

# Using RAMPAGE to identify and annotate promoters in insect genomes

R. Taylor Raborn<sup>\*1</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

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](mailto: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. 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]. 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.

## 96 2 Materials

97 The analyses described herein require a workstation capable of doing modern  
 98 bioinformatics; minimally a reasonably-appointed laptop. An intermediate un-  
 99 derstanding of the Linux/Unix command line will be extremely useful, although  
 100 we make efforts to explain the procedures with clarity. In addition, it will likely  
 101 be necessary for the participant to have superuser privileges on the machine.  
 102 If you do not have a machine (or have access to one) that meets these re-  
 103 quirements, it is recommended that you consider cloud-based cyberinfrastruc-  
 104 ture, including Amazon Web Services (AWS; <https://aws.amazon.com/>), Cy-  
 105 Verse (<http://www.cyverse.org/>) [19], or JetStream (<https://jetstream-cloud.org/>)  
 106 [20]. The former is a well-known pay-per-use solution, while the latter two are  
 107 NSF-funded resources that makes compute allocations freely available to the  
 108 public.

### 109 2.1 Hardware

- 110 1. x86-64 compatible processors
- 111 2. At least 8GB RAM
- 112 3. 30GB+ hard disk space

### 113 2.2 Operating System

- 114 – 64 bit Linux (preferred) or Mac OS X (with Command Line Tools from  
 115 XCode)

### 116 2.3 Software

117 Below is a list of the software packages required for this demonstration (*see Note*  
 118 **1**).

#### 119 Sequence retrieval

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

#### 122 GoRAMPAGE

- 123 1. GoRAMPAGE [3] (<https://github.com/brendelGroup/GoRAMPAGE>)
- 124 2. fastq-multx [22] (<https://github.com/brwnj/fastq-multx>)
- 125 3. FASTX-Toolkit [23] ([http://hannonlab.cshl.edu/fastx\\_toolkit/Index.html](http://hannonlab.cshl.edu/fastx_toolkit/Index.html))
- 126 4. TagDust2 [24] (<https://sourceforge.net/projects/tagdust/>)
- 127 5. Samtools [25] (<http://www.htslib.org/doc/samtools.html>)
- 128 6. STAR [26] (<https://github.com/alexdobin/STAR>)

#### 129 TSRchitect

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

## 134 2.4 Demonstration

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

## 139 2.5 Installation of R packages

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

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

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

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

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

```
160 library(TSRchitect) #loading TSRchitect
```

## 161 3 Methods

### 162 3.1 Retrieving the RAMPAGE sequence data from NCBI

163 To begin our analysis, we must download the RAMPAGE data to our worksta-  
 164 tion. We will utilize tools provided by the SRA Toolkit, which should already  
 165 be installed on your machine (see **Materials**). The command *fastq-dump* al-  
 166 lows one to directly retrieve data from the GEO database using the appropriate  
 167 identifier(s). While there are 36 RAMPAGE libraries in the Batut *et al.* pa-  
 168 per, we will select a subset of these to analyze here. We will compare samples  
 169 from selected embryonic (E01h-E03h) and larval (L1-L3) tissues, representing  
 170 the beginning and end of embryonic development. For more information about

171 the experiment and the available RAMPAGE libraries, please see the following  
 172 link: <https://www.ncbi.nlm.nih.gov/Traces/study/?acc=SRP011193>.

173  
 174 First, let's proceed with downloading the libraries from early embryonic tissues  
 175 (see **See Note 4**). We will make a new folder (entitled "fastq\_files/") to  
 176 house these files.

```
177 mkdir fastq_files
178 cd fastq_files
179
180 fastq-dump --split-files SRR424683
181 fastq-dump --split-files SRR424684
182 fastq-dump --split-files SRR424685
```

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

```
184 fastq-dump --split-files SRR424707
185 fastq-dump --split-files SRR424708
186 fastq-dump --split-files SRR424709
```

187 Once the download of the aforementioned files are complete, you should see a  
 188 total of 12 (6 x 2) separate fastq files in your current working directory:

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

### 190 3.2 Creating symlinks to the files

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

```
194 cd ..
195 mkdir -p output/reads/clipped
196 cd output/reads/clipped
197
198 #embryonic libraries
199 ln -s ../../../../fastq-files/SRR424683_1.fastq E01h.R1.clipped.fq
200 ln -s ../../../../fastq-files/SRR424683_2.fastq E01h.R2.clipped.fq
201 ln -s ../../../../fastq-files/SRR424684_1.fastq E02h.R1.clipped.fq
202 ln -s ../../../../fastq-files/SRR424684_2.fastq E02h.R2.clipped.fq
203 ln -s ../../../../fastq-files/SRR424685_1.fastq E03h.R1.clipped.fq
204 ln -s ../../../../fastq-files/SRR424685_2.fastq E03h.R2.clipped.fq
205
206 #larval libraries
207 ln -s ../../../../fastq-files/SRR424707_1.fastq L1.R1.clipped.fq
208 ln -s ../../../../fastq-files/SRR424707_2.fastq L1.R2.clipped.fq
209 ln -s ../../../../fastq-files/SRR424708_1.fastq L2.R1.clipped.fq
```

```

210 ln -s ../../../../fastq-files/SRR424708_2.fastq L2.R2.clipped.fq
211 ln -s ../../../../fastq-files/SRR424709_1.fastq L3.R1.clipped.fq
212 ln -s ../../../../fastq-files/SRR424709_2.fastq L3.R2.clipped.fq
213
214 cd ../../.. #returning to the output directory

```

### 215 3.3 Downloading genomic data from *D. melanogaster*

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

223  
 224 Please download the rRNA sequences from the demo/additional\_files folder  
 225 in the demo. These sequences were retrieved separately from Genbank at the  
 226 NCBI database. Please navigate to the rRNA file "Dmel\_rRNA.fasta" found in  
 227 the Demo.

```

228 head -n 3
229 >ref|NR_133562.1| Drosophila melanogaster 28S ribosomal RNA (28SrRNA:CR45844), rRNA
230 TTATATACAACCTCAACTCATATGGGACTACCCCTGAATTAAAGCATATTAATTAGGGGAGGAAAAGAA
231 ACTAACAAGGATTTTCTTAGTAGCGGCGAGCGAAAAGAAAACAGTTCAGCACTAAGTCACTTTGTCTATA

```

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

```

235 mkdir genome
236 cd genome
237 wget ftp://ftp.ensembl.org/pub/release-78/fasta/drosophila_melanogaster/dna/Drosophila_m
238 #uncompressing the file
239 gzip -d Drosophila_melanogaster.BDGP5.dna.toplevel.fa.gz
240 cd ..

```

### 241 3.4 Filtering and alignment of RAMPAGE reads using 242 GoRAMPAGE

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

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

```
253 #vi GoRAMPAGE_script_MMB.sh #updating with a text editor
254 ./GoRAMPAGE_script_MMB.sh
```

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

```
258 less -S errScript
```

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

262 **Inspecting the rRNA filtering results.** To evaluate the results from Step  
 263 3 (rRNA filtering), please navigate to the top level of the "output" directory  
 264 and open the file "LOGFILES". You'll see the recorded progress of the program  
 265 Tagdust and a record of the results. We notice that (for the L3h library) 1046448  
 266 of reads (78.1%) were "extracted", meaning that slightly more than 20% of  
 267 reads were removed because of matches with ribosomal sequences. The removed  
 268 reads from all libraries are found in the "dusted\_discard" directory, and the  
 269 extracted reads are found in the current directory. Due to their sheer abundance  
 270 within cells, ribosomal RNA sequences are an inevitable contaminant within TSS  
 271 profiling libraries. For analysis purposes, it is important that these sequences be  
 272 removed, which is what has been completed here.  
 273 Since this step was conducted appropriately, we can proceed to the next step.

274 **Evaluating the alignments.** The folder "alignments/" in your GoRAMPAGE  
 275 output folder will now contain 6 .bam files, each representing the distinct RAM-  
 276 PAGE libraries selected for our analysis. Typing "ls -l" from the command line  
 277 will show that these files are symlinks to the original alignment files found  
 278 in the "STARoutput/" directory. "STARoutput/", as its name suggests, con-  
 279 tains the output from the STAR alignment, and this includes the alignment files  
 280 "\*.sortedByCoord.out.bam", and four additional log files. The files with the suf-  
 281 fix "\*.STAR.Log.final.out" each contain a summary of the alignment, such as  
 282 the number of input reads, the percentage of uniquely-mapped reads and the  
 283 percentage of unmapped reads. An inspection of these log files indicates that  
 284 the alignments have similar mapping rates ( 70-80%), a reasonable outcome for  
 285 our purposes.

286  
 287 Now that our RAMPAGE libraries are filtered and aligned, we can commence  
 288 with the second half of our analysis.



### 289 3.5 Promoter identification from aligned RAMPAGE libraries

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

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

305  
 306 There are two general ways one can choose to run *TSRchitect*. The first is in-  
 307 teractively *i.e.* typing the instructions directly into an R console. While this  
 308 is a perfectly acceptable way to run analyses using package, for larger jobs  
 309 it will likely be more efficient (and likely more reproducible) to run a dedi-  
 310 cated R script. We have provided a sample script "MMB\_chapter\_TSRchitect.R"  
 311 to make it easier for you to set up an R script. In the section to follow, we  
 312 will go through the output of the analysis. For further details on how to use  
 313 *TSRchitect*, please see its documentation at its Bioconductor page found here:  
 314 <https://www.bioconductor.org/packages/release/bioc/html/TSRchitect.html>.

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

320  
 321 Once this is complete, we can run the batch script from the Linux command-line  
 322 as follows:

```
323 R CMD BATCH MMB_chapter_TSRchitect.R
324 #assumes variables BAMDIR and DmAnnot have already been set
325 bg #puts this job in the background
```

326 Once the job is underway, you can monitor its progress by looking at the contents  
 327 of the .Rout file (in this case, "MMB\_chapter\_TSRchitect.Rout").

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

334  
 335 The inputs in this case are BAM files (*inputType*="bam"); *TSRchitect* also ac-  
 336 cepts input in BED format.

```
337 DmRAMPAGE <- loadTSSobj(experimentTitle = "RAMPAGE Tutorial", \
338   inputDir=BAMDIR, inputType="bam", isPairedEnd=TRUE, \
339   sampleNames=c("E1h","E2h", "E3h", "L1", "L2", "L3"), \
340   replicateIDs=c(1,1,1,2,2,2))
```

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

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

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

```
356 DmRAMPAGE <- addTagCountsToTSR(experimentName=DmRAMPAGE, \
357   tsrSetType="replicates", tsrSet=1, tagCountThreshold=10, \
358   writeTable=TRUE)
```

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

```
364 DmRAMPAGE <- importAnnotationExternal(experimentName=DmRAMPAGE, \
365   fileType="gff3", annotFile=DmAnnot)
```

```
366  

367 DmRAMPAGE <- addAnnotationToTSR(experimentName=DmRAMPAGE, \
368   tsrSetType="replicates", tsrSet=1, \
369   upstreamDist=1000, downstreamDist=200, feature="gene", \
370   featureColumnID="ID", writeTable=TRUE)
```

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

```
#merging the sample data into two groups
DmRAMPAGE <- mergeSampleData(DmRAMPAGE)

# ... identifying TSRs from the merged samples:
DmRAMPAGE <- determineTSR(experimentName=DmRAMPAGE, \
  n.cores=4, tsrSetType="merged", \
  tssSet="all", tagCountThreshold=40, \
  clustDist=20, writeTable=TRUE)
```

**Evaluating the results** Our analysis using *TSRchitect* is now complete. Your working directory should now contain the following:

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

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

```
wc -l *.tab
8377 TSRset-1.tab
6159 TSRset-2.tab
4814 TSRset-3.tab
17924 TSRset-4.tab
11851 TSRset-5.tab
3242 TSRset-6.tab
13986 TSRsetCombined.tab
7344 TSRsetMerged-1.tab
12126 TSRsetMerged-2.tab
85823 total
```

We will see that we have identified between roughly 3,200 and 18,000 TSRs within the individual RAMPAGE samples, which is attributable to the differences in library sizes. We detect 7,344 TSRs within the early embryonic samples ("TSRsetMerged-1.tab") and 12,126 TSRs in the late larval samples ("TSRsetMerged-2.tab"). Within the combined samples ("TSRsetCombined.tab")

we find 13,986 TSRs, which is similar to the number reported by Hoskins *et. al.* [1].

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

### 3.6 Summary

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

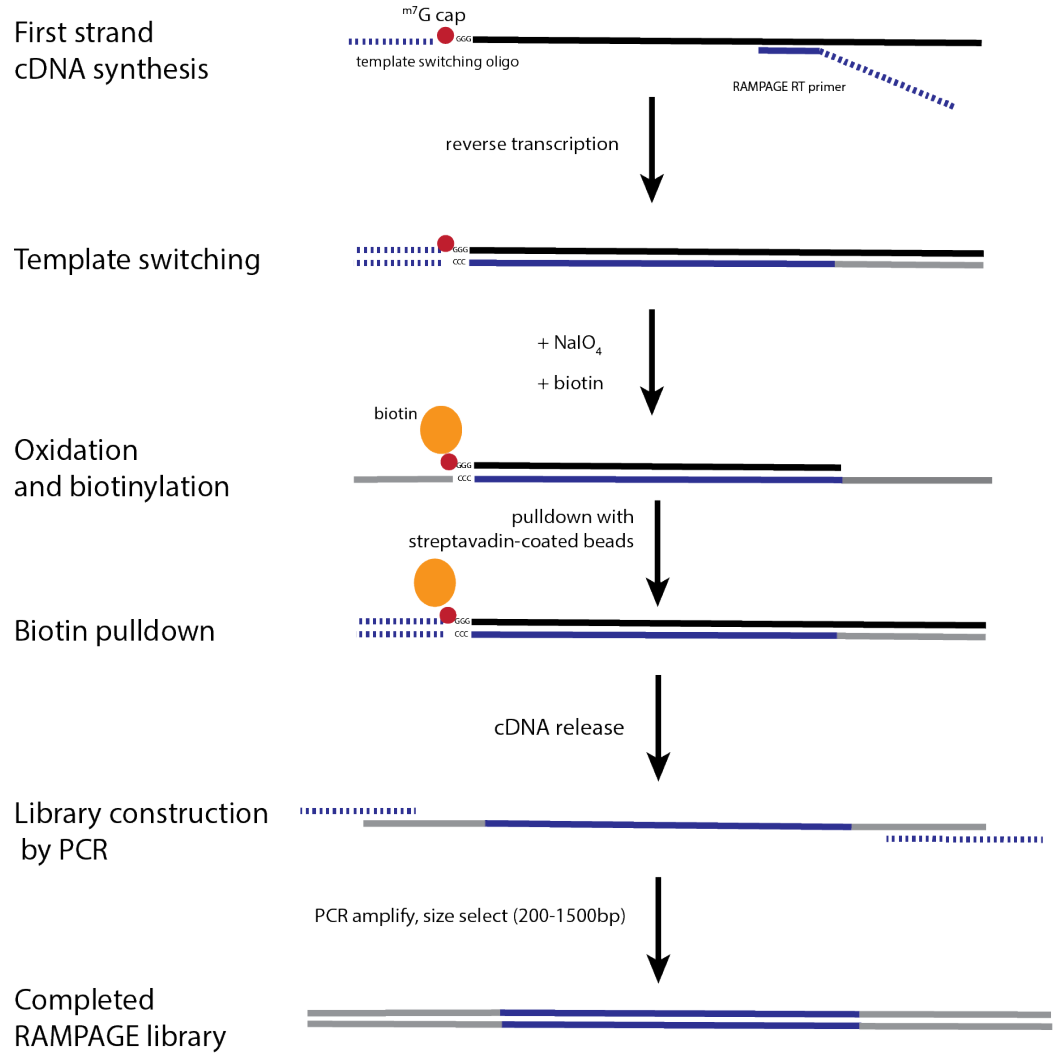
### 3.7 Figures

## 4 Notes

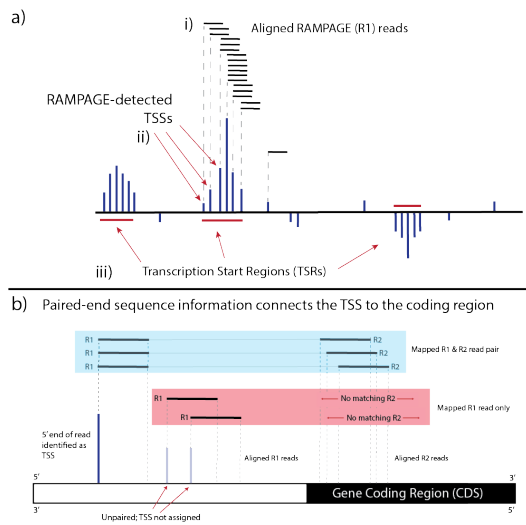
1. Please consult the GoRAMPAGE documentation found here:  
<https://github.com/BrendelGroup/GoRAMPAGE>.  
 Installation instructions for the prerequisites of GoRAMPAGE (which includes some of the items listed) are found at the following link:  
<https://github.com/BrendelGroup/GoRAMPAGE/tree/master/src>.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

```
apt-get install libssl-dev
apt-get install libcurl4-openssl-dev
apt-get install libxml2-dev
```

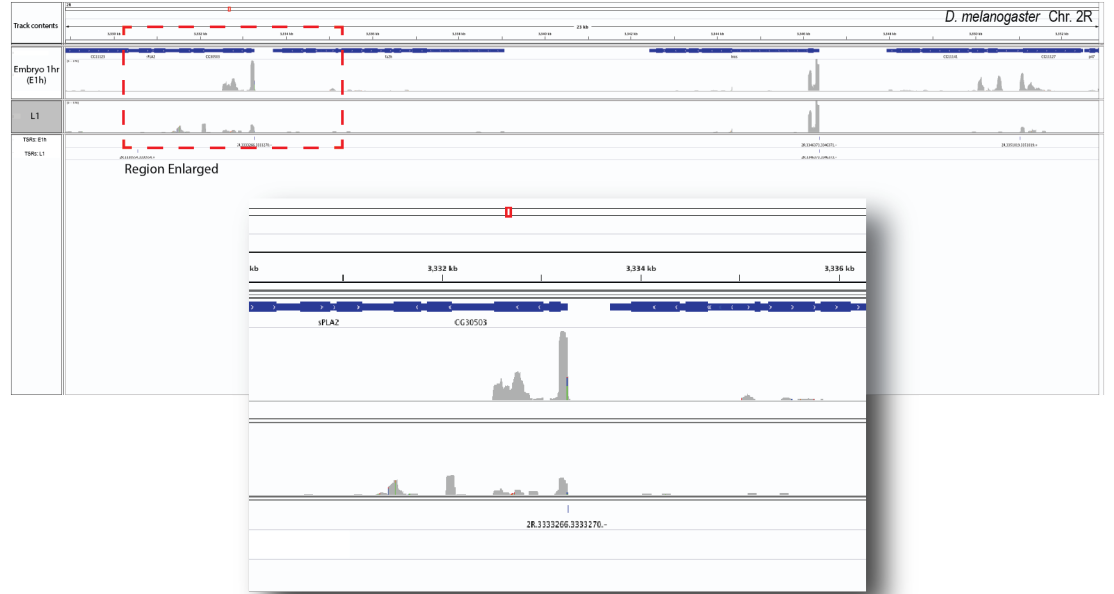
 If you do not Ubuntu, use the commands necessary to install the above packages on your Linux distribution.



**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.

3. You can clone the entire GoRAMPAGE repository (which includes the contents of the Demo) to your workspace on the command line using git, as follows:
 

```
git clone https://github.com/brendelgroup/GoRAMPAGE/
cd demo/MMB
```

The "scripts/" folder in the Demo contains code for you to run the two major workflows described in this chapter. The "additional\_files/" folder contains the following files which are necessary for the analysis: i) a fasta file containing ribosomal RNA sequences for *D. melanogaster* (*Dmel\_rRNA.fasta*) and ii) a gene annotation for *D. melanogaster* (*Drosophila\_melanogaster.BDGP5.78.gff*).
4. Since these fastq files are paired-end, we use the argument *-split-files* to generate separate files for each read pair.
5. 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.
6. For parallel execution, GoRAMPAGE uses the Linux package *GNU parallel* [32]. Please see the GoRAMPAGE documentation for more information.
7. To do this, please edit the batch script *TSRchitect\_script\_MMB.R* with a text editor of your choice.
8. Because the samples provided derive from related developmental stages, we will merge them for annotation purposes using the argument *replicateIDs*, (though it must be emphasized that they are not replicates).
9. All of *TSRchitect*'s output files are labeled according to the order that they are loaded onto the *tssObject*. For example, *TSSset-1.txt* corresponds to the first RAMPAGE dataset (in our case E1h), and *TSSset-2.txt* corresponds to the second RAMPAGE dataset (for this example E2h), and so on. You can check which datasets are loaded on the *tssObject* by simply entering it on an 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,



- 489 T. Lassmann, C. Yu, C. Yu, B. W. Booth, B. W. Booth, D. Zhang, D. Zhang,  
490 K. H. Wan, K. H. Wan, L. Yang, L. Yang, N. Boley, N. Boley, J. Andrews, J. An-  
491 drews, T. C. Kaufman, T. C. Kaufman, B. R. Graveley, B. R. Graveley, P. J.  
492 Bickel, P. J. Bickel, P. Carninci, J. W. Carlson, J. W. Carlson, S. E. Celniker,  
493 and S. E. Celniker, "Genome-wide analysis of promoter architecture in *Drosophila*  
494 *melanogaster*." *Genome Research*, vol. 21, no. 2, pp. 182–192, Feb. 2011.
- 495 2. P. J. Batut, A. Dobin, C. Plessy, P. Carninci, and T. R. Gingeras, "High-fidelity  
496 promoter profiling reveals widespread alternative promoter usage and transposon-  
497 driven developmental gene expression." *Genome Research*, Aug. 2012.
- 498 3. V. P. Brendel and R. T. Raborn, "Gorampage- a workflow for promoter detection  
499 by 5'-read mapping," <https://github.com/brendelGroup/GoRAMPAGE>, 2016.
- 500 4. R. T. Raborn and V. Brendel, *TSRchitect: Promoter identification from large-scale*  
501 *TSS profiling data*, 2017, r Bioconductor package version 1.0.0. [Online]. Available:  
502 <http://bioconductor.org/packages/release/bioc/html/TSRchitect.html>
- 503 5. J. T. Kadonaga, "Perspectives on the RNA polymerase II core promoter." *Wiley*  
504 *Interdisciplinary Reviews: Developmental Biology*, vol. 1, no. 1, pp. 40–51, Jan.  
505 2012.
- 506 6. R. Kodzius, M. Kojima, H. Nishiyori, M. Nakamura, S. Fukuda, M. Tagami,  
507 D. Sasaki, K. Imamura, C. Kai, M. Harbers, Y. Hayashizaki, and P. Carninci,  
508 "CAGE: cap analysis of gene expression." *Nature Methods*, vol. 3, no. 3, pp. 211–  
509 222, Mar. 2006.
- 510 7. P. Carninci, T. Kasukawa, S. Katayama, J. Gough, M. C. Frith, N. Maeda,  
511 R. Oyama, T. Ravasi, B. Lenhard, C. Wells, R. Kodzius, K. Shimokawa, V. B.  
512 Bajic, S. E. Brenner, S. Batalov, A. R. R. Forrest, M. Zavolan, M. J. Davis, L. G.  
513 Wilming, V. Aidinis, J. E. Allen, A. Ambesi-Impimbato, R. Apweiler, R. N. Atu-  
514 raliya, T. L. Bailey, M. Bansal, L. Baxter, K. W. Beisel, T. Bersano, H. Bono, A. M.  
515 Chalk, K. P. Chiu, V. Choudhary, A. Christoffels, D. R. Clutterbuck, M. L. Crowe,  
516 E. Dalla, B. P. Dalrymple, B. de Bono, G. Della Gatta, D. di Bernardo, T. Down,  
517 P. Engstrom, M. Fagiolini, G. Faulkner, C. F. Fletcher, T. Fukushima, M. Furuno,  
518 S. Futaki, M. Gariboldi, P. Georgii-Hemming, T. R. Gingeras, T. Gojobori, R. E.  
519 Green, S. Gustincich, M. Harbers, Y. Hayashi, T. K. Hensch, N. Hirokawa, D. Hill,  
520 L. Huminiecki, M. Iacono, K. Ikeo, A. Iwama, T. Ishikawa, M. Jakt, A. Kanapin,  
521 M. Katoh, Y. Kawasaki, J. Kelso, H. Kitamura, H. Kitano, G. Kollias, S. P. T. Kr-  
522 ishnan, A. Kruger, S. K. Kummerfeld, I. V. Kurochkin, L. F. Lareau, D. Lazarevic,  
523 L. Lipovich, J. Liu, S. Liuni, S. McWilliam, M. Madan Babu, M. Madera, L. Mar-  
524 chionni, H. Matsuda, S. Matsuzawa, H. Miki, F. Mignone, S. Miyake, K. Mor-  
525 ris, S. Mottagui-Tabar, N. Mulder, N. Nakano, H. Nakauchi, P. Ng, R. Nilsson,  
526 S. Nishiguchi, S. Nishikawa, F. Nori, O. Ohara, Y. Okazaki, V. Orlando, K. C.  
527 Pang, W. J. Pavan, G. Pavesi, G. Pesole, N. Petrovsky, S. Piazza, J. Reed, J. F.  
528 Reid, B. Z. Ring, M. Ringwald, B. Rost, Y. Ruan, S. L. Salzberg, A. Sandelin,  
529 C. Schneider, C. Schönbach, K. Sekiguchi, C. A. M. Semple, S. Seno, L. Sessa,  
530 Y. Sheng, Y. Shibata, H. Shimada, K. Shimada, D. Silva, B. Sinclair, S. Sperling,  
531 E. Stupka, K. Sugiura, R. Sultana, Y. Takenaka, K. Taki, K. Tammoja, S. L. Tan,  
532 S. Tang, M. S. Taylor, J. Tegner, S. A. Teichmann, H. R. Ueda, E. van Nimwegen,  
533 R. Verardo, C. L. Wei, K. Yagi, H. Yamanishi, E. Zabarovsky, S. Zhu, A. Zim-  
534 mer, W. Hide, C. Bult, S. M. Grimmond, R. D. Teasdale, E. T. Liu, V. Brusic,  
535 J. Quackenbush, C. Wahlestedt, J. S. Mattick, D. A. Hume, C. Kai, D. Sasaki,  
536 Y. Tomaru, S. Fukuda, M. Kanamori-Katayama, M. Suzuki, J. Aoki, T. Arakawa,  
537 J. Iida, K. Imamura, M. Itoh, T. Kato, H. Kawaji, N. Kawagashira, T. Kawashima,  
538 M. Kojima, S. Kondo, H. Konno, K. Nakano, N. Ninomiya, T. Nishio, M. Okada,

- 539 C. Plessy, K. Shibata, T. Shiraki, S. Suzuki, M. Tagami, K. Waki, A. Watahiki,  
540 Y. Okamura-Oho, H. Suzuki, J. Kawai, Y. Hayashizaki, F. Consortium, R. G. E. R.  
541 Group, and G. S. G. G. N. P. C. Group, "The transcriptional landscape of the mam-  
542 malian genome," *Science (New York, NY)*, vol. 309, no. 5740, pp. 1559–1563, Sep.  
543 2005.
- 544 8. E. A. Rach, H.-Y. Yuan, W. H. Majoros, P. Tomancak, and U. Ohler, "Motif  
545 composition, conservation and condition-specificity of single and alternative tran-  
546 scription start sites in the *Drosophila* genome." *Genome Biology*, vol. 10, no. 7, p.  
547 R73, 2009.
- 548 9. B. Lenhard, A. Sandelin, and P. Carninci, "Metazoan promoters: emerging char-  
549 acteristics and insights into transcriptional regulation." *Nature Reviews Genetics*,  
550 vol. 13, no. 4, pp. 233–245, Apr. 2012.
- 551 10. T. Ni, D. L. Corcoran, E. A. Rach, S. Song, E. P. Spana, Y. Gao, U. Ohler,  
552 and J. Zhu, "A paired-end sequencing strategy to map the complex landscape of  
553 transcription initiation." *Nature Methods*, vol. 7, no. 7, pp. 521–527, Jul. 2010.
- 554 11. U. Ohler, G.-c. Liao, H. Niemann, and G. M. Rubin, "Computational analysis of  
555 core promoters in the *Drosophila* genome." *Genome Biology*, vol. 3, no. 12, pp.  
556 research0087.1–0087.12, 2002.
- 557 12. R. T. Raborn, K. Spitze, V. P. Brendel, and M. Lynch, "Promoter Architecture  
558 and Sex-Specific Gene Expression in *Daphnia pulex*." *Genetics*, vol. 204, no. 2, pp.  
559 593–612, Aug. 2016.
- 560 13. C. Nepal, Y. Hadzhiev, C. Previti, V. Haberle, N. Li, H. Takahashi, A. M. M.  
561 Suzuki, Y. Sheng, R. F. Abdelhamid, S. Anand, J. Gehrig, A. Akalin, C. E. M.  
562 Kockx, A. A. J. van der Sloot, W. F. J. van IJcken, O. Armant, S. Rastegar,  
563 C. Watson, U. Strahle, E. Stupka, P. Carninci, B. Lenhard, and F. Muller, "Dy-  
564 namic regulation of the transcription initiation landscape at single nucleotide res-  
565 olution during vertebrate embryogenesis," *Genome Research*, vol. 23, no. 11, pp.  
566 1938–1950, Nov. 2013.
- 567 14. P. Carninci, A. Sandelin, B. Lenhard, S. Katayama, K. Shimokawa, J. Ponjavic,  
568 C. A. M. Semple, M. S. Taylor, P. G. Engström, M. C. Frith, A. R. R. For-  
569 rest, W. B. Alkema, S. L. Tan, C. Plessy, R. Kodzius, T. Ravasi, T. Kasukawa,  
570 S. Fukuda, M. Kanamori-Katayama, Y. Kitazume, H. Kawaji, C. Kai, M. Naka-  
571 mura, H. Konno, K. Nakano, S. Mottagui-Tabar, P. Arner, A. Chesi, S. Gustincich,  
572 F. Persichetti, H. Suzuki, S. M. Grimmond, C. A. Wells, V. Orlando, C. Wahlest-  
573 edt, E. T. Liu, M. Harbers, J. Kawai, V. B. Bajic, D. A. Hume, and Y. Hayashizaki,  
574 "Genome-wide analysis of mammalian promoter architecture and evolution," *Nat-  
575 ure Genetics*, vol. 38, no. 6, pp. 626–635, Apr. 2006.
- 576 15. S. Mwangi, G. Attardo, Y. Suzuki, S. Aksoy, and A. Christoffels, "TSS seq based  
577 core promoter architecture in blood feeding Tsetse fly (*Glossina morsitans mor-  
578 sitans*) vector of Trypanosomiasis," *BMC Genomics*, vol. 16, no. 1, p. 722, Sep.  
579 2015.
- 580 16. K. Tsuchihara, Y. Suzuki, H. Wakaguri, T. Irie, K. Tanimoto, S.-i. Hashimoto,  
581 K. Matsushima, J. Mizushima-Sugano, R. Yamashita, K. Nakai, D. Bentley, H. Es-  
582 umi, and S. Sugano, "Massive transcriptional start site analysis of human genes in  
583 hypoxia cells," *Nucleic Acids Research*, vol. 37, no. 7, pp. 2249–2263, Apr. 2009.
- 584 17. N. Cvetesic and B. Lenhard, "Core promoters across the genome," *Nature Biotech-  
585 nology*, vol. 35, no. 2, pp. 123–124, Feb. 2017.
- 586 18. P. J. Batut and T. R. Gingeras, "RAMPAGE: Promoter Activity Profiling by  
587 Paired-End Sequencing of 5'-Complete cDNAs." in *Current Protocols in Molecular  
588 Biology*. Current protocols in molecular biology / edited by Frederick M Ausubel  
589 [et al], 2013, pp. 25B.11.1–25B.11.16.

- 590 19. N. Merchant, E. Lyons, S. Goff, M. Vaughn, D. Ware, D. Micklos, and P. Antin,  
591 “The iPlant Collaborative: Cyberinfrastructure for Enabling Data to Discovery for  
592 the Life Sciences.” *PLoS Biology*, vol. 14, no. 1, p. e1002342, Jan. 2016.
- 593 20. C. A. Stewart, T. M. Cockerill, I. Foster, D. Hancock, N. Merchant,  
594 E. Skidmore, D. Stanzione, J. Taylor, S. Tuecke, G. Turner, M. Vaughn,  
595 and N. I. Gaffney, “Jetstream: A self-provisioned, scalable science and  
596 engineering cloud environment,” in *Proceedings of the 2015 XSEDE Conference:  
597 Scientific Advancements Enabled by Enhanced Cyberinfrastructure*, ser. XSEDE  
598 ’15. New York, NY, USA: ACM, 2015, pp. 29:1–29:8. [Online]. Available:  
599 <http://doi.acm.org/10.1145/2792745.2792774>
- 600 21. R. Leinonen, H. Sugawara, M. Shumway, and International Nucleotide Sequence  
601 Database Collaboration, “The sequence read archive.” *Nucleic Acids Research*,  
602 vol. 39, no. Database issue, pp. D19–21, Jan. 2011.
- 603 22. E. Aronesty, “Comparison of Sequencing Utility Programs,” *The Open Bioinform-*  
604 *atics Journal*, vol. 7, no. 1, pp. 1–8, Jan. 2013.
- 605 23. H. Lab, “FASTX Toolkit.” [Online]. Available:  
606 [http://hannonlab.cshl.edu/fastx\\_toolkit/](http://hannonlab.cshl.edu/fastx_toolkit/)
- 607 24. T. Lassmann, “TagDust2: a generic method to extract reads from sequencing data,”  
608 *BMC Bioinformatics*, vol. 16, no. 1, p. 1, Jan. 2015.
- 609 25. H. Li, B. Handsaker, A. Wysoker, T. Fennell, J. Ruan, N. Homer, G. Marth, G. R.  
610 Abecasis, R. Durbin, and 1000 Genome Project Data Processing Subgroup, “The  
611 Sequence Alignment/Map format and SAMtools,” *Bioinformatics (Oxford, Eng-*  
612 *land)*, vol. 25, no. 16, pp. 2078–2079, Aug. 2009.
- 613 26. A. Dobin and T. R. Gingeras, “Optimizing RNA-Seq Mapping with STAR,” in  
614 *Transcription Factor Regulatory Networks*. New York, NY: Springer New York,  
615 Apr. 2016, pp. 245–262.
- 616 27. R Core Team, *R: A Language and Environment for Statistical Computing*, R  
617 Foundation for Statistical Computing, Vienna, Austria, 2017. [Online]. Available:  
618 <https://www.R-project.org>
- 619 28. M. Lawrence and M. Morgan, “Scalable Genomics with R and Bioconductor,”  
620 *Statistical Science*, vol. 29, no. 2, pp. 214–226, May 2014.
- 621 29. A. Yates, W. Akanni, M. R. Amodè, D. Barrell, K. Billis, D. Carvalho-Silva,  
622 C. Cummins, P. Clapham, S. Fitzgerald, L. Gil, C. G. Girão, L. Gordon,  
623 T. Hourlier, S. E. Hunt, S. H. Janacek, N. Johnson, T. Juettemann, S. Keenan,  
624 I. Lavidas, F. J. Martin, T. Maurel, W. McLaren, D. N. Murphy, R. Nag,  
625 M. Nuhn, A. Parker, M. Patricio, M. Pignatelli, M. Rahtz, H. S. Riat,  
626 D. Sheppard, K. Taylor, A. Thormann, A. Vullo, S. P. Wilder, A. Zadissa,  
627 E. Birney, J. Harrow, M. Muffato, E. Perry, M. Ruffier, G. Spudich, S. J.  
628 Trevanion, F. Cunningham, B. L. Aken, D. R. Zerbino, and P. Flicek, “Ensembl  
629 2016,” *Nucleic Acids Research*, vol. 44, no. D1, pp. D710–D716, 2016. [Online].  
630 Available: + <http://dx.doi.org/10.1093/nar/gkv1157>
- 631 30. V. Haberle, A. R. R. Forrest, Y. Hayashizaki, P. Carninci, and B. Lenhard,  
632 “CAGEr: precise TSS data retrieval and high-resolution promoterome mining for  
633 integrative analyses.” *Nucleic Acids Research*, vol. 43, no. 8, pp. gkv054–e51, Feb.  
634 2015.
- 635 31. H. Thorvaldsdottir, J. T. Robinson, and J. P. Mesirov, “Integrative Genomics  
636 Viewer (IGV): high-performance genomics data visualization and exploration,”  
637 *Briefings in Bioinformatics* (), vol. 14, no. 2, pp. 178–192, Mar. 2013.
- 638 32. O. Tange, “Gnu parallel - the command-line power tool,” *login: The*  
639 *USENIX Magazine*, vol. 36, no. 1, pp. 42–47, Feb 2011. [Online]. Available:  
640 <http://www.gnu.org/s/parallel>

## 641 **6 Checklist of Items to be Sent to Volume Editors**

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

643 ☐ The final L<sup>A</sup>T<sub>E</sub>X source files

644 ☐ A final PDF file

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

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