

# Using RAMPAGE to identify and annotate promoters in insect genomes

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

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

<sup>2</sup>School of Informatics and Computing, Indiana University

Department of Biology and School of Informatics and Computing,  
Indiana University

212 S. Hawthorne Drive 205 Simon Hall, Bloomington, IN 47401, USA  
<http://www.brendelgroup.org>

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

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

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

## 1 Introduction

### 1.1 TSS Profiling Identifies Promoters at Genome-Scale

The promoter, defined in eukaryotes as the genomic region bound by RNA Polymerase II immediately prior to transcription initiation [?], is the site where regulatory signals unite to direct gene expression. The identification of promoter regions is a valuable step for understanding the *cis*-regulatory signals that are present in an organism, and is 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—empirical mapping of TSSs—precisely identifying sequence motifs that demarcate the promoter is unreliable. In contrast with current *in silico*

---

\* Correspondence: [rtraborn@indiana.edu](mailto:rtraborn@indiana.edu)

approaches, direct mapping of TSSs identifies the location of the core promoter. Cap Analysis of Gene Expression (CAGE) [?], 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 [?], providing the first glimpse of the global landscape of transcription initiation. At the onset of the NGS era, CAGE was coupled with massively-parallel sequencing to generate 5'-ends of mRNAs at substantially higher scale. This advance provided more extensive coverage of the expressed transcriptome, and provided increased sensitivity for quantitative measurements *i.e.* measurement of promoter activity.

## 1.2 Promoter Architecture of *Drosophila melanogaster*

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

## 1.3 Promoter Structure of Insects

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

see [?]), the authors identified 3134 mapping to 1424 genes. The authors found a preference for CA and AA dinucleotides at the TSS, and observe the major core promoter elements observed in *Drosophila*: TATA, Inr, DPE, in addition to MTE (Motif Ten Element). As in *D. melanogaster*, peaked promoters were more likely to contain TATA and Inr than broad promoters. While the taxonomic sampling of species for TSS profiling has been limited, the existing studies are sufficient to provide a general picture of insect promoter architecture. A major demarcation between the promoter architecture of insects and mammals appears to be the large fraction of mammalian promoters found in CpG islands [?]. CpG island promoters (CPIs) form the largest class of promoter in mammals [?]; by contrast, CPIs are not known to exist as a class in invertebrates.

#### 1.4 Paired-end TSS Profiling with RAMPAGE

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

## 2 Materials

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

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

## 2.1 Hardware Requirements

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

## 2.2 Software Requirements

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

## 2.3 Online Appendix

We created an online appendix to serve as a companion to this chapter. Please find the repository at [https://github.com/rtraborn/MMB\\_appendix](https://github.com/rtraborn/MMB_appendix). You can clone this appendix to your workspace using git, as follows:

```
git clone https://github.com/rtraborn/MMB_appendix.git
```

The "scripts/" folder in the Appendix 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*).

## 2.4 Installation of R packages

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

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

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

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

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

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

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

```
146 library(TSRchitect)
```

## 147 3 Methods

### 148 3.1 Retrieving the RAMPAGE sequence data from NCBI's Gene 149 Expression Omnibus (GEO)

150 To begin our analysis, we must download the RAMPAGE data to our worksta-  
 151 tion. We will utilize tools provided by the SRA Toolkit, which should already  
 152 be installed on your machine (see **Materials**). The command *fastq-dump* allows  
 153 one to directly retrieve data from the GEO database using the appropriate iden-  
 154 tifier(s). While there are 36 RAMPAGE libraries in the Batut *et al.* dataset,  
 155 we will select a subset of these to analyze here. We will compare samples from  
 156 selected embryonic (E01h-E03h) and larval (L1-L3) tissues, representing the be-  
 157 ginning and end of embryonic development. For more information about the  
 158 experiment and the available RAMPAGE libraries, please see the following link:  
 159 <https://www.ncbi.nlm.nih.gov/Traces/study/?acc=SRP011193>

160 First, let's proceed with the libraries from early embryonic tissues. Note that  
 161 since these fastq files are paired-end, we use the argument *-split-files* to generate  
 162 separate files for each read pair.

```
163 mkdir fastq_files #creating a new folder to house the downloaded files
164 cd fastq_files #moving into this directory
165 fastq-dump --split-files SRR424683
166 fastq-dump --split-files SRR424684
167 fastq-dump --split-files SRR424685
```

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

```

170 fastq-dump --split-files SRR424707
171 fastq-dump --split-files SRR424708
172 fastq-dump --split-files SRR424709

```

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

```

175 ls -l *.fastq | wc -l

```

### 176 3.2 Creating symlinks to the files

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

```

180 mkdir symlinks
181
182 #embryonic libraries
183 ln -s SRR424683_1.fastq symlinks/E01h.R1.clipped.fq
184 ln -s SRR424683_2.fastq symlinks/E01h.R2.clipped.fq
185 ln -s SRR424684_1.fastq symlinks/E02h.R1.clipped.fq
186 ln -s SRR424684_2.fastq symlinks/E02h.R2.clipped.fq
187 ln -s SRR424685_1.fastq symlinks/E03h.R1.clipped.fq
188 ln -s SRR424685_2.fastq symlinks/E03h.R2.clipped.fq
189
190 #larval libraries
191 ln -s SRR424707_1.fastq symlinks/L1.R1.clipped.fq
192 ln -s SRR424707_2.fastq symlinks/L1.R2.clipped.fq
193 ln -s SRR424708_1.fastq symlinks/L2.R1.clipped.fq
194 ln -s SRR424708_2.fastq symlinks/L2.R2.clipped.fq
195 ln -s SRR424709_1.fastq symlinks/L3.R1.clipped.fq
196 ln -s SRR424709_2.fastq symlinks/L3.R2.clipped.fq

```

### 197 3.3 Downloading genomic data from *D. melanogaster*

198 Now that we have the fastq files from the RAMPAGE libraries downloaded and  
 199 named appropriately, we now must retrieve the genome assembly and rRNA  
 200 sequences from *D. melanogaster*. The genome assembly is required for aligning  
 201 the RAMPAGE reads, and the rRNA sequences are required to filter out match-  
 202 ing reads in the sequenced RAMPAGE libraries, since our sample is intended  
 203 to contain only capped RNA transcripts. Please download the rRNA sequences  
 204 from the link we provide below. These sequences were retrieved separately from  
 205 Genbank at the NCBI database.

206     Please download the assembly from the ENSEMBL database as follows:

```

207 wget ftp://ftp.ensembl.org/pub/release-78/fasta/drosophila_melanogaster/dna/Drosophila_m
208 #uncompressing the file
209 gzip -d Drosophila_melanogaster.BDGP5.dna.toplevel.fa.gz

```

210 Please navigate to the rRNA file `texttt"Dmel_rRNA.fasta"` found in the  
 211 Appendix.

```
212 head -n 3
213 >ref|NR_133562.1| Drosophila melanogaster 28S ribosomal RNA (28SrRNA:CR45844), rRNA
214 TTATATACAACCTCAACTCATATGGGACTACCCCTGAATTAAAGCATATTAATTAGGGGAGGAAAAGAA
215 ACTAACAAGGATTTTCTTAGTAGCGGCGAGCGAAAAGAAAACAGTTCAGCACTAAGTCACTTTGTCTATA
```

### 216 3.4 Filtering and alignment of RAMPAGE reads using 217 GoRAMPAGE

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

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

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

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

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

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

```
244 less -S errScript
```

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

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

Since this step was conducted appropriately, we can proceed to the next step.

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

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

### 3.5 Promoter identification from aligned RAMPAGE libraries

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

**Setting up the Analysis** *TSRchitect*, the package we'll use for this analysis, is an R package available in the Bioconductor suite of genomics tools [?]. It makes use of existing packages and data structures within this environment, where available, to identify promoters from sequence alignments. Since you have



289 already installed *TSRchitect* and its dependencies (see section 2.3), we are set  
 290 to proceed.

291 There are two general ways one can choose to run *TSRchitect*. The first is in-  
 292 teractively *i.e.* typing the instructions directly into an R console. While this  
 293 is a perfectly acceptable way to run analyses using package, for larger jobs  
 294 it will likely be more efficient (and likely more reproducible) to run a dedi-  
 295 cated R script. We have provided a sample script "MMB\_chapter\_TSRchitect.R"  
 296 to make it easier for you to set up an R script. In the section to follow, we  
 297 will go through the output of the analysis. For further details on how to use  
 298 *TSRchitect*, please see its documentation at its Bioconductor page found here  
 299 <https://www.bioconductor.org/packages/release/bioc/html/TSRchitect.html>.  
 300

301 **Running the Analysis** To run *TSRchitect* using the batch script provided,  
 302 first provide full paths for the variables "BAMDIR" and "DmAnnot" in "MMB\_chapter\_TSRchitect.R"  
 303 using a text editor. *BAMDIR* should be a path to the subdirectory "alignments/"  
 304 in RAMPAGE output directory you specified earlier, and *DmAnnot* should be  
 305 a full path to the *D. melanogaster* gene annotation listed above. Once this is  
 306 complete, we can run the batch script from the Linux command-line as follows:

```
307 R CMD BATCH MMB_chapter_TSRchitect.R
308 #assumes variables BAMDIR and DmAnnot have already been set
309 bg #puts this job in the background
```

310 Once the job is underway, you can monitor its progress by looking at the  
 311 contents of the .Rout file (in this case, "MMB\_chapter\_TSRchitect.Rout"). The  
 312 job should complete within an hour on most systems.  
 313

314 Before we evaluate the results (which will have been written to your working  
 315 directory after running the batch script), there are some important parameters  
 316 to review. First, we must initialize the *tssObject* (which stores the information  
 317 about the experiment) appropriately.  
 318

319 Note that since the samples provided derive from related developmental  
 320 stages, we will merge them for annotation purposes using the argument *repli-*  
 321 *cateIDs*, (though they emphaize that they are not replicates). The input in this  
 322 case are BAM files (*inputType*="bam"); *TSRchitect* also accepts input in BED  
 323 format.

```
324 DmRAMPAGE <- loadTSSobj(experimentTitle = "RAMPAGE Tutorial", \
325   inputDir=BAMDIR, inputType="bam", isPairedEnd=TRUE, \
326   sampleNames=c("E1h", "E2h", "E3h", "L1", "L2", "L3"), \
327   replicateIDs=c(1,1,1,2,2,2))
```

328 A critical step in our analysis is identifying TSRs from the aligned TSS  
 329 data; to do this we use the function *determineTSR*. We have selected the job  
 330 to run on 4 cores in this example (*n.cores*=4). Please enter the number of cores

appropriate for your system. Because we want to identify TSRs from every one of the selected RAMPAGE libraries, we specify *tssSet*="all". The parameter *tagCountThreshold* was set to 25, meaning that only TSSs supported by 25 or more 5' RAMPAGE reads will be included within a TSR. Setting *writeTable* to "TRUE" means that the identified TSRs from each set will be written to the working directory.

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

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

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

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

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

DmRAMPAGE <- addAnnotationToTSR(experimentName=DmRAMPAGE,
tsrSetType="replicates", tsrSet=1, \
upstreamDist=1000, downstreamDist=200, feature="gene", \
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)
```

373 **Evaluating the results** Our analysis using *TSRchitect* is now complete. For  
 374 comparison, the example batch script we provide took just under 44 minutes to  
 375 run.

376 Your working directory should now contain the following:

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

382 Let's briefly review the files. We can quickly obtain the counts on the com-  
 383 mand line, as follows:

```
384 wc -l *.tab
385 8377 TSRset-1.tab
386 6159 TSRset-2.tab
387 4814 TSRset-3.tab
388 17924 TSRset-4.tab
389 11851 TSRset-5.tab
390 3242 TSRset-6.tab
391 13986 TSRsetCombined.tab
392 7344 TSRsetMerged-1.tab
393 12126 TSRsetMerged-2.tab
394 85823 total
```

395 We will see that we have identified between roughly 3,200 and 18,000 TSRs  
 396 within the individual RAMPAGE samples, which is attributable to the dif-  
 397 ferences in library sizes. We detect 7,344 TSRs within the early embryonic  
 398 samples ("TSRsetMerged-1.tab") and 12,126 TSRs in the late larval samples  
 399 ("TSRsetMerged-2.tab"). Within the combined samples ("TSRsetCombined.tab")  
 400 we find 13,986 TSRs, which is similar to the number reported by Hoskins *et. al.*  
 401 [?].

402  
 403 In addition to identifying the position of a given TSRs, *TSRchitect* records  
 404 other useful information about its properties. The *width* of a TSR refers the span  
 405 of the genomic region it occupies (in bp), and the *Shape Index* (SI) is measure  
 406 of the relative peakedness of the TSR. We can see an example of this in the file  
 407 "TSRsetMerged-1.txt".

408	seq	start	end	strand	nTSSs	tsrWidth	shapeIndex	featureID
409	2L.67043.67044.+			2L	67043	67044 +	270 2	1 NA
410	2L.74089.74115.+			2L	74089	74115 +	341 27	0.13 NA
411	2L.94739.94752.+			2L	94739	94752 +	1650 14	0.55 FBgn0031
412	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.

### 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.

## 4 References

### References

1. J. T. Kadonaga, "Perspectives on the RNA polymerase II core promoter." *Wiley Interdisciplinary Reviews: Developmental Biology*, vol. 1, no. 1, pp. 40–51, Jan. 2012.
2. R. Kodzius, M. Kojima, H. Nishiyori, M. Nakamura, S. Fukuda, M. Tagami, D. Sasaki, K. Imamura, C. Kai, M. Harbers, Y. Hayashizaki, and P. Carninci, "CAGE: cap analysis of gene expression." *Nature Methods*, vol. 3, no. 3, pp. 211–222, Mar. 2006.
3. P. Carninci, T. Kasukawa, S. Katayama, J. Gough, M. C. Frith, N. Maeda, R. Oyama, T. Ravasi, B. Lenhard, C. Wells, R. Kodzius, K. Shimokawa, V. B. Bajic, S. E. Brenner, S. Batalov, A. R. R. Forrest, M. Zavolan, M. J. Davis, L. G. Wilming, V. Aidinis, J. E. Allen, A. Ambesi-Impiombato, R. Apweiler, R. N. Aturaliya, T. L. Bailey, M. Bansal, L. Baxter, K. W. Beisel, T. Bersano, H. Bono, A. M. Chalk, K. P. Chiu, V. Choudhary, A. Christoffels, D. R. Clutterbuck, M. L. Crowe, E. Dalla, B. P. Dalrymple, B. de Bono, G. Della Gatta, D. di Bernardo, T. Down, P. Engstrom, M. Fagiolini, G. Faulkner, C. F. Fletcher, T. Fukushima, M. Furuno, S. Futaki, M. Gariboldi, P. Georgii-Hemming, T. R. Gingeras, T. Gojobori, R. E. Green, S. Gustincich, M. Harbers, Y. Hayashi, T. K. Hensch, N. Hirokawa, D. Hill, L. Huminiecki, M. Iacono, K. Ikeo, A. Iwama, T. Ishikawa, M. Jakt, A. Kanapin, M. Katoh, Y. Kawasaki, J. Kelso, H. Kitamura, H. Kitano, G. Kollias, S. P. T. Krishnan, A. Kruger, S. K. Kummerfeld, I. V. Kurochkin, L. F. Lareau, D. Lazarevic,

- L. Lipovich, J. Liu, S. Liuni, S. McWilliam, M. Madan Babu, M. Madera, L. Marchionni, H. Matsuda, S. Matsuzawa, H. Miki, F. Mignone, S. Miyake, K. Morris, S. Mottagui-Tabar, N. Mulder, N. Nakano, H. Nakauchi, P. Ng, R. Nilsson, S. Nishiguchi, S. Nishikawa, F. Nori, O. Ohara, Y. Okazaki, V. Orlando, K. C. Pang, W. J. Pavan, G. Pavesi, G. Pesole, N. Petrovsky, S. Piazza, J. Reed, J. F. Reid, B. Z. Ring, M. Ringwald, B. Rost, Y. Ruan, S. L. Salzberg, A. Sandelin, C. Schneider, C. Schönbach, K. Sekiguchi, C. A. M. Semple, S. Seno, L. Sessa, Y. Sheng, Y. Shibata, H. Shimada, K. Shimada, D. Silva, B. Sinclair, S. Sperling, E. Stupka, K. Sugiura, R. Sultana, Y. Takenaka, K. Taki, K. Tammoja, S. L. Tan, S. Tang, M. S. Taylor, J. Tegner, S. A. Teichmann, H. R. Ueda, E. van Nimwegen, R. Verardo, C. L. Wei, K. Yagi, H. Yamanishi, E. Zabarovsky, S. Zhu, A. Zimmer, W. Hide, C. Bult, S. M. Grimmond, R. D. Teasdale, E. T. Liu, V. Brusic, J. Quackenbush, C. Wahlestedt, J. S. Mattick, D. A. Hume, C. Kai, D. Sasaki, Y. Tomaru, S. Fukuda, M. Kanamori-Katayama, M. Suzuki, J. Aoki, T. Arakawa, J. Iida, K. Imamura, M. Itoh, T. Kato, H. Kawaji, N. Kawagashira, T. Kawashima, M. Kojima, S. Kondo, H. Konno, K. Nakano, N. Ninomiya, T. Nishio, M. Okada, C. Plessy, K. Shibata, T. Shiraki, S. Suzuki, M. Tagami, K. Waki, A. Watahiki, Y. Okamura-Oho, H. Suzuki, J. Kawai, Y. Hayashizaki, F. Consortium, R. G. E. R. Group, and G. S. G. G. N. P. C. Group, "The transcriptional landscape of the mammalian genome," *Science (New York, NY)*, vol. 309, no. 5740, pp. 1559–1563, Sep. 2005.
4. R. A. Hoskins, R. A. Hoskins, J. M. Landolin, J. M. Landolin, J. B. Brown, J. B. Brown, J. E. Sandler, J. E. Sandler, H. Takahashi, H. Takahashi, T. Lassmann, T. Lassmann, C. Yu, C. Yu, B. W. Booth, B. W. Booth, D. Zhang, D. Zhang, K. H. Wan, K. H. Wan, L. Yang, L. Yang, N. Boley, N. Boley, J. Andrews, J. Andrews, T. C. Kaufman, T. C. Kaufman, B. R. Graveley, B. R. Graveley, P. J. Bickel, P. J. Bickel, P. Carninci, J. W. Carlson, J. W. Carlson, S. E. Celniker, and S. E. Celniker, "Genome-wide analysis of promoter architecture in *Drosophila melanogaster*," *Genome Research*, vol. 21, no. 2, pp. 182–192, Feb. 2011.
  5. E. A. Rach, H.-Y. Yuan, W. H. Majoros, P. Tomancak, and U. Ohler, "Motif composition, conservation and condition-specificity of single and alternative transcription start sites in the *Drosophila* genome," *Genome Biology*, vol. 10, no. 7, p. R73, 2009.
  6. B. Lenhard, A. Sandelin, and P. Carninci, "Metazoan promoters: emerging characteristics and insights into transcriptional regulation," *Nature Reviews Genetics*, vol. 13, no. 4, pp. 233–245, Apr. 2012.
  7. T. Ni, D. L. Corcoran, E. A. Rach, S. Song, E. P. Spana, Y. Gao, U. Ohler, and J. Zhu, "A paired-end sequencing strategy to map the complex landscape of transcription initiation," *Nature Methods*, vol. 7, no. 7, pp. 521–527, Jul. 2010.
  8. U. Ohler, G.-c. Liao, H. Niemann, and G. M. Rubin, "Computational analysis of core promoters in the *Drosophila* genome," *Genome Biology*, vol. 3, no. 12, pp. research0087.1–0087.12, 2002.
  9. R. T. Raborn, K. Spitze, V. P. Brendel, and M. Lynch, "Promoter Architecture and Sex-Specific Gene Expression in *Daphnia pulex*," *Genetics*, vol. 204, no. 2, pp. 593–612, Aug. 2016.
  10. C. Nepal, Y. Hadzhiev, C. Previti, V. Haberle, N. Li, H. Takahashi, A. M. M. Suzuki, Y. Sheng, R. F. Abdelhamid, S. Anand, J. Gehrig, A. Akalin, C. E. M. Kockx, A. A. J. van der Sloot, W. F. J. van IJcken, O. Armant, S. Rastegar, C. Watson, U. Strahle, E. Stupka, P. Carninci, B. Lenhard, and F. Muller, "Dynamic regulation of the transcription initiation landscape at single nucleotide res-

- olution during vertebrate embryogenesis,” *Genome Research*, vol. 23, no. 11, pp. 1938–1950, Nov. 2013.
11. P. Carninci, A. Sandelin, B. Lenhard, S. Katayama, K. Shimokawa, J. Ponjavic, C. A. M. Semple, M. S. Taylor, P. G. Engström, M. C. Frith, A. R. R. Forrest, W. B. Alkema, S. L. Tan, C. Plessy, R. Kodzius, T. Ravasi, T. Kasukawa, S. Fukuda, M. Kanamori-Katayama, Y. Kitazume, H. Kawaji, C. Kai, M. Nakamura, H. Konno, K. Nakano, S. Mottagui-Tabar, P. Arner, A. Chesi, S. Gustincich, F. Persichetti, H. Suzuki, S. M. Grimmond, C. A. Wells, V. Orlando, C. Wahlestedt, E. T. Liu, M. Harbers, J. Kawai, V. B. Bajic, D. A. Hume, and Y. Hayashizaki, “Genome-wide analysis of mammalian promoter architecture and evolution,” *Nature Genetics*, vol. 38, no. 6, pp. 626–635, Apr. 2006.
  12. S. Mwangi, G. Attardo, Y. Suzuki, S. Aksoy, and A. Christoffels, “TSS seq based core promoter architecture in blood feeding Tsetse fly (*Glossina morsitans morsitans*) vector of Trypanosomiasis,” *BMC Genomics*, vol. 16, no. 1, p. 722, Sep. 2015.
  13. K. Tsuchihara, Y. Suzuki, H. Wakaguri, T. Irie, K. Tanimoto, S.-i. Hashimoto, K. Matsushima, J. Mizushima-Sugano, R. Yamashita, K. Nakai, D. Bentley, H. Esumi, and S. Sugano, “Massive transcriptional start site analysis of human genes in hypoxia cells,” *Nucleic Acids Research*, vol. 37, no. 7, pp. 2249–2263, Apr. 2009.
  14. N. Cvetesic and B. Lenhard, “Core promoters across the genome,” *Nature Biotechnology*, vol. 35, no. 2, pp. 123–124, Feb. 2017.
  15. T. Lassmann, “TagDust2: a generic method to extract reads from sequencing data,” *BMC Bioinformatics*, vol. 16, no. 1, p. 1, Jan. 2015.
  16. A. Dobin and T. R. Gingeras, “Optimizing RNA-Seq Mapping with STAR,” in *Transcription Factor Regulatory Networks*. New York, NY: Springer New York, Apr. 2016, pp. 245–262.
  17. V. Haberle, “CAGER: an R package for CAGE (Cap Analysis of Gene Expression) data analysis and promoterome mining,” 2013.
  18. M. Lawrence and M. Morgan, “Scalable Genomics with R and Bioconductor,” *Statistical Science*, vol. 29, no. 2, pp. 214–226, May 2014.

## 5 Checklist of Items to be Sent to Volume Editors

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

- ☐ The final L<sup>A</sup>T<sub>E</sub>X 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.