

Using RAMPAGE to identify and annotate promoters in insect genomes

R. Taylor Raborn^{*1} and Volker P. Brendel^{1,2}

¹Department of Biology, Indiana University

²School of Informatics and Computing, Indiana University

Department of Biology

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 *Drosophila melanogaster* [1] and subsequently was employed to reveal widespread transposon-driven alternative promoter usage in the fruit fly [2].

In this chapter we discuss the computational analysis of the experimental data derived from one TSS profiling method, RAMPAGE (RNA Annotation and Mapping of Promoters for Analysis of Gene Expression), that can be used for the precise, quantitative identification of promoters in insect genomes. We demonstrate this using the software tools GoRAMPAGE [3] and TSRchitect [4], providing detailed 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, which is defined in eukaryotes as the genomic region bound by RNA Polymerase II immediately prior to transcription initiation [5], is the primary locus of the regulation of gene expression. The identification of promoter regions is necessary for understanding the *cis*-regulatory signals controlling gene expression in an organism, and is also 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—absent empirically-defined information—precisely identifying

* Correspondence: rtraborn@indiana.edu

sequence motifs that demarcate the promoter is unreliable. In contrast with current *in silico* approaches, direct mapping of TSSs identifies the location of the core promoter. Cap Analysis of Gene Expression (CAGE) [6], 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 [7], providing the first glimpse of the global landscape of transcription initiation. At the onset of the next-generation sequencing (NGS) era, CAGE was coupled with massively-parallel sequencing to define 5'-mRNA ends at large scale. This advance provided more extensive coverage of the expressed transcriptome and provided increased sensitivity for quantitative measurements of promoter activity.

1.2 Promoter Architecture of *Drosophila melanogaster*

Hoskins and colleagues [1] 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. The authors found that TSS distributions at *Drosophila* promoters exhibit a range of shapes that can be generally grouped into two major classes: *peaked* and *broad*. This confirmed the original finding of Rach and colleagues [8], which was done using publicly-available expressed sequence tags (ESTs). 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 [8, 9]. The authors also showed a strong association between promoter class and motif composition (consistent with previous findings [8, 10]). Peaked promoters were associated with positionally-enriched *cis*-regulatory motifs including TATA, Initiator (Inr) and DPE (Downstream Promoter Element), while broad promoters contained an enrichment of less-well characterized motifs, including *Ohler6* and *Ohler7* [11]. The existence of at least two promoter classes appears to be conserved among metazoans and has been reported (using TSS profiling methods) in insects, cladocerans [12], fish [13] and mammals [14, 9].

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: 81% of TSSs 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 only other insect species for which TSS profiling has been applied is the Tsetse fly (*Glossina morsitans morsitans*) [15]. Using TSS-seq (specifically Oligo-capping; for details see [16]), the authors identified 3134 promoters associated with 1424 genes. The authors found a preference for CA and AA dinucleotides at the TSSs 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 [15]. CpG island promoters (CPIs) form the largest class of promoter in mammals [17]; 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 [2, 18], a protocol for 5'-cDNA sequencing that combines cap trapping and template-switching with paired-end sequence information (see Figure 1). As with CAGE and other TSS profiling methods, RAMPAGE reads are aligned, to obtain TSSs and clustered to identify Transcription Start Regions (TSRs), which are enrichments of TSSs consistent with promoters (Figure 2A). 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 [2] (Figure 2B). 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 and improves interpretation of transcript species.

Batut and colleagues [2] generated libraries from total RNA isolated from 36 stages across the life cycle of *D. melanogaster*, generating a comprehensive gene expression and promoter atlas for fruit fly and 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 model system.

In anticipation of the future application of TSS profiling into other insect model systems, we discuss in this chapter a well-documented protocol for the computational processing and analysis of RAMPAGE data, using selected libraries from Batut *et al.* [2]. This method consists of two parts: first, we discuss how to process, filter and align the sequenced RAMPAGE libraries to the *D. melanogaster* genome. Second, we show how to 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 of doing modern bioinformatics; minimally a reasonably-appointed laptop. 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 have 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/>), CyVerse (<http://www.cyverse.org/>) [19], or JetStream (<https://jetstream-cloud.org/>) [20]. The former is a well-known pay-per-use solution, while the latter two are NSF-funded resources that makes compute allocations freely available to the public.

2.1 Hardware

1. x86-64 compatible processors
2. At least 8GB RAM
3. 80GB+ hard disk space

2.2 Operating System

- 64 bit Linux (preferred) or Mac OS X (with Command Line Tools from XCode)

2.3 Software

Below is a list of the software packages required for this demonstration (*see Note 1*).

Sequence retrieval

1. SRA Toolkit [21] (<https://www.ncbi.nlm.nih.gov/sra/docs/toolkitsoft/>)

GoRAMPAGE

1. GoRAMPAGE [3] (<https://github.com/brendelGroup/GoRAMPAGE>)
2. fastq-multx [22] (<https://github.com/brwnj/fastq-multx>)
3. FASTX-Toolkit [23] (http://hannonlab.cshl.edu/fastx_toolkit/Index.html)
4. TagDust2 [24] (<https://sourceforge.net/projects/tagdust/>)
5. Samtools [25] (<http://www.htslib.org/doc/samtools.html>)

131 6. STAR [26] (<https://github.com/alexdobin/STAR>)

132 **TSRchitect**

- 133 1. R (v. 3.4 and up) [27] (<https://www.r-project.org/>)
- 134 2. Bioconductor (v. 3.5 and up) [28] (<http://bioconductor.org/>)
- 135 3. TSRchitect [4] (<http://bioconductor.org/packages/release/bioc/html/TSRchitect.html>)
- 136 4. Various R package dependencies (see **Methods**)

137 **2.4 Demonstration**

138 We created an online demonstration (demo) to serve as a companion to this
 139 chapter, which contains both scripts and select files to assist you in completing
 140 this tutorial. Please find the repository here:
 141 <https://github.com/brendelgroup/GoRAMPAGE/demo/MMB> (see **Note 2**).

142 **2.5 Installation of R packages**

143 For installation of the software listed above, please follow the instructions pro-
 144 vided by each respective package. Part of our analysis will require the use of R
 145 packages found in the Bioconductor suite [28] (see **Note 3**). To install Biocon-
 146 ductor, please type the following from an R console:

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

149 We will use the R package *TSRchitect* to identify promoters from aligned RAM-
 150 PAGE libraries. Prior to running the analysis, it will be necessary to install a
 151 series of prerequisite packages to *TSRchitect* from Bioconductor. Please install
 152 these packages, followed by *TSRchitect* (as before, from an R console):

```
153 source("https://bioconductor.org/biocLite.R")
154 biocLite(c("AnnotationHub", "BiocGenerics", "BiocParallel",
155 "ENCODEExplorer", "GenomicAlignments", "GenomeInfoDb",
156 "GenomicRanges", "IRanges", "methods",
157 "Rsamtools", "rtracklayer", "S4Vectors",
158 "SummarizedExperiment"))
159
160 biocLite("TSRchitect")
```

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

```
163 library(TSRchitect) #loading TSRchitect
```

164 3 Methods

165 3.1 Retrieving the RAMPAGE sequence data from NCBI

166 To begin our analysis, we must download the RAMPAGE data to our worksta-
 167 tion. We will utilize tools provided by the SRA Toolkit, which should already
 168 be installed on your machine (see **Materials**). The command *fastq-dump* al-
 169 lows one to directly retrieve data from the GEO database using the appropriate
 170 identifier(s). While there are 36 RAMPAGE libraries in the Batut *et al.* pa-
 171 per, we will select a subset of these to analyze here. We will compare samples
 172 from selected embryonic (E01h-E03h) and larval (L1-L3) tissues, representing
 173 the beginning and end of embryonic development. For more information about
 174 the experiment and the available RAMPAGE libraries, please see the following
 175 link: <https://www.ncbi.nlm.nih.gov/Traces/study/?acc=SRP011193>.

176
 177 First, let's proceed with downloading the libraries from early embryonic tissues
 178 (see **See Note 4**). We will make a new folder (entitled "**fastq_files/**") to
 179 house these files.

```
180 mkdir fastq_files
181 cd fastq_files
182
183 fastq-dump --split-files SRR424683
184 fastq-dump --split-files SRR424684
185 fastq-dump --split-files SRR424685
```

186 We continue by downloading the data from late larval tissues.

```
187 fastq-dump --split-files SRR424707
188 fastq-dump --split-files SRR424708
189 fastq-dump --split-files SRR424709
```

190 Once the download of the aforementioned files are complete, you should see a
 191 total of 12 (6 \times 2) separate fastq files in your current working directory:

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

193 3.2 Creating symlinks to the files

194 Our workflow expects fastq files that have the format "***.R1/R2.clipped.fq**".
 195 Rather than rename them, we can simply create brand new symbolic links (sym-
 196 links) to the files, as follows:

```
197 cd ..
198 mkdir -p output/reads/clipped
199 cd output/reads/clipped
200
201 #embryonic libraries
```

```

202 ln -s ../../../../fastq-files/SRR424683_1.fastq E01h.R1.clipped.fq
203 ln -s ../../../../fastq-files/SRR424683_2.fastq E01h.R2.clipped.fq
204 ln -s ../../../../fastq-files/SRR424684_1.fastq E02h.R1.clipped.fq
205 ln -s ../../../../fastq-files/SRR424684_2.fastq E02h.R2.clipped.fq
206 ln -s ../../../../fastq-files/SRR424685_1.fastq E03h.R1.clipped.fq
207 ln -s ../../../../fastq-files/SRR424685_2.fastq E03h.R2.clipped.fq
208
209 #larval libraries
210 ln -s ../../../../fastq-files/SRR424707_1.fastq L1.R1.clipped.fq
211 ln -s ../../../../fastq-files/SRR424707_2.fastq L1.R2.clipped.fq
212 ln -s ../../../../fastq-files/SRR424708_1.fastq L2.R1.clipped.fq
213 ln -s ../../../../fastq-files/SRR424708_2.fastq L2.R2.clipped.fq
214 ln -s ../../../../fastq-files/SRR424709_1.fastq L3.R1.clipped.fq
215 ln -s ../../../../fastq-files/SRR424709_2.fastq L3.R2.clipped.fq
216
217 cd ../../.. #returning to the output directory

```

218 3.3 Downloading genomic data from *D. melanogaster*

219 Now that we have the fastq files from the RAMPAGE libraries downloaded and
 220 named appropriately, we now must retrieve the genome assembly and rRNA se-
 221 quences from *D. melanogaster*. The genome assembly is required for aligning the
 222 RAMPAGE reads, and the rRNA sequences are required to filter out matching
 223 reads in the sequenced RAMPAGE libraries. Because our sample is intended to
 224 contain only capped RNAs, any rRNA sequences we observe in these RAMPAGE
 225 libraries are contaminants that must be removed.

226
 227 Please download the rRNA sequences from the demo/additional_files folder
 228 in the demo. These sequences were retrieved separately from Genbank at the
 229 NCBI database. Please navigate to the rRNA file "Dmel_rRNA.fasta" found in
 230 the Demo.

```

231 head -n 3
232 >ref|NR_133562.1| Drosophila melanogaster 28S ribosomal RNA (28SrRNA:CR45844), rRNA
233 TTATATACAACCTCAACTCATATGGGACTACCCCTGAATTTAAGCATATTAATTAGGGGAGGAAAAGAA
234 ACTAACAAGGATTTTCTTAGTAGCGCGAGCGAAAAGAAAACAGTTCAGCACTAAGTCACTTTGTCTATA

```

235 We will then download a version of the *D. melanogaster* genome assem-
 236 bly from ENSEMBL (www.ensembl.org) [29]. To retrieve the genome assembly,
 237 please do the following:

```

238 mkdir genome
239 cd genome
240 wget ftp://ftp.ensembl.org/pub/release-78/fasta/
241 drosophila_melanogaster/dna/Drosophila_melanogaster.BDGP5.dna.toplevel.fa.gz
242 #uncompressing the file
243 gzip -d Drosophila_melanogaster.BDGP5.dna.toplevel.fa.gz
244 cd ..

```

245 3.4 Filtering and alignment of RAMPAGE reads using 246 GoRAMPAGE

247 At this stage we are ready to commence with the rRNA filtering and alignment
248 of the RAMPAGE libraries. We will use GoRAMPAGE, a tool we developed, to
249 perform these tasks in a concerted workflow. GoRAMPAGE runs TagDust [24]
250 to remove rRNA and low-complexity reads and STAR [26] to align RAMPAGE
251 (or other paired-end) reads to a given genome assembly.

252 **Setting up the GoRAMPAGE job.** Please refer to the script "GoRAMPAGE_script_MMB.sh"
253 and (using a text editor) provide the appropriate paths to the genome assembly,
254 output directory (see above) and rRNA sequences (*see Note 5*). GoRAMPAGE
255 jobs can optionally be run in parallel (*see Note 6*). The script can be executed
256 as follows:

```
257 #vi GoRAMPAGE_script_MMB.sh #updating with a text editor
258 ./GoRAMPAGE_script_MMB.sh
```

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

```
262 less -S errScript
```

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

266 **Inspecting the rRNA filtering results.** To evaluate the results from Step
267 3 (rRNA filtering), please navigate to the top level of the "output" directory
268 and open the file "LOGFILES". You'll see the recorded progress of the program
269 Tagdust and a record of the results. We notice that (for the L3h library) 1046448
270 of reads (78.1%) were "extracted", meaning that slightly more than 20% of
271 reads were removed because of matches with ribosomal sequences. The removed
272 reads from all libraries are found in the "dusted_discard" directory, and the
273 extracted reads are found in the current directory. Due to their sheer abundance
274 within cells, ribosomal RNA sequences are an inevitable contaminant within TSS
275 profiling libraries. For analysis purposes, it is important that these sequences be
276 removed, which is what has been completed here.
277 Since this step was conducted appropriately, we can proceed to the next step.

278 **Evaluating the alignments.** The folder "alignments/" in your GoRAMPAGE
279 output folder will now contain 6 .bam files, each representing the distinct RAM-
280 PAGE libraries selected for our analysis. Typing "ls -l" from the command line
281 will show that these files are symlinks to the original alignment files found
282 in the "STARoutput/" directory. "STARoutput/", as its name suggests, con-
283 tains the output from the STAR alignment, and this includes the alignment files

284 "*.sortedByCoord.out.bam", and four additional log files. The files with the suf-
 285 fix "*.STAR.Log.final.out" each contain a summary of the alignment, such as
 286 the number of input reads, the percentage of uniquely-mapped reads and the
 287 percentage of unmapped reads. An inspection of these log files indicates that
 288 the alignments have similar mapping rates (70-80%), a reasonable outcome for
 289 our purposes.

290

291 Now that our RAMPAGE libraries are filtered and aligned, we can commence
 292 with the second half of our analysis.

293 3.5 Promoter identification from aligned RAMPAGE libraries

294 We can now use the prepared alignment files to identify TSSs and promoters from
 295 the selected RAMPAGE libraries. There are currently several tools available
 296 for this purpose. *CAGEr*, developed by Haberle [30], was utilized to perform
 297 TSS identification as part of the FANTOM5 efforts. We will use *TSRchitect* in
 298 this demonstration, since it was specifically designed to analyze paired-end TSS
 299 profiling datasets, and also because it is more flexible with respect to model
 300 system (*i.e.* it does not require a corresponding *BSGenome* package). The latter
 301 feature will be helpful when analyzing the non-*D. melanogaster* TSS profiling
 302 datasets that we expect to be generated in the near future.

303 **Setting up the Analysis.** *TSRchitect*, the package we'll use for this analy-
 304 sis, is an R package available in the Bioconductor suite of genomics tools [28].
 305 It makes use of existing packages and data structures within this environment,
 306 where available, to identify promoters from sequence alignments. Since you have
 307 already installed *TSRchitect* and its dependencies (see section 2.3), we are set
 308 to proceed.

309

310 There are two general ways one can choose to run *TSRchitect*. The first is in-
 311 teractively *i.e.* typing the instructions directly into an R console. While this
 312 is a perfectly acceptable way to run analyses using package, for larger jobs
 313 it will likely be more efficient (and likely more reproducible) to run a dedi-
 314 cated R script. We have provided a sample script "*MMB_chapter_TSRchitect.R*"
 315 to make it easier for you to set up an R script. In the section to follow, we
 316 will go through the output of the analysis. For further details on how to use
 317 *TSRchitect*, please see its documentation at its Bioconductor page found here:
 318 <https://www.bioconductor.org/packages/release/bioc/html/TSRchitect.html>.

319 **Running the Analysis.** To run *TSRchitect* using the batch script, provide
 320 full paths for the variables "BAMDIR" and "DmAnnot" in the script provided
 321 (see **Note 7**). *BAMDIR* should be a path to the subdirectory "alignments/" in
 322 RAMPAGE output directory you specified earlier, and *DmAnnot* should be a
 323 full path to the *D. melanogaster* gene annotation listed above.

324

325 Once this is complete, we can run the batch script from the Linux command-line
 326 as follows:

```
327 R CMD BATCH MMB_chapter_TSRchitect.R
328 #assumes variables BAMDIR and DmAnnot have already been set
329 bg #puts this job in the background
```

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

332 **Reviewing the *TSRchitect* script.** Before we evaluate the results (which
 333 will have been written to your working directory after running the batch script),
 334 there are some important aspects of the analysis to review. We discuss these for
 335 informational purposes only; it will not necessary to perform these commands
 336 separate from the batch script provided. First, we must initialize the *tssObject*
 337 (which stores the information about the experiment) appropriately (*see Note 8*).

338
 339 The inputs in this case are BAM files (*inputType*="bam"); *TSRchitect* also ac-
 340 cepts input in BED format.

```
341 DmRAMPAGE <- loadTSSobj(experimentTitle = "RAMPAGE Tutorial", \
342   inputDir=BAMDIR, inputType="bam", isPairedEnd=TRUE, \
343   sampleNames=c("E1h", "E2h", "E3h", "L1", "L2", "L3"), \
344   replicateIDs=c(1,1,1,2,2,2))
```

345 A critical step in our analysis is identifying TSRs from the aligned TSS data;
 346 to do this we use the function *determineTSR*. We have selected the job to run
 347 on 4 cores in this example (*n.cores*=4). Please enter the number of cores ap-
 348 propriate for your system. Because we want to identify TSRs from every one
 349 of the selected RAMPAGE libraries, we specify *tssSet*="all". The parameter
 350 *tagCountThreshold* was set to 25, meaning that only TSSs supported by 25 or
 351 more 5' RAMPAGE reads will be included within a TSR. Setting *writeTable* to
 352 "TRUE" means that the identified TSRs from each set will be written to the
 353 working directory.

```
354 DmRAMPAGE <- determineTSR(experimentName=DmRAMPAGE, n.cores=4, \
355   tsrSetType="replicates", tssSet="all", tagCountThreshold=25, \
356   clustDist=20, writeTable=TRUE)
```

357 *TSRchitect* can incorporate the tag abundances from each of the samples
 358 and append them to the list of identified TSRs. This is useful for downstream
 359 analysis of differential expression.

```
360 DmRAMPAGE <- addTagCountsToTSR(experimentName=DmRAMPAGE, \
361   tsrSetType="replicates", tsrSet=1, tagCountThreshold=10, \
362   writeTable=TRUE)
```

363 We can use *TSRchitect* to import an annotation file (or, alternatively, use an
 364 existing one from *AnnotationHub*) and use it to associate our set of identified
 365 TSRs with coding genes. We can specify the maximum distances (both up-
 366 and downstream) between the TSR and the annotation using the arguments
 367 *upstreamDist* and *downstreamDist*.

```
368 DmRAMPAGE <- importAnnotationExternal(experimentName=DmRAMPAGE, \
369   fileType="gff3", annotFile=DmAnnot)
370
371 DmRAMPAGE <- addAnnotationToTSR(experimentName=DmRAMPAGE, \
372   tsrSetType="replicates", tsrSet=1, \
373   upstreamDist=1000, downstreamDist=200, feature="gene", \
374   featureColumnID="ID", writeTable=TRUE)
```

375 Now we have generated a set of identified TSSs, TSRs from all 6 RAMPAGE
 376 libraries, and have associated the identified TSRs with annotated genes. Next, we
 377 will merge the libraries into two samples according to condition: early embryonic
 378 (E1h, E2h, E3h) and late larval (L1, L2, L3) using the information we provided
 379 when we initialized the *tssObject* at the start of this section. After merging, we
 380 identify promoters i) within the merged samples and ii) within the entire dataset
 381 combined, and associate with the *D. melanogaster* gene annotation as described
 382 previously (not shown).

```
383 #merging the sample data into two groups
384 DmRAMPAGE <- mergeSampleData(DmRAMPAGE)
385
386 # ... identifying TSRs from the merged samples:
387 DmRAMPAGE <- determineTSR(experimentName=DmRAMPAGE, \
388   n.cores=4, tsrSetType="merged", \
389   tssSet="all", tagCountThreshold=40, \
390   clustDist=20, writeTable=TRUE)
```

391 **Evaluating the results** Our analysis using *TSRchitect* is now complete. A
 392 snapshot of a representative sample of small set of aligned RAMPAGE libraries
 393 is shown in Figure 3. Your working directory should now contain the following:

- 394 – TSSs from each sample *e.g.* TSSset-1.txt: (6)
- 395 – TSRs from each sample (in both .txt and .tab formats): (12)
- 396 – TSRs from each merged group (in both .txt and .tab formats): *e.g.* TSRsetMerged-
 397 1.txt: (4)
- 398 – TSRs from the combined set of TSSs: TSRsetCombined.tab: (1)

399 Let's briefly review the files (*see Note 9*). We can quickly obtain the counts on
 400 the command line, as follows:

```
401 wc -l *.tab
402 8377 TSRset-1.tab
```

```

403 6159 TSRset-2.tab
404 4814 TSRset-3.tab
405 17924 TSRset-4.tab
406 11851 TSRset-5.tab
407 3242 TSRset-6.tab
408 13986 TSRsetCombined.tab
409 7344 TSRsetMerged-1.tab
410 12126 TSRsetMerged-2.tab
411 85823 total

```

412 We will see that we have identified between roughly 3,200 and 18,000 TSRs
413 within the individual RAMPAGE samples, which is attributable to the dif-
414 ferences in library sizes. We detect 7,344 TSRs within the early embryonic
415 samples ("TSRsetMerged-1.tab") and 12,126 TSRs in the late larval samples
416 ("TSRsetMerged-2.tab"). Within the combined samples ("TSRsetCombined.tab")
417 we find 13,986 TSRs, which is similar to the number reported by Hoskins *et. al.*
418 [1].

419
420 In addition to identifying the position of a given TSRs, *TSRchitect* records other
421 useful information about its properties. The *width* of a TSR refers the span of
422 the genomic region it occupies (in bp), and the *Shape Index* (SI) is measure of
423 the relative peakedness of the TSR. We can see an example of this in the file
424 "TSRsetMerged-1.txt".

seq	start	end	strand	nTSSs	tsrWidth	shapeIndex	featureID
2L.67043.67044.+			2L	67043	67044 +	270 2	1 NA
2L.74089.74115.+			2L	74089	74115 +	341 27	0.13 NA
2L.94739.94752.+			2L	94739	94752 +	1650 14	0.55 FBgn0031
2L.102386.102386.+			2L	102386	102386 +	284 1	2 FBgn0031

430 3.6 Summary

431 The workflow provided here is intended to serve as a useful entry point for the
432 analysis of TSS profiling data in insects. On the computational side, we have
433 provided an open source set of tools so that the uninitiated genome scientist
434 can begin to analyze RAMPAGE (or other forms of TSS profiling data) quickly.
435 While the analysis centered on *D. melanogaster* via the use of public datasets,
436 it is anticipated that this will assist groups who may be interested in performing
437 TSS profiling in their preferred insect model system. The application of TSS
438 profiling technology across a more representative sample of insect diversity will
439 improve our understanding of the positions and general structure *cis*-regulatory
440 regions in this phylum.

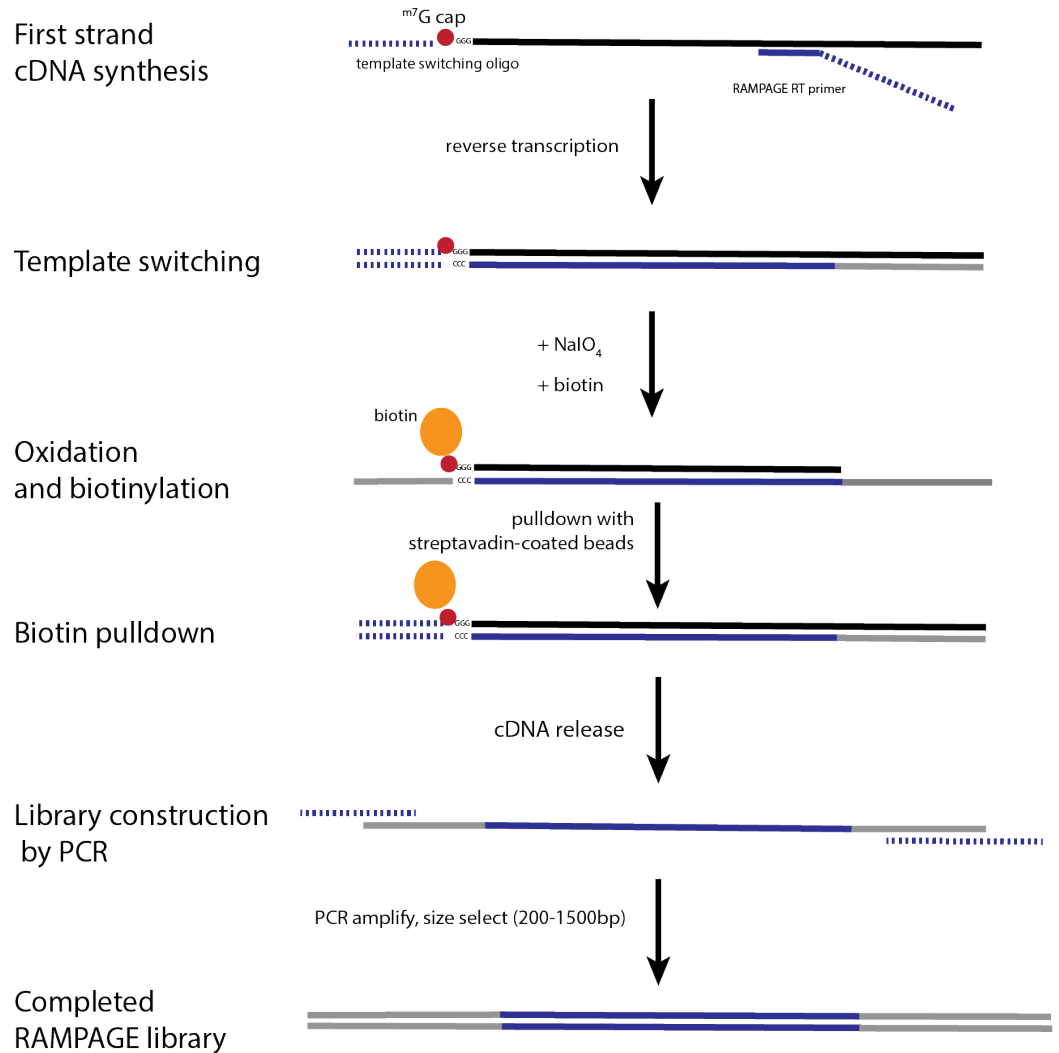


Fig. 1. A brief summary of the RAMPAGE protocol. Starting with high-quality total RNA, first-strand cDNA synthesis is initiated using a cap-bound oligonucleotide and a custom RAMPAGE RT primer, creating a double-stranded DNA-RNA hybrid molecule. Next, the 5'-m⁷G cap is oxidized, bound with biotin and pulled down with streptavidin-coated beads. The single-stranded cDNA molecules is released and the final RAMPAGE library construction is completed with PCR using custom oligonucleotides, followed by size-selection. This illustration was adapted from [18].

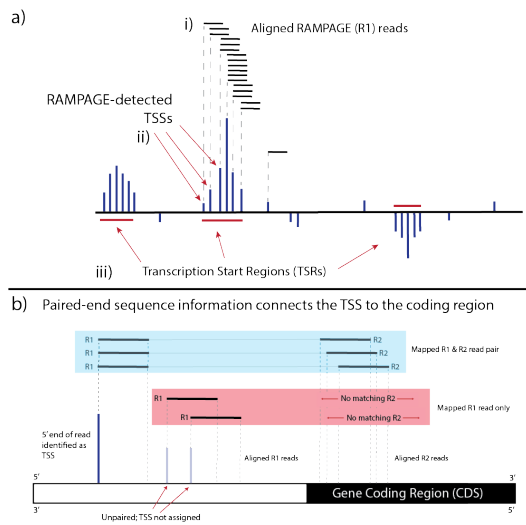


Fig. 2. An overview of promoter identification using RAMPAGE. a) RAMPAGE reads are aligned to the genome. The 5'-most genomic coordinate from each properly-paired R1 read is estimated as a TSS. The abundance of mapped 5'-ends at a given TSS is a measure of its abundance. TSSs above a minimum threshold will be clustered into TSRs. b) RAMPAGE-derived Paired-end sequence information provides a connection between a 5'-mRNA end and a gene coding region. Only properly-paired R1 reads (*i.e.* with an aligned R2 read) are identified as TSSs and then included in the downstream clustering procedure described in part a).

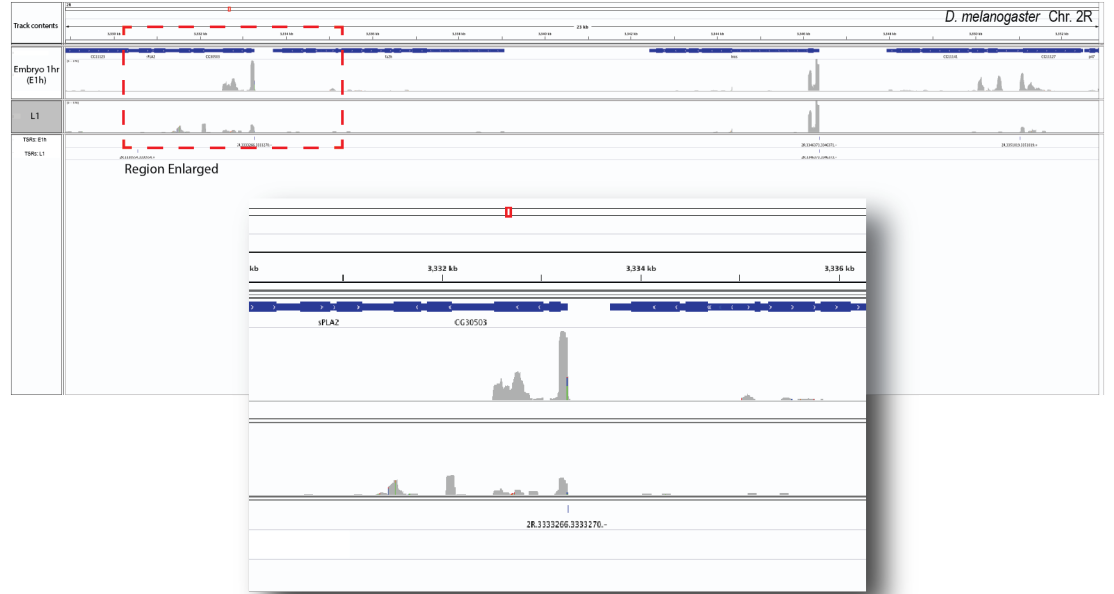


Fig. 3. An overview of the TSS profiling information provided by RAMPAGE. A representative visualization of RAMPAGE peaks (*i.e.* clusters of properly-aligned RAMPAGE reads) within an arbitrarily-selected genomic region of *D. melanogaster* chromosome 2R is shown, along with the corresponding gene annotation within this region. RAMPAGE data from two RAMPAGE libraries from Batut *et al* [2] are shown, which were generated from RNA isolated from developmental stages E1h and L1 *see Methods*. For each library, the abundance of RAMPAGE reads that align to a given site within the genome is represented by density plots (shown in gray). Gene models are shown in blue, where the thickened line represents exons and thin lines represent introns. The locations of TSRs identified by *TSRchitect* are shown in the two tracks from the bottom of the image. A single region, highlighted with the red dashed line is enlarged (the *Inset*) to show further detail of a selected gene and RAMPAGE signals. In some cases, the expression of 5'-ends between the two samples is roughly equivalent, whereas in others the observed signal is substantially higher (*see Inset*). The original images are screenshots generated in the Integrated Genomic Viewer (IGV; <http://software.broadinstitute.org/software/igv/>) [31]. Where necessary, additional annotation was added using Adobe Illustrator.

441 **3.7 Figures**442 **4 Notes**

- 443 1. Please consult the GoRAMPAGE documentation found here:
 444 <https://github.com/BrendelGroup/GoRAMPAGE>.
 445 Installation instructions for the prerequisites of GoRAMPAGE (which in-
 446 cludes some of the items listed) are found at the following link:
 447 <https://github.com/BrendelGroup/GoRAMPAGE/tree/master/src>.
 448 2. On Linux, the installation of a few packages are necessary in order to install
 449 Bioconductor packages using *biocLite*.
 450 To install them using Ubuntu:
 451

```
apt-get install libssl-dev
```


 452

```
apt-get install libcurl4-openssl-dev
```


 453

```
apt-get install libxml2-dev
```


 454 If you do not Ubuntu, use the commands necessary to install the above
 455 packages on your Linux distribution.
 456 3. You can clone the entire GoRAMPAGE repository (which includes the con-
 457 tents of the Demo) to your workspace on the command line using git, as
 458 follows:
 459

```
git clone https://github.com/brendelgroup/GoRAMPAGE/
```


 460

```
cd demo/MMB
```


 461 The "scripts/" folder in the Demo contains code for you to run the two major
 462 workflows described in this chapter. The "additional_files/" folder con-
 463 tains the following files which are necessary for the analysis: i) a fasta file con-
 464 taining ribosomal RNA sequences for *D. melanogaster* (*Dmel_rRNA.fasta*)
 465 and ii) a gene annotation for *D. melanogaster* (*Drosophila_melanogaster.BDGP5.78.gff*).
 466 4. Since these fastq files are paired-end, we use the argument *-split-files* to
 467 generate separate files for each read pair.
 468 5. If you are running this on a cluster with a job scheduler you'll need to add
 469 the necessary headers to the top of the script and submit the job in the
 470 appropriate manner.
 471 6. For parallel execution, GoRAMPAGE uses the Linux package *GNU parallel*
 472 [32]. Please see the GoRAMPAGE documentation for more information.
 473 7. To do this, please edit the batch script *TSRchitect_script_MMB.R* with a
 474 text editor of your choice.
 475 8. Because the samples provided derive from related developmental stages, we
 476 will merge them for annotation purposes using the argument *replicateIDs*,
 477 (though it must be emphasized that they are not replicates).
 478 9. All of *TSRchitect*'s output files are labeled according to the order that they
 479 are loaded onto the *tssObject*. For example, *TSSset-1.txt* corresponds to the
 480 first RAMPAGE dataset (in our case E1h), and *TSSset-2.txt* corresponds to
 481 the second RAMPAGE dataset (for this example E2h), and so on. You can
 482 check which datasets are loaded on the *tssObject* by simply entering it on an
 483 R console. Please see the *TSRchitect* documentation for more information.

Acknowledgments

The authors would like to thank Philippe Batut for generous technical assistance with the RAMPAGE protocol, and to Nathan Keith for his help establishing the protocol in our laboratory.

Disclosure Declaration

The authors declare that they have no competing interests.

5 References

References

1. 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.
2. P. J. Batut, A. Dobin, C. Plessy, P. Carninci, and T. R. Gingeras, "High-fidelity promoter profiling reveals widespread alternative promoter usage and transposon-driven developmental gene expression." *Genome Research*, Aug. 2012.
3. V. P. Brendel and R. T. Raborn, "Gorampage- a workflow for promoter detection by 5'-read mapping," <https://github.com/brendelGroup/GoRAMPAGE>, 2016.
4. R. T. Raborn and V. Brendel, *TSRchitect: Promoter identification from large-scale TSS profiling data*, 2017, r Bioconductor package version 1.0.0. [Online]. Available: <http://bioconductor.org/packages/release/bioc/html/TSRchitect.html>
5. 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.
6. 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.
7. 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. Huminiecchi, M. Iacono, K. Ikeo, A. Iwama, T. Ishikawa, M. Jakt, A. Kanapin,

- 526 M. Katoh, Y. Kawasawa, J. Kelso, H. Kitamura, H. Kitano, G. Kollias, S. P. T. Kr-
 527 ishnan, A. Kruger, S. K. Kummerfeld, I. V. Kurochkin, L. F. Lareau, D. Lazarevic,
 528 L. Lipovich, J. Liu, S. Liuni, S. McWilliam, M. Madan Babu, M. Madera, L. Mar-
 529 chionni, H. Matsuda, S. Matsuzawa, H. Miki, F. Mignone, S. Miyake, K. Mor-
 530 ris, S. Mottagui-Tabar, N. Mulder, N. Nakano, H. Nakauchi, P. Ng, R. Nilsson,
 531 S. Nishiguchi, S. Nishikawa, F. Nori, O. Ohara, Y. Okazaki, V. Orlando, K. C.
 532 Pang, W. J. Pavan, G. Pavesi, G. Pesole, N. Petrovsky, S. Piazza, J. Reed, J. F.
 533 Reid, B. Z. Ring, M. Ringwald, B. Rost, Y. Ruan, S. L. Salzberg, A. Sandelin,
 534 C. Schneider, C. Schönbach, K. Sekiguchi, C. A. M. Semple, S. Seno, L. Sessa,
 535 Y. Sheng, Y. Shibata, H. Shimada, K. Shimada, D. Silva, B. Sinclair, S. Sperling,
 536 E. Stupka, K. Sugiura, R. Sultana, Y. Takenaka, K. Taki, K. Tammoja, S. L. Tan,
 537 S. Tang, M. S. Taylor, J. Tegner, S. A. Teichmann, H. R. Ueda, E. van Nimwegen,
 538 R. Verardo, C. L. Wei, K. Yagi, H. Yamanishi, E. Zabarovsky, S. Zhu, A. Zim-
 539 mer, W. Hide, C. Bult, S. M. Grimmond, R. D. Teasdale, E. T. Liu, V. Brusic,
 540 J. Quackenbush, C. Wahlestedt, J. S. Mattick, D. A. Hume, C. Kai, D. Sasaki,
 541 Y. Tomaru, S. Fukuda, M. Kanamori-Katayama, M. Suzuki, J. Aoki, T. Arakawa,
 542 J. Iida, K. Imamura, M. Itoh, T. Kato, H. Kawaji, N. Kawagashira, T. Kawashima,
 543 M. Kojima, S. Kondo, H. Konno, K. Nakano, N. Ninomiya, T. Nishio, M. Okada,
 544 C. Plessy, K. Shibata, T. Shiraki, S. Suzuki, M. Tagami, K. Waki, A. Watahiki,
 545 Y. Okamura-Oho, H. Suzuki, J. Kawai, Y. Hayashizaki, F. Consortium, R. G. E. R.
 546 Group, and G. S. G. G. N. P. C. Group, “The transcriptional landscape of the mam-
 547 malian genome,” *Science (New York, NY)*, vol. 309, no. 5740, pp. 1559–1563, Sep.
 548 2005.
- 549 8. E. A. Rach, H.-Y. Yuan, W. H. Majoros, P. Tomancak, and U. Ohler, “Motif
 550 composition, conservation and condition-specificity of single and alternative tran-
 551 scription start sites in the *Drosophila* genome.” *Genome Biology*, vol. 10, no. 7, p.
 552 R73, 2009.
- 553 9. B. Lenhard, A. Sandelin, and P. Carninci, “Metazoan promoters: emerging char-
 554 acteristics and insights into transcriptional regulation.” *Nature Reviews Genetics*,
 555 vol. 13, no. 4, pp. 233–245, Apr. 2012.
- 556 10. T. Ni, D. L. Corcoran, E. A. Rach, S. Song, E. P. Spana, Y. Gao, U. Ohler,
 557 and J. Zhu, “A paired-end sequencing strategy to map the complex landscape of
 558 transcription initiation.” *Nature Methods*, vol. 7, no. 7, pp. 521–527, Jul. 2010.
- 559 11. U. Ohler, G.-c. Liao, H. Niemann, and G. M. Rubin, “Computational analysis of
 560 core promoters in the *Drosophila* genome.” *Genome Biology*, vol. 3, no. 12, pp.
 561 research0087.1–0087.12, 2002.
- 562 12. R. T. Raborn, K. Spitze, V. P. Brendel, and M. Lynch, “Promoter Architecture
 563 and Sex-Specific Gene Expression in *Daphnia pulex*.” *Genetics*, vol. 204, no. 2, pp.
 564 593–612, Aug. 2016.
- 565 13. C. Nepal, Y. Hadzhiev, C. Previti, V. Haberle, N. Li, H. Takahashi, A. M. M.
 566 Suzuki, Y. Sheng, R. F. Abdelhamid, S. Anand, J. Gehrig, A. Akalin, C. E. M.
 567 Kockx, A. A. J. van der Sloot, W. F. J. van IJcken, O. Armant, S. Rastegar,
 568 C. Watson, U. Strahle, E. Stupka, P. Carninci, B. Lenhard, and F. Muller, “Dy-
 569 namic regulation of the transcription initiation landscape at single nucleotide res-
 570 olution during vertebrate embryogenesis,” *Genome Research*, vol. 23, no. 11, pp.
 571 1938–1950, Nov. 2013.
- 572 14. P. Carninci, A. Sandelin, B. Lenhard, S. Katayama, K. Shimokawa, J. Ponjavic,
 573 C. A. M. Semple, M. S. Taylor, P. G. Engström, M. C. Frith, A. R. R. For-
 574 rest, W. B. Alkema, S. L. Tan, C. Plessy, R. Kodzius, T. Ravasi, T. Kasukawa,
 575 S. Fukuda, M. Kanamori-Katayama, Y. Kitazume, H. Kawaji, C. Kai, M. Naka-

- mura, 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.
15. 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.
16. 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.
17. N. Cvetesic and B. Lenhard, "Core promoters across the genome," *Nature Biotechnology*, vol. 35, no. 2, pp. 123–124, Feb. 2017.
18. P. J. Batut and T. R. Gingeras, "RAMPAGE: Promoter Activity Profiling by Paired-End Sequencing of 5'-Complete cDNAs." in *Current Protocols in Molecular Biology*. Current protocols in molecular biology / edited by Frederick M Ausubel [et al], 2013, pp. 25B.11.1–25B.11.16.
19. N. Merchant, E. Lyons, S. Goff, M. Vaughn, D. Ware, D. Micklos, and P. Antin, "The iPlant Collaborative: Cyberinfrastructure for Enabling Data to Discovery for the Life Sciences." *PLoS Biology*, vol. 14, no. 1, p. e1002342, Jan. 2016.
20. C. A. Stewart, T. M. Cockerill, I. Foster, D. Hancock, N. Merchant, E. Skidmore, D. Stanzione, J. Taylor, S. Tuecke, G. Turner, M. Vaughn, and N. I. Gaffney, "Jetstream: A self-provisioned, scalable science and engineering cloud environment," in *Proceedings of the 2015 XSEDE Conference: Scientific Advancements Enabled by Enhanced Cyberinfrastructure*, ser. XSEDE '15. New York, NY, USA: ACM, 2015, pp. 29:1–29:8. [Online]. Available: <http://doi.acm.org/10.1145/2792745.2792774>
21. R. Leinonen, H. Sugawara, M. Shumway, and International Nucleotide Sequence Database Collaboration, "The sequence read archive." *Nucleic Acids Research*, vol. 39, no. Database issue, pp. D19–21, Jan. 2011.
22. E. Aronesty, "Comparison of Sequencing Utility Programs," *The Open Bioinformatics Journal*, vol. 7, no. 1, pp. 1–8, Jan. 2013.
23. H. Lab, "FASTX Toolkit." [Online]. Available: http://hannonlab.cshl.edu/fastx_toolkit/
24. T. Lassmann, "TagDust2: a generic method to extract reads from sequencing data," *BMC Bioinformatics*, vol. 16, no. 1, p. 1, Jan. 2015.
25. H. Li, B. Handsaker, A. Wysoker, T. Fennell, J. Ruan, N. Homer, G. Marth, G. R. Abecasis, R. Durbin, and 1000 Genome Project Data Processing Subgroup, "The Sequence Alignment/Map format and SAMtools," *Bioinformatics (Oxford, England)*, vol. 25, no. 16, pp. 2078–2079, Aug. 2009.
26. 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.
27. R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2017. [Online]. Available: <https://www.R-project.org>
28. M. Lawrence and M. Morgan, "Scalable Genomics with R and Bioconductor," *Statistical Science*, vol. 29, no. 2, pp. 214–226, May 2014.

29. A. Yates, W. Akanni, M. R. Amode, D. Barrell, K. Billis, D. Carvalho-Silva,
C. Cummins, P. Clapham, S. Fitzgerald, L. Gil, C. G. GirÅşn, L. Gordon,
T. Hourlier, S. E. Hunt, S. H. Janacek, N. Johnson, T. Juettemann, S. Keenan,
I. Lavidas, F. J. Martin, T. Maurel, W. McLaren, D. N. Murphy, R. Nag,
M. Nuhn, A. Parker, M. Patricio, M. Pignatelli, M. Rahtz, H. S. Riat,
D. Sheppard, K. Taylor, A. Thormann, A. Vullo, S. P. Wilder, A. Zadissa,
E. Birney, J. Harrow, M. Muffato, E. Perry, M. Ruffier, G. Spudich, S. J.
Trevanion, F. Cunningham, B. L. Aken, D. R. Zerbino, and P. Flicek, "Ensembl
2016," *Nucleic Acids Research*, vol. 44, no. D1, pp. D710–D716, 2016. [Online].
Available: + <http://dx.doi.org/10.1093/nar/gkv1157>
30. V. Haberle, A. R. R. Forrest, Y. Hayashizaki, P. Carninci, and B. Lenhard,
"CAGEr: precise TSS data retrieval and high-resolution promoterome mining for
integrative analyses." *Nucleic Acids Research*, vol. 43, no. 8, pp. gkv054–e51, Feb.
2015.
31. H. Thorvaldsdottir, J. T. Robinson, and J. P. Mesirov, "Integrative Genomics
Viewer (IGV): high-performance genomics data visualization and exploration,"
Briefings in Bioinformatics (), vol. 14, no. 2, pp. 178–192, Mar. 2013.
32. O. Tange, "Gnu parallel - the command-line power tool," *login: The
USENIX Magazine*, vol. 36, no. 1, pp. 42–47, Feb 2011. [Online]. Available:
<http://www.gnu.org/s/parallel>

6 Checklist of Items to be Sent to Volume Editors

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

- ☐ The final L^AT_EX 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.