

# Using RAMPAGE to identify and annotate regulatory elements in insect genomes

R. Taylor Raborn<sup>\*1,2</sup> and Volker P. Brendel<sup>1,2</sup>

<sup>1</sup>Department of Biology, Indiana University

<sup>2</sup>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](mailto:rtraborn@indiana.edu)

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

## 24 1.2 Promoter Architecture of *Drosophila melanogaster*

25 Hoskins and colleagues [4] performed CAGE in *D. melanogaster* as part of the  
 26 modENCODE consortium, identifying promoters at large-scale and character-  
 27 izing the promoter architecture of an insect genome for the first time. Hoskins  
 28 [4] indicated that TSS distributions at *Drosophila* promoters exhibit a range  
 29 of shapes that can be generally grouped into two major classifications: *peaked*  
 30 and *broad*. Peaked promoters have a single, major TSS position occupying a  
 31 narrow genomic region, whereas broad promoters lack a single, major TSS and  
 32 contain TSSs across a wider region [5][6]. The authors also showed a strong asso-  
 33 ciation between promoter class and motif composition (consistent with previous  
 34 findings [5, 7]). Peaked promoters were associated with positionally-enriched *cis*-  
 35 regulatory motifs including TATA, Initiator (Inr) and DPE, while broad promot-  
 36 ers contained an enrichment of less-well characterized motifs, including *Ohler6*  
 37 and *Ohler7* [8]. The existence of two promoter classes appears to be conserved  
 38 among metazoans, and has been reported (using TSS profiling methodologies) in  
 39 insects, cladocerans [9], fish [10] and mammals [11, 6].

## 40 1.3 Promoter Structure of Insects

41 Beyond *D. melanogaster*, few investigations have utilized TSS profiling in insect  
 42 genomes. As a consequence, what is known about promoter architecture in in-  
 43 sects is largely restricted to the *Drosophila* genus. As part of the modENCODE  
 44 effort, CAGE was performed in multiple tissues and developmental stages of the  
 45 *Drosophila pseudoobscura*. TSSs were found to be highly similar between species:  
 46 more than 80% of TSSs (81%) of aligned, CAGE-identified TSSs from *D. pseu-*  
 47 *doobscura* were positioned within 20nt of their counterparts in *D. melanogaster*.  
 48 An enrichment of the CA dinucleotide was detected at the TSS ([-1, +1]), and  
 49 the motifs corresponding to TATA, Inr and DPE were positioned at the same  
 50 locations relative to the TSS in both species. The one other insect species for  
 51 which TSS profiling has been applied is the Tsetse fly (*Glossina morsitans morsi-*  
 52 *tans*) [12]. Using TSS-seq (specifically Oligo-capping; for details on this method  
 53 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

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

## 2.1 Hardware Requirements

- x86-64 compatible processors
- At least 8GB RAM
- 30GB+ hard disk space

## 2.2 Software Requirements

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

## 2.3 Installation of R packages

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

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

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

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

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

```

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

```

```

134 Finally, please confirm that TSRchitect has been installed correctly by load-
135 ing it from your R console as follows:

```

```

136
137 library(TSRchitect)

```

### 138 3 Methods

#### 139 3.1 Retrieving the RAMPAGE sequence data from NCBI's 140 Gene Expression Omnibus (GEO)

```

141 To begin our analysis, we must download the RAMPAGE data to our work-
142 station. We will utilize tools provided by the SRA Toolkit, which should
143 already be installed on your machine (see Materials). The command fastq-
144 dump allows one to directly retrieve data from the GEO database using
145 the appropriate identifier(s). While there are 36 RAMPAGE libraries in the
146 Batut et al. dataset, we will select a subset of these to analyze here. We
147 will compare samples from selected embryonic (E01h-E03h) and larval (L1-
148 L3) tissues, representing the beginning and end of embryonic development.
149 For more information about the experiment and the available RAMPAGE li-
150 braries, please see the following link: https://www.ncbi.nlm.nih.gov/Traces/study/?acc=SRP011193
151 First, let's proceed with the libraries from early embryonic tissues. Note
152 that since these fastq files are paired-end, we use the argument -split-files
153 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

```

```

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

```

```

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

```

```

164 Once the download of the aforementioned files are complete, you should see
165 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/dm
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 TTATATACAACCTCAACTCATATGGGACTACCCCTGAATTTAAGCATATTAATTAGGGGAGGAAAAGAA
203 ACTAACAAGGATTTTCTTAGTAGCGGCGAGCGAAAAAGAAAACAGTTCAGCACTAAGTCACTTTGTCTATA
```

### 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:

```
\texttt{./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)

1046448 of reads (78.1%) were "extracted", meaning that slightly more than 20% of reads were removed because of matches with ribosomal sequences. The removed reads from all libraries are found in the "dusted\_discard" directory, and the extracted reads are found in the current directory. Due to their sheer abundance within cells, ribosomal RNA sequences are an inevitable contaminant within TSS profiling libraries. For analysis purposes, it is important that these sequences be removed, which is what we have done here. Confident that this step was conducted successfully, we can proceed to the next step.

**Evaluating the alignments** The folder "alignments/" in your GoRAMAPGE output folder will now contain 6 .bam files, each representing the distinct RAMAPGE libraries selected for our analysis. Typing "ls -l" from the command line will show that these files are symlinks to the original alignment files found in the "STARoutput/" directory. "STARoutput/", as its name suggests, contains the output from the STAR alignment, and this includes the alignment files "\*.sortedByCoord.out.bam", and four additional log files. The files with the suffix "\*.STAR.Log.final.out" each contain a summary of the alignment, such as the number of input reads, the percentage of uniquely-mapped reads and the percentage of unmapped reads. An inspection of these log files indicates that the alignments have similar mapping rates ( 70-80%), a reasonable outcome for our purposes.

Now that our RAMAPGE libraries are filtered and aligned, we can commence with the second half of our analysis, which is identifying TSSs and promoters from RAMAPGE libraries.

### 3.5 Promoter identification from aligned RAMAPGE libraries

## 4 Notes

### Acknowledgments

### Disclosure Declaration

The authors declare that they have no competing interests.

## 5 Figures

For L<sup>A</sup>T<sub>E</sub>X users, we recommend using the *graphics* or *graphicx* package and the `\includegraphics` command.

Please check that the lines in line drawings are not interrupted and are of a constant width. Grids and details within the figures must be clearly legible and may not be written one on top of the other. Line drawings should have

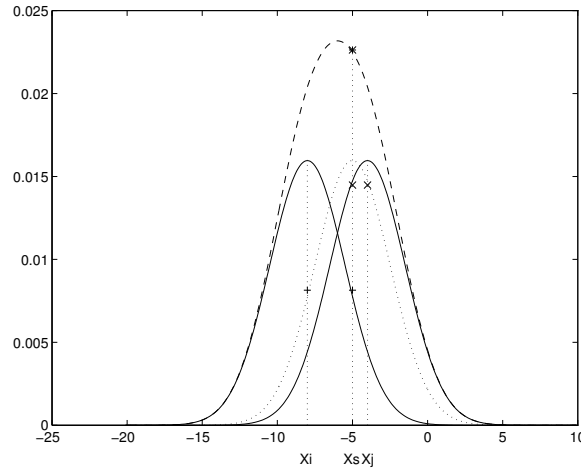


279 a resolution of at least 800 dpi (preferably 1200 dpi). The lettering in figures  
 280 should have a height of 2 mm (10-point type). Figures should be numbered and  
 281 should have a caption which should always be positioned *under* the figures, in  
 282 contrast to the caption belonging to a table, which should always appear *above*  
 283 the table; this is simply achieved as matter of sequence in your source.

284 Please center the figures or your tabular material by using the `\centering`  
 285 declaration. Short captions are centered by default between the margins and  
 286 typeset in 9-point type (Fig. 1 shows an example). The distance between text  
 287 and figure is preset to be about 8 mm, the distance between figure and caption  
 288 about 6 mm.

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

292 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.

293 Please define figures (and tables) as floating objects. Please avoid using op-  
 294 tional location parameters like “[h]” for “here”.  
 295

## 296 5.1 Formulas

297 Displayed equations or formulas are centered and set on a separate line (with an  
 298 extra line or halfline space above and below). Displayed expressions should be  
 299 numbered for reference. The numbers should be consecutive within each section

or within the contribution, with numbers enclosed in parentheses and set on the right margin – which is the default if you use the *equation* environment, e.g.,

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

Equations should be punctuated in the same way as ordinary text but with 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.<sup>1</sup>

## 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)

---

<sup>1</sup> The footnote numeral is set flush left and the text follows with the usual word spacing.

## 336 5.4 Citations

337 For citations in the text please use square brackets and consecutive numbers:  
338 [?], [?], [?] – provided automatically by L<sup>A</sup>T<sub>E</sub>X's `\cite ... \bibitem` mechanism.

## 339 5.5 Page Numbering and Running Heads

340 There is no need to include page numbers. If your paper title is too long to serve  
341 as a running head, it will be shortened. Your suggestion as to how to shorten it  
342 would be most welcome.

## 343 6 References

## 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-Impimbato, 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. Kawasaki, 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. 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,

- 392 and S. E. Celniker, "Genome-wide analysis of promoter architecture in *Drosophila*  
393 *melanogaster*." *Genome Research*, vol. 21, no. 2, pp. 182–192, Feb. 2011.
- 394 5. E. A. Rach, H.-Y. Yuan, W. H. Majoros, P. Tomancak, and U. Ohler, "Motif  
395 composition, conservation and condition-specificity of single and alternative tran-  
396 scription start sites in the *Drosophila* genome." *Genome Biology*, vol. 10, no. 7, p.  
397 R73, 2009.
- 398 6. B. Lenhard, A. Sandelin, and P. Carninci, "Metazoan promoters: emerging char-  
399 acteristics and insights into transcriptional regulation." *Nature Reviews Genetics*,  
400 vol. 13, no. 4, pp. 233–245, Apr. 2012.
- 401 7. T. Ni, D. L. Corcoran, E. A. Rach, S. Song, E. P. Spana, Y. Gao, U. Ohler,  
402 and J. Zhu, "A paired-end sequencing strategy to map the complex landscape of  
403 transcription initiation." *Nature Methods*, vol. 7, no. 7, pp. 521–527, Jul. 2010.
- 404 8. U. Ohler, G.-c. Liao, H. Niemann, and G. M. Rubin, "Computational analysis of  
405 core promoters in the *Drosophila* genome." *Genome Biology*, vol. 3, no. 12, pp.  
406 research0087.1–0087.12, 2002.
- 407 9. R. T. Raborn, K. Spitze, V. P. Brendel, and M. Lynch, "Promoter Architecture  
408 and Sex-Specific Gene Expression in *Daphnia pulex*." *Genetics*, vol. 204, no. 2, pp.  
409 593–612, Aug. 2016.
- 410 10. C. Nepal, Y. Hadzhiev, C. Previti, V. Haberle, N. Li, H. Takahashi, A. M. M.  
411 Suzuki, Y. Sheng, R. F. Abdelhamid, S. Anand, J. Gehrig, A. Akalin, C. E. M.  
412 Kockx, A. A. J. van der Sloot, W. F. J. van IJcken, O. Armant, S. Rastegar,  
413 C. Watson, U. Strahle, E. Stupka, P. Carninci, B. Lenhard, and F. Muller, "Dy-  
414 namic regulation of the transcription initiation landscape at single nucleotide res-  
415 olution during vertebrate embryogenesis," *Genome Research*, vol. 23, no. 11, pp.  
416 1938–1950, Nov. 2013.
- 417 11. P. Carninci, A. Sandelin, B. Lenhard, S. Katayama, K. Shimokawa, J. Ponjavic,  
418 C. A. M. Semple, M. S. Taylor, P. G. Engström, M. C. Frith, A. R. R. For-  
419 rest, W. B. Alkema, S. L. Tan, C. Plessy, R. Kodzius, T. Ravasi, T. Kasukawa,  
420 S. Fukuda, M. Kanamori-Katayama, Y. Kitazume, H. Kawaji, C. Kai, M. Naka-  
421 mura, H. Konno, K. Nakano, S. Mottagui-Tabar, P. Arner, A. Chesi, S. Gustincich,  
422 F. Persichetti, H. Suzuki, S. M. Grimmond, C. A. Wells, V. Orlando, C. Wahlest-  
423 edt, E. T. Liu, M. Harbers, J. Kawai, V. B. Bajic, D. A. Hume, and Y. Hayashizaki,  
424 "Genome-wide analysis of mammalian promoter architecture and evolution," *Na-  
425 ture Genetics*, vol. 38, no. 6, pp. 626–635, Apr. 2006.
- 426 12. S. Mwangi, G. Attardo, Y. Suzuki, S. Aksoy, and A. Christoffels, "TSS seq based  
427 core promoter architecture in blood feeding Tsetse fly (*Glossina morsitans mor-  
428 sitans*) vector of Trypanosomiasis," *BMC Genomics*, vol. 16, no. 1, p. 722, Sep.  
429 2015.
- 430 13. K. Tsuchihara, Y. Suzuki, H. Wakaguri, T. Irie, K. Tanimoto, S.-i. Hashimoto,  
431 K. Matsushima, J. Mizushima-Sugano, R. Yamashita, K. Nakai, D. Bentley, H. Es-  
432 umi, and S. Sugano, "Massive transcriptional start site analysis of human genes in  
433 hypoxia cells," *Nucleic Acids Research*, vol. 37, no. 7, pp. 2249–2263, Apr. 2009.
- 434 14. N. Cvetesic and B. Lenhard, "Core promoters across the genome," *Nature Biotech-  
435 nology*, vol. 35, no. 2, pp. 123–124, Feb. 2017.
- 436 15. T. Lassmann, "TagDust2: a generic method to extract reads from sequencing data,"  
437 *BMC Bioinformatics*, vol. 16, no. 1, p. 1, Jan. 2015.
- 438 16. A. Dobin and T. R. Gingeras, "Optimizing RNA-Seq Mapping with STAR," in  
439 *Transcription Factor Regulatory Networks*. New York, NY: Springer New York,  
440 Apr. 2016, pp. 245–262.

441 In order to permit cross referencing within LNCS-Online, and eventually  
 442 between different publishers and their online databases, LNCS will, from now  
 443 on, be standardizing the format of the references. This new feature will increase  
 444 the visibility of publications and facilitate academic research considerably. Please  
 445 base your references on the examples below. References that don't adhere to this  
 446 style will be reformatted by Springer. You should therefore check your references  
 447 thoroughly when you receive the final pdf of your paper. The reference section  
 448 must be complete. You may not omit references. Instructions as to where to find  
 449 a fuller version of the references are not permissible.

450 The following section shows a sample reference list with entries for journal  
 451 articles [?], an LNCS chapter [?], a book [?], proceedings without editors [?] and  
 452 [?], as well as a URL [?]. Please note that proceedings published in LNCS are  
 453 not cited with their full titles, but with their acronyms!

## 454 7 Checklist of Items to be Sent to Volume Editors

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

- 456 ☐ The final L<sup>A</sup>T<sub>E</sub>X source files
- 457 ☐ A final PDF file
- 458 ☐ A copyright form, signed by one author on behalf of all of the authors of the  
 459 paper.
- 460 ☐ A readme giving the name and email address of the corresponding author.