotation of the repetitive regions for the organism under study (e.g., *hg19_repeats.bed*, *see* **Note 8**); a gtf file containing Ensembl gene annotation (e.g., *Homo_sapiens.GRCh37.66 gtf*, *see* **Note 9**)*PROmiRNA/Data*, it contains all intermediate output files and it is divided into four subdirectories, namely *gff_files*, where all tissue-specific, as well as pooled TCs are stored, *fasta*, where fasta sequences of TC regions are stored, *background*, where TCs and computed sequence properties for the background TCs are stored, *overlap_files*, which stores intermediate overlap files between miRNA genomic coordinates and CAGE tags in different tissues and *matrix_file*, where intermediate matrices of read counts before and after quantile normalization are stored. This *Data* directory contains many other intermediate files, the most important being discussed in the next session. The PROmiRNA software can be used in two different modes, depending on the application:

1. Testing mode. Given a set of genomic regions in gff format, test if they contain one or more miRNA promoters, based on a pretrained PROmiRNA model. After defining the output directory where all intermediate files and final results will be placed, run

   ```
   > python test_new_regions.py <out_dir> <input_file>
   ```

   For example: given the *test_regions.gff* provided inside the PROmiRNA directory, and setting output_dir = *test_regions*, run:

```
> python test_new_regions.py test_regions
test_regions.gff
```

In the output directory, the main result files from this command are:

output_EM.txt, it contains promoter regions, as well as background TCs, with their respective genomic coordinates, normalized tag counts, values of the computed features and prior and posterior probabilities from the model.

miRNA_predicted_promoters.txt, it reports, for each miRNA in the input file, genomic coordinates of the predicted promoter TCs, together with normalized CAGE tag counts and genomic distance of the TC from the miRNA precursor.

2. Training mode. The original PROmiRNA model is trained on FANTOM4 data on the human assembly hg19. If you want to use PROmiRNA with new CAGE libraries (e.g., FANTOM5 or Encode data) or on a new assembly / organism, we strongly suggest to re-train the PROmiRNA model.

   After downloading the necessary files (*see* **Notes 4–9**), retrieve the miRNA annotation from miRBase [51]:

   ```
   > python download_mirbase_annotation.py <org> <v>
   ```

   Where *org* is the official three-letter code for the organism identifier (e.g., *hsa* for human) and *v* indicates the miRBase release number. This command downloads the following annotation files in the PROmiRNA/miRBase directory:

   [org].gff2, a gff file containing the genomic coordinates of all precursor miRNAs for a specified organism *org*;

   miRNA.txt, annotation file containing information about each mature miRNA (e.g., accession, species, genomic sequence..);

   mirna_context.txt, annotation of the genomic context of a miRNA (intergenic, intron, exon, 3′ UTR, 5′ UTR)

   The training itself is done via:

   ```
   > python PROmiRNA.py <genome>
   ```

   Where `<genome>` refers to the genome assembly specified for promoter prediction, e.g., hg19.

3.6 The PROmiRNA Output

Although PROmiRNA produces many intermediate files during both testing and training, the most important output files are summarized below and illustrated in Fig. 3b.

1. Files reporting the overlap between CAGE tags and regions upstream of miRNAs / random intergenic regions for each tissue, sorted by genomic position:

   ```
   PROmiRNA/Data/overlap_files/[back_tags/tags_
   mirna]_overlap_<tissue>_sorted.txt
   ```

2. Matrix file of the normalized CAGE tag counts across tissues for both putative TSS positions and random intergenic regions:
   `PROmiRNA/Data/matrix_file/[count_matrix/background]_normalized_mean.txt`
3. TC cluster files for candidate promoter regions (both tissue-specific and pooled across tissues), further distinguished in intragenic, intergenic, and all TCs:
   `PROmiRNA/Data/gff_files/tss_<tissue>_[inter/intra/all]_filtered.gff`
   A similar file is provided for background TCs:
   `PROmiRNA/Data/background/background_filtered.gff`
4. Files of computed sequence properties for putative TSSs and background TCs:
   `PROmiRNA/Data/gff_files/CpG_tss_sorted.tpm, PROmiRNA/Data/gff_files/TATA_box_affinity_tss_sorted.tmp, PROmiRNA/Data/gff_files/conservation_av_tss_sorted.tmp PROmiRNA/Data/background/CpG_back_sorted.tpm, PROmiRNA/Data/backgrouns/TATA_box_affinity_back_sorted.tmp, PROmiRNA/Data/backgoruns/conservation_av_back_sorted.tmp`
5. File listing the final model's parameters learned during PROmiRNA training:
   `PROmiRNA/Data/final_parameters.txt`
6. Files reporting the final promoter predictions (see previous section)
   `PROmiRNA/Data/output_EM_complete.txt, PROmiRNA/Data/mirnas_with_predicted_promoters.txt`

3.7 Case Study: Prediction of miRNA Promoters in Gm12878

To show an example of application of both DPI and PROmiRNA we used both tools to identify miRNA promoters in the B-lymphoblastoid cell line Gm12878. Although DPI is designed to define CAGE peaks genome wide, and not tuned to specifically find miRNA promoters, we can nonetheless assign DPI peaks to miRNA genes by looking at the defined DPI peaks in the 50 kb region upstream of annotated miRNAs.

3.7.1 Application of PROmiRNA to Gm12878 CAGE Data

For PROmiRNA, alignment files in bam format (two biological replicates) for the Gm12878 cell line were downloaded at

http://fantom.gsc.riken.jp/5/datafiles/latest/basic/human.cell_line.hCAGE/

For the sake of simplicity we will rename these files to *Gm12878_rep1.bam* and *Gm12878_rep2.bam*. As PROmiRNA

requires input alignments in bed format, the bam file were converted to bed format using the Bedtools:

```
> bamToBed -i Gm12878_[rep1/rep2].bam >
Gm12878_[rep1/rep2].bed
```

The two bed files were placed in the *PROmiRNA/external_data/bed_files* directory

miRNA promoters in the Gm12878 cell line from miRBase version 20 and human assembly hg19 were predicted as follows:

```
> python download_mirnabse_annotation hsa 20
> python PROmiRNA hg19
```

The miRNA promoter predictions were listed in the output file *PROmiRNA/Data/mirnas_with_predicted_promoters.txt*. This file reports the union of predicted promoters from the two replicates. In order to derive a specific and strict list of promoters, and minimize the number of false positives we applied the following constraints:

- only promoter predictions common to the two replicates were retained
- only promoter predictions in open chromatin regions were retained

In order to fulfill the second criterion we downloaded DNAseI hypersensitivity peak sites for the Gm12878 cell line from the ENCODE website http://ftp.ebi.ac.uk/pub/databases/ensembl/encode/integration_data_jan2011/byDataType/openchrom/jan2011/fdrPeaks/. For the sake of simplicity we will rename this file to *DNAseI_ Gm12878.bed*.

After converting PROmiRNA predicted promoters to gff format using a customized simple script (*promoters_Gm12878.gff file*) we computed the overlap between DNAseI hypersensitivity sites and promoters' genomic coordinates (extended by 100 bp upstream and downstream) by means of the Bedtools:

```
> windowBed -a promoters_Gm12878.gff file -b DNAseI_Gm12878.bed -w 100 -u >
promoters_Gm12878_dnase_validated.gff
```

3.7.2 Application of DPI to Gm12878 CAGE Data

The cell-specific ctss file is downloaded at http://fantom.gsc.riken.jp/5/datafiles/latest/basic/human.cell_line.hCAGE/ and given as input file to DPI.

```
> identify_tss_peaks.sh -g human.hg19.genome -i'DATA FOLDER/*.ctss.bed.gz'
-o ./result -d Y
```

The output files of the predicted Tag Clusters genome wide are:

```
tc.decompose_smoothing_merged.ctssMaxCounts11_ctssMaxTpm1.bed (robust cutoff)
```

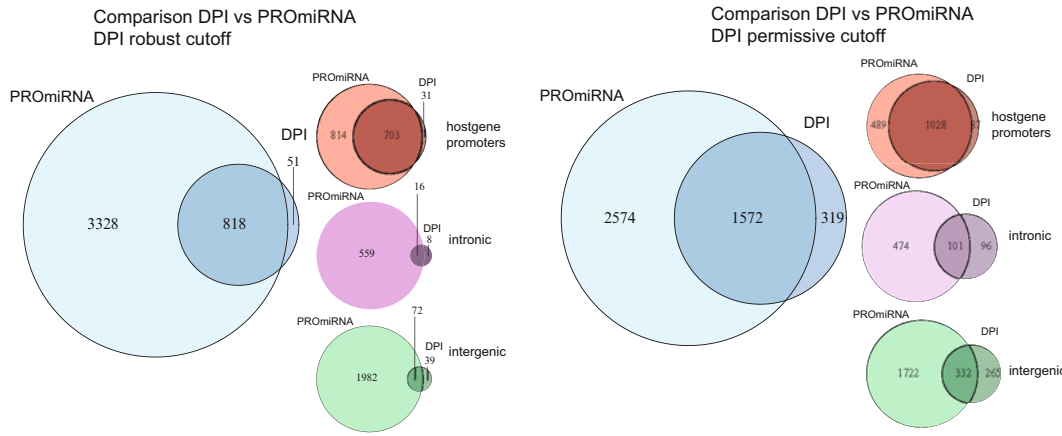

Fig. 4 Comparison between DPI and PROmiRNA for the miRNA promoter prediction task in the Gm12878 cell line. Overlap between predictions is shown for different promoter classes and for two sets of DPI predictions: set 1—where a robust cutoff of 2.0 TPM expression has been applied to the representative CTSS of a tag cluster and set 2—where a more permissive cutoff of 0.7 TPM has been applied. The overlap between DPI and PROmiRNA is improved when considering the DPI set 2 (permissive cutoff) given the lower expression values of miRNA promoters compared to gene promoters. The biggest overlap is observed for miRNA host gene promoters in both cases (DPI set 1 and set 2), whereas the overlap between the two tools is limited when it comes to the prediction of intergenic and intronic miRNA promoters

tc.decompose_smoothing_merged.ctssMaxCounts.bed (permissive cutoff)

In order to compare DPI predictions with PROmiRNA predictions we considered only DPI peaks up to 50 kb upstream of annotated miRNAs from miRBase v 20. We also filtered DPI peaks according to their overlap with DNAseI hypersensitivity regions as done above. The procedure was repeated for the two DPI output files corresponding to both the robust and permissive cutoff on the read counts.

The results from the comparison are summarized in Fig. 4. First of all, we observe that the largest overlap between DPI and PROmiRNA predictions is reached with the sets of DPI peaks at the permissive cutoff. This strengthen the argument that miRNA promoters are lowly detected compared to the protein-coding gene promoters due to fast processing of the miRNA primary transcripts, and a strict cutoff on the read counts will not allow their genome-wide identification.

The overlap between the tools is very high for miRNA host gene promoters, i.e., the promoters of protein-coding genes hosting miRNA hairpins inside their transcripts, and lower for intergenic and independent intragenic miRNA promoters, This underlines the fact that miRNA promoter prediction is still a challenging task compared to protein-coding gene promoter prediction.

Overall, PROmiRNA returns many more predictions than DPI: this might be due to the fact that PROmiRNA, unlike DPI, does not filter out genomic positions where only one tag maps, but includes them in the subsequent analysis, generating inevitably more predictions. While this is necessary in order to capture TSSs of lowly expressed miRNA primary transcripts which harbor promoter features in their sequences, it can happen that a certain fraction of PROmiRNA predictions represent false positives. However, most of predictions might represent real alternative miRNA promoters which need further investigation and validation.

4 Notes

1. The input files to DPI are CTSS files in bed format. CTSS stand for CAGE Transcription Start Site, and a CTSS file stores the 5′-end positions of the representative CAGE tags which start at the same genomic position on the same strand, together with the total number of representative CAGE tags at that position. A representative CAGE tag is a group of tags which have identical sequence (and therefore identical genomic mapping) [30].
2. A Tag Cluster (TC) is a cluster of overlapping TSSs, which are within 20–21 bp of each other. A TC genomic regions spans from the 5′-end of its most 5′-end tag, to the 3′ end of its most 3′-end tag. Two adjacent but non-overlapping tags contribute to separate TCs unless they are bridged by another tag. For a more detailed definition and some examples see [30].
3. Independent component analysis (ICA) is a useful method in signal processing, which is used to decompose a multivariate signal into subsignals (*see* Fig. 5), when knowing/assuming that the subsignals are independent and non-Gaussian, and by maximizing the statistical independency of the subsignals. The input of ICA is n observations of mixed signal, and each observation is a linear mixture of the original signals. The ICA technique is exemplified in Fig. 5. The top panel shows a simulated signal, which consists of $m = 2$ independent non-Gaussian source signal components, over time. Assume that we observed $n = 3$ independent observations of this mixed signal (middle panel). By applying the ICA technique we are able to decompose the observed mixed signal in two estimated signal components, which correspond to the original signal we want to reconstruct (lower panel).
   DPI assumes that each long tag cluster peak corresponds to a mixed signal (i.e., independent CAGE profiles). The m subsignals come from different transcriptional starting sites. The expression level of each TSS is independent from the others, and the signals are assumed to be non-Gaussian. Thus, ICA is

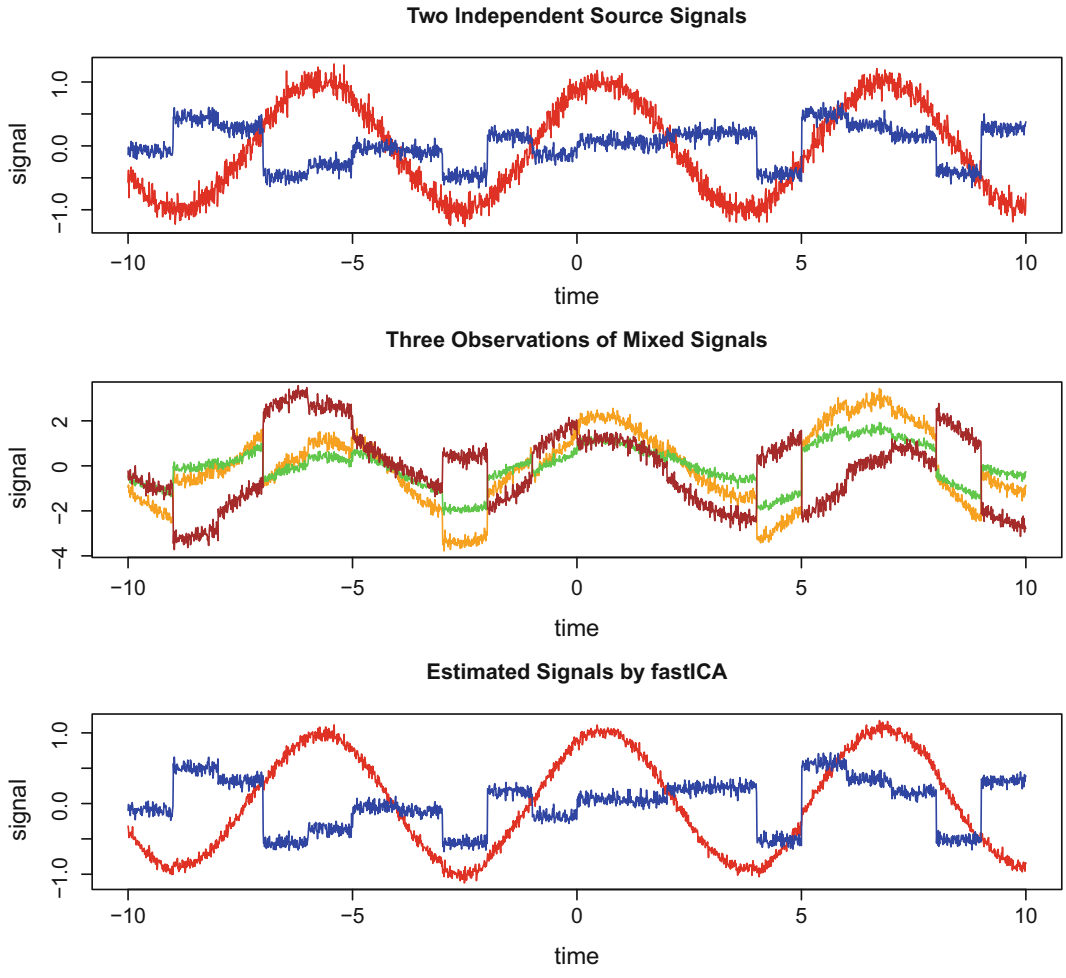

Fig. 5 Upper panel. Simulation of a mixed signal over time. Middle panel. Independent observations of a noisy mixed signals. Lower panel. Reconstruction of the two independent components of the noisy mixed signal using FastICA, an efficient R implementation of Independent Component analysis

suitable to separate mixed TSSs peaks into single TSS peaks. Different tissue types correspond here to the *n* observations.

DPI calculates ICA using the R package *fastICA*, an efficient and popular algorithm for finding an orthogonal rotation of the data [52]. In *fastICA*, the non-Gaussianity is measured as a proxy for the statistical independency using approximations to negative-entropy, which is robust and fast to compute.

4. The input files to PROmiRNA are CAGE tag alignments on the genome of interest in bed format (one bed file for each library), with six columns: *chromosome* (in UCSC format, e.g., chr1), *start* (5′-end of the aligned tag), *stop* (3′-end of the aligned tag), *tag_identifier* (or any other string), *number of tags* (number of identical tags mapping exactly at those position), strand.

5. When training PROmiRNA with new CAGE libraries on a new organism, the external annotation data (provided for hg19 with the current version of PROmiRNA) has to be built from scratch. Phastcons files in WigFix format for each chromosome can be downloaded at http://hgdownload.cse.ucsc.edu/goldenPath/hg19.

   For example, for hg19 the link is: http://hgdownload.cse.ucsc.edu/goldenPath/hg19/phastCons46way/vertebrate

   WigFix files need to be converted to the binary wib format. This can be done with the *wigFix2wib.pl* script provided in *PROmiRNA/src*. Example of usage:

   ```
   > wigFix2wib.pl inFile1.wigFix[.gz][in-
   File2.wigFix]…]
   ```

   The generated *.wib files have to be placed in the directory *PROmiRNA/external_data/Phastcons* before using the software.

6. To retrieve the *chrom.sizes* file for your organism of interest use the *fetchChromSizes* script from http://hgdownload.cse.ucsc.edu/admin/exe/linux.x86_64/fetchChromSizes.

   Example of usage:

   ```
   > fetchchromSizes <db> <db>.chrom.sizes
   ```

   <db> corresponds to one of the ucsc databases (e.g., hg18, hg19, mm9, etc.). Place the *<db>.chrom.sizes* file in the *PROmiRNA/external_data* directory.

7. Sequence fasta files for each chromosome can be downloaded at http://hgdownload.cse.ucsc.edu/goldenPath/<db>/chromosomes/, where <db> corresponds to one of the ucsc databases (e.g., hg18, hg19, mm9, etc.). Pool the individual *chrom.fa* files into a common file *<db>.fa* using the Linux command *cat*. For example:

   ```
   > cat chr1.fa, chr2.fa, …… > hg19.fa
   ```

   Place the <db>.fa file in the *PROmiRNA/external_data* directory.

   Afterwards, create a fasta index file from <db>.fa using the samtools [53]:

   ```
   > samtools faidx PROmiRNA/external_
   data/<bd.fa>
   ```

8. PROmiRNA excludes repetitive regions when forming TCs from CAGE tags. In order to allow that, it requires a file in bed format listing the genomic coordinates of annotated repetitive regions for the genome of interest. A repeat file can be downloaded from the UCSC Genome Browser with the following instructions:
   - Go on the UCSC website (https://genome.ucsc.edu) and select *Table Browser* on the left menu;

- Select the organism and the genome assembly. For example, for the human assembly hg19 select *Mammal* in the *clade* field, "Human" in the *genome* field, *GRCh37/hg19* in the *assembly* field, *RepeatMasker* in the *track* field, "genome" in the *region* field and *BED-browser extensible data* in the *output format* field.
- Click the *get output* button to retrieve the desired file in bed format
- The repeat file has to placed in the *PROmiRNA/external_data* directory.

9. In order to annotate host gene and intronic promoters for intragenic miRNAs, PROmiRNA requires a gene annotation file in gtf Ensembl format. Such a file can be downloaded from the Ensembl ftp site (www.ensembl.org/info/data/ftp/index.html) and has to be placed in the *PROmiRNA/external_data* directory.

Acknowledgements

We would like to thank Xintian You for useful criticism and proofreading of the manuscript. The project was supported by the Freie Universität Berlin within the Excellence Initiative of the German Research Foundation.

References

1. Fickett JW, Hatzigeorgiou AG (1997) Eukaryotic promoter recognition. Genome Res 7
2. Lenhard B, Sandelin A, Carninci P (2012) Metazoan promoters: emerging characteristics and insights into transcriptional regulation. Nat Rev Genet 6
3. Wasserman WW, Sandelin A (2004) Applied bioinformatics for the identification of regulatory elements. Nat Rev Genet 5
4. Yella VR, Bansal M (2014) In silico Identification of Eukaryotic Promoters. In: Systems and synthetic biology
5. Abeel T, Saeys Y, Bonnet E et al (2008) Generic eukaryotic core promoter prediction using structural features of DNA. Genome Res 18
6. Sandelin A, Carninci P P, Lenhard B et al (2007) Mammalian RNA polymerase II core promoters: insights from genome-wide studies. Nat Rev Genet 8
7. Zeng J, Zhu S, Yan H (2009) Towards accurate human promoter recognition: a review of currently used sequence features and classification methods. Brief Bioinform 10
8. Kondrakhin YV, Kel AE, Kolchanov NA et al (1995) Eukaryotic promoter recognition by binding sites for transcription factors. Comput Appl Biosci 11
9. Hutchinson GB (1996) The prediction of vertebrate promoter regions using differential hexamer frequency analysis. Comput Appl Biosci 12
10. Prestridge DS (1995) Predicting Pol II promoter sequences using transcription factor binding sites. J Mol Biol 249
11. Matys V, Kel-Margoulis OV, Fricke E et al (2006) TRANSFAC and its module TRANSCompel: transcriptional gene regulation in eukaryotes. Nucleic Acids Res 1
12. Mathelier A, Zhao X, Zhang AW et al (2014) JASPAR 2014: an extensively expanded and updated open-access database of transcription factor binding profiles. Nucleic Acids Res
13. Scherf M, Klingenhoff A, Werner T (2000) Highly specific localization of promoter regions in large genomic sequences by PromoterInspector: a novel context analysis approach. J Mol Biol 297

14. Knudsen S (1999) Promoter2.0: for the recognition of PolII promoter sequences. Bioinformatics 15
15. Down TA, Hubbard TJ (2002) Computational detection and location of transcription start sites in mammalian genomic DNA. Genome Res 12
16. Ohler U, Niemann H, Liao G et al (2001) Joint modeling of DNA sequence and physical properties to improve eukaryotic promoter recognition. Bioinformatics 17
17. Abeel T, Saeys Y, Rouzé P et al (2008) ProSOM: core promoter prediction based on unsupervised clustering of DNA physical profiles. Bioinformatics 24
18. Sonnenburg S, Zien A, Rätsch A (2006) ARTS: accurate recognition of transcription starts in human. Bioinformatics 22
19. Xie X, Wu S, Lam KM et al (2006) PromoterExplorer: an effective promoter identification method based on the AdaBoost algorithm. Bioinformatics 22
20. Zhao X, Xuan Z, Zhang MQ (2007) Boosting with stumps for predicting transcription start sites. Genome Biol 8
21. Wang J, Ungar LH, Tseng H et al (2007) MetaProm: a neural network based meta-predictor for alternative human promoter prediction. BMC Genomics 8
22. Won HH, Kim MJ, Kim S et al (2008) EnsemPro: an ensemble approach to predicting transcription start sites in human genomic DNA sequences. Genomics 91
23. Valen E, Sandelin A (2011) Genomic and chromatin signals underlying transcription start-site selection. Trends Genet 27
24. Johnson DS, Mortazavi A, Myers AM et al (2007) Genome-wide mapping of in vivo protein-DNA interactions. Science 316
25. Shiraki T, Kondo S, Katayama S et al (2003) Cap analysis gene expression for high-throughput analysis of transcriptional starting point and identification of promoter usage. Proc Natl Acad Sci U S A 100
26. Ravasi T, Suzuki H, Cannistraci CV et al (2010) An atlas of combinatorial transcriptional regulation in mouse and man. Cell 140
27. Core LJ, Waterfall JJ, Lis JT (2008) Nascent RNA sequencing reveals widespread pausing and divergent initiation at human promoters. Science 322
28. Wang X, Xuan Z, Zhao X et al (2009) High-resolution human core-promoter prediction with CoreBoost_HM. Genome Res 19
29. Megraw M, Pereira F, Jensen TH et al (2009) A transcription factor affinity-based code for mammalian transcription initiation. Genome Res 19
30. Carninci P, Sandelin A, Lenhard B et al (2006) Genome-wide analysis of mammalian promoter architecture and evolution. Nat Genet 38
31. (Dgt) FCaTRPaC (2014) A promoter-level mammalian expression atlas. Nature 507
32. Marsico A, Huska MR, Lasserre J et al (2013) PROmiRNA: a new miRNA promoter recognition method uncovers the complex regulation of intronic miRNAs. Genome Biol 14
33. Gustincich S, Sandelin A, Plessy C et al (2006) The complexity of the mammalian transcriptome. J Physiol 575
34. Valen E, Pascarella G, Chalk A et al (2009) Genome-wide detection and analysis of hippocampus core promoters using DeepCAGE. Genome Res 19
35. Consortium F (2009) The transcriptional network that controls growth arrest and differentiation in a human myeloid leukemia cell line. Nat Genet 41
36. Kanamori-Katayama M, Itoh M, Kawaji H et al (2011) Unamplified cap analysis of gene expression on a single-molecule sequencer. Genome Res 21
37. Andersson R, Gebhard C, Miguel-Escalada I et al (2014) An atlas of active enhancers across human cell types and tissues. Nature 507
38. Heinz S, Benner C, Spann N et al (2010) Simple combinations of lineage-determining transcription factors prime cis-regulatory elements required for macrophage and B cell identities. Mol Cell 38
39. Frith MC, Valen E, Krogh A et al (2008) A code for transcription initiation in mammalian genomes. Genome Res 18
40. Balwierz PJ, Carninci P, Daub CO et al (2009) Methods for analyzing deep sequencing expression data: constructing the human and mouse promoterome with deepCAGE data. Genome Biol 10
41. Haberle V, Forrest AR, Hayashizaki Y et al (2015) CAGEr: precise TSS data retrieval and high-resolution promoterome mining for integrative analyses. Nucleic Acids Res 43
42. Altschul SF, Madden TL, Schäffer AA et al (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res 25
43. Faulkner GJ, Forrest AR, Chalk AM et al (2008) A rescue strategy for multimapping short sequence tags refines surveys of transcriptional activity by CAGE. Genomics 91
44. Lassmann T, Frings O, Sonnhammer EL (2009) Kalign2: high-performance multiple

alignment of protein and nucleotide sequences allowing external features. Nucleic Acids Res 37

45. Djebali S, Davis CA, Merkel A et al (2012) Landscape of transcription in human cells. Nature 489
46. Kadota K, Nishiyama T, Shimizu K (2012) A normalization strategy for comparing tag count data. Algorithms Mol Biol 7
47. Severin J, Waterhouse AM, Kawaji H et al (2009) FANTOM4 EdgeExpressDB: an integrated database of promoters, genes, microRNAs, expression dynamics and regulatory interactions. Genome Biol 10
48. Severin J, Lizio M, Harshbarger J et al. Interactive visualization and analysis of large-scale sequencing datasets using ZENBU. Nat Biotechnol 32
49. Lizio M, Harshbarger J, Shimoji H et al (2015) Gateways to the FANTOM5 promoter level mammalian expression atlas. Genome Biol
50. Robinson MD, Mccarthy DJ, Smyth GK (2010) edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. Bioinformatics 26
51. Griffiths-Jones S, Grocock RJ, Van Dongen S et al (2006) miRBase: microRNA sequences, targets and gene nomenclature. Nucleic Acids Res 1
52. Hyvärinen A (1999) Fast and robust fixed-point algorithms for independent component analysis. IEEE Trans Neural Netw 10
53. Li H, Handsaker B, Wysoker A et al (2009) The sequence alignment/map format and SAMtools. Bioinformatics 25