

Using RAMPAGE to identify and annotate promoters in insect genomes

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

¹Department of Biology, Indiana University

²School 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

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 only other insect species for which TSS profiling has been applied is the

53 Tsetse fly (*Glossina morsitans morsitans*) [15]. Using TSS-seq (specifically Oligo-
 54 capping; for details see [16]), the authors identified 3134 mapping to 1424 genes.
 55 The authors found a preference for CA and AA dinucleotides at the TSS, and
 56 observe the major core promoter elements observed in *Drosophila*: TATA, Inr,
 57 DPE, in addition to MTE (Motif Ten Element). As in *D. melanogaster*, peaked
 58 promoters were more likely to contain TATA and Inr than broad promoters.
 59 While the taxonomic sampling of species for TSS profiling has been limited, the
 60 existing studies are sufficient to provide a general picture of insect promoter ar-
 61 chitecture. A major demarcation between the promoter architecture of insects
 62 and mammals appears to be the large fraction of mammalian promoters found
 63 in CpG islands [15]. CpG island promoters (CPIs) form the largest class of pro-
 64 moter in mammals [17]; by contrast, CPIs are not known to exist as a class in
 65 invertebrates.

66 1.4 Paired-end TSS Profiling with RAMPAGE

67 The most recent major methodological advance in TSS Profiling is RAMPAGE
 68 (RNA Annotation and Mapping of Promoters for the Analysis of Gene Expres-
 69 sion) [2, 18]. RAMPAGE is a protocol for 5'-cDNA sequencing that combines cap
 70 trapping and template-switching with paired-end sequence information. A key
 71 advantage of generating paired-end sequence is transcript connectivity, which
 72 provides a direct link between a given 5'-end and its associated mRNA molecule
 73 [2]. Because short or spurious RNAs are found within the transcriptome, tran-
 74 script connectivity allows the TSSs (and thus promoters) of full-length mRNAs
 75 to be unambiguously identified, which benefits genome annotation and improves
 76 interpretation of transcript species.
 77 Batut and colleagues [2] generated libraries from total RNA isolated from 36
 78 stages across the life cycle of *D. melanogaster* providing a comprehensive gene
 79 expression and promoter atlas for fruit fly and in the process demonstrating the
 80 utility of RAMPAGE. RAMPAGE is currently being applied as part of the latest
 81 iteration of ENCODE to identify promoters in human, but as of this writing it
 82 has not been applied to any non-*Drosophila* insect model system. In anticipation
 83 of the future application of TSS profiling into other insect model systems here
 84 we provide a documented protocol for the computational processing RAMPAGE
 85 data, using selected libraries from Batut *et al.* [2]. This method will consist of two
 86 parts: first, we will process, filter and align the sequenced RAMPAGE libraries to
 87 the *D. melanogaster* genome. Second, we will identify TSSs and promoters from
 88 the aligned sequences and associate them with coding regions. In closing, we will
 89 consider further applications of this data and discuss the utility of reproducible
 90 workflows in bioinformatic analysis.

91 2 Materials

92 The analyses described herein require a workstation capable of doing modern
 93 bioinformatics, including a reasonably-appointed laptop. An intermediate un-
 94 derstanding 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*).

Sequence retrieval

1. SRA Toolkit [20] (<https://www.ncbi.nlm.nih.gov/sra/docs/toolkitsoft/>)

GoRAMPAGE

1. GoRAMPAGE [3] (<https://github.com/brendelGroup/GoRAMPAGE>)
2. fastq-multx [21] (<https://github.com/brwnj/fastq-multx>)
3. FASTX-Toolkit [22] (http://hannonlab.cshl.edu/fastx_toolkit/Index.html)
4. TagDust2 [23] (<https://sourceforge.net/projects/tagdust/>)
5. Samtools [24] (<http://www.htslib.org/doc/samtools.html>)
6. STAR [25] (<https://github.com/alexdobin/STAR>)

TSRchitect

1. R (v. 3.4 and up) [26] (<https://www.r-project.org/>)
2. Bioconductor (v. 3.5 and up) [27] (<http://bioconductor.org/>)
3. TSRchitect [4] (<http://bioconductor.org/packages/release/bioc/html/TSRchitect.html>)
4. Various R package dependencies (see **Methods**)

128 2.4 Online Appendix

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

133 2.5 Installation of R packages

134 For installation of the software listed above, please follow the instructions pro-
 135 vided by each respective package. Part of our analysis will require the use of R
 136 packages found in the Bioconductor suite [27]. To install Bioconductor, please
 137 type the following from an R console:

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

140 We will use the R package *TSRchitect* to identify promoters from aligned RAM-
 141 PAGE libraries. Prior to running the analysis, it will be necessary to install a
 142 series of prerequisite packages to *TSRchitect* from Bioconductor. Please install
 143 these packages as follows (as before, from an R console):

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

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

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

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

```
155 library(TSRchitect) #installing TSRchitect
```

156 3 Methods

157 3.1 Retrieving the RAMPAGE sequence data from NCBI

158 To begin our analysis, we must download the RAMPAGE data to our worksta-
 159 tion. We will utilize tools provided by the SRA Toolkit, which should already
 160 be installed on your machine (see **Materials**). The command *fastq-dump* al-
 161 lows one to directly retrieve data from the GEO database using the appropriate
 162 identifier(s). While there are 36 RAMPAGE libraries in the Batut *et al.* pa-
 163 per, we will select a subset of these to analyze here. We will compare samples

164 from selected embryonic (E01h-E03h) and larval (L1-L3) tissues, representing
 165 the beginning and end of embryonic development. For more information about
 166 the experiment and the available RAMPAGE libraries, please see the following
 167 link: <https://www.ncbi.nlm.nih.gov/Traces/study/?acc=SRP011193>.

168

169 First, let's proceed with downloading the libraries from early embryonic tis-
 170 sues (see **See Note 3**). We will make a new folder (entitled "**fastq_files/**")
 171 to house these files.

```
172 mkdir fastq_files
173 cd fastq_files
174
175 fastq-dump --split-files SRR424683
176 fastq-dump --split-files SRR424684
177 fastq-dump --split-files SRR424685
```

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

```
179
180 fastq-dump --split-files SRR424707
181 fastq-dump --split-files SRR424708
182 fastq-dump --split-files SRR424709
183
184
185
```

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

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

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
 218 sequences from *D. melanogaster*. The genome assembly is required for aligning
 219 the RAMPAGE reads, and the rRNA sequences are required to filter out match-
 220 ing reads in the sequenced RAMPAGE libraries, since our sample is intended
 221 to contain only capped RNA transcripts. Please download the rRNA sequences
 222 from the link we provide below. These sequences were retrieved separately from
 223 Genbank at the NCBI database.

224
 225 To retrieve the genome assembly from the ENSEMBL database, please do the
 226 following:

```

227 mkdir genome
228 cd genome
229 wget ftp://ftp.ensembl.org/pub/release-78/fasta/drosophila_melanogaster/dna/Drosophila_m
230 #uncompressing the file
231 gzip -d Drosophila_melanogaster.BDGP5.dna.toplevel.fa.gz
232 cd ..

```

233 Please navigate to the rRNA file "Dmel_rRNA.fasta" found in the Appendix.

```

234 head -n 3
235 >ref|NR_133562.1| Drosophila melanogaster 28S ribosomal RNA (28SrRNA:CR45844), rRNA
236 TTATATACAACCTCAACTCATATGGGACTACCCCTGAATTTAAGCATATTAATTAGGGGAGGAAAAGAA
237 ACTAACAAGGATTTTCTTAGTAGCGGCGAGCGAAAAGAAAACAGTTCAGCACTAAGTCACTTTGTCTATA

```

238 3.4 Filtering and alignment of RAMPAGE reads using 239 GoRAMPAGE

240 At this stage we are ready to commence with the rRNA filtering and alignment
 241 of the RAMPAGE libraries. We will use GoRAMPAGE, a tool we developed,
 242 to perform these tasks in a concerted workflow. GoRAMPAGE runs TagDust

243 [23] to remove rRNA and low-complexity reads, and uses STAR [25] to align
 244 RAMPAGE (or other paired-end) reads to a given genome assembly.

245 **Setting up the GoRAMPAGE job.** Please refer to the script "GoRAMPAGE_script_MMB.sh"
 246 and (using a text editor) provide the appropriate paths to the genome assembly,
 247 output directory (see above) and rRNA sequences (*see Note 4*). GoRAMPAGE
 248 jobs can optionally be run in parallel (*see Note 5*). The script can be executed
 249 as follows:

```
250 #vi GoRAMPAGE_script_MMB.sh #updating with a text editor
251 ./GoRAMPAGE_script_MMB.sh
```

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

```
255 less -S errScript
```

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

259 **Inspecting the rRNA filtering results.** To evaluate the results from Step
 260 3 (rRNA filtering), please navigate to the top level of the "output" directory
 261 and open the file "LOGFILES". You'll see the recorded progress of the program
 262 Tagdust and a record of the results. We notice that (for the L3h library) 1046448
 263 of reads (78.1%) were "extracted", meaning that slightly more than 20% of
 264 reads were removed because of matches with ribosomal sequences. The removed
 265 reads from all libraries are found in the "dusted_discard" directory, and the
 266 extracted reads are found in the current directory. Due to their sheer abundance
 267 within cells, ribosomal RNA sequences are an inevitable contaminant within TSS
 268 profiling libraries. For analysis purposes, it is important that these sequences be
 269 removed, which is what has been completed here.
 270 Since this step was conducted appropriately, we can proceed to the next step.

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

282 our purposes.

283

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

286 3.5 Promoter identification from aligned RAMPAGE libraries

287 We can now use the prepared alignment files to identify TSSs and promoters from
288 the selected RAMPAGE libraries. There are currently several tools available
289 for this purpose. *CAGEr*, developed by Haberle [28], was utilized to perform
290 TSS identification as part of the FANTOM5 efforts. We will use *TSRchitect* in
291 this demonstration, since it was specifically designed to analyze paired-end TSS
292 profiling datasets, and also because it is more flexible with respect to model
293 system (*i.e.* it does not require a corresponding *BSGenome* package). The latter
294 feature will be helpful when analyzing the non-*D. melanogaster* TSS profiling
295 datasets that we expect to be generated in the near future.

296 **Setting up the Analysis.** *TSRchitect*, the package we'll use for this analy-
297 sis, is an R package available in the Bioconductor suite of genomics tools [27].
298 It makes use of existing packages and data structures within this environment,
299 where available, to identify promoters from sequence alignments. Since you have
300 already installed *TSRchitect* and its dependencies (see section 2.3), we are set
301 to proceed.

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

312 **Running the Analysis.** To run *TSRchitect* using the batch script, provide
313 full paths for the variables "`BAMDIR`" and "`DmAnnot`" in the script provided
314 (*see Note 6*). *BAMDIR* should be a path to the subdirectory "`alignments/`" in
315 RAMPAGE output directory you specified earlier, and *DmAnnot* should be a
316 full path to the *D. melanogaster* gene annotation listed above.
317 Once this is complete, we can run the batch script from the Linux command-line
318 as follows:

```
319 R CMD BATCH MMB_chapter_TSRchitect.R
320 #assumes variables BAMDIR and DmAnnot have already been set
321 bg #puts this job in the background
```

322 Once the job is underway, you can monitor its progress by looking at the con-
 323 tents of the .Rout file (in this case, "MMB_chapter_TSRchitect.Rout"). The job
 324 should complete within an hour on most systems.

325

326 **Reviewing the *TSRchitect* script.** Before we evaluate the results (which
 327 will have been written to your working directory after running the batch script),
 328 there are some important aspects of the analysis to review. We discuss these for
 329 informational purposes only; it will not necessary to perform these commands
 330 separate from the batch script provided. First, we must initialize the *tssObject*
 331 (which stores the information about the experiment) appropriately (*see Note 7*).

332

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

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

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

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

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

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

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

```

362 DmRAMPAGE <- importAnnotationExternal(experimentName=DmRAMPAGE, \
363   fileType="gff3", annotFile=DmAnnot)
364
365 DmRAMPAGE <- addAnnotationToTSR(experimentName=DmRAMPAGE, \
366   tsrSetType="replicates", tsrSet=1, \
367   upstreamDist=1000, downstreamDist=200, feature="gene", \
368   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).

```

377 #merging the sample data into two groups
378 DmRAMPAGE <- mergeSampleData(DmRAMPAGE)
379
380 # ... identifying TSRs from the merged samples:
381 DmRAMPAGE <- determineTSR(experimentName=DmRAMPAGE, \
382   n.cores=4, tsrSetType="merged", \
383   tssSet="all", tagCountThreshold=40, \
384   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. We can quickly obtain the counts on the command line, as follows:

```

394 wc -l *.tab
395 8377 TSRset-1.tab
396 6159 TSRset-2.tab
397 4814 TSRset-3.tab
398 17924 TSRset-4.tab
399 11851 TSRset-5.tab
400 3242 TSRset-6.tab
401 13986 TSRsetCombined.tab
402 7344 TSRsetMerged-1.tab

```

403 12126 TSRsetMerged-2.tab
 404 85823 total

405 We will see that we have identified between roughly 3,200 and 18,000 TSRs
 406 within the individual RAMPAGE samples, which is attributable to the dif-
 407 ferences in library sizes. We detect 7,344 TSRs within the early embryonic
 408 samples ("TSRsetMerged-1.tab") and 12,126 TSRs in the late larval samples
 409 ("TSRsetMerged-2.tab"). Within the combined samples ("TSRsetCombined.tab")
 410 we find 13,986 TSRs, which is similar to the number reported by Hoskins *et. al.*
 411 [1].

412
 413 In addition to identifying the position of a given TSRs, *TSRchitect* records other
 414 useful information about its properties. The *width* of a TSR refers the span of
 415 the genomic region it occupies (in bp), and the *Shape Index* (SI) is measure of
 416 the relative peakedness of the TSR. We can see an example of this in the file
 417 "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

423 3.6 Summary

424 The workflow provided here is intended to serve as a useful entry point for the
 425 analysis of TSS profiling data in insects. On the computational side, we have
 426 provided an open source set of tools so that the uninitiated genome scientist
 427 can begin to analyze RAMPAGE (or other forms of TSS profiling data) quickly.
 428 While the analysis centered on *D. melanogaster* via the use of public datasets,
 429 it is anticipated that this will assist groups who may be interested in performing
 430 TSS profiling in their preferred insect model system.

431 The application of TSS profiling technology across a more representative sample
 432 of insect diversity will improve our understanding of the positions and general
 433 structure *cis*-regulatory regions in this phylum.

434 3.7 Figures

435 4 Notes

- 436 1. Please consult the GoRAMPAGE documentation found here:
 437 <https://github.com/BrendelGroup/GoRAMPAGE>.
 438 Installation instructions for the prerequisites of GoRAMPAGE (which in-
 439 cludes some of the items listed) are found at the following link:
 440 <https://github.com/BrendelGroup/GoRAMPAGE/tree/master/src>.
 441 2. You can clone this appendix to your workspace on the command line using
 442 git, as follows:

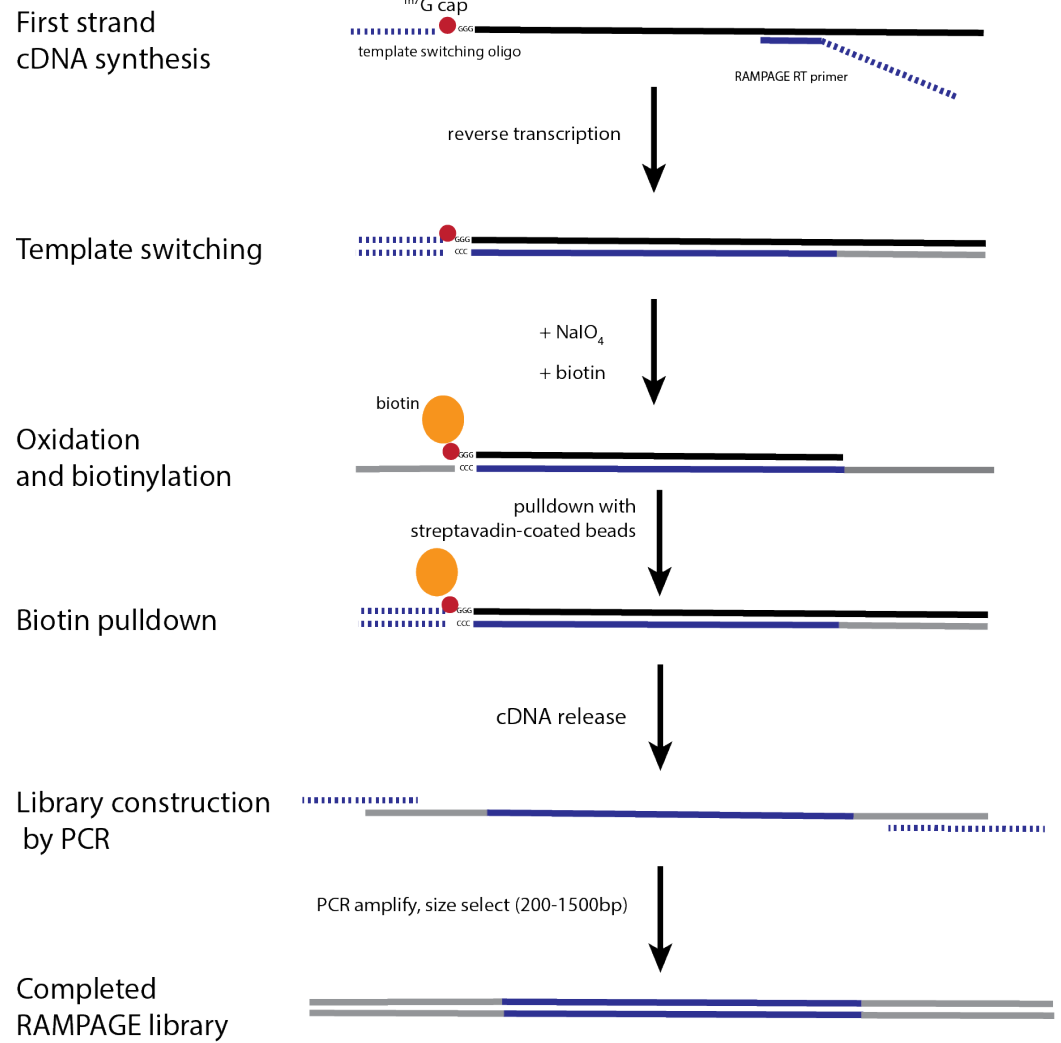


Fig. 1. Test caption for figure 1

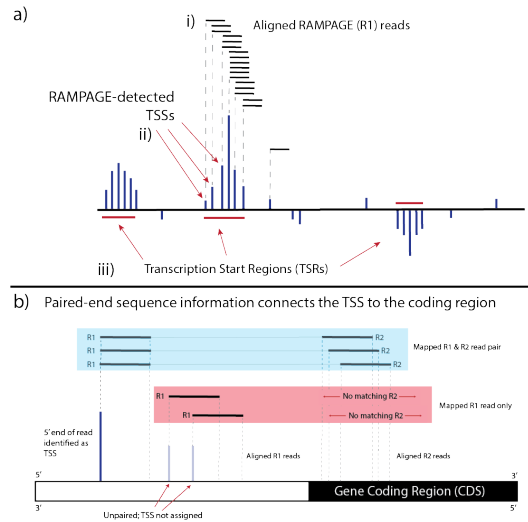


Fig. 2. Test caption for figure 2

```
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*).

3. Since these fastq files are paired-end, we use the argument `-split-files` to generate separate files for each read pair.
4. 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.
5. For parallel execution, GoRAMPAGE uses the Linux package *GNU parallel* [29]. Please see the GoRAMPAGE documentation for more information.
6. To do this, please edit the batch script `TSRchitect_script_MMB.R` with a text editor of your choice.
7. 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).

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.

References

References

1. R. A. Hoskins, R. A. Hoskins, J. M. Landolin, J. M. Landolin, J. B. Brown, J. B. Brown, J. E. Sandler, J. E. Sandler, H. Takahashi, H. Takahashi, T. Lassmann, T. Lassmann, C. Yu, C. Yu, B. W. Booth, B. W. Booth, D. Zhang, D. Zhang, K. H. Wan, K. H. Wan, L. Yang, L. Yang, N. Boley, N. Boley, J. Andrews, J. Andrews, T. C. Kaufman, T. C. Kaufman, B. R. Graveley, B. R. Graveley, P. J. Bickel, P. J. Bickel, P. Carninci, J. W. Carlson, J. W. Carlson, S. E. Celniker, and S. E. Celniker, "Genome-wide analysis of promoter architecture in *Drosophila melanogaster*." *Genome Research*, vol. 21, no. 2, pp. 182–192, Feb. 2011.
2. P. J. Batut, A. Dobin, C. Plessy, P. Carninci, and T. R. Gingeras, "High-fidelity promoter profiling reveals widespread alternative promoter usage and transposon-driven developmental gene expression." *Genome Research*, Aug. 2012.
3. V. P. Brendel and R. T. Raborn, "Gorampage- a workflow for promoter detection by 5'-read mapping," <https://github.com/brendelGroup/GoRAMPAGE>, 2016.
4. R. T. Raborn and V. Brendel, *TSRchitect: Promoter identification from large-scale TSS profiling data*, 2017, r Bioconductor package version 1.0.0. [Online]. Available: <http://bioconductor.org/packages/release/bioc/html/TSRchitect.html>
5. J. T. Kadonaga, "Perspectives on the RNA polymerase II core promoter." *Wiley Interdisciplinary Reviews: Developmental Biology*, vol. 1, no. 1, pp. 40–51, Jan. 2012.
6. R. Kodzius, M. Kojima, H. Nishiyori, M. Nakamura, S. Fukuda, M. Tagami, D. Sasaki, K. Imamura, C. Kai, M. Harbers, Y. Hayashizaki, and P. Carninci, "CAGE: cap analysis of gene expression." *Nature Methods*, vol. 3, no. 3, pp. 211–222, Mar. 2006.
7. P. Carninci, T. Kasukawa, S. Katayama, J. Gough, M. C. Frith, N. Maeda, R. Oyama, T. Ravasi, B. Lenhard, C. Wells, R. Kodzius, K. Shimokawa, V. B. Bajic, S. E. Brenner, S. Batalov, A. R. R. Forrest, M. Zavolan, M. J. Davis, L. G. Wilming, V. Aidinis, J. E. Allen, A. Ambesi-Impiombato, R. Apweiler, R. N. Aturaliya, T. L. Bailey, M. Bansal, L. Baxter, K. W. Beisel, T. Bersano, H. Bono, A. M. Chalk, K. P. Chiu, V. Choudhary, A. Christoffels, D. R. Clutterbuck, M. L. Crowe, E. Dalla, B. P. Dalrymple, B. de Bono, G. Della Gatta, D. di Bernardo, T. Down, P. Engstrom, M. Fagiolini, G. Faulkner, C. F. Fletcher, T. Fukushima, M. Furuno, S. Futaki, M. Gariboldi, P. Georgii-Hemming, T. R. Gingeras, T. Gojobori, R. E. Green, S. Gustincich, M. Harbers, Y. Hayashi, T. K. Hensch, N. Hirokawa, D. Hill, L. Huminiecki, M. Iacono, K. Ikeo, A. Iwama, T. Ishikawa, M. Jakt, A. Kanapin, M. Katoh, Y. Kawasawa, 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.

- 510 Reid, B. Z. Ring, M. Ringwald, B. Rost, Y. Ruan, S. L. Salzberg, A. Sandelin,
511 C. Schneider, C. Schönbach, K. Sekiguchi, C. A. M. Semple, S. Seno, L. Sessa,
512 Y. Sheng, Y. Shibata, H. Shimada, K. Shimada, D. Silva, B. Sinclair, S. Sperling,
513 E. Stupka, K. Sugiura, R. Sultana, Y. Takenaka, K. Taki, K. Tammoja, S. L. Tan,
514 S. Tang, M. S. Taylor, J. Tegner, S. A. Teichmann, H. R. Ueda, E. van Nimwegen,
515 R. Verardo, C. L. Wei, K. Yagi, H. Yamanishi, E. Zabarovsky, S. Zhu, A. Zimmer,
516 W. Hide, C. Bult, S. M. Grimmond, R. D. Teasdale, E. T. Liu, V. Brusic,
517 J. Quackenbush, C. Wahlestedt, J. S. Mattick, D. A. Hume, C. Kai, D. Sasaki,
518 Y. Tomaru, S. Fukuda, M. Kanamori-Katayama, M. Suzuki, J. Aoki, T. Arakawa,
519 J. Iida, K. Imamura, M. Itoh, T. Kato, H. Kawaji, N. Kawagashira, T. Kawashima,
520 M. Kojima, S. Kondo, H. Konno, K. Nakano, N. Ninomiya, T. Nishio, M. Okada,
521 C. Plessy, K. Shibata, T. Shiraki, S. Suzuki, M. Tagami, K. Waki, A. Watahiki,
522 Y. Okamura-Oho, H. Suzuki, J. Kawai, Y. Hayashizaki, F. Consortium, R. G. E. R.
523 Group, and G. S. G. G. N. P. C. Group, "The transcriptional landscape of the mam-
524 malian genome," *Science (New York, NY)*, vol. 309, no. 5740, pp. 1559–1563, Sep.
525 2005.
- 526 8. E. A. Rach, H.-Y. Yuan, W. H. Majoros, P. Tomancak, and U. Ohler, "Motif
527 composition, conservation and condition-specificity of single and alternative tran-
528 scription start sites in the *Drosophila* genome." *Genome Biology*, vol. 10, no. 7, p.
529 R73, 2009.
- 530 9. B. Lenhard, A. Sandelin, and P. Carninci, "Metazoan promoters: emerging char-
531 acteristics and insights into transcriptional regulation." *Nature Reviews Genetics*,
532 vol. 13, no. 4, pp. 233–245, Apr. 2012.
- 533 10. T. Ni, D. L. Corcoran, E. A. Rach, S. Song, E. P. Spana, Y. Gao, U. Ohler,
534 and J. Zhu, "A paired-end sequencing strategy to map the complex landscape of
535 transcription initiation." *Nature Methods*, vol. 7, no. 7, pp. 521–527, Jul. 2010.
- 536 11. U. Ohler, G.-c. Liao, H. Niemann, and G. M. Rubin, "Computational analysis of
537 core promoters in the *Drosophila* genome." *Genome Biology*, vol. 3, no. 12, pp.
538 research0087.1–0087.12, 2002.
- 539 12. R. T. Raborn, K. Spitze, V. P. Brendel, and M. Lynch, "Promoter Architecture
540 and Sex-Specific Gene Expression in *Daphnia pulex*." *Genetics*, vol. 204, no. 2, pp.
541 593–612, Aug. 2016.
- 542 13. C. Nepal, Y. Hadzhiev, C. Previti, V. Haberle, N. Li, H. Takahashi, A. M. M.
543 Suzuki, Y. Sheng, R. F. Abdelhamid, S. Anand, J. Gehrig, A. Akalin, C. E. M.
544 Kockx, A. A. J. van der Sloot, W. F. J. van IJcken, O. Armant, S. Rastegar,
545 C. Watson, U. Strahle, E. Stupka, P. Carninci, B. Lenhard, and F. Muller, "Dy-
546 namic regulation of the transcription initiation landscape at single nucleotide res-
547 olution during vertebrate embryogenesis," *Genome Research*, vol. 23, no. 11, pp.
548 1938–1950, Nov. 2013.
- 549 14. P. Carninci, A. Sandelin, B. Lenhard, S. Katayama, K. Shimokawa, J. Ponjavic,
550 C. A. M. Semple, M. S. Taylor, P. G. Engström, M. C. Frith, A. R. R. For-
551 rest, W. B. Alkema, S. L. Tan, C. Plessy, R. Kodzius, T. Ravasi, T. Kasukawa,
552 S. Fukuda, M. Kanamori-Katayama, Y. Kitazume, H. Kawaji, C. Kai, M. Naka-
553 mura, H. Konno, K. Nakano, S. Mottagui-Tabar, P. Arner, A. Chesi, S. Gustincich,
554 F. Persichetti, H. Suzuki, S. M. Grimmond, C. A. Wells, V. Orlando, C. Wahle-
555 stedt, E. T. Liu, M. Harbers, J. Kawai, V. B. Bajic, D. A. Hume, and Y. Hayashizaki,
556 "Genome-wide analysis of mammalian promoter architecture and evolution," *Nat-
557 ure Genetics*, vol. 38, no. 6, pp. 626–635, Apr. 2006.
- 558 15. S. Mwangi, G. Attardo, Y. Suzuki, S. Aksoy, and A. Christoffels, "TSS seq based
559 core promoter architecture in blood feeding Tsetse fly (*Glossina morsitans mor-*

- sitans) vector of Trypanosomiasis,” *BMC Genomics*, vol. 16, no. 1, p. 722, Sep. 2015.
16. K. Tsuchihara, Y. Suzuki, H. Wakaguri, T. Irie, K. Tanimoto, S.-i. Hashimoto, K. Matsushima, J. Mizushima-Sugano, R. Yamashita, K. Nakai, D. Bentley, H. Es-umi, and S. Sugano, “Massive transcriptional start site analysis of human genes in hypoxia cells,” *Nucleic Acids Research*, vol. 37, no. 7, pp. 2249–2263, Apr. 2009.
 17. N. Cvetesic and B. Lenhard, “Core promoters across the genome,” *Nature Biotech-nology*, vol. 35, no. 2, pp. 123–124, Feb. 2017.
 18. P. J. Batut and T. R. Gingeras, “RAMPAGE: Promoter Activity Profiling by Paired-End Sequencing of 5’-Complete cDNAs.” in *Current Protocols in Molecular Biology*. Current protocols in molecular biology / edited by Frederick M Ausubel [et al], 2013, pp. 25B.11.1–25B.11.16.
 19. N. Merchant, E. Lyons, S. Goff, M. Vaughn, D. Ware, D. Micklos, and P. Antin, “The iPlant Collaborative: Cyberinfrastructure for Enabling Data to Discovery for the Life Sciences.” *PLoS Biology*, vol. 14, no. 1, p. e1002342, Jan. 2016.
 20. R. Leinonen, H. Sugawara, M. Shumway, and International Nucleotide Sequence Database Collaboration, “The sequence read archive.” *Nucleic Acids Research*, vol. 39, no. Database issue, pp. D19–21, Jan. 2011.
 21. E. Aronesty, “Comparison of Sequencing Utility Programs,” *The Open Bioinform-atics Journal*, vol. 7, no. 1, pp. 1–8, Jan. 2013.
 22. H. Lab, “FASTX Toolkit.” [Online]. Available: http://hannonlab.cshl.edu/fastx_toolkit/
 23. T. Lassmann, “TagDust2: a generic method to extract reads from sequencing data,” *BMC Bioinformatics*, vol. 16, no. 1, p. 1, Jan. 2015.
 24. H. Li, B. Handsaker, A. Wysoker, T. Fennell, J. Ruan, N. Homer, G. Marth, G. R. Abecasis, R. Durbin, and 1000 Genome Project Data Processing Subgroup, “The Sequence Alignment/Map format and SAMtools,” *Bioinformatics (Oxford, Eng-land)*, vol. 25, no. 16, pp. 2078–2079, Aug. 2009.
 25. 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.
 26. R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2017. [Online]. Available: <https://www.R-project.org>
 27. M. Lawrence and M. Morgan, “Scalable Genomics with R and Bioconductor,” *Statistical Science*, vol. 29, no. 2, pp. 214–226, May 2014.
 28. V. Haberle, A. R. R. Forrest, Y. Hayashizaki, P. Carninci, and B. Lenhard, “CAGER: precise TSS data retrieval and high-resolution promoterome mining for integrative analyses.” *Nucleic Acids Research*, vol. 43, no. 8, pp. gkv054–e51, Feb. 2015.
 29. O. Tange, “Gnu parallel - the command-line power tool,” *login: The USENIX Magazine*, vol. 36, no. 1, pp. 42–47, Feb 2011. [Online]. Available: <http://www.gnu.org/s/parallel>

6 Checklist of Items to be Sent to Volume Editors

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

- ☐ The final L^AT_EX source files

- 606 ☐ A final PDF file
- 607 ☐ A copyright form, signed by one author on behalf of all of the authors of the
608 paper.
- 609 ☐ A readme giving the name and email address of the corresponding author.