

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 *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 [1] and TSRchitect [2], 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 [3], 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*

* Correspondence: rtraborn@indiana.edu

12 *silico* approaches, direct mapping of TSSs identifies the location of the core pro-
 13 moter. Cap Analysis of Gene Expression (CAGE) [4], one of the first methods
 14 devised to identify 5'-ends of mRNAs at large-scale, involves selective capture
 15 of 5'-capped transcripts, first-strand reverse-transcription and ligation of a short
 16 oligonucleotide (CAGE tag).
 17 CAGE was initially utilized by the FANTOM (Functional Annotation of the
 18 Mammalian Genome) consortium to identify promoter architecture in human
 19 and mouse [5], providing the first glimpse of the global landscape of transcrip-
 20 tion initiation. At the onset of the NGS era, CAGE was coupled with massively-
 21 parallel sequencing to generate 5'-ends of mRNAs at substantially higher scale.
 22 This advance provided more extensive coverage of the expressed transcriptome,
 23 and provided increased sensitivity for quantitative measurements *i.e.* measure-
 24 ment of promoter activity.

25 1.2 Promoter Architecture of *Drosophila melanogaster*

26 Hoskins and colleagues [6] performed CAGE in *D. melanogaster* as part of the
 27 modENCODE consortium, identifying promoters at large-scale and character-
 28 izing the promoter architecture of an insect genome for the first time. Hoskins
 29 [6] indicated that TSS distributions at *Drosophila* promoters exhibit a range
 30 of shapes that can be generally grouped into two major classifications: *peaked*
 31 and *broad*. Peaked promoters have a single, major TSS position occupying a
 32 narrow genomic region, whereas broad promoters lack a single, major TSS and
 33 contain TSSs across a wider region [7][8]. The authors also showed a strong asso-
 34 ciation between promoter class and motif composition (consistent with previous
 35 findings [7, 9]). Peaked promoters were associated with positionally-enriched *cis*-
 36 regulatory motifs including TATA, Initiator (Inr) and DPE, while broad promot-
 37 ers contained an enrichment of less-well characterized motifs, including *Ohler6*
 38 and *Ohler7* [10]. The existence of two promoter classes appears to be conserved
 39 among metazoans, and has been reported (using TSS profiling methodologies) in
 40 insects, cladocerans [11], fish [12] and mammals [13, 8].

41 1.3 Promoter Structure of Insects

42 Beyond *D. melanogaster*, few investigations have utilized TSS profiling in insect
 43 genomes. As a consequence, what is known about promoter architecture in in-
 44 sects is largely restricted to the *Drosophila* genus. As part of the modENCODE
 45 effort, CAGE was performed in multiple tissues and developmental stages of the
 46 *Drosophila pseudoobscura*. TSSs were found to be highly similar between species:
 47 more than 80% of TSSs (81%) of aligned, CAGE-identified TSSs from *D. pseu-*
 48 *doobscura* were positioned within 20nt of their counterparts in *D. melanogaster*.
 49 An enrichment of the CA dinucleotide was detected at the TSS ($[-1, +1]$), and
 50 the motifs corresponding to TATA, Inr and DPE were positioned at the same
 51 locations relative to the TSS in both species. The one other insect species for
 52 which TSS profiling has been applied is the Tsetse fly (*Glossina morsitans morsi-*
 53 *tans*) [14]. Using TSS-seq (specifically Oligo-capping; for details on this method

see [15]), 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 [14]. CpG island promoters (CPIs) form the largest class of promoter in mammals [16]; 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 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

1. Bioconductor (v. 3.5 and up) [17] (<http://bioconductor.org/>)
2. FASTX-Toolkit [18] (http://hannonlab.cshl.edu/fastx_toolkit/index.html)
3. GoRAMPAGE [1] (<https://github.com/brendelGroup/GoRAMPAGE>)
4. R (v. 3.4 and up) [19] (<https://www.r-project.org/>)
5. Samtools [20] (<http://www.htslib.org/doc/samtools.html>)
6. SRA Toolkit [21] (<https://www.ncbi.nlm.nih.gov/sra/docs/toolkitsoft/>)
7. STAR [22] (<https://github.com/alexdobin/STAR>)
8. TagDust2 [23] (<https://sourceforge.net/projects/tagdust/>)
9. TSRchitect [2] (<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 (see **Note 1**).

2.5 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()
```

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

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

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

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

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

```
144 library(TSRchitect)
```

145 3 Methods

146 3.1 Retrieving the RAMPAGE sequence data from NCBI's Gene 147 Expression Omnibus (GEO)

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

158
 159 First, let's proceed with the libraries from early embryonic tissues (see **See**
 160 **Note 2**).

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

166 We continue by downloading the RAMPAGE libraries from late embryonic
 167 tissues:

```

168 fastq-dump --split-files SRR424707
169 fastq-dump --split-files SRR424708
170 fastq-dump --split-files SRR424709

```

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

```

173 ls -l *.fastq | wc -l

```

3.2 Creating symlinks to the files

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

```

178 mkdir symlinks
179
180 #embryonic libraries
181 ln -s SRR424683_1.fastq symlinks/E01h.R1.clipped.fq
182 ln -s SRR424683_2.fastq symlinks/E01h.R2.clipped.fq
183 ln -s SRR424684_1.fastq symlinks/E02h.R1.clipped.fq
184 ln -s SRR424684_2.fastq symlinks/E02h.R2.clipped.fq
185 ln -s SRR424685_1.fastq symlinks/E03h.R1.clipped.fq
186 ln -s SRR424685_2.fastq symlinks/E03h.R2.clipped.fq
187
188 #larval libraries
189 ln -s SRR424707_1.fastq symlinks/L1.R1.clipped.fq
190 ln -s SRR424707_2.fastq symlinks/L1.R2.clipped.fq
191 ln -s SRR424708_1.fastq symlinks/L2.R1.clipped.fq
192 ln -s SRR424708_2.fastq symlinks/L2.R2.clipped.fq
193 ln -s SRR424709_1.fastq symlinks/L3.R1.clipped.fq
194 ln -s SRR424709_2.fastq symlinks/L3.R2.clipped.fq

```

3.3 Downloading genomic data from *D. melanogaster*

Now that we have the fastq files from the RAMPAGE libraries downloaded and named appropriately, we now must retrieve the genome assembly and rRNA 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:

```

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

```

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

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

213 3.4 Filtering and alignment of RAMPAGE reads using 214 GoRAMPAGE

215 At this stage we are ready to commence with the rRNA filtering and alignment
216 of the RAMPAGE libraries. We will use GoRAMPAGE, a tool we developed,
217 to perform these tasks in a concerted workflow. GoRAMPAGE runs TagDust
218 [23] to remove rRNA and low-complexity reads, and uses STAR [22] to align
219 RAMPAGE (or other paired-end) reads to a given genome assembly.

220

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

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

228 **Setting up the GoRAMPAGE job** Now, once this is complete, please
229 copy the contents of the "symlinks" directory that you created earlier (*i.e.*
230 all of the *.fq files) into the "clipped/" directory. Please refer to the script
231 "GoRAMPAGE_script_MMB.sh" and (using a text editor) provide the appropriate
232 paths to the genome assembly, output directory (see above) and rRNA sequences
233 (*see Note 3*). The script can be executed as follows:

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

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

```
239 less -S errScript
```

240 Should the run fail before completion, any associated error messages will be
241 printed to the errScript file. Once the job is complete, you should see the message
242 "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 [24], 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 [17]. It makes use of existing packages and data structures within this environment, where available, to identify promoters from sequence alignments. Since you have

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

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

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

```
302 R CMD BATCH MMB_chapter_TSRchitect.R
303 #assumes variables BAMDIR and DmAnnot have already been set
304 bg #puts this job in the background
```

305 Once the job is underway, you can monitor its progress by looking at the
306 contents of the `.Rout` file (in this case, "`MMB_chapter_TSRchitect.Rout`"). The
307 job should complete within an hour on most systems.
308

309 Before we evaluate the results (which will have been written to your working
310 directory after running the batch script), there are some important parameters
311 to review. First, we must initialize the *tssObject* (which stores the information
312 about the experiment) appropriately (see **Note 4**).
313

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

```
316 DmRAMPAGE <- loadTSSobj(experimentTitle = "RAMPAGE Tutorial", \
317   inputDir=BAMDIR, inputType="bam", isPairedEnd=TRUE, \
318   sampleNames=c("E1h", "E2h", "E3h", "L1", "L2", "L3"), \
319   replicateIDs=c(1,1,1,2,2,2))
```

320 A critical step in our analysis is identifying TSRs from the aligned TSS
321 data; to do this we use the function *determineTSR*. We have selected the job
322 to run on 4 cores in this example (*n.cores*=4). Please enter the number of cores
323 appropriate for your system. Because we want to identify TSRs from every one
324 of the selected RAMPAGE libraries, we specify *tssSet*="all". The parameter
325 *tagCountThreshold* was set to 25, meaning that only TSSs supported by 25 or

326 more 5' RAMPAGE reads will be included within a TSR. Setting *writeTable* to
 327 "TRUE" means that the identified TSRs from each set will be written to the
 328 working directory.

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

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

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

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

```
342 DmRAMPAGE <- importAnnotationExternal(experimentName=DmRAMPAGE, \
343 fileType="gff3", annotFile=DmAnnot)
344
345 DmRAMPAGE <- addAnnotationToTSR(experimentName=DmRAMPAGE, \
346 tsrSetType="replicates", tsrSet=1, \
347 upstreamDist=1000, downstreamDist=200, feature="gene", \
348 featureColumnID="ID", writeTable=TRUE)
```

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

```
357 #merging the sample data into two groups
358 DmRAMPAGE <- mergeSampleData(DmRAMPAGE)
359
360 # ... identifying TSRs from the merged samples:
361 DmRAMPAGE <- determineTSR(experimentName=DmRAMPAGE, \
362 n.cores=4, tsrSetType="merged", \
363 tssSet="all", tagCountThreshold=40, \
364 clustDist=20, writeTable=TRUE)
```

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

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:

```

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

```

We will see that we have identified between roughly 3,200 and 18,000 TSRs within the individual RAMPAGE samples, which is attributable to the differences in library sizes. We detect 7,344 TSRs within the early embryonic samples ("TSRsetMerged-1.tab") and 12,126 TSRs in the late larval samples ("TSRsetMerged-2.tab"). Within the combined samples ("TSRsetCombined.tab") we find 13,986 TSRs, which is similar to the number reported by Hoskins *et. al.* [6].

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

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

405 3.6 Summary

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

416 3.7 Figures

417 4 Notes

- 418 1. You can clone this appendix to your workspace on the command line using
 419 git, as follows:

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

- 421 The "scripts/" folder in the Appendix contains code for you to run the two
 422 major workflows described in this chapter. The "additional_files/" folder
 423 contains the following files which are necessary for the analysis: i) a fasta file
 424 containing ribosomal RNA sequences for *D. melanogaster* (*Dmel_rRNA.fasta*)
 425 and ii) a gene annotation for *D. melanogaster* (*Drosophila_melanogaster.BDGP5.78.gff*).
 426 2. Since these fastq files are paired-end, we use the argument *-split-files* to
 427 generate separate files for each read pair.
 428 3. If you are running this on a cluster with a job scheduler you'll need to add
 429 the necessary headers to the top of the script and submit the job in the
 430 appropriate manner.
 431 4. Because the samples provided derive from related developmental stages, we
 432 will merge them for annotation purposes using the argument *replicateIDs*,
 433 (though it must be emphasized that they are not replicates).

434 Acknowledgments

435 The authors would like to thank Philippe Batut for generous technical as-
 436 sistance with the RAMPAGE protocol, and to Nathan Keith for his help
 437 establishing the protocol in our laboratory.

438 Disclosure Declaration

439 The authors declare that they have no competing interests.

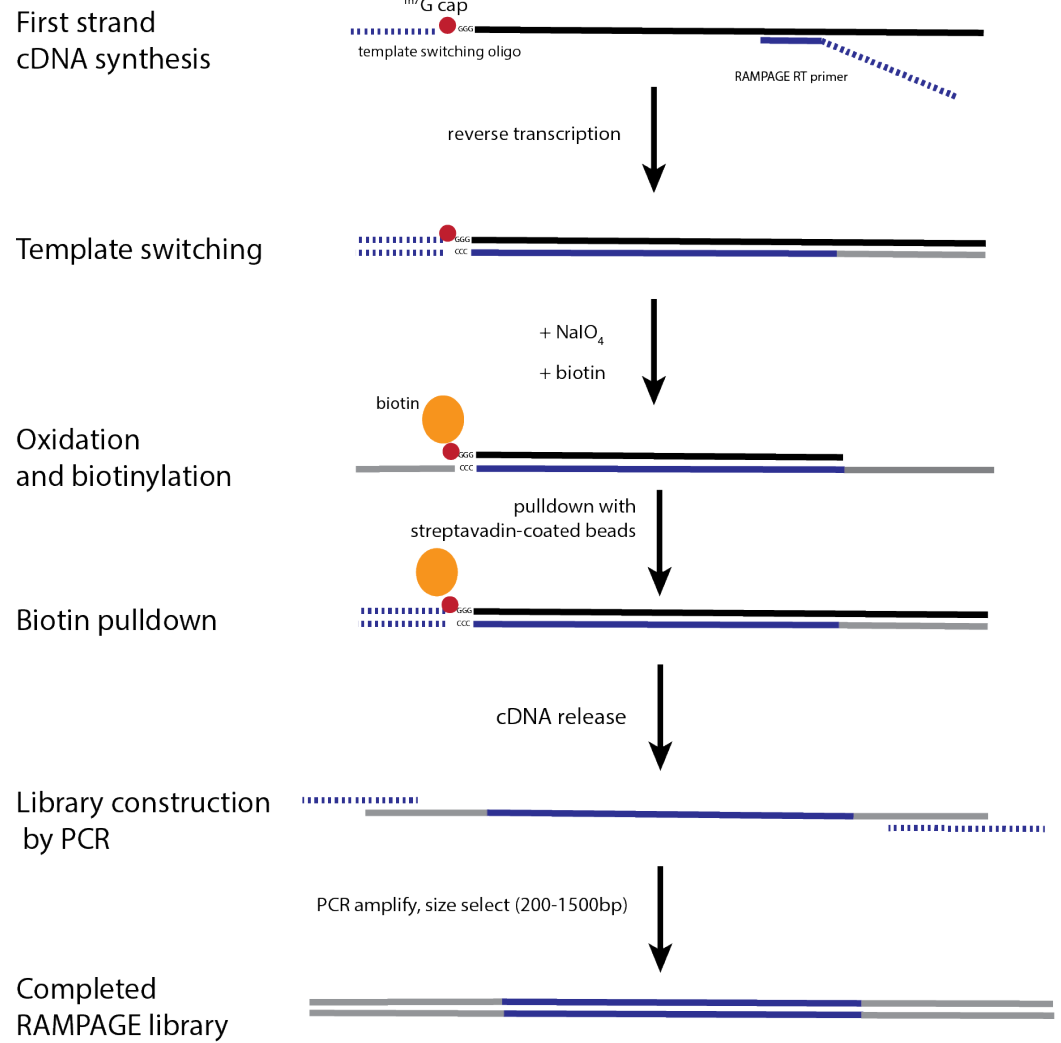


Fig. 1. Test caption for figure 1

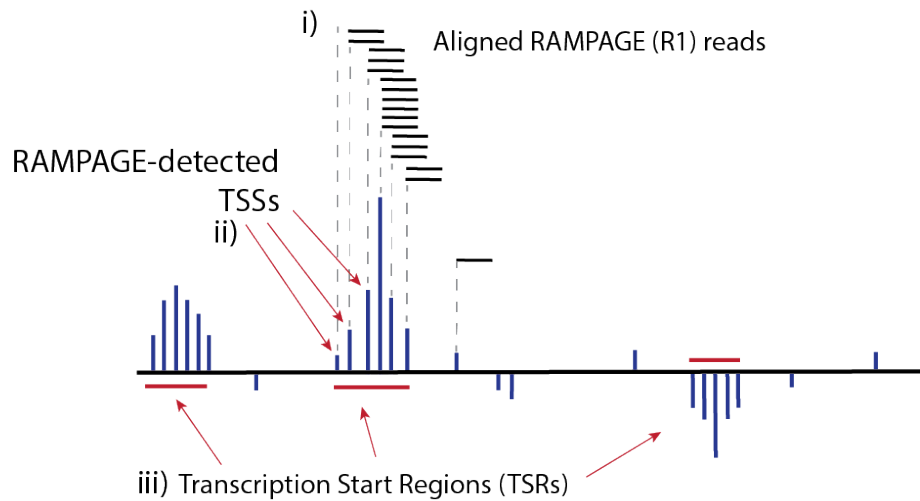


Fig. 2. Test caption for figure 2

5 References

References

1. V. P. Brendel and R. T. Raborn, "Gorampage- a workflow for promoter detection by 5'-read mapping," <https://github.com/brendelGroup/GoRAMPAGE>, 2016.
2. 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>
3. 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.
4. 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.
5. 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,

- 465 M. Katoh, Y. Kawasawa, J. Kelso, H. Kitamura, H. Kitano, G. Kollias, S. P. T. Kr-
 466 ishnan, A. Kruger, S. K. Kummerfeld, I. V. Kurochkin, L. F. Lareau, D. Lazarevic,
 467 L. Lipovich, J. Liu, S. Liuni, S. McWilliam, M. Madan Babu, M. Madera, L. Mar-
 468 chionni, H. Matsuda, S. Matsuzawa, H. Miki, F. Mignone, S. Miyake, K. Mor-
 469 ris, S. Mottagui-Tabar, N. Mulder, N. Nakano, H. Nakauchi, P. Ng, R. Nilsson,
 470 S. Nishiguchi, S. Nishikawa, F. Nori, O. Ohara, Y. Okazaki, V. Orlando, K. C.
 471 Pang, W. J. Pavan, G. Pavesi, G. Pesole, N. Petrovsky, S. Piazza, J. Reed, J. F.
 472 Reid, B. Z. Ring, M. Ringwald, B. Rost, Y. Ruan, S. L. Salzberg, A. Sandelin,
 473 C. Schneider, C. Schönbach, K. Sekiguchi, C. A. M. Semple, S. Seno, L. Sessa,
 474 Y. Sheng, Y. Shibata, H. Shimada, K. Shimada, D. Silva, B. Sinclair, S. Sperling,
 475 E. Stupka, K. Sugiura, R. Sultana, Y. Takenaka, K. Taki, K. Tammoja, S. L. Tan,
 476 S. Tang, M. S. Taylor, J. Tegner, S. A. Teichmann, H. R. Ueda, E. van Nimwegen,
 477 R. Verardo, C. L. Wei, K. Yagi, H. Yamanishi, E. Zabarovsky, S. Zhu, A. Zim-
 478 mer, W. Hide, C. Bult, S. M. Grimmond, R. D. Teasdale, E. T. Liu, V. Brusic,
 479 J. Quackenbush, C. Wahlestedt, J. S. Mattick, D. A. Hume, C. Kai, D. Sasaki,
 480 Y. Tomaru, S. Fukuda, M. Kanamori-Katayama, M. Suzuki, J. Aoki, T. Arakawa,
 481 J. Iida, K. Imamura, M. Itoh, T. Kato, H. Kawaji, N. Kawagashira, T. Kawashima,
 482 M. Kojima, S. Kondo, H. Konno, K. Nakano, N. Ninomiya, T. Nishio, M. Okada,
 483 C. Plessy, K. Shibata, T. Shiraki, S. Suzuki, M. Tagami, K. Waki, A. Watahiki,
 484 Y. Okamura-Oho, H. Suzuki, J. Kawai, Y. Hayashizaki, F. Consortium, R. G. E. R.
 485 Group, and G. S. G. G. N. P. C. Group, "The transcriptional landscape of the mam-
 486 malian genome," *Science (New York, NY)*, vol. 309, no. 5740, pp. 1559–1563, Sep.
 487 2005.
- 488 6. R. A. Hoskins, R. A. Hoskins, J. M. Landolin, J. M. Landolin, J. B. Brown, J. B.
 489 Brown, J. E. Sandler, J. E. Sandler, H. Takahashi, H. Takahashi, T. Lassmann,
 490 T. Lassmann, C. Yu, C. Yu, B. W. Booth, B. W. Booth, D. Zhang, D. Zhang,
 491 K. H. Wan, K. H. Wan, L. Yang, L. Yang, N. Boley, N. Boley, J. Andrews, J. An-
 492 drews, T. C. Kaufman, T. C. Kaufman, B. R. Graveley, B. R. Graveley, P. J.
 493 Bickel, P. J. Bickel, P. Carninci, J. W. Carlson, J. W. Carlson, S. E. Celniker,
 494 and S. E. Celniker, "Genome-wide analysis of promoter architecture in *Drosophila*
 495 *melanogaster*," *Genome Research*, vol. 21, no. 2, pp. 182–192, Feb. 2011.
- 496 7. E. A. Rach, H.-Y. Yuan, W. H. Majoros, P. Tomancak, and U. Ohler, "Motif
 497 composition, conservation and condition-specificity of single and alternative tran-
 498 scription start sites in the *Drosophila* genome," *Genome Biology*, vol. 10, no. 7, p.
 499 R73, 2009.
- 500 8. B. Lenhard, A. Sandelin, and P. Carninci, "Metazoan promoters: emerging char-
 501 acteristics and insights into transcriptional regulation," *Nature Reviews Genetics*,
 502 vol. 13, no. 4, pp. 233–245, Apr. 2012.
- 503 9. T. Ni, D. L. Corcoran, E. A. Rach, S. Song, E. P. Spana, Y. Gao, U. Ohler,
 504 and J. Zhu, "A paired-end sequencing strategy to map the complex landscape of
 505 transcription initiation," *Nature Methods*, vol. 7, no. 7, pp. 521–527, Jul. 2010.
- 506 10. U. Ohler, G.-c. Liao, H. Niemann, and G. M. Rubin, "Computational analysis of
 507 core promoters in the *Drosophila* genome," *Genome Biology*, vol. 3, no. 12, pp.
 508 research0087.1–0087.12, 2002.
- 509 11. R. T. Raborn, K. Spitze, V. P. Brendel, and M. Lynch, "Promoter Architecture
 510 and Sex-Specific Gene Expression in *Daphnia pulex*," *Genetics*, vol. 204, no. 2, pp.
 511 593–612, Aug. 2016.
- 512 12. C. Nepal, Y. Hadzhiev, C. Previti, V. Haberle, N. Li, H. Takahashi, A. M. M.
 513 Suzuki, Y. Sheng, R. F. Abdelhamid, S. Anand, J. Gehrig, A. Akalin, C. E. M.
 514 Kockx, A. A. J. van der Sloot, W. F. J. van IJcken, O. Armant, S. Rastegar,

- 515 C. Watson, U. Strahle, E. Stupka, P. Carninci, B. Lenhard, and F. Muller, "Dy-
516 namic regulation of the transcription initiation landscape at single nucleotide res-
517 olution during vertebrate embryogenesis," *Genome Research*, vol. 23, no. 11, pp.
518 1938–1950, Nov. 2013.
- 519 13. P. Carninci, A. Sandelin, B. Lenhard, S. Katayama, K. Shimokawa, J. Ponjavic,
520 C. A. M. Semple, M. S. Taylor, P. G. Engström, M. C. Frith, A. R. R. For-
521 rest, W. B. Alkema, S. L. Tan, C. Plessy, R. Kodzius, T. Ravasi, T. Kasukawa,
522 S. Fukuda, M. Kanamori-Katayama, Y. Kitazume, H. Kawaji, C. Kai, M. Naka-
523 mura, H. Konno, K. Nakano, S. Mottagui-Tabar, P. Arner, A. Chesi, S. Gustincich,
524 F. Persichetti, H. Suzuki, S. M. Grimmond, C. A. Wells, V. Orlando, C. Wahlest-
525 edt, E. T. Liu, M. Harbers, J. Kawai, V. B. Bajic, D. A. Hume, and Y. Hayashizaki,
526 "Genome-wide analysis of mammalian promoter architecture and evolution," *Nat-
527 ure Genetics*, vol. 38, no. 6, pp. 626–635, Apr. 2006.
- 528 14. S. Mwangi, G. Attardo, Y. Suzuki, S. Aksoy, and A. Christoffels, "TSS seq based
529 core promoter architecture in blood feeding Tsetse fly (*Glossina morsitans mor-
530 sitans*) vector of Trypanosomiasis," *BMC Genomics*, vol. 16, no. 1, p. 722, Sep.
531 2015.
- 532 15. K. Tsuchihara, Y. Suzuki, H. Wakaguri, T. Irie, K. Tanimoto, S.-i. Hashimoto,
533 K. Matsushima, J. Mizushima-Sugano, R. Yamashita, K. Nakai, D. Bentley, H. Es-
534 umi, and S. Sugano, "Massive transcriptional start site analysis of human genes in
535 hypoxia cells," *Nucleic Acids Research*, vol. 37, no. 7, pp. 2249–2263, Apr. 2009.
- 536 16. N. Cvetesic and B. Lenhard, "Core promoters across the genome," *Nature Biotech-
537 nology*, vol. 35, no. 2, pp. 123–124, Feb. 2017.
- 538 17. M. Lawrence and M. Morgan, "Scalable Genomics with R and Bioconductor,"
539 *Statistical Science*, vol. 29, no. 2, pp. 214–226, May 2014.
- 540 18. H. Lab, "FASTX Toolkit." [Online]. Available:
541 http://hannonlab.cshl.edu/fastx_toolkit/
- 542 19. R Core Team, *R: A Language and Environment for Statistical Computing*, R
543 Foundation for Statistical Computing, Vienna, Austria, 2017. [Online]. Available:
544 <https://www.R-project.org>
- 545 20. H. Li, B. Handsaker, A. Wysoker, T. Fennell, J. Ruan, N. Homer, G. Marth, G. R.
546 Abecasis, R. Durbin, and 1000 Genome Project Data Processing Subgroup, "The
547 Sequence Alignment/Map format and SAMtools," *Bioinformatics (Oxford, Eng-
548 land)*, vol. 25, no. 16, pp. 2078–2079, Aug. 2009.
- 549 21. R. Leinonen, H. Sugawara, M. Shumway, and International Nucleotide Sequence
550 Database Collaboration, "The sequence read archive." *Nucleic Acids Research*,
551 vol. 39, no. Database issue, pp. D19–21, Jan. 2011.
- 552 22. A. Dobin and T. R. Gingeras, "Optimizing RNA-Seq Mapping with STAR," in
553 *Transcription Factor Regulatory Networks*. New York, NY: Springer New York,
554 Apr. 2016, pp. 245–262.
- 555 23. T. Lassmann, "TagDust2: a generic method to extract reads from sequencing data,"
556 *BMC Bioinformatics*, vol. 16, no. 1, p. 1, Jan. 2015.
- 557 24. V. Hablerle, "CAGER: an R package for CAGE (Cap Analysis of Gene Expression)
558 data analysis and promoterome mining," 2013.

559 6 Checklist of Items to be Sent to Volume Editors

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

- 561 ☐ The final L^AT_EX source files

- 562 ☐ A final PDF file
- 563 ☐ A copyright form, signed by one author on behalf of all of the authors of the
- 564 paper.
- 565 ☐ A readme giving the name and email address of the corresponding author.