Using RAMPAGE to identify and annotate regulatory elements in insect genomes

R. Taylor Raborn[*,1,2] and Volker P. Brendel[1,2]

[1]Department of Biology, Indiana University
[2]School of Informatics and Computing, Indiana University

Department of Biology and School of Informatics and Computing,
Indiana University
212 S. Hawthorne Drive 205 Simon Hall, Bloomington, IN 47401, USA
http://www.brendelgroup.org

Abstract. Application of Transcription Start Site (TSS) profiling technologies, coupled with large-scale next-generation sequencing (NGS) has yielded valuable insights into the location, structure and activity of promoters across diverse metazoan model systems. In insects, TSS profiling has been used to characterize the promoter architecture of *D. melanogaster*, and, shortly thereafter, to reveal widespread transposon-driven alternative promoter usage.

In this chapter we highlight the utility of one TSS profiling method, RAMPAGE (RNA annotation and mapping of promoters for analysis of gene expression), for the precise, quantitative identification of promoters in insect genomes. We demonstrate this using our tools GoRAMPAGE and TSRchitect, providing details instructions with the aim of taking the user from raw reads to processed results.

Keywords: *cis*-regulatory regions, promoter architecture, transcription initiation, transcription start sites (TSSs)

1 Introduction

1.1 TSS Profiling Identifies Promoters at Genome-Scale

The promoter, defined in eukaryotes as the genomic region bound by RNA Polymerase II immediately prior to transcription initiation [1], is the site where regulatory signals unite to direct gene expression. The identification of promoter regions is a valuable step for understanding the *cis*-regulatory signals that are present in an organism, and is important for genome annotation. However, despite the rapid accumulation of genome sequences across metazoan and arthropod diversity, accurate annotation of promoter regions remains sparse. This is because—empirical mapping of TSSs—precisely identifying sequence motifs that demarcate the promoter is unreliable. In contrast with current *in*

---

* Correspondence: rtraborn@indiana.edu

silico approaches, direct mapping of TSSs identifies the location of the core promoter. Cap Analysis of Gene Expression (CAGE) [2], one of the first methods devised to identify 5'-ends of mRNAs at large-scale, involves selective capture of 5'-capped transcripts, first-strand reverse-transcription and ligation of a short oligonucleotide (CAGE tag). CAGE was initially utilized by the FANTOM (Functional Annotation of the Mammalian Genome) consortium to identify promoter architecture in human and mouse [3], providing the first glimpse of the global landscape of transcription initiation. At the onset of the NGS era, CAGE was coupled with massively-parallel sequencing to generate 5'-ends of mRNAs at substantially higher scale. This advance provided more extensive coverage of the expressed transcriptome, and provided increased sensitivity for quantitative measurements *i.e.* measurement of promoter activity.

1.2   Promoter Architecture of *Drosophila melanogaster*

Hoskins and colleagues [4] performed CAGE in *D. melanogaster* as part of the modENCODE consortium, identifying promoters at large-scale and characterizing the promoter architecture of an insect genome for the first time. Hoskins [4] indicated that TSS distributions at *Drosophila* promoters exhibit a range of shapes that can be generally grouped into two major classifications: *peaked* and *broad*. Peaked promoters have a single, major TSS position occupying a narrow genomic region, whereas broad promoters lack a single, major TSS and contain TSSs across a wider region [5][6]. The authors also showed a strong association between promoter class and motif composition (consistent with previous findings [5, 7]). Peaked promoters were associated with positionally-enriched *cis*-regulatory motifs including TATA, Initiator (Inr) and DPE, while broad promoters contained an enrichment of less-well characterized motifs, including *Ohler6* and *Ohler7* [8]. The existence of two promoter classes appears to be conserved among metazoans, and has been reported (using TSS profiling methodolgies) in insects, cladocerans [9], fish [10] and mammals [11, 6].

1.3   Promoter Structure of Insects

Beyond *D. melanogaster*, few investigations have utilized TSS profiling in insect genomes. As a consequence, what is known about promoter architecture in insects is largely restricted to the *Drosophila* genus. As part of the modENCODE effort, CAGE was performed in multiple tissues and developmental stages of the *Drosophila pseudoobscura*. TSSs were found to be highly similar between species: more than 80% of TSSs (81%) of aligned, CAGE-identified TSSs from *D. pseudoobscura* were positioned within 20nt of their counterparts in *D. melanogaster*. An enrichment of the CA dinucleotide was detected at the TSS ([-1, +1]), and the motifs corresponding to TATA, Inr and DPE were positioned at the same locations relative to the TSS in both species. The one other insect species for which TSS profiling has been applied is the Tsetse fly (*Glossina morsitans morsitans*) [12]. Using TSS-seq (specifically Oligo-capping; for details on this method see [13]), the authors identified 3134 mapping to 1424 genes. The authors found

a preference for CA and AA dinucleotides at the TSS, and observe the major core promoter elements observed in *Drosophila*: TATA, Inr, DPE, in addition to MTE (Motif Ten Element). As in *D. melanogaster*, peaked promoters were more likely to contain TATA and Inr than broad promoters. While the taxonomic sampling of species for TSS profiling has been limited, the existing studies are sufficient to provide a general picture of insect promoter architecture. A major demarcation between the promoter architecture of insects and mammals appears to be the large fraction of mammalian promoters found in CpG islands [12]. CpG island promoters (CPIs) form the largest class of promoter in mammals [14]; by contrast, CPIs are not known to exist as a class in invertebrates.

1.4   Paired-end TSS Profiling with RAMPAGE

The most recent major methodological advance in TSS Profiling is RAMPAGE (RNA Annotation and Mapping of Promoters for the Analysis of Gene Expression) . RAMPAGE is a protocol for 5'-cDNA sequencing that combines cap trapping and template-switching with paired-end sequence information. A key advantage of generating paired-end sequence is transcript connectivity, which provides a direct link between a given 5'-end and its associated mRNA molecule. Because short or spurious RNAs are found within the transcriptome, transcript connectivity allows the TSSs (and thus promoters) of full-length mRNAs to be unambiguously identified, which benefits genome annotation. Batut and colleagues generated libraries from total RNA isolated from 36 stages across the life cycle of *D. melanogaster* providing a comprehensive gene expression and promoter atlas for fruit fly and in the process demonstrating the utility of RAMPAGE. RAMPAGE is currently being applied as part of the latest iteration of ENCODE to identify promoters in human, but as of this writing it has not been applied to any non-*Drosophila* insect species. In anticipation of the future application of TSS profiling into other insect model systems here we provide a documented protocol for the computational processing RAMPAGE data, using selected libraries from Batut *et al.*. This method will consist of two parts: first, we will process, filter and align the sequenced RAMPAGE libraries to the *D. melanogaster* genome. Second, we will identify TSSs and promoters from the aligned sequences and associate them with coding regions. In closing, we will consider further applications of this data and discuss the utility of reproducible workflows in bioinformatic analysis.

2   Materials

The analyses described herein require a workstation capable for modern bioinformatics. An intermediate understanding of the Linux/Unix command line will be extremely useful, although we make efforts to explain the procedures with clarity. In addition, it will likely be necessary for the participant to have superuser privileges on the machine. If you do not have a machine (or access to one) that meets

94 these requirements, it is recommended that you consider cloud-based cyberin-
95 frastructure, including Amazon Web Services (AWS; https://aws.amazon.com/)
96 or CyVerse (http://www.cyverse.org/). The former is a well-known pay-per-use
97 solution, while the latter is an NSF-funded resource that is made freely available
98 to the public.

2.1   Hardware Requirements

100 − x86-64 compatible processors
101 − At least 8GB RAM
102 − 30GB+ hard disk space

2.2   Software Requirements

104 − Operating system: 64 bit Linux (preferred) or Mac OS X (with Command
105   Line Tools from XCode)
106 − R (version 3.4)
107 − Bioconductor (version 3.5)
108 − FASTX-Toolit (version 0.0.13)
109 − Samtools (version 1.3 or above)
110 − SRA Toolkit (version 2.3.4-2 or above)
111 − STAR aligner (version 2.4 or above)
112 − TagDust (version 2.33)

2.3   Installation of R packages

114 For installation of the software listed above, please follow the instructions
115 provided by each respective package. Part of our analysis will require the
116 use of R packages found in the Bioconductor suite. To install Bioconductor,
117 please type the following from an R console:

```
source("https://bioconductor.org/biocLite.R")
biocLite()
```

120 We will use the R package *TSRchitect* to identify promoters from aligned
121 RAMAPGE libraries. First, we will need to install a series of prerequisite
122 packages to *TSRchitect* from Bioconductor. Please install these packages as
123 follows (as before, from an R console):

```
source("https://bioconductor.org/biocLite.R")
biocLite(c("AnnotationHub", "BiocGenerics", "BiocParallel",
  "ENCODExplorer",  "GenomicAlignments", "GenomeInfoDb",
  "GenomicRanges", "IRanges", "methods",
  "Rsamtools", "rtracklayer", "S4Vectors",
  "SummarizedExperiment"))
```

130 To install *TSRchitect*, please type the following from an R console:
131

```
132    source ("https://bioconductor.org/biocLite.R")
133    biocLite ("TSRchitect")
```

Finally, please confirm that TSRchitect has been installed correctly by loading it from your R console as follows:

```
137    library (TSRchitect)
```

3   Methods

3.1   Retrieving the RAMPAGE sequence data from NCBI's Gene Expression Omnibus (GEO)

To begin our analysis, we must download the RAMPAGE data to our workstation. We will utilize tools provided by the SRA Toolkit, which should already be installed on your machine (see **Materials**). The command *fastq-dump* allows one to directly retrieve data from the GEO database using the appropriate identifier(s). While there are 36 RAMPAGE libraries in the Batut *et al.* dataset, we will select a subset of these to analyze here. We will compare samples from selected embryonic (E01h-E03h) and larval (L1-L3) tissues, representing the beginning and end of embryonic development. For more information about the experiment and the available RAMPAGE libraries, please see the following link: https://www.ncbi.nlm.nih.gov/Traces/study/?acc=SRP011193 First, let's proceed with the libraries from early embryonic tissues. Note that since these fastq files are paired-end, we use the argument *–split-files* to generate separate files for each read pair.

```
154    mkdir fastq_files #creating a new folder to house the downloaded files
155    cd fastq_files #moving into this directory
156    fastq-dump --split-files SRR424683
157    fastq-dump --split-files SRR424684
158    fastq-dump --split-files SRR424685
```

We continue by downloading the RAMPAGE libraries from late embryonic tissues:

```
161    fastq-dump --split-files SRR424707
162    fastq-dump --split-files SRR424708
163    fastq-dump --split-files SRR424709
```

Once the download of the aforementioned files are complete, you should see a total of 12 (6x2) separate fastq files in your current working directory:

```
166    ls -l *.fastq | wc -l
```

167 ## 3.2 Creating symlinks to the files

168 Our workflow expects fastq files that have the format "*.R1/R2.clipped.fq".
169 Rather than rename them, we can simply create brand new symbolic links
170 to the files, as follows:

```
171  mkdir symlinks
172  ln −s SRR424683_1.fastq symlinks/E01h.R1.clipped.fq #embryonic libraries
173  ln −s SRR424683_2.fastq symlinks/E01h.R2.clipped.fq
174  ln −s SRR424684_1.fastq symlinks/E02h.R1.clipped.fq
175  ln −s SRR424684_2.fastq symlinks/E02h.R2.clipped.fq
176  ln −s SRR424685_1.fastq symlinks/E03h.R1.clipped.fq
177  ln −s SRR424685_2.fastq symlinks/E03h.R2.clipped.fq
178
179  ln −s SRR424707_1.fastq symlinks/L1.R1.clipped.fq #larval libraries
180  ln −s SRR424707_2.fastq symlinks/L1.R2.clipped.fq
181  ln −s SRR424708_1.fastq symlinks/L2.R1.clipped.fq
182  ln −s SRR424708_2.fastq symlinks/L2.R2.clipped.fq
183  ln −s SRR424709_1.fastq symlinks/L3.R1.clipped.fq
184  ln −s SRR424709_2.fastq symlinks/L3.R2.clipped.fq
```

185 ## 3.3 Downloading genomic data from *D. melanogaster*

186 Now that we have the fastq files from the RAMPAGE libraries downloaded
187 and named appropriately, we now must retrieve the genome assembly and
188 rRNA sequences from *D. melanogaster*. The genome assembly is required
189 for aligning the RAMPAGE reads, and the rRNA sequences are required to
190 filter out matching reads in the sequenced RAMPAGE libraries, since our
191 sample is intended to contain only capped RNA transcripts. Please download
192 the rRNA sequences from the link we provide below. These sequences were
193 retrieved separately from Genbank at the NCBI database.
194 Please download the assembly from the ENSEMBL database as follows:

```
195  wget ftp://ftp.ensembl.org/pub/release−78/fasta/drosophila_melanogaster/dna
196  #uncompressing the file
197  gzip −d Drosophila_melanogaster.BDGP5.dna.toplevel.fa.gz
```

198 The rRNA sequences are found at the following link: https://iu.box.com/s/3a5lqbo58qlykhmqxw00h2uo
199 You should see a file entitled "Dmel_rRNA.fasta" in your current directory.

```
200  head −n 3
201  >ref|NR_133562.1| Drosophila melanogaster 28S ribosomal RNA (28SrRNA:CR458
202  TTATATACAACCTCAACTCATATGGGACTACCCCCTGAATTTAAGCATATTAATTAGGGGAGGAAAAGAA
203  ACTAACAAGGATTTTCTTAGTAGCGGCGAGCGAAAAGAAAACAGTTCAGCACTAAGTCACTTTGTCTATA
```

3.4   Filtering and alignment of RAMPAGE reads using GoRAMPAGE

At this stage we are ready to commence with the rRNA filtering and alignment of the RAMPAGE libraries. We will use GoRAMPAGE, a tool we developed, to perform these tasks in a concerted workflow. GoRAMPAGE runs TagDust [15] to remove rRNA and low-complexity reads, and uses STAR [16] to align RAMPAGE (or other paired-end) reads to a given genome assembly.

Preparing the output directory It will also be necessary to create an output directory under "outputDir" for the results. GoRAMPAGE expects the results of a given step to be in place prior to initiating a run, so we'll need to create the appropriate folders before proceeding. Please do this as follows:

```
mkdir output #omit if you already have an output directory selected
mkdir output/reads
mkdir output/reads/clipped
```

Setting up the GoRAMPAGE job Now, once this is complete, please copy the contents of the "symlinks" directory that you created earlier (*i.e.* all of the *.fq files) into the "clipped/" directory. Please refer to the script "GoRAMPAGE_script_MMB.sh" and (using a text editor) provide the appropriate paths to the genome assembly, output directory (see above) and rRNA sequences. Note that if you are running this on a cluster with a job scheduler you'll need to add the necessary headers to the top of the script and submit the job in the appropriate manner. The script can be executed as follows:

```
./GoRAMPAGE_script_MMB.sh
#alternatively 'sh GoRAMPAGE_script_MMB.sh'
```

If everything is working correctly you should start to see the results of the job being written to the file "errScript". You can inspect the progress during the run using the *less* command.

```
less -S errScript
```

Should the run fail before completion, any associated error messages will be printed to the errScript file. Once the job is complete, you should see the message "GoRAMPAGE job is complete!" appear on the command-line terminal.

Inspecting the rRNA filtering results To evaluate the results from Step 3 (rRNA filtering), please navigate to the top level of the "output" directory and open the file "LOGFILES". You'll see the recorded progress of the program Tagdust and a record of the results. We notice that (for the L3h library)

242 1046448 of reads (78.1%) were "extracted", meaning that slightly more than
243 20% of reads were removed because of matches with ribosomal sequences.
244 The removed reads from all libraries are found in the `"dusted_discard"`
245 directory, and the extracted reads are found in the current directory. Due
246 to their sheer abundance within cells, ribosomal RNA sequences are an in-
247 evitable contaminant within TSS profiling libraries. For analysis purposes,
248 it is important that these sequences be removed, which is what has been
249 completed here.
250 Confident that this step was conducted successfully, we can proceed to the
251 next step.

252 **Evaluating the alignments** The folder "alignments/" in your GoRAMAPGE
253 output folder will now contain 6 .bam files, each representing the distinct
254 RAMAPGE libraries selected for our analysis. Typing "ls -l" from the com-
255 mand line will show that these files are symlinks to the original alignment
256 files found in the "STARoutput/" directory. "STARoutput/", as its name
257 suggests, contains the output from the STAR alignment, and this includes
258 the alignment files "*.sortedByCoord.out.bam", and four additional log files.
259 The files with the suffix "*.STAR.Log.final.out" each contain a summary of
260 the alignment, such as the number of input reads, the percentage of uniquely-
261 mapped reads and the percentage of unmapped reads. An inspection of these
262 log files indicates that the alignments have similar mapping rates (70-80%),
263 a reasonable outcome for our purposes.
264
265 Now that our RAMPAGE libraries are filtered and aligned, we can commence
266 with the second half of our analysis.

267 ### 3.5 Promoter identification from aligned RAMPAGE libraries

268 We can now use the prepared alignment files to identify TSSs and promot-
269 ers from the selected RAMPAGE libraries. There are currently several tools
270 available for this purpose. *CAGEr*, developed by Haberle [17], was utilized
271 to perform TSS identification as part of the FANTOM5 efforts. We will
272 use *TSRchitect* in this demonstration, since it was specifically designed to
273 analyze paired-end TSS profiling datasets, and also because it is more flex-
274 ible with respect to model system (*i.e.* it does not require a corresponding
275 *BSGenome* package). The latter feature will be helpful when analyzing the
276 non-*D. melanagaster* TSS profiling datasets that we expect to be generated
277 in the near future.

278 **Setting up the Analysis** *TSRchitect*, the package we'll use for this anal-
279 ysis, is an R package available in the Bioconductor suite of genomics tools
280 [18]. It makes use of existing packages and data structures within this envi-
281 ronment, where available, to identify promoters from sequence alignments.

²⁸² Since you have already installed *TSRchitect* and its dependencies (see sec-
²⁸³ tion 2.3), we are set to proceed.
²⁸⁴ There are two general ways one can choose to run *TSRchitect*. The first is in-
²⁸⁵ teractively *i.e.* typing the instructions directly into an R console. While this
²⁸⁶ is certainly an acceptable way to run this package, for larger jobs it will likely
²⁸⁷ be more efficient to run a dedicated R script. We have provided a sample
²⁸⁸ script `"MMB_chapter_TSRchitect.R"` to make it easier for you to set up an R
²⁸⁹ script. In the section to follow, we will go through the output of the analysis.
²⁹⁰ For further details on how to use *TSRchitect*, please see its documentation at
²⁹¹ its Bioconductor page here https://www.bioconductor.org/packages/release/bioc/html/TSRchitect.html
²⁹²
²⁹³ To run TSRchitect using the batch script provided, first provide full paths
²⁹⁴ for "BAMDIR" and "DmAnnot" in `"MMB_chapter_TSRchitect.R"` using a
²⁹⁵ text editor. *BAMDIR* should be a path to the subdirectory "alignments/"
²⁹⁶ in RAMPAGE output directory you specified earlier, and *DmAnnot* should
²⁹⁷ be a full path to the *D. melanogaster* gene annotation listed above. Once
²⁹⁸ this is complete, we can run the batch script from the Linux command-line
²⁹⁹ as follows:

³⁰⁰ R CMD BATCH MMB_chapter_TSRchitect.R #assumes variables BAMDIR and DmAnnot
³⁰¹ bg #puts the job in the background

³⁰² Once the job is underway, you can monitor its progress by looking at the
³⁰³ contents of the .Rout file (in this case, `"MMB_chapter_TSRchitect.Rout"`).

³⁰⁴ ## 4 Notes

³⁰⁵ **Acknowledgments**

³⁰⁶ **Disclosure Declaration**

³⁰⁷ The authors declare that they have no competing interests.

³⁰⁸ # 5 Figures

³⁰⁹ For LATEX users, we recommend using the *graphics* or *graphicx* package and the
³¹⁰ `\includegraphics` command.
³¹¹ Please check that the lines in line drawings are not interrupted and are of
³¹² a constant width. Grids and details within the figures must be clearly legible
³¹³ and may not be written one on top of the other. Line drawings should have
³¹⁴ a resolution of at least 800 dpi (preferably 1200 dpi). The lettering in figures
³¹⁵ should have a height of 2 mm (10-point type). Figures should be numbered and
³¹⁶ should have a caption which should always be positioned *under* the figures, in
³¹⁷ contrast to the caption belonging to a table, which should always appear *above*
³¹⁸ the table; this is simply achieved as matter of sequence in your source.
³¹⁹ Please center the figures or your tabular material by using the `\centering`
³²⁰ declaration. Short captions are centered by default between the margins and

321 typeset in 9-point type (Fig. 1 shows an example). The distance between text
322 and figure is preset to be about 8 mm, the distance between figure and caption
323 about 6 mm.

324   To ensure that the reproduction of your illustrations is of a reasonable quality,
325 we advise against the use of shading. The contrast should be as pronounced as
326 possible.

327   If screenshots are necessary, please make sure that you are happy with the
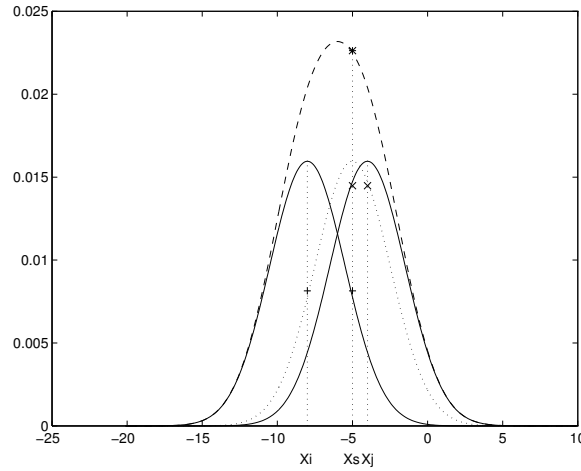print quality before you send the files.

Fig. 1. One kernel at x_s (*dotted kernel*) or two kernels at x_i and x_j (*left and right*)
lead to the same summed estimate at x_s. This shows a figure consisting of different
types of lines. Elements of the figure described in the caption should be set in italics,
in parentheses, as shown in this sample caption.

328
329   Please define figures (and tables) as floating objects. Please avoid using op-
330 tional location parameters like " [h] " for "here".

5.1   Formulas

332 Displayed equations or formulas are centered and set on a separate line (with an
333 extra line or halfline space above and below). Displayed expressions should be
334 numbered for reference. The numbers should be consecutive within each section
335 or within the contribution, with numbers enclosed in parentheses and set on the
336 right margin – which is the default if you use the *equation* environment, e.g.,

$$\psi(u) = \int_o^T \left[\frac{1}{2} \left(\Lambda_o^{-1} u, u \right) + N^*(-u) \right] dt \ .$$
(1)

337   Equations should be punctuated in the same way as ordinary text but with
338 a small space before the end punctuation mark.

5.2   Footnotes

The superscript numeral used to refer to a footnote appears in the text either directly after the word to be discussed or – in relation to a phrase or a sentence – following the punctuation sign (comma, semicolon, or period). Footnotes should appear at the bottom of the normal text area, with a line of about 2 cm set immediately above them.[1]

5.3   Program Code

Program listings or program commands in the text are normally set in typewriter font, e.g., CMTT10 or Courier.

Example of a Computer Program

```
program Inflation (Output)
   {Assuming annual inflation rates of 7%, 8%, and 10%,...
    years};
    const
      MaxYears = 10;
    var
      Year: 0..MaxYears;
      Factor1, Factor2, Factor3: Real;
    begin
      Year := 0;
      Factor1 := 1.0; Factor2 := 1.0; Factor3 := 1.0;
      WriteLn('Year  7% 8% 10%'); WriteLn;
      repeat
        Year := Year + 1;
        Factor1 := Factor1 * 1.07;
        Factor2 := Factor2 * 1.08;
        Factor3 := Factor3 * 1.10;
        WriteLn(Year:5,Factor1:7:3,Factor2:7:3,Factor3:7:3)
      until Year = MaxYears
end.
```

(Example from Jensen K., Wirth N. (1991) Pascal user manual and report. Springer, New York)

5.4   Citations

For citations in the text please use square brackets and consecutive numbers: [?], [?], [?] – provided automatically by LaTeX's \cite ... \bibitem mechanism.

6   References

---

[1] The footnote numeral is set flush left and the text follows with the usual word spacing.

References

1. J. T. Kadonaga, "Perspectives on the RNA polymerase II core promoter." *Wiley Interdisciplinary Reviews: Developmental Biology*, vol. 1, no. 1, pp. 40–51, Jan. 2012.

2. R. Kodzius, M. Kojima, H. Nishiyori, M. Nakamura, S. Fukuda, M. Tagami, D. Sasaki, K. Imamura, C. Kai, M. Harbers, Y. Hayashizaki, and P. Carninci, "CAGE: cap analysis of gene expression." *Nature Methods*, vol. 3, no. 3, pp. 211–222, Mar. 2006.

3. P. Carninci, T. Kasukawa, S. Katayama, J. Gough, M. C. Frith, N. Maeda, R. Oyama, T. Ravasi, B. Lenhard, C. Wells, R. Kodzius, K. Shimokawa, V. B. Bajic, S. E. Brenner, S. Batalov, A. R. R. Forrest, M. Zavolan, M. J. Davis, L. G. Wilming, V. Aidinis, J. E. Allen, A. Ambesi-Impiombato, R. Apweiler, R. N. Aturaliya, T. L. Bailey, M. Bansal, L. Baxter, K. W. Beisel, T. Bersano, H. Bono, A. M. Chalk, K. P. Chiu, V. Choudhary, A. Christoffels, D. R. Clutterbuck, M. L. Crowe, E. Dalla, B. P. Dalrymple, B. de Bono, G. Della Gatta, D. di Bernardo, T. Down, P. Engstrom, M. Fagiolini, G. Faulkner, C. F. Fletcher, T. Fukushima, M. Furuno, S. Futaki, M. Gariboldi, P. Georgii-Hemming, T. R. Gingeras, T. Gojobori, R. E. Green, S. Gustincich, M. Harbers, Y. Hayashi, T. K. Hensch, N. Hirokawa, D. Hill, L. Huminiecki, M. Iacono, K. Ikeo, A. Iwama, T. Ishikawa, M. Jakt, A. Kanapin, M. Katoh, Y. Kawasawa, J. Kelso, H. Kitamura, H. Kitano, G. Kollias, S. P. T. Krishnan, A. Kruger, S. K. Kummerfeld, I. V. Kurochkin, L. F. Lareau, D. Lazarevic, L. Lipovich, J. Liu, S. Liuni, S. McWilliam, M. Madan Babu, M. Madera, L. Marchionni, H. Matsuda, S. Matsuzawa, H. Miki, F. Mignone, S. Miyake, K. Morris, S. Mottagui-Tabar, N. Mulder, N. Nakano, H. Nakauchi, P. Ng, R. Nilsson, S. Nishiguchi, S. Nishikawa, F. Nori, O. Ohara, Y. Okazaki, V. Orlando, K. C. Pang, W. J. Pavan, G. Pavesi, G. Pesole, N. Petrovsky, S. Piazza, J. Reed, J. F. Reid, B. Z. Ring, M. Ringwald, B. Rost, Y. Ruan, S. L. Salzberg, A. Sandelin, C. Schneider, C. Schönbach, K. Sekiguchi, C. A. M. Semple, S. Seno, L. Sessa, Y. Sheng, Y. Shibata, H. Shimada, K. Shimada, D. Silva, B. Sinclair, S. Sperling, E. Stupka, K. Sugiura, R. Sultana, Y. Takenaka, K. Taki, K. Tammoja, S. L. Tan, S. Tang, M. S. Taylor, J. Tegner, S. A. Teichmann, H. R. Ueda, E. van Nimwegen, R. Verardo, C. L. Wei, K. Yagi, H. Yamanishi, E. Zabarovsky, S. Zhu, A. Zimmer, W. Hide, C. Bult, S. M. Grimmond, R. D. Teasdale, E. T. Liu, V. Brusic, J. Quackenbush, C. Wahlestedt, J. S. Mattick, D. A. Hume, C. Kai, D. Sasaki, Y. Tomaru, S. Fukuda, M. Kanamori-Katayama, M. Suzuki, J. Aoki, T. Arakawa, J. Iida, K. Imamura, M. Itoh, T. Kato, H. Kawaji, N. Kawagashira, T. Kawashima, M. Kojima, S. Kondo, H. Konno, K. Nakano, N. Ninomiya, T. Nishio, M. Okada, C. Plessy, K. Shibata, T. Shiraki, S. Suzuki, M. Tagami, K. Waki, A. Watahiki, Y. Okamura-Oho, H. Suzuki, J. Kawai, Y. Hayashizaki, F. Consortium, R. G. E. R. Group, and G. S. G. G. N. P. C. Group, "The transcriptional landscape of the mammalian genome," *Science (New York, NY)*, vol. 309, no. 5740, pp. 1559–1563, Sep. 2005.

4. R. A. Hoskins, R. A. Hoskins, J. M. Landolin, J. M. Landolin, J. B. Brown, J. B. Brown, J. E. Sandler, J. E. Sandler, H. Takahashi, H. Takahashi, T. Lassmann, T. Lassmann, C. Yu, C. Yu, B. W. Booth, B. W. Booth, D. Zhang, D. Zhang, K. H. Wan, K. H. Wan, L. Yang, L. Yang, N. Boley, N. Boley, J. Andrews, J. Andrews, T. C. Kaufman, T. C. Kaufman, B. R. Graveley, B. R. Graveley, P. J. Bickel, P. J. Bickel, P. Carninci, J. W. Carlson, J. W. Carlson, S. E. Celniker,

and S. E. Celniker, "Genome-wide analysis of promoter architecture in Drosophila melanogaster." *Genome Research*, vol. 21, no. 2, pp. 182–192, Feb. 2011.

5. E. A. Rach, H.-Y. Yuan, W. H. Majoros, P. Tomancak, and U. Ohler, "Motif composition, conservation and condition-specificity of single and alternative transcription start sites in the Drosophila genome." *Genome Biology*, vol. 10, no. 7, p. R73, 2009.

6. B. Lenhard, A. Sandelin, and P. Carninci, "Metazoan promoters: emerging characteristics and insights into transcriptional regulation." *Nature Reviews Genetics*, vol. 13, no. 4, pp. 233–245, Apr. 2012.

7. T. Ni, D. L. Corcoran, E. A. Rach, S. Song, E. P. Spana, Y. Gao, U. Ohler, and J. Zhu, "A paired-end sequencing strategy to map the complex landscape of transcription initiation." *Nature Methods*, vol. 7, no. 7, pp. 521–527, Jul. 2010.

8. U. Ohler, G.-c. Liao, H. Niemann, and G. M. Rubin, "Computational analysis of core promoters in the Drosophila genome." *Genome Biology*, vol. 3, no. 12, pp. research0087.1–0087.12, 2002.

9. R. T. Raborn, K. Spitze, V. P. Brendel, and M. Lynch, "Promoter Architecture and Sex-Specific Gene Expression in Daphnia pulex." *Genetics*, vol. 204, no. 2, pp. 593–612, Aug. 2016.

10. C. Nepal, Y. Hadzhiev, C. Previti, V. Haberle, N. Li, H. Takahashi, A. M. M. Suzuki, Y. Sheng, R. F. Abdelhamid, S. Anand, J. Gehrig, A. Akalin, C. E. M. Kockx, A. A. J. van der Sloot, W. F. J. van IJcken, O. Armant, S. Rastegar, C. Watson, U. Strahle, E. Stupka, P. Carninci, B. Lenhard, and F. Muller, "Dynamic regulation of the transcription initiation landscape at single nucleotide resolution during vertebrate embryogenesis," *Genome Research*, vol. 23, no. 11, pp. 1938–1950, Nov. 2013.

11. P. Carninci, A. Sandelin, B. Lenhard, S. Katayama, K. Shimokawa, J. Ponjavic, C. A. M. Semple, M. S. Taylor, P. G. Engström, M. C. Frith, A. R. R. Forrest, W. B. Alkema, S. L. Tan, C. Plessy, R. Kodzius, T. Ravasi, T. Kasukawa, S. Fukuda, M. Kanamori-Katayama, Y. Kitazume, H. Kawaji, C. Kai, M. Nakamura, H. Konno, K. Nakano, S. Mottagui-Tabar, P. Arner, A. Chesi, S. Gustincich, F. Persichetti, H. Suzuki, S. M. Grimmond, C. A. Wells, V. Orlando, C. Wahlestedt, E. T. Liu, M. Harbers, J. Kawai, V. B. Bajic, D. A. Hume, and Y. Hayashizaki, "Genome-wide analysis of mammalian promoter architecture and evolution," *Nature Genetics*, vol. 38, no. 6, pp. 626–635, Apr. 2006.

12. S. Mwangi, G. Attardo, Y. Suzuki, S. Aksoy, and A. Christoffels, "TSS seq based core promoter architecture in blood feeding Tsetse fly (Glossina morsitans morsitans) vector of Trypanosomiasis," *BMC Genomics*, vol. 16, no. 1, p. 722, Sep. 2015.

13. K. Tsuchihara, Y. Suzuki, H. Wakaguri, T. Irie, K. Tanimoto, S.-i. Hashimoto, K. Matsushima, J. Mizushima-Sugano, R. Yamashita, K. Nakai, D. Bentley, H. Esumi, and S. Sugano, "Massive transcriptional start site analysis of human genes in hypoxia cells," *Nucleic Acids Research*, vol. 37, no. 7, pp. 2249–2263, Apr. 2009.

14. N. Cvetesic and B. Lenhard, "Core promoters across the genome," *Nature Biotechnology*, vol. 35, no. 2, pp. 123–124, Feb. 2017.

15. T. Lassmann, "TagDust2: a generic method to extract reads from sequencing data," *BMC Bioinformatics*, vol. 16, no. 1, p. 1, Jan. 2015.

16. A. Dobin and T. R. Gingeras, "Optimizing RNA-Seq Mapping with STAR," in *Transcription Factor Regulatory Networks*. New York, NY: Springer New York, Apr. 2016, pp. 245–262.

17. V. Haberle, "CAGEr: an R package for CAGE (Cap Analysis of Gene Expression) data analysis and promoterome mining," 2013.

474   18. M. Lawrence and M. Morgan, "Scalable Genomics with R and Bioconductor,"
475   *Statistical Science*, vol. 29, no. 2, pp. 214–226, May 2014.

7   Checklist of Items to be Sent to Volume Editors

Here is a checklist of everything the volume editor requires from you:

☐ The final LaTeX source files

☐ A final PDF file

☐ A copyright form, signed by one author on behalf of all of the authors of the paper.

☐ A readme giving the name and email address of the corresponding author.