

# 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 *Drosophila melanogaster* [1], and, shortly thereafter, to reveal widespread transposon-driven alternative promoter usage in *D. melanogaster* [2].

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 [3] and TSRchitect [4], 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 [5], 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—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 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 [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. Hoskins [1] 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 [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, while broad promoters contained an enrichment of less-well characterized motifs, including *Ohler6* and *Ohler7* [11]. The existence of two promoter classes appears to be conserved among metazoans, and has been reported (using TSS profiling methodologies) 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: 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*) [15]. Using TSS-seq (specifically Oligo-capping; for details see [16]), 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 [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 (RNA Annotation and Mapping of Promoters for the Analysis of Gene Expression) [2, 18]. 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 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* 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 model system. 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.* [2]. 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 of doing modern bioinformatics, including 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/>) or CyVerse (<http://www.cyverse.org/>) [19]. The former is a well-known pay-per-use solution, while the latter is an NSF-funded resource that makes compute allocations freely available to the public.

## 2.1 Hardware

1. x86-64 compatible processors
2. At least 8GB RAM
3. 30GB+ 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*).

1. Bioconductor (v. 3.5 and up) [20] (<http://bioconductor.org/>)
2. FASTX-Toolkit [21] ([http://hannonlab.cshl.edu/fastx\\_toolkit/Index.html](http://hannonlab.cshl.edu/fastx_toolkit/Index.html))
3. GoRAMPAGE [3] (<https://github.com/brendelGroup/GoRAMPAGE>)
4. R (v. 3.4 and up) [22] (<https://www.r-project.org/>)
5. Samtools [23] (<http://www.htslib.org/doc/samtools.html>)
6. SRA Toolkit [24] (<https://www.ncbi.nlm.nih.gov/sra/docs/toolkitsoft/>)
7. STAR [25] (<https://github.com/alexdobin/STAR>)
8. TagDust2 [26] (<https://sourceforge.net/projects/tagdust/>)
9. TSRchitect [4] (<http://bioconductor.org/packages/release/bioc/html/TSRchitect.html>)

## 2.4 Online Appendix

We created an online appendix to serve as a companion to this chapter, which contains both scripts and select files to assist you in completing this tutorial. Please find the repository at [https://github.com/rtraborn/MMB\\_appendix](https://github.com/rtraborn/MMB_appendix) (*see Note 2*).

## 125 2.5 Installation of R packages

126 For installation of the software listed above, please follow the instructions pro-  
 127 vided by each respective package. Part of our analysis will require the use of R  
 128 packages found in the Bioconductor suite [20]. To install Bioconductor, please  
 129 type the following from an R console:

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

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

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

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

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

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

```
147 library(TSRchitect)
```

## 148 3 Methods

### 149 3.1 Retrieving the RAMPAGE sequence data from NCBI's Gene 150 Expression Omnibus (GEO)

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

161  
 162 First, let's proceed with the libraries from early embryonic tissues (see **See**  
 163 **Note 3**).

```

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

```

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

```

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

```

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

```

176 ls -l *.fastq | wc -l

```

### 177 3.2 Creating symlinks to the files

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

```

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

```

### 198 3.3 Downloading genomic data from *D. melanogaster*

199 Now that we have the fastq files from the RAMPAGE libraries downloaded and  
 200 named appropriately, we now must retrieve the genome assembly and rRNA  
 201 sequences from *D. melanogaster*. The genome assembly is required for aligning

the RAMPAGE reads, and the rRNA sequences are required to filter out matching reads in the sequenced RAMPAGE libraries, since our sample is intended to contain only capped RNA transcripts. Please download the rRNA sequences from the link we provide below. These sequences were retrieved separately from Genbank at the NCBI database.

Please download the assembly from the ENSEMBL database as follows:

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

Please navigate to the rRNA file "Dmel\_rRNA.fasta" found in the Appendix.

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

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

At this stage we are ready to commence with the rRNA filtering and alignment of the RAMPAGE libraries. We will use GoRAMPAGE, a tool we developed, to perform these tasks in a concerted workflow. GoRAMPAGE runs TagDust [26] to remove rRNA and low-complexity reads, and uses STAR [25] to align RAMPAGE (or other paired-end) reads to a given genome assembly.

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

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

**Setting up the GoRAMPAGE job** Now, once this is complete, please copy the contents of the "symlinks" directory that you created earlier (*i.e.* all of the \*.fq files) into the "clipped/" directory. Please refer to the script "GoRAMPAGE\_script\_MMB.sh" and (using a text editor) provide the appropriate paths to the genome assembly, output directory (see above) and rRNA sequences (*see Note 4*). GoRAMPAGE jobs can optionally be run in parallel (*see Note 5*). The script can be executed as follows:

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

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

```
less -S errScript
```

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

**Inspecting the rRNA filtering results** To evaluate the results from Step 3 (rRNA filtering), please navigate to the top level of the "output" directory and open the file "LOGFILES". You'll see the recorded progress of the program Tagdust and a record of the results. We notice that (for the L3h library) 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 GoRAMPAGE output folder will now contain 6 .bam files, each representing the distinct RAMPAGE 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 [27], was utilized to perform TSS identification as part of the FANTOM5 efforts. We will use *TSRchitect* in



279 this demonstration, since it was specifically designed to analyze paired-end TSS  
 280 profiling datasets, and also because it is more flexible with respect to model  
 281 system (*i.e.* it does not require a corresponding *BSGenome* package). The latter  
 282 feature will be helpful when analyzing the non-*D. melanogaster* TSS profiling  
 283 datasets that we expect to be generated in the near future.

284 **Setting up the Analysis** *TSRchitect*, the package we'll use for this analy-  
 285 sis, is an R package available in the Bioconductor suite of genomics tools [20].  
 286 It makes use of existing packages and data structures within this environment,  
 287 where available, to identify promoters from sequence alignments. Since you have  
 288 already installed *TSRchitect* and its dependencies (see section 2.3), we are set  
 289 to proceed.

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

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

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

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

312

313 **Reviewing the *TSRchitect* script** Before we evaluate the results (which will  
 314 have been written to your working directory after running the batch script),  
 315 there are some important aspects of the analysis to review. We discuss these for  
 316 informational purposes only; it will not necessary to perform these commands  
 317 separate from the batch script provided. First, we must initialize the *tssObject*  
 318 (which stores the information about the experiment) appropriately (*see Note 6*).  
 319

320 The input in this case are BAM files (*inputType*="bam"); *TSRchitect* also  
 321 accepts input in BED format.

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

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

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

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

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

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

```
348 DmRAMPAGE <- importAnnotationExternal(experimentName=DmRAMPAGE, \
349   fileType="gff3", annotFile=DmAnnot)
350
351 DmRAMPAGE <- addAnnotationToTSR(experimentName=DmRAMPAGE, \
352   tsrSetType="replicates", tsrSet=1, \
353   upstreamDist=1000, downstreamDist=200, feature="gene", \
354   featureColumnID="ID", writeTable=TRUE)
```

355 Now we have generated a set of identified TSSs, TSRs from all 6 RAMPAGE  
 356 libraries, and have associated the identified TSRs with annotated genes. Next, we  
 357 will merge the libraries into two samples according to condition: early embryonic  
 358 (E1h, E2h, E3h) and late larval (L1, L2, L3) using the information we provided  
 359 when we initialized the *tssObject* at the start of this section. After merging, we  
 360 identify promoters i) within the merged samples and ii) within the entire dataset

361 combined, and associate with the *D. melanogaster* gene annotation as described  
 362 previously (not shown).

```
363 #merging the sample data into two groups
364 DmRAMPAGE <- mergeSampleData(DmRAMPAGE)
365
366 # ... identifying TSRs from the merged samples:
367 DmRAMPAGE <- determineTSR(experimentName=DmRAMPAGE, \
368   n.cores=4, tsrSetType="merged", \
369   tssSet="all", tagCountThreshold=40, \
370   clustDist=20, writeTable=TRUE)
```

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

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

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

```
380 wc -l *.tab
381 8377 TSRset-1.tab
382 6159 TSRset-2.tab
383 4814 TSRset-3.tab
384 17924 TSRset-4.tab
385 11851 TSRset-5.tab
386 3242 TSRset-6.tab
387 13986 TSRsetCombined.tab
388 7344 TSRsetMerged-1.tab
389 12126 TSRsetMerged-2.tab
390 85823 total
```

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

398  
 399 In addition to identifying the position of a given TSRs, *TSRchitect* records  
 400 other useful information about its properties. The *width* of a TSR refers the span  
 401 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.

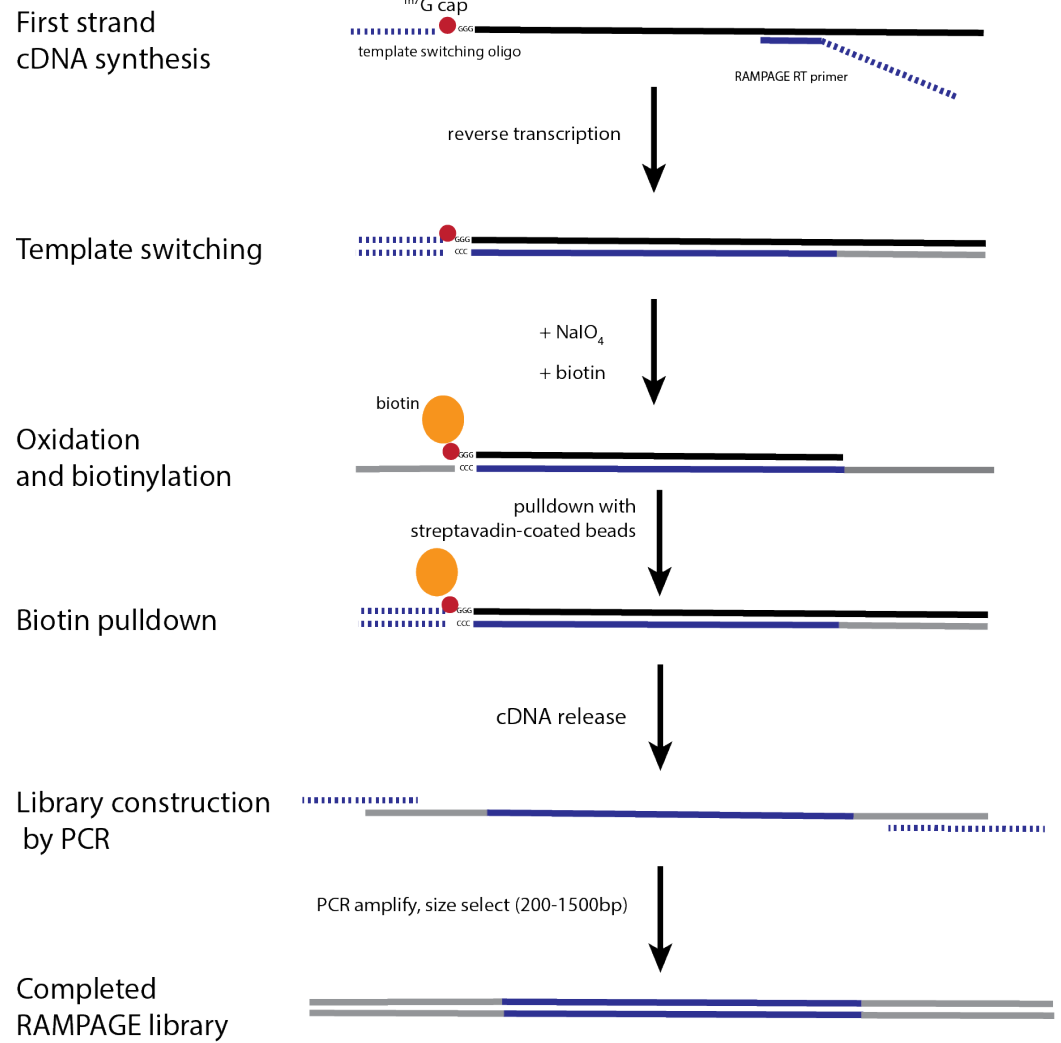
### 3.7 Figures

## 4 Notes

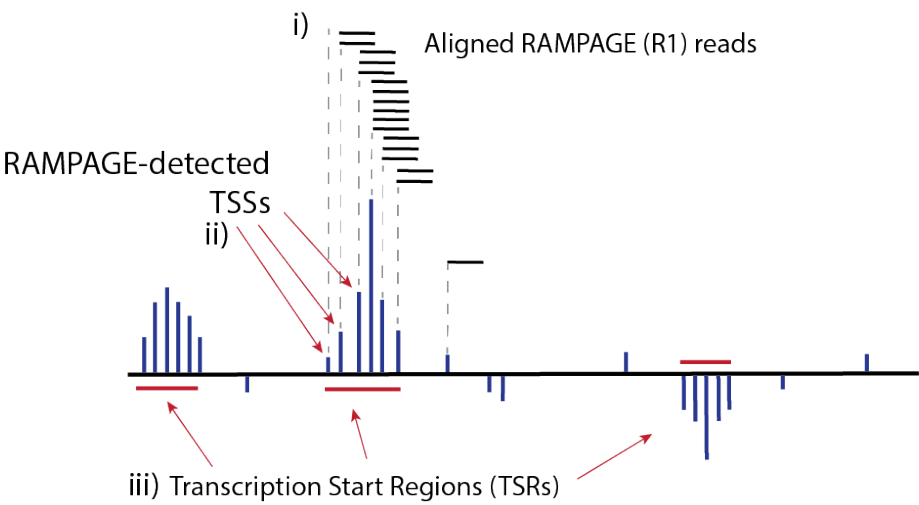
- Please consult the GoRAMPAGE documentation found here: <https://github.com/BrendelGroup/GoRAMPAGE>. Installation instructions for the prerequisites of GoRAMPAGE (which includes some of the items listed) are found at the following link: <https://github.com/BrendelGroup/GoRAMPAGE#installation>.
- You can clone this appendix to your workspace on the command line 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*).
- Since these fastq files are paired-end, we use the argument *-split-files* to generate separate files for each read pair.
- 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.
- For parallel execution, GoRAMPAGE uses the Linux package *GNU parallel* [28]. Please see the GoRAMPAGE documentation for more information.
- 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).



**Fig. 1.** Test caption for figure 1



**Fig. 2.** Test caption for figure 2

**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-Impimbato, R. Apweiler, R. N. Aturaliya, T. L. Bailey, M. Bansal, L. Baxter, K. W. Beisel, T. Bersano, H. Bono, A. M. Chalk, K. P. Chiu, V. Choudhary, A. Christoffels, D. R. Clutterbuck, M. L. Crowe, E. Dalla, B. P. Dalrymple, B. de Bono, G. Della Gatta, D. di Bernardo, T. Down, P. Engstrom, M. Fagiolini, G. Faulkner, C. F. Fletcher, T. Fukushima, M. Furuno, S. Futaki, M. Gariboldi, P. Georgii-Hemming, T. R. Gingeras, T. Gojobori, R. E. Green, S. Gustincich, M. Harbers, Y. Hayashi, T. K. Hensch, N. Hirokawa, D. Hill, L. Huminiecki, M. Iacono, K. Ikeo, A. Iwama, T. Ishikawa, M. Jakt, A. Kanapin, M. Katoh, Y. Kawasaki, J. Kelso, H. Kitamura, H. Kitano, G. Kollias, S. P. T. Krishnan, A. Kruger, S. K. Kummerfeld, I. V. Kurochkin, L. F. Lareau, D. Lazarevic, L. Lipovich, J. Liu, S. Liuni, S. McWilliam, M. Madan Babu, M. Madera, L. Marchionni, H. Matsuda, S. Matsuzawa, H. Miki, F. Mignone, S. Miyake, K. Morris, S. Mottagui-Tabar, N. Mulder, N. Nakano, H. Nakauchi, P. Ng, R. Nilsson, S. Nishiguchi, S. Nishikawa, F. Nori, O. Ohara, Y. Okazaki, V. Orlando, K. C. Pang, W. J. Pavan, G. Pavesi, G. Pesole, N. Petrovsky, S. Piazza, J. Reed, J. F. Reid, B. Z. Ring, M. Ringwald, B. Rost, Y. Ruan, S. L. Salzberg, A. Sandelin, C. Schneider, C. Schönbach, K. Sekiguchi, C. A. M. Semple, S. Seno, L. Sessa, Y. Sheng, Y. Shibata, H. Shimada, K. Shimada, D. Silva, B. Sinclair, S. Sperling, E. Stupka, K. Sugiura, R. Sultana, Y. Takenaka, K. Taki, K. Tammoja, S. L. Tan, S. Tang, M. S. Taylor, J. Tegner, S. A. Teichmann, H. R. Ueda, E. van Nimwegen, R. Verardo, C. L. Wei, K. Yagi, H. Yamanishi, E. Zabarovsky, S. Zhu, A. Zimmer, W. Hide, C. Bult, S. M. Grimmond, R. D. Teasdale, E. T. Liu, V. Brusic, J. Quackenbush, C. Wahlestedt, J. S. Mattick, D. A. Hume, C. Kai, D. Sasaki, Y. Tomaru, S. Fukuda, M. Kanamori-Katayama, M. Suzuki, J. Aoki, T. Arakawa, J. Iida, K. Imamura, M. Itoh, T. Kato, H. Kawaji, N. Kawagashira, T. Kawashima, M. Kojima, S. Kondo, H. Konno, K. Nakano, N. Ninomiya, T. Nishio, M. Okada, C. Plessy, K. Shibata, T. Shiraki, S. Suzuki, M. Tagami, K. Waki, A. Watahiki, Y. Okamura-Oho, H. Suzuki, J. Kawai, Y. Hayashizaki, F. Consortium, R. G. E. R. Group, and G. S. G. 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.
8. 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.

- 512 9. B. Lenhard, A. Sandelin, and P. Carninci, "Metazoan promoters: emerging char-  
513 acteristics and insights into transcriptional regulation." *Nature Reviews Genetics*,  
514 vol. 13, no. 4, pp. 233–245, Apr. 2012.
- 515 10. T. Ni, D. L. Corcoran, E. A. Rach, S. Song, E. P. Spana, Y. Gao, U. Ohler,  
516 and J. Zhu, "A paired-end sequencing strategy to map the complex landscape of  
517 transcription initiation." *Nature Methods*, vol. 7, no. 7, pp. 521–527, Jul. 2010.
- 518 11. U. Ohler, G.-c. Liao, H. Niemann, and G. M. Rubin, "Computational analysis of  
519 core promoters in the *Drosophila* genome." *Genome Biology*, vol. 3, no. 12, pp.  
520 research0087.1–0087.12, 2002.
- 521 12. R. T. Raborn, K. Spitze, V. P. Brendel, and M. Lynch, "Promoter Architecture  
522 and Sex-Specific Gene Expression in *Daphnia pulex*." *Genetics*, vol. 204, no. 2, pp.  
523 593–612, Aug. 2016.
- 524 13. C. Nepal, Y. Hadzhiev, C. Previti, V. Haberle, N. Li, H. Takahashi, A. M. M.  
525 Suzuki, Y. Sheng, R. F. Abdelhamid, S. Anand, J. Gehrig, A. Akalin, C. E. M.  
526 Kockx, A. A. J. van der Sloot, W. F. J. van IJcken, O. Armant, S. Rastegar,  
527 C. Watson, U. Strahle, E. Stupka, P. Carninci, B. Lenhard, and F. Muller, "Dy-  
528 namic regulation of the transcription initiation landscape at single nucleotide res-  
529 olution during vertebrate embryogenesis," *Genome Research*, vol. 23, no. 11, pp.  
530 1938–1950, Nov. 2013.
- 531 14. P. Carninci, A. Sandelin, B. Lenhard, S. Katayama, K. Shimokawa, J. Ponjavic,  
532 C. A. M. Semple, M. S. Taylor, P. G. Engström, M. C. Frith, A. R. R. For-  
533 rest, W. B. Alkema, S. L. Tan, C. Plessy, R. Kodzius, T. Ravasi, T. Kasukawa,  
534 S. Fukuda, M. Kanamori-Katayama, Y. Kitazume, H. Kawaji, C. Kai, M. Naka-  
535 mura, H. Konno, K. Nakano, S. Mottagui-Tabar, P. Arner, A. Chesi, S. Gustincich,  
536 F. Persichetti, H. Suzuki, S. M. Grimmond, C. A. Wells, V. Orlando, C. Wahlest-  
537 edt, E. T. Liu, M. Harbers, J. Kawai, V. B. Bajic, D. A. Hume, and Y. Hayashizaki,  
538 "Genome-wide analysis of mammalian promoter architecture and evolution," *Na-  
539 ture Genetics*, vol. 38, no. 6, pp. 626–635, Apr. 2006.
- 540 15. S. Mwangi, G. Attardo, Y. Suzuki, S. Aksoy, and A. Christoffels, "TSS seq based  
541 core promoter architecture in blood feeding Tsetse fly (*Glossina morsitans mor-  
542 sitans*) vector of Trypanosomiasis," *BMC Genomics*, vol. 16, no. 1, p. 722, Sep.  
543 2015.
- 544 16. K. Tsuchihara, Y. Suzuki, H. Wakaguri, T. Irie, K. Tanimoto, S.-i. Hashimoto,  
545 K. Matsushima, J. Mizushima-Sugano, R. Yamashita, K. Nakai, D. Bentley, H. Es-  
546 umi, and S. Sugano, "Massive transcriptional start site analysis of human genes in  
547 hypoxia cells," *Nucleic Acids Research*, vol. 37, no. 7, pp. 2249–2263, Apr. 2009.
- 548 17. N. Cvetesic and B. Lenhard, "Core promoters across the genome," *Nature Biotech-  
549 nology*, vol. 35, no. 2, pp. 123–124, Feb. 2017.
- 550 18. P. J. Batut and T. R. Gingeras, "RAMPAGE: Promoter Activity Profiling by  
551 Paired-End Sequencing of 5'-Complete cDNAs." in *Current Protocols in Molecular  
552 Biology*. Current protocols in molecular biology / edited by Frederick M Ausubel  
553 [et al], 2013, pp. 25B.11.1–25B.11.16.
- 554 19. N. Merchant, E. Lyons, S. Goff, M. Vaughn, D. Ware, D. Micklos, and P. Antin,  
555 "The iPlant Collaborative: Cyberinfrastructure for Enabling Data to Discovery for  
556 the Life Sciences." *PLoS Biology*, vol. 14, no. 1, p. e1002342, Jan. 2016.
- 557 20. M. Lawrence and M. Morgan, "Scalable Genomics with R and Bioconductor,"  
558 *Statistical Science*, vol. 29, no. 2, pp. 214–226, May 2014.
- 559 21. H. Lab, "FASTX Toolkit." [Online]. Available:  
560 [http://hannonlab.cshl.edu/fastx\\_toolkit/](http://hannonlab.cshl.edu/fastx_toolkit/)



- 561 22. R. Core Team, *R: A Language and Environment for Statistical Computing*, R  
562 Foundation for Statistical Computing, Vienna, Austria, 2017. [Online]. Available:  
563 <https://www.R-project.org>
- 564 23. H. Li, B. Handsaker, A. Wysoker, T. Fennell, J. Ruan, N. Homer, G. Marth, G. R.  
565 Abecasis, R. Durbin, and 1000 Genome Project Data Processing Subgroup, “The  
566 Sequence Alignment/Map format and SAMtools,” *Bioinformatics (Oxford, Eng-*  
567 *land)*, vol. 25, no. 16, pp. 2078–2079, Aug. 2009.
- 568 24. R. Leinonen, H. Sugawara, M. Shumway, and International Nucleotide Sequence  
569 Database Collaboration, “The sequence read archive.” *Nucleic Acids Research*,  
570 vol. 39, no. Database issue, pp. D19–21, Jan. 2011.
- 571 25. A. Dobin and T. R. Gingeras, “Optimizing RNA-Seq Mapping with STAR,” in  
572 *Transcription Factor Regulatory Networks*. New York, NY: Springer New York,  
573 Apr. 2016, pp. 245–262.
- 574 26. T. Lassmann, “TagDust2: a generic method to extract reads from sequencing data,”  
575 *BMC Bioinformatics*, vol. 16, no. 1, p. 1, Jan. 2015.
- 576 27. V. Haberle, A. R. R. Forrest, Y. Hayashizaki, P. Carninci, and B. Lenhard,  
577 “CAGER: precise TSS data retrieval and high-resolution promoterome mining for  
578 integrative analyses.” *Nucleic Acids Research*, vol. 43, no. 8, pp. gkv054–e51, Feb.  
579 2015.
- 580 28. O. Tange, “Gnu parallel - the command-line power tool,” *login: The*  
581 *USENIX Magazine*, vol. 36, no. 1, pp. 42–47, Feb 2011. [Online]. Available:  
582 <http://www.gnu.org/s/parallel>

## 583 6 Checklist of Items to be Sent to Volume Editors

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

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