**Using RAMPAGE to identify and annotate promoters in insect genomes**

R. Taylor Raborn*\**1 and Volker P. Brendel1,2 1Department of Biology, Indiana University

2School of Informatics and Computing, Indiana University

Department of Biology Indiana University

212 S. Hawthorne Drive 205 Simon Hall, Bloomington, IN 47401, USA [http://www.brendelgroup.org](http://www.brendelgroup.org/)

**Abstract.** Application of Transcription Start Site (TSS) profiling tech- nologies, coupled with large-scale next-generation sequencing (NGS) has yielded valuable insights into the location, structure and activity of pro- moters across diverse metazoan model systems. In insects, TSS profiling has been used to characterize the promoter architecture of *Drosophila melanogaster* [1] and subsequently was employed to reveal widespread transposon-driven alternative promoter usage in the fruit fly [2].

In this chapter we discuss the computational analysis of the experimental data derived from of one TSS profiling method, RAMPAGE (RNA Anno- tation and Mapping of Promoters for Analysis of Gene Expression), that can be used for the precise, quantitative identification of promoters in insect genomes. We demonstrate this using the software tools GoRAM- PAGE [3] and TSRchitect [4], providing detailed instructions with the aim of taking the user from raw reads to processed results.

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

1 **1 Introduction**

# 2 1.1 TSS Profiling Identifies Promoters at Genome-Scale

3 The promoter, which is defined in eukaryotes as the genomic region bound by

4 RNA Polymerase II immediately prior to transcription initiation [5], is the pri-

5 mary locus of the regulation of gene expression. The identification of promoter

6 regions is necessary for understanding the *cis*-regulatory signals controlling gene

7 expression in an organism, and is also important for genome annotation. How-

8 ever, despite the rapid accumulation of genome sequences across metazoan and

9 arthropod diversity, accurate annotation of promoter regions remains sparse.

10 This is because—absent empirically-defined information—precisely identifying

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

11 sequence motifs that demarcate the promoter is unreliable. In contrast with cur-

12 rent *in silico* approaches, direct mapping of TSSs identifies the location of the

13 core promoter. Cap Analysis of Gene Expression (CAGE) [6], one of the first

14 methods devised to identify 5*1*-ends of mRNAs at large-scale, involves selective

15 capture of 5*1*-capped transcripts, first-strand reverse-transcription and ligation

16 of a short oligonucleotide (CAGE tag).

17

18 CAGE was initially utilized by the FANTOM (Functional Annotation of the

19 Mammalian Genome) consortium to identify promoter architecture in human

20 and mouse [7], providing the first glimpse of the global landscape of transcrip-

21 tion initiation. At the onset of the next-generation sequencing (NGS) era, CAGE

22 was coupled with massively-parallel sequencing to define 5*1*-mRNA ends at large

23 scale. This advance provided more extensive coverage of the expressed tran-

24 scriptome and provided increased sensitivity for quantitative measurements of

25 promoter activity.

26 **1.2 Promoter Architecture of *Drosophila melanogaster***

27 Hoskins and colleagues [1] performed CAGE in *D. melanogaster* as part of the

28 modENCODE consortium, identifying promoters at large-scale and characteriz-

29 ing the promoter architecture of an insect genome for the first time. The authors

30 found that TSS distributions at *Drosophila* promoters exhibit a range of shapes

31 that can be generally grouped into two major classes: *peaked* and *broad*. This

32 confirmed the original finding of Rach and colleagues [8], which was done us-

33 ing publicly-available expressed sequence tags (ESTs). Peaked promoters have a

34 single, major TSS position occupying a narrow genomic region, whereas broad

35 promoters lack a single, major TSS and contain TSSs across a wider region [8, 9].

36 The authors also showed a strong association between promoter class and motif

37 composition (consistent with previous findings [8, 10]). Peaked promoters were

38 associated with positionally-enriched *cis*-regulatory motifs including TATA, Ini-

39 tiator (Inr) and DPE (Downstream Promoter Element), while broad promoters

40 contained an enrichment of less-well characterized motifs, including *Ohler6* and

41 *Ohler7* [11]. The existence of at least two promoter classes appears to be con-

42 served among metazoans and has been reported (using TSS profiling methods)

43 in insects, cladocerans [12], fish [13] and mammals [14, 9].

44 **1.3 Promoter Structure of Insects**

45 Beyond *D. melanogaster*, few investigations have utilized TSS profiling in insect

46 genomes. As a consequence, what is known about promoter architecture in in-

47 sects is largely restricted to the *Drosophila* genus. As part of the modENCODE

48 effort, CAGE was performed in multiple tissues and developmental stages of the

49 *Drosophila pseudoobscura*. TSSs were found to be highly similar between species:

50 81% of TSSs of aligned, CAGE-identified TSSs from *D. pseudoobscura* were po-

51 sitioned within 20nt of their counterparts in *D. melanogaster*. An enrichment of

52 the CA dinucleotide was detected at the TSS ([-1, +1]), and the motifs corre-

53 sponding to TATA, Inr and DPE were positioned at the same locations relative

54 to the TSS in both species.

55

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

57 Tsetse fly (*Glossina morsitans morsitans*) [15]. Using TSS-seq (specifically Oligo-

58 capping; for details see [16]), the authors identified 3134 promoters associated

59 with 1424 genes. The authors found a preference for CA and AA dinucleotides at

60 the TSSs and observe the major core promoter elements observed in *Drosophila*:

61 TATA, Inr, DPE, in addition to MTE (Motif Ten Element). As in *D. melanogaster*,

62 peaked promoters were more likely to contain TATA and Inr than broad promot-

63 ers. While the taxonomic sampling of species for TSS profiling has been limited,

64 the existing studies are sufficient to provide a general picture of insect promoter

65 architecture. A major demarcation between the promoter architecture of insects

66 and mammals appears to be the large fraction of mammalian promoters found

67 in CpG islands [15]. CpG island promoters (CPIs) form the largest class of pro-

68 moter in mammals [17]; by contrast, CPIs are not known to exist as a class in

69 invertebrates.

# 70 1.4 Paired-end TSS Profiling with RAMPAGE

71 A notable recent methodological advance in TSS Profiling is RAMPAGE [2, 18],

72 a protocol for 5*1*-cDNA sequencing that combines cap trapping and template-

73 switching with paired-end sequence information (see Figure 1). As with CAGE

74 and other TSS profiling methods, RAMPAGE reads are aligned, to obtain TSSs

75 and clustered to identify Transcription Start Regions (TSRs), which are en-

76 richments of TSSs consistent with promoters (Figure 2a). A key advantage of

77 generating paired-end sequence is transcript connectivity, which provides a di-

78 rect link between a given 5*1*-end and its associated mRNA molecule [2] (Figure

79 2b). Because short or spurious RNAs are found within the transcriptome, tran-

80 script connectivity allows the TSSs (and thus promoters) of full-length mRNAs

81 to be unambiguously identified, which benefits genome annotation and improves

82 interpretation of transcript species. There are other TSS profiling methodologies

83 provide paired-end information, although these methods differ (with each other

84 and with RAMPAGE) in the ways capped RNA is captured and processed into

85 finished libraries. These include PEAT (Paired-end analysis of transcription)

86 [10] and nanoCAGE [19, 20]. PEAT has been applied in two species to date:

87 *D. melanogaster* [10] and the model plant *Arabidopsis thaliana* [21], whereas

88 nanoCAGE has been applied to mammalian systems. While this chapter will

89 discuss the processing and analysis of RAMPAGE libraries, the code and tools

90 we present here are capable of handling any other TSS profiling read datasets.

91

92 Batut and colleagues [2] generated libraries from total RNA isolated from 36

93 stages across the life cycle of *D. melanogaster*, generating a comprehensive gene

94 expression and promoter atlas for fruit fly and demonstrating the utility of RAM-

95 PAGE. RAMPAGE is currently being applied as part of the latest iteration of

96 ENCODE [22] to identify promoters in diverse human tissues [23], but as of this

97 writing it has not been applied to any non-*Drosophila* insect model system.

98

99 In anticipation of the future application of TSS profiling into other insect model

100 systems, we discuss in this chapter a well-documented protocol for the computa-

101 tional processing and analysis of RAMPAGE data, using selected libraries from

102 Batut *et al.* [2]. This method consists of two parts: first, we discuss how to pro-

103 cess, filter and align the sequenced RAMPAGE libraries to the *D. melanogaster*

104 genome. Second, we show how to identify TSSs and promoters from the aligned

105 sequences and associate them with coding regions. In closing, we will consider

106 further applications of this data and discuss the utility of reproducible workflows

107 in bioinformatic analysis.

108 **2 Materials**

109 The example analyses described herein require a workstation capable of doing

110 modern bioinformatics; minimally a reasonably-appointed laptop. An interme-

111 diate understanding of the Linux/Unix command line will be extremely useful,

112 although we make efforts to explain the procedures with clarity. In addition, it

113 will likely be necessary for the participant to have superuser privileges on the

114 machine. If you do not have a machine (or have access to one) that meets these

115 requirements, it is recommended that you consider cloud-based cyberinfrastruc-

116 ture, including Amazon Web Services (AWS; https://aws.amazon.com/), Cy-

117 Verse [(http://www.cyverse.org/)](http://www.cyverse.org/)) [24], or JetStream (https://jetstream-cloud.org/)

118 [25]. The former is a well-known pay-per-use solution, while the latter two are

119 NSF-funded resources that make compute allocations freely available to the pub-

120 lic.

121 For many users, the cyberinfrastructure approach is a convenient solution, par-

122 ticularly when providers offer task-dedicated virtual machines. In that case,

123 the user essentially rents a fully equipped computer with all necessary soft-

124 ware pre-installed and sufficient resources for the intended job. For the com-

125 putational workflows discussed here, researchers can check out an instance of

126 the "bgRAMOSE" image at JetStream which comes with all Brendel Group

127 software (https://brendelgroup.github.io/) as well as other useful bioinformatics

128 tools enabled.

129 **2.1 Hardware**

130 1. x86-64 compatible processors

131 2. 16GB RAM

132 3. 80GB+ hard disk space

133 **2.2 Operating System**

134 **–** 64 bit Linux (preferred) or Mac OS X (with Command Line Tools from

135 XCode)

|  |  |
| --- | --- |
| 136 | **2.3 Software** |
| 137 | Below is a list of the software packages required for this demonstration (*see* **Note** |
| 138 | **1**). |
| 139 |  |
| 140 | **Sequence retrieval** |
| 141 | 1. SRA Toolkit [26] (https[://www.ncbi.nlm.nih.gov/sra/docs/toolkitsoft/)](http://www.ncbi.nlm.nih.gov/sra/docs/toolkitsoft/)) |
| 142 | **GoRAMPAGE** |
| 143 | 1. GoRAMPAGE [3] (https://github.com/brendelGroup/GoRAMPAGE) |
| 144 | 2. fastq-multx [27] (https://github.com/brwnj/fastq-multx) |
| 145 | 3. FASTX-Toolkit [28] (<http://hannonlab.cshl.edu/fastx_toolkit/Index.html)> |
| 146 | 4. TagDust2 [29] (https://sourceforge.net/projects/tagdust/) |
| 147 | 5. Samtools [30] [(http://www.htslib.org/doc/samtools.html)](http://www.htslib.org/doc/samtools.html)) |
| 148 | 6. STAR [31] (https://github.com/alexdobin/STAR) |
| 149 | **TSRchitect** |
| 150 | 1. R (v. 3.4 and up) [32] (h[ttps://www.r-project.org/)](http://www.r-project.org/)) |
| 151 | 2. Bioconductor (v. 3.5 and up) [33] [(http://bioconductor.org/)](http://bioconductor.org/)) |
| 152 | 3. TSRchitect [4] [(http://bioconductor.org/pac](http://bioconductor.org/packages/release/bioc/html/TSRchitect.html))k[ages/release/bioc/html/TSRchitect.html)](http://bioconductor.org/packages/release/bioc/html/TSRchitect.html)) |
| 153 | 4. Various R package dependencies (see **Methods**) |

154 **2.4 Demonstration**

155 We created an online demonstration (demo) to serve as a companion to this

156 chapter, which contains both scripts and select files to assist you in completing

157 this tutorial. Please find the repository at here (in the folder "demo/MMB/"):

158 https://github.com/brendelgroup/GoRAMPAGE (*see* **Note 2**).

159 **2.5 Installation of R packages**

160 For installation of the software listed above, please follow the instructions pro-

161 vided by each respective package. Part of our analysis will require the use of R

162 packages found in the Bioconductor suite [33] (*see* **Note 3**). To install Biocon-

163 ductor, please type the following from an R console:

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

165 biocLite()

166 We will use the R package *TSRchitect* to identify promoters from aligned RAM-

167 PAGE libraries. Prior to running the analysis, it will be necessary to install a

168 series of prerequisite packages to *TSRchitect* from Bioconductor. Please install

169 these packages, followed by *TSRchitect* (as before, from an R console):

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

171 biocLite(c("AnnotationHub", "BiocGenerics", "BiocParallel",

172 "ENCODExplorer", "GenomicAlignments", "GenomeInfoDb",

173 "GenomicRanges", "IRanges", "methods",

174 "Rsamtools", "rtracklayer", "S4Vectors",

175 "SummarizedExperiment"))

176

177 biocLite("TSRchitect")

178 Finally, please confirm that TSRchitect has been installed correctly by loading

179 it from your R console as follows:

180 library(TSRchitect) #loading TSRchitect

181 **3 Methods**

# 182 3.1 Retrieving the RAMPAGE sequence data from NCBI

183 To begin our analysis, we must download the RAMPAGE data to our worksta-

184 tion. We will utilize tools provided by the SRA Toolkit, which should already

185 be installed on your machine (see **Materials**). The command *fastq-dump* al-

186 lows one to directly retrieve data from the GEO database using the appropriate

187 identifier(s). While there are 36 RAMPAGE libraries in the Batut *et al.* pa-

188 per, we will select a subset of these to analyze here. We will compare samples

189 from selected embryonic (E01h-E03h) and larval (L1-L3) tissues, representing

190 the beginning and end of embryonic development. For more information about

191 the experiment and the available RAMPAGE libraries, please see the following

192 link: http[s://www.ncbi.nlm.nih.go](http://www.ncbi.nlm.nih.gov/Traces/study/?acc=SRP011193)v/T[races/study/?acc=SRP011193.](http://www.ncbi.nlm.nih.gov/Traces/study/?acc=SRP011193)

193

194 First, let’s proceed with downloading the libraries from early embryonic tissues

195 (*see* **See Note 4**). We will make a new folder (entitled "fastq\_files/") to

196 house these files.

197 mkdir fastq\_files

198 cd fastq\_files

199

200 fastq-dump --split-files SRR424683

201 fastq-dump --split-files SRR424684

202 fastq-dump --split-files SRR424685

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

204 fastq-dump --split-files SRR424707

205 fastq-dump --split-files SRR424708

206 fastq-dump --split-files SRR424709

207 Once the download of the aforementioned files are complete, you should see a

208 total of 12 (6 *x* 2) separate fastq files in your current working directory:

209 ls -l \*.fastq | wc -l

210 **3.2 Creating symlinks to the files**

211 Our workflow expects fastq files that have the format "\*.R1/R2.clipped.fq".

212 Rather than rename them, we can simply create brand new symbolic links (sym-

213 links) to the files, as follows:

214 cd ..

215 mkdir -p output/reads/clipped

216 cd output/reads/clipped

217

218 #embryonic libraries

219 ln -s ../../../fastq\_files/SRR424683\_1.fastq E01h.R1.clipped.fq

220 ln -s ../../../fastq\_files/SRR424683\_2.fastq E01h.R2.clipped.fq

221 ln -s ../../../fastq\_files/SRR424684\_1.fastq E02h.R1.clipped.fq

222 ln -s ../../../fastq\_files/SRR424684\_2.fastq E02h.R2.clipped.fq

223 ln -s ../../../fastq\_files/SRR424685\_1.fastq E03h.R1.clipped.fq

224 ln -s ../../../fastq\_files/SRR424685\_2.fastq E03h.R2.clipped.fq

225

226 #larval libraries

227 ln -s ../../../fastq\_files/SRR424707\_1.fastq L1.R1.clipped.fq

228 ln -s ../../../fastq\_files/SRR424707\_2.fastq L1.R2.clipped.fq

229 ln -s ../../../fastq\_files/SRR424708\_1.fastq L2.R1.clipped.fq

230 ln -s ../../../fastq\_files/SRR424708\_2.fastq L2.R2.clipped.fq

231 ln -s ../../../fastq\_files/SRR424709\_1.fastq L3.R1.clipped.fq

232 ln -s ../../../fastq\_files/SRR424709\_2.fastq L3.R2.clipped.fq

233

234 cd ../../.. #returning to the output directory

235 **3.3 Downloading genomic data from *D. melanogaster***

236 Now that we have the fastq files from the RAMPAGE libraries downloaded and

237 named appropriately, we now must retrieve the genome assembly and rRNA se-

238 quences from *D. melanogaster*. The genome assembly is required for aligning the

239 RAMPAGE reads, and the rRNA sequences are required to filter out matching

240 reads in the sequenced RAMPAGE libraries. Because our sample is intended to

241 contain only capped RNAs, any rRNA sequences we observe in these RAMPAGE

242 libraries are contaminants that must be removed.

243

244 Please make note of the rRNA sequences, found in the file "Dmel\_rRNA.fasta",

245 from the folder additional\_files folder in the demo (*see* **Note 5**).

246

247 We will then download a version of the *D. melanogaster* genome assembly from

248 ENSEMBL (www.ensembl.org) [34]. To retrieve the genome assembly, please do

249 the following:

250 mkdir genome

251 cd genome

252 wget ftp://ftp.ensembl.org/pub/release-78/fasta/

253 drosophila\_melanogaster/dna/Drosophila\_melanogaster.BDGP5.dna.toplevel.fa.gz

254 #uncompressing the file

255 gzip -d Drosophila\_melanogaster.BDGP5.dna.toplevel.fa.gz

256 cd ..

# 257 3.4 Filtering and alignment of RAMPAGE reads using

258 **GoRAMPAGE**

259 At this stage we are ready to commence with the rRNA filtering and alignment

260 of the RAMPAGE libraries. We will use GoRAMPAGE, a tool we developed, to

261 perform these tasks in a concerted workflow. GoRAMPAGE runs TagDust [29]

262 to remove rRNA and low-complexity reads and STAR [31] to align RAMPAGE

263 (or other paired-end) reads to a given genome assembly.

264 **Setting up the GoRAMPAGE job.** Please refer to the script

265 "GoRAMPAGE\_script\_MMB.sh" and (using a text editor) provide the appropriate

266 paths to the genome assembly, output directory (see above) and rRNA sequences

267 (*see* **Note 6**). GoRAMPAGE jobs can optionally be run in parallel (*see* **Note**

268 **7**). The script can be executed as follows:

269 #vi GoRAMPAGE\_script\_MMB.sh #updating with a text editor

270 ./GoRAMPAGE\_script\_MMB.sh

271 If everything is working correctly you should start to see the results of the job

272 being written to the file "errScript". You can inspect the progress during the

273 run using the *less* command.

274 less -S errScript

275 Should the run fail before completion, any associated error messages will be

276 printed to the errScript file. Once the job is complete, you should see the message

277 "GoRAMPAGE job is complete!" appear on the command-line terminal.

278 **Inspecting the rRNA filtering results.** To evaluate the results from Step

279 3 (rRNA filtering), please navigate to the top level of the "output" directory

280 and open the file "LOGFILES". You’ll see the recorded progress of the program

281 Tagdust and a record of the results. We notice that (for the L3h library) 1046448

282 of reads (78.1%) were "extracted", meaning that slightly more than 20% of

283 reads were removed because of matches with ribosomal sequences. The removed

284 reads from all libraries are found in the "dusted\_discard" directory, and the

285 extracted reads are found in the current directory. Due to their sheer abundance

286 within cells, ribosomal RNA sequences are an inevitable contaminant within TSS

287 profiling libraries. For analysis purposes, it is important that these sequences be

288 removed, which is what has been completed here.

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

290 **Evaluating the alignments.** The folder "alignments/" in your GoRAMPAGE

291 output folder will now contain 6 .bam files, each representing the distinct RAM-

292 PAGE libraries selected for our analysis. Typing "ls -l" from the command line

293 will show that these files are symlinks to the original alignment files found

294 in the "STARoutput/" directory. "STARoutput/", as its name suggests, con-

295 tains the output from the STAR alignment, and this includes the alignment files

296 "\*.sortedByCoord.out.bam", and four additional log files. The files with the suf-

297 fix "\*.STAR.Log.final.out" each contain a summary of the alignment, such as

298 the number of input reads, the percentage of uniquely-mapped reads and the

299 percentage of unmapped reads. An inspection of these log files indicates that

300 the alignments have similar mapping rates (~70-80%), a reasonable outcome for

301 our purposes.

302

303 Now that our RAMPAGE libraries are filtered and aligned, we can commence

304 with the second half of our analysis.

# 305 3.5 Promoter identification from aligned RAMPAGE libraries

306 We can now use the prepared alignment files to identify TSSs and promoters

307 from the selected RAMPAGE libraries. There are currently several tools avail-

308 able for this purpose. *CAGEr*, developed by Haberle [35], was utilized to perform

309 TSS identification as part of the FANTOM5 efforts. We will use *TSRchitect* in

310 this demonstration, since it was specifically designed to analyze paired-end TSS

311 profiling datasets, and also because it is more flexible with respect to model sys-

312 tem (*i.e.* it does not require a corresponding *BSGenome* [36] package). The latter

313 feature will be helpful when analyzing the non-*D. melanagaster* TSS profiling

314 datasets that we expect to be generated in the near future.

315 **Setting up the Analysis.** *TSRchitect*, the package we’ll use for this analy-

316 sis, is an R package available in the Bioconductor suite of genomics tools [33].

317 It makes use of existing packages and data structures within this environment,

318 where available, to identify promoters from sequence alignments. Since you have

319 already installed *TSRchitect* and its dependencies (see section 2.3), we are set

320 to proceed.

321

322 There are two general ways one can choose to run *TSRchitect*. The first is in-

323 teractively *i.e.* typing the instructions directly into an R console. While this

324 is a perfectly acceptable way to run analyses using package, for larger jobs

325 it will likely be more efficient (and likely more reproducible) to run a ded-

326 icated R script. We have provided sample scripts to make it easier for you

327 to set up an R script. The two scripts are identical with a single exception:

328 one is set up to run in parallel ("TSRchitect\_parallel\_MMB.R"), while the

329 other is written to run in serial ("TSRchitect\_serial\_MMB.R"). Please select

330 the script that best suits your computing resources. In the section to follow, we

331 will go through the output of the analysis. For further details on how to use

332 *TSRchitect*, please see its documentation at its Bioconductor page found here:

333 h[ttps://www.bioconductor.org/pac](http://www.bioconductor.org/packages/release/bioc/html/TSRchitect.html)k[ages/release/bio](http://www.bioconductor.org/packages/release/bioc/html/TSRchitect.html)c/h[tml/TSRchitec](http://www.bioconductor.org/packages/release/bioc/html/TSRchitect.html)t.h[tml.](http://www.bioconductor.org/packages/release/bioc/html/TSRchitect.html)

334 **Running the Analysis.** To run TSRchitect using the batch script, provide

335 full paths for the variables "BAMDIR" and "DmAnnot" in the script provided

336 (*see* **Note 8**). *BAMDIR* should be a path to the subdirectory "alignments/" in

337 RAMPAGE output directory you specified earlier, and *DmAnnot* should be a

338 full path to the *D. melanogaster* gene annotation listed above.

339

340 Once this is complete, we can run the batch script from the Linux command-line

341 as follows:

342 R CMD BATCH TSRchitect\_parallel\_MMB.R #or use ’serial script

343 #assumes variables BAMDIR and DmAnnot have already been set

344 bg #puts this job in the background

345 Once the job is underway, you can monitor its progress by looking at the contents

346 of the .Rout file (in this case, "TSRchitect\_parallel\_MMB.Rout").

347 **Reviewing the *TSRchitect* script.** Before we evaluate the results (which

348 will have been written to your working directory after running the batch script),

349 there are some important aspects of the analysis to review. We discuss these for

350 informational purposes only; it will not necessary to perform these commands

351 separate from the batch script provided. First, we must initialize the *tssObject*

352 (which stores the information about the experiment) appropriately (*see* **Note 9**).

353

354 The inputs in this case are BAM files; *TSRchitect* also accepts input in BED

355 format.

356 DmRAMPAGE <- loadTSSobj(experimentTitle = "RAMPAGE Tutorial", \

357 inputDir=BAMDIR, isPairedBAM=TRUE, \

358 sampleNames=c("E1h","E2h", "E3h", "L1", "L2", "L3"), \

359 replicateIDs=c(1,1,1,2,2,2))

360 Next, we extract TSSs from the alignment files:

361 DmRAMPAGE <- inputToTSS(DmRAMPAGE)

362 A critical step in our analysis is identifying TSRs from the aligned TSS data;

363 to do this we use the function *determineTSR*. We have selected the job to run

364 on 4 cores in this example (*n.cores*=4). Please enter the number of cores ap-

365 propriate for your system. Because we want to identify TSRs from every one

366 of the selected RAMPAGE libraries, we specify *tssSet* ="all". The parameter

367 *tagCountThreshold* was set to 25, meaning that only TSSs supported by 25 or

368 more 5*1* RAMPAGE reads will be included within a TSR. Setting *writeTable* to

369 "TRUE" means that the identified TSRs from each set will be written to the

370 working directory.

\

|  |  |
| --- | --- |
| 371 | DmRAMPAGE <- determineTSR(experimentName=DmRAMPAGE, n.cores=4, |
| 372 | tsrSetType="replicates", tssSet="all", tagCountThreshold=25, \ |
| 373 | clustDist=20, writeTable=TRUE) |

374 *TSRchitect* can incorporate the tag abundances from each of the samples and

375 append them to the list of identified TSRs. This is useful for downstream analysis

376 of differential expression.

377 DmRAMPAGE <- addTagCountsToTSR(experimentName=DmRAMPAGE, \

378 tsrSetType="replicates", tsrSet=1, tagCountThreshold=10, \

379 writeTable=TRUE)

380 We can use *TSRchitect* to import an annotation file (or, alternatively, use an

381 existing one from *AnnotationHub*) and use it to associate our set of identified

382 TSRs with coding genes. We can specify the maximum distances (both up-

383 and downstream) between the TSR and the annotation using the arguments

384 *upstreamDist* and *downstreamDist*.

importAnnotationExternal(experimentName=DmRAMPAGE, \

|  |  |
| --- | --- |
| 385 | DmRAMPAGE <- |
| 386 | fileType="gff3", annotFile=DmAnnot) |
| 387 |  |
| 388 | DmRAMPAGE <- |
| 389 | tsrSetType="replicates", tsrSet=1, \ |
| 390 | upstreamDist=1000, downstreamDist=200, |
| 391 | featureColumnID="ID", writeTable=TRUE) |

addAnnotationToTSR(experimentName=DmRAMPAGE, \

feature="gene", \

392 Now we have generated a set of identified TSSs, TSRs from all 6 RAMPAGE

393 libraries, and have associated the identified TSRs with annotated genes. Next, we

394 will merge the libraries into two samples according to condition: early embryonic

395 (E1h, E2h, E3h) and late larval (L1, L2, L3) using the information we provided

396 when we initialized the *tssObject* at the start of this section. After merging, we

397 identify promoters i) within the merged samples and ii) within the entire dataset

398 combined, and associate with the *D. melanogaster* gene annotation as described

399 previously (not shown).

400 #merging the sample data into two groups

401 DmRAMPAGE <- mergeSampleData(DmRAMPAGE)

402

403 # ... identifying TSRs from the merged samples:

404 DmRAMPAGE <- determineTSR(experimentName=DmRAMPAGE, \

|  |  |
| --- | --- |
| 405 | n.cores=4, tsrSetType="merged", \ |
| 406 | tssSet="all", tagCountThreshold=40, \ |
| 407 | clustDist=20, writeTable=TRUE) |

408 **Evaluating the results** Our analysis using *TSRchitect* is now complete. A

409 snapshot of a representative sample of small set of aligned RAMPAGE libraries

410 is shown in Figure 3. Your working directory should now contain the following:

447 **3.6 Summary**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 411 | **–** TSSs from each sample *e.g.* TSSset-1.txt: (6) | | | | | | |
| 412 | **–** TSRs from each sample (in both .txt and .tab formats): (12) | | | | | | |
| 413 | **–** TSRs from each merged group (in both .txt and .tab formats): *e.g.* TSRsetMerged- | | | | | | |
| 414 | 1.txt: (4) | | | | | | |
| 415 | **–** TSRs from the combined set of TSSs: TSRsetCombined.tab: (1) | | | | | | |
| 416 | Let’s briefly review the files (*see* **Note 10**). We can quickly obtain the counts | | | | | | |
| 417 | on the command line, as follows: | | | | | | |
| 418 | wc -l \*.tab | | | | | | |
| 419 | 8377 TSRset-1.tab | | | | | | |
| 420 | 6159 TSRset-2.tab | | | | | | |
| 421 | 4814 TSRset-3.tab | | | | | | |
| 422 | 17924 TSRset-4.tab | | | | | | |
| 423 | 11851 TSRset-5.tab | | | | | | |
| 424 | 3242 TSRset-6.tab | | | | | | |
| 425 | 13986 TSRsetCombined.tab | | | | | | |
| 426 | 7344 TSRsetMerged-1.tab | | | | | | |
| 427 | 12126 TSRsetMerged-2.tab | | | | | | |
| 428 | 85823 total | | | | | | |
| 429 | We will see that we have identified between roughly 3,200 and 18,000 TSRs | | | | | | |
| 430 | within the individual RAMPAGE samples, which is attributable to the dif- | | | | | | |
| 431 | ferences in library sizes. We detect 7,344 TSRs within the early embryonic | | | | | | |
| 432 | samples ("TSRsetMerged-1.tab") and 12,126 TSRs in the late larval samples | | | | | | |
| 433 | ("TSRsetMerged-2.tab"). Within the combined samples ("TSRsetCombined.tab") | | | | | | |
| 434 | we find 13,986 TSRs, which is similar to the number reported by Hoskins *et. al.* | | | | | | |
| 435 | [1]. | | | | | | |
| 436 |  | | | | | | |
| 437 | In addition to identifying the position of a given TSRs, *TSRchitect* records other | | | | | | |
| 438 | useful information about its properties. The *width* of a TSR refers the span of | | | | | | |
| 439 | the genomic region it occupies (in bp), and the *Shape Index* (SI) is measure of | | | | | | |
| 440 | the relative peakedness of the TSR. We can see an example of this in the file | | | | | | |
| 441 | "TSRsetMerged-1.txt". | | | | | | |
| 442 | seq start end strand nTSSs tsrWidth shapeIndex featureID | | | | | | |
| 443 | 2L.67043.67044.+ | 2L | 67043 67044 + | 270 | 2 | 1 | NA |
| 444 | 2L.74089.74115.+ | 2L | 74089 74115 + | 341 | 27 | 0.13 | NA |
| 445 | 2L.94739.94752.+ | 2L | 94739 94752 + | 1650 | 14 | 0.55 | FBgn0031 |
| 446 | 2L.102386.102386.+ | 2L | 102386 102386 + | 284 | 1 | 2 | FBgn0031 |

448 The workflow provided here is intended to serve as a useful entry point for the

449 analysis of TSS profiling data in insects. On the computational side, we have

450 provided an open source set of tools so that the uninitiated genome scientist

451 can begin to analyze RAMPAGE (or other forms of TSS profiling data) quickly.

452 While the analysis centered on *D. melanogaster* via the use of public datasets,

453 it is anticipated that this will assist groups who may be interested in performing

454 TSS profiling in their preferred insect model system.The application of TSS

455 profiling technology across a more representative sample of insect diversity will

456 improve our understanding of the positions and general structure *cis*-regulatory

457 regions in this phylum.

458 **4 Figures**

459 **5 Notes**

460 1. Please consult the GoRAMPAGE documentation found here:

461 https://github.com/BrendelGroup/GoRAMPAGE.

462 Installation instructions for the prerequisites of GoRAMPAGE (which in-

463 cludes some of the items listed) are found at the following link:

464 https://github.com/BrendelGroup/GoRAMPAGE/tree/master/src.

465 2. On Linux, the installation of a few packages are necessary in order to install

466 Bioconductor packages using *biocLite()*.

467 To install them using Ubuntu:

468 apt-get install libssl-dev

469 apt-get install libcurl4-openssl-dev

470 apt-get install libxml2-dev

471 If you do not Ubuntu, use the commands necessary to install the above

472 packages on your Linux distribution.

473 3. You can clone the entire GoRAMPAGE repository (which includes the con-

474 tents of the demo) to your workspace on the command line using git, as

475 follows:

476 git clone https://github.com/brendelgroup/GoRAMPAGE/

477 cd demo/MMB

478 The "scripts/" folder in the demo contains code for you to run the two major

479 workflows described in this chapter. The "additional\_files/" folder con-

480 tains the following files which are necessary for the analysis: i) a fasta file con-

481 taining ribosomal RNA sequences for *D. melanogaster* (Dmel\_rRNA.fasta)

482 and ii) a gene annotation for *D. melanogaster*

483 (Drosophila\_melanogaster.BDGP5.78.gff).

484 4. Since these fastq files are paired-end, we use the argument *–split-files* to

485 generate separate files for each read pair.

486 5. If you are running this on a cluster with a job scheduler you’ll need to add

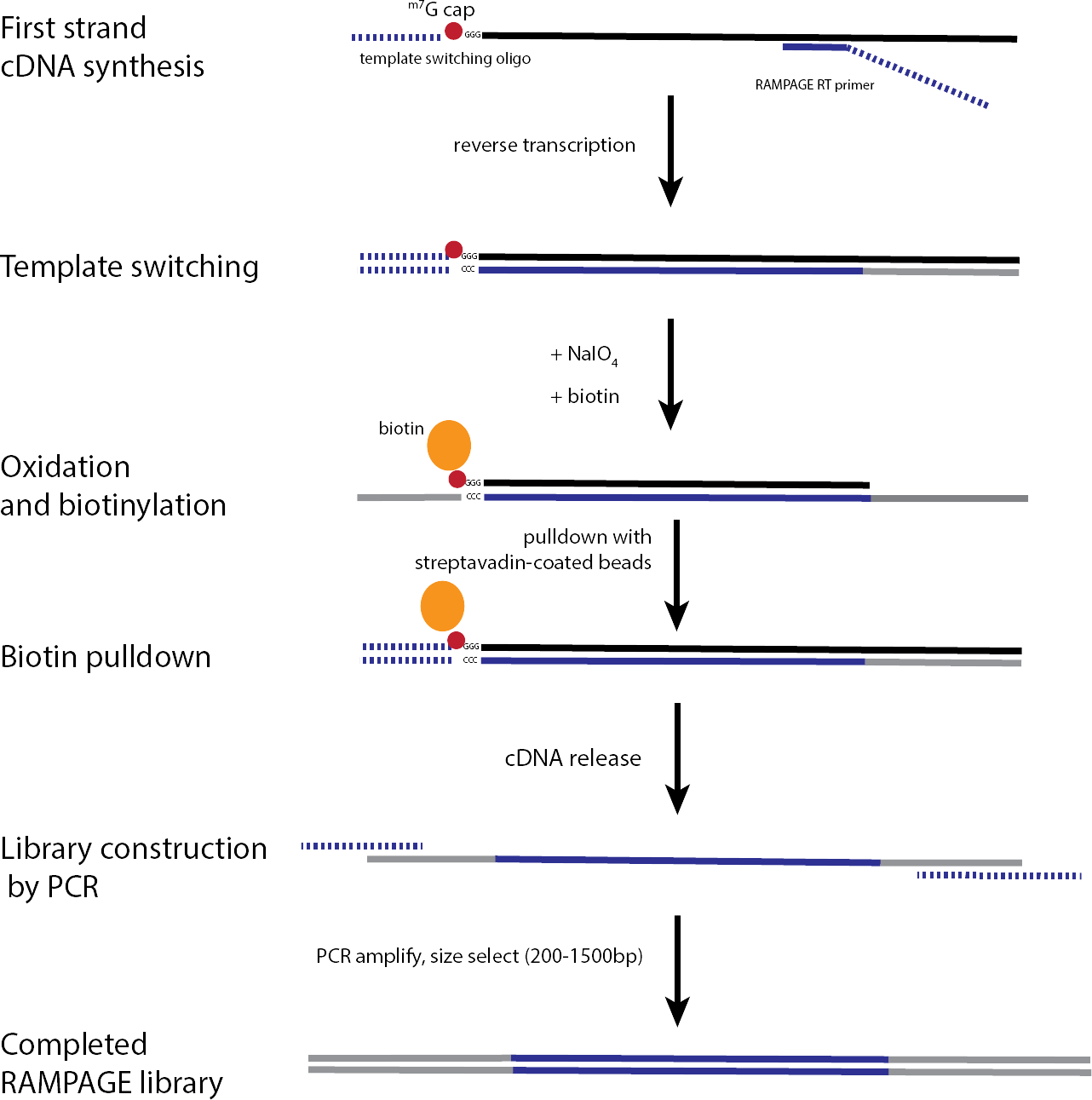
487 the necessary headers to the top of the script and submit the job in the

488 appropriate manner.

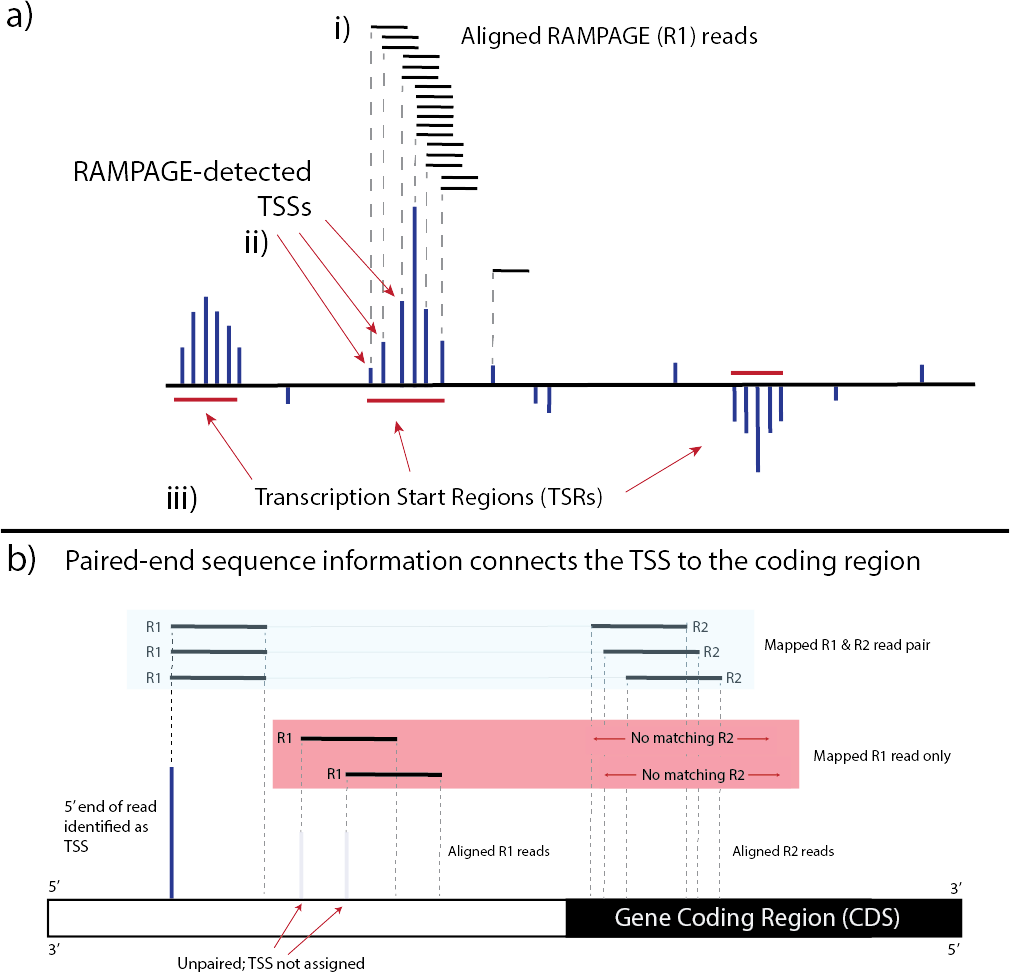
489 6. The rRNA sequences were retrieved separately from Genbank at NCBI [38].

490 7. For parallel execution, GoRAMPAGE uses the Linux package *GNU parallel*

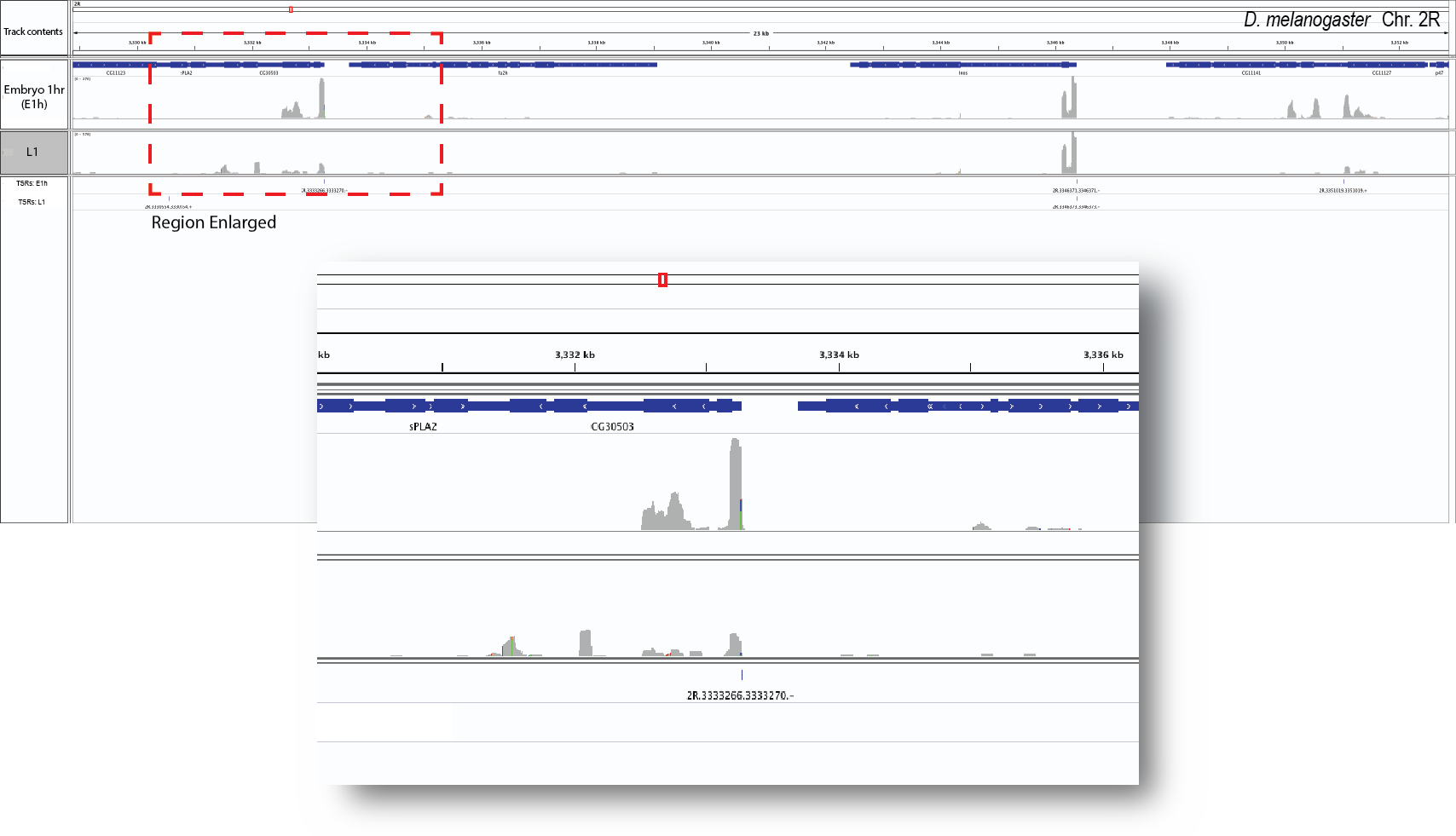
491 [39]. Please see the GoRAMPAGE documentation for more information.



**Fig. 1.** A brief summary of the RAMPAGE protocol. Starting with high-quality to- tal RNA, first-strand cDNA synthesis is initiated using a cap-bound oligonucleotide and a custom RAMPAGE RT primer, creating a double-stranded DNA-RNA hybrid molecule. Next, the 5*l*-m7G cap is oxidized, bound with biotin and pulled down with streptavadin-coated beads. The single-stranded cDNA molecules is released and the final RAMPAGE library construction is completed with PCR using custom oligonu- cleotides, followed by size-selection. This illustration was adapted from [18].



**Fig. 2.** An overview of promoter identification using RAMPAGE. a) RAMPAGE reads are aligned to the genome. The 5*l*-most genomic coordinate from each properly-paired R1 read is estimated as a TSS. The abundance of mapped 5*l*-ends at a given TSS is a measure of its abundance. TSSs above a minimum threshold will be clustered into TSRs. b) RAMPAGE-derived Paired-end sequence information provides a connection between a 5*l*-mRNA end and a gene coding region. Only properly-paired R1 reads (*i.e.* with an aligned R2 read) are identified as TSSs and then included in the downstream clustering procedure described in part *a*.



**Fig. 3.** An overview of the TSS profiling information provided by RAMPAGE. A rep- resentative visualization of RAMPAGE peaks (*i.e.* clusters of properly-aligned RAM- PAGE reads) within an arbitrarily-selected genomic region of *D. melanogaster* chro- mosome 2R is shown, along with the corresponding gene annotation within this region. RAMPAGE data from two RAMPAGE libraries from Batut *et al* [2] are shown, which were generated from RNA isolated from developmental stages E1h and L1 *see* **Meth- ods**. For each library, the abundance of RAMPAGE reads that align to a given site within the genome is represented by density plots (shown in gray). Gene models are shown in blue, where the thickened line represents exons and thin lines represent in- trons. The locations of TSRs identified by *TSRchitect* are shown in the two tracks from the bottom of the image. A single region, highlighted with the red dashed line is enlarged (the *Inset* ) to show further detail of a selected gene and RAMAPGE signals. In some cases, the expression of 5*l*-ends between the two samples is roughly equiv- alent, whereas in others the observed signal is substantially higher (*see* Inset). The original images are screenshots generated in the Integrated Genomic Viewer (IGV; <http://software.broadinstitute.org/software/igv/)>[37]. Where necessary, additional an- notation was added using Adobe Illustrator.

492 8. To do this, please edit the batch script TSRchitect\_serial\_MMB.R with a

493 text editor of your choice.

|  |  |  |
| --- | --- | --- |
| 494 | 9. | Because the samples provided derive from related developmental stages, we |
| 495 |  | will merge them for annotation purposes using the argument *replicateIDs*, |
| 496 |  | (though it must be emphasized that they are not replicates). |
| 497 | 10. | All of *TSRchitect’s* output files are labeled according to the order that they |
| 498 |  | are loaded onto the *tssObject*. For example, *TSSset-1.txt* corresponds to the |
| 499 |  | first RAMPAGE dataset (in our case E1h), and *TSSset-2.txt* corresponds to |
| 500 |  | the second RAMPAGE dataset (for this example E2h), and so on. You can |
| 501 |  | check which datasets are loaded on the *tssObject* by simply entering it on an |
| 502 |  | R console. Please see the *TSRchitect* documentation for more information. |
| 503 |  | **Acknowledgments** |
| 504 |  | The authors would like to thank Philippe Batut for generous technical assis- |
| 505 |  | tance with the RAMPAGE protocol, and to Nathan Keith for his help estab- |
| 506 |  | lishing the protocol in our laboratory. The authors are grateful to Thomas |
| 507 |  | W. McCarthy for his help testing the code and providing editorial feedback. |
| 508 |  | **Disclosure Declaration** |
| 509 |  | The authors declare that they have no competing interests. |
| 510 | **6** | **References** |

511 **References**

|  |  |  |
| --- | --- | --- |
| 512 | 1. | R. A. Hoskins, R. A. Hoskins, J. M. Landolin, J. M. Landolin, J. B. Brown, J. B. |
| 513 |  | Brown, J. E. Sandler, J. E. Sandler, H. Takahashi, H. Takahashi, T. Lassmann, |
| 514 |  | T. Lassmann, C. Yu, C. Yu, B. W. Booth, B. W. Booth, D. Zhang, D. Zhang, |
| 515 |  | K. H. Wan, K. H. Wan, L. Yang, L. Yang, N. Boley, N. Boley, J. Andrews, J. An- |
| 516 |  | drews, T. C. Kaufman, T. C. Kaufman, B. R. Graveley, B. R. Graveley, P. J. |
| 517 |  | Bickel, P. J. Bickel, P. Carninci, J. W. Carlson, J. W. Carlson, S. E. Celniker, |
| 518 |  | and S. E. Celniker, “Genome-wide analysis of promoter architecture in Drosophila |
| 519 |  | melanogaster.” *Genome Research*, vol. 21, no. 2, pp. 182–192, Feb. 2011. |
| 520 | 2. | P. J. Batut, A. Dobin, C. Plessy, P. Carninci, and T. R. Gingeras, “High-fidelity |
| 521 |  | promoter profiling reveals widespread alternative promoter usage and transposon- |
| 522 |  | driven developmental gene expression.” *Genome Research*, Aug. 2012. |
| 523 | 3. | V. P. Brendel and R. T. Raborn, *GoRAMPAGE- A Workflow For Promoter Detec-* |
| 524 |  | *tion by 5’-Read Mapping*, https://github.com/brendelGroup/GoRAMPAGE, 2016. |
| 525 | 4. | R. T. Raborn and V. Brendel, *TSRchitect: Promoter identification from large-scale* |
| 526 |  | *TSS profiling data*, 2017, r Bioconductor package version 1.0.0. [Online]. Available: |
| 527 |  | <http://bioconductor.org/packages/release/bioc/html/TSRchitect.html> |
| 528 | 5. | J. T. Kadonaga, “Perspectives on the RNA polymerase II core promoter.” *Wiley* |
| 529 |  | *Interdisciplinary Reviews: Developmental Biology*, vol. 1, no. 1, pp. 40–51, Jan. |
| 530 |  | 2012. |

531 6. R. Kodzius, M. Kojima, H. Nishiyori, M. Nakamura, S. Fukuda, M. Tagami,

532 D. Sasaki, K. Imamura, C. Kai, M. Harbers, Y. Hayashizaki, and P. Carninci,

533 “CAGE: cap analysis of gene expression.” *Nature Methods*, vol. 3, no. 3, pp. 211–

534 222, Mar. 2006.

535 7. P. Carninci, T. Kasukawa, S. Katayama, J. Gough, M. C. Frith, N. Maeda,

536 R. Oyama, T. Ravasi, B. Lenhard, C. Wells, R. Kodzius, K. Shimokawa, V. B.

537 Bajic, S. E. Brenner, S. Batalov, A. R. R. Forrest, M. Zavolan, M. J. Davis, L. G.

538 Wilming, V. Aidinis, J. E. Allen, A. Ambesi-Impiombato, R. Apweiler, R. N. Atu-

539 raliya, T. L. Bailey, M. Bansal, L. Baxter, K. W. Beisel, T. Bersano, H. Bono, A. M.

540 Chalk, K. P. Chiu, V. Choudhary, A. Christoffels, D. R. Clutterbuck, M. L. Crowe,

541 E. Dalla, B. P. Dalrymple, B. de Bono, G. Della Gatta, D. di Bernardo, T. Down,

542 P. Engstrom, M. Fagiolini, G. Faulkner, C. F. Fletcher, T. Fukushima, M. Furuno,

543 S. Futaki, M. Gariboldi, P. Georgii-Hemming, T. R. Gingeras, T. Gojobori, R. E.

544 Green, S. Gustincich, M. Harbers, Y. Hayashi, T. K. Hensch, N. Hirokawa, D. Hill,

545 L. Huminiecki, M. Iacono, K. Ikeo, A. Iwama, T. Ishikawa, M. Jakt, A. Kanapin,

546 M. Katoh, Y. Kawasawa, J. Kelso, H. Kitamura, H. Kitano, G. Kollias, S. P. T. Kr-

547 ishnan, A. Kruger, S. K. Kummerfeld, I. V. Kurochkin, L. F. Lareau, D. Lazarevic,

548 L. Lipovich, J. Liu, S. Liuni, S. McWilliam, M. Madan Babu, M. Madera, L. Mar-

549 chionni, H. Matsuda, S. Matsuzawa, H. Miki, F. Mignone, S. Miyake, K. Mor-

550 ris, S. Mottagui-Tabar, N. Mulder, N. Nakano, H. Nakauchi, P. Ng, R. Nilsson,

551 S. Nishiguchi, S. Nishikawa, F. Nori, O. Ohara, Y. Okazaki, V. Orlando, K. C.

552 Pang, W. J. Pavan, G. Pavesi, G. Pesole, N. Petrovsky, S. Piazza, J. Reed, J. F.

553 Reid, B. Z. Ring, M. Ringwald, B. Rost, Y. Ruan, S. L. Salzberg, A. Sandelin,

554 C. Schneider, C. Schönbach, K. Sekiguchi, C. A. M. Semple, S. Seno, L. Sessa,

555 Y. Sheng, Y. Shibata, H. Shimada, K. Shimada, D. Silva, B. Sinclair, S. Sperling,

556 E. Stupka, K. Sugiura, R. Sultana, Y. Takenaka, K. Taki, K. Tammoja, S. L. Tan,

557 S. Tang, M. S. Taylor, J. Tegner, S. A. Teichmann, H. R. Ueda, E. van Nimwegen,

558 R. Verardo, C. L. Wei, K. Yagi, H. Yamanishi, E. Zabarovsky, S. Zhu, A. Zim-

559 mer, W. Hide, C. Bult, S. M. Grimmond, R. D. Teasdale, E. T. Liu, V. Brusic,

560 J. Quackenbush, C. Wahlestedt, J. S. Mattick, D. A. Hume, C. Kai, D. Sasaki,

561 Y. Tomaru, S. Fukuda, M. Kanamori-Katayama, M. Suzuki, J. Aoki, T. Arakawa,

562 J. Iida, K. Imamura, M. Itoh, T. Kato, H. Kawaji, N. Kawagashira, T. Kawashima,

563 M. Kojima, S. Kondo, H. Konno, K. Nakano, N. Ninomiya, T. Nishio, M. Okada,

564 C. Plessy, K. Shibata, T. Shiraki, S. Suzuki, M. Tagami, K. Waki, A. Watahiki,

565 Y. Okamura-Oho, H. Suzuki, J. Kawai, Y. Hayashizaki, F. Consortium, R. G. E. R.

566 Group, and G. S. G. G. N. P. C. Group, “The transcriptional landscape of the mam-

567 malian genome,” *Science (New York, NY)*, vol. 309, no. 5740, pp. 1559–1563, Sep.

568 2005.

569 8. E. A. Rach, H.-Y. Yuan, W. H. Majoros, P. Tomancak, and U. Ohler, “Motif

570 composition, conservation and condition-specificity of single and alternative tran-

571 scription start sites in the Drosophila genome.” *Genome Biology*, vol. 10, no. 7, p.

572 R73, 2009.

573 9. B. Lenhard, A. Sandelin, and P. Carninci, “Metazoan promoters: emerging char-

574 acteristics and insights into transcriptional regulation.” *Nature Reviews Genetics*,

575 vol. 13, no. 4, pp. 233–245, Apr. 2012.

576 10. T. Ni, D. L. Corcoran, E. A. Rach, S. Song, E. P. Spana, Y. Gao, U. Ohler,

577 and J. Zhu, “A paired-end sequencing strategy to map the complex landscape of

578 transcription initiation.” *Nature Methods*, vol. 7, no. 7, pp. 521–527, Jul. 2010.

579 11. U. Ohler, G.-c. Liao, H. Niemann, and G. M. Rubin, “Computational analysis of

580 core promoters in the Drosophila genome.” *Genome Biology*, vol. 3, no. 12, pp.

research0087.1–0087.12, 2002.

|  |  |
| --- | --- |
| 581 |  |
| 582 | 12. |
| 583 |  |
| 584 |  |
| 585 | 13. |
| 586 |  |
| 587 |  |
| 588 |  |
| 589 |  |
| 590 |  |
| 591 |  |
| 592 | 14. |
| 593 |  |
| 594 |  |
| 595 |  |
| 596 |  |
| 597 |  |
| 598 |  |
| 599 |  |
| 600 |  |
| 601 | 15. |
| 602 |  |
| 603 |  |
| 604 |  |
| 605 | 16. |
| 606 |  |
| 607 |  |
| 608 |  |
| 609 | 17. |
| 610 |  |
| 611 | 18. |
| 612 |  |
| 613 |  |
| 614 |  |
| 615 | 19. |
| 616 |  |
| 617 |  |
| 618 |  |
| 619 |  |
| 620 |  |
| 621 | 20. |
| 622 |  |
| 623 |  |
| 624 | 21. |
| 625 |  |
| 626 |  |
| 627 |  |
| 628 | 22. |
| 629 |  |
| 630 | 23. |
| 631 |  |

R. T. Raborn, K. Spitze, V. P. Brendel, and M. Lynch, “Promoter Architecture and Sex-Specific Gene Expression in Daphnia pulex.” *Genetics*, vol. 204, no. 2, pp. 593–612, Aug. 2016.

C. Nepal, Y. Hadzhiev, C. Previti, V. Haberle, N. Li, H. Takahashi, A. M. M.

Suzuki, Y. Sheng, R. F. Abdelhamid, S. Anand, J. Gehrig, A. Akalin, C. E. M. Kockx, A. A. J. van der Sloot, W. F. J. van IJcken, O. Armant, S. Rastegar,

C. Watson, U. Strahle, E. Stupka, P. Carninci, B. Lenhard, and F. Muller, “Dy- namic regulation of the transcription initiation landscape at single nucleotide res- olution during vertebrate embryogenesis,” *Genome Research*, vol. 23, no. 11, pp. 1938–1950, Nov. 2013.

P. Carninci, A. Sandelin, B. Lenhard, S. Katayama, K. Shimokawa, J. Ponjavic,

C. A. M. Semple, M. S. Taylor, P. G. Engström, M. C. Frith, A. R. R. For- rest, W. B. Alkema, S. L. Tan, C. Plessy, R. Kodzius, T. Ravasi, T. Kasukawa,

S. Fukuda, M. Kanamori-Katayama, Y. Kitazume, H. Kawaji, C. Kai, M. Naka- mura, H. Konno, K. Nakano, S. Mottagui-Tabar, P. Arner, A. Chesi, S. Gustincich,

F. Persichetti, H. Suzuki, S. M. Grimmond, C. A. Wells, V. Orlando, C. Wahlest- edt, E. T. Liu, M. Harbers, J. Kawai, V. B. Bajic, D. A. Hume, and Y. Hayashizaki, “Genome-wide analysis of mammalian promoter architecture and evolution,” *Na- ture Genetics*, vol. 38, no. 6, pp. 626–635, Apr. 2006.

S. Mwangi, G. Attardo, Y. Suzuki, S. Aksoy, and A. Christoffels, “TSS seq based

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.

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.

N. Cvetesic and B. Lenhard, “Core promoters across the genome,” *Nature Biotech-*

*nology*, vol. 35, no. 2, pp. 123–124, Feb. 2017.

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.

C. Plessy, N. Bertin, H. Takahashi, R. Simone, M. Salimullah, T. Lassmann,

M. Vitezic, J. Severin, S. Olivarius, D. Lazarevic, N. Hornig, V. Orlando, I. Bell,

H. Gao, J. Dumais, P. Kapranov, H. Wang, C. A. Davis, T. R. Gingeras, J. Kawai,

C. O. Daub, Y. Hayashizaki, S. Gustincich, and P. Carninci, “Linking promoters to functional transcripts in small samples with nanoCAGE and CAGEscan.” *Nature Methods*, vol. 7, no. 7, pp. 528–534, Jul. 2010.

J. S. Cumbie, M. G. Ivanchenko, and M. Megraw, “NanoCAGE-XL and CapFilter:

an approach to genome wide identification of high confidence transcription start sites,” *BMC Genomics*, vol. 16, no. 1, p. 528, Aug. 2015.

T. Morton, J. Petricka, D. L. Corcoran, S. Li, C. M. Winter, A. Carda, P. N.

Benfey, U. Ohler, and M. Megraw, “Paired-end analysis of transcription start sites in Arabidopsis reveals plant-specific promoter signatures.” *The Plant cell*, vol. 26, no. 7, pp. 2746–2760, Jul. 2014.

ENCODE Project Consortium, “An integrated encyclopedia of DNA elements in

the human genome.” *Nature*, vol. 489, no. 7414, pp. 57–74, Sep. 2012.

E. Consortium. (2017) Rampage and cage data standards and processing pipeline. [Online]. Available: h[ttps://www.enco](http://www.encodeproject.org/rampage/)depro[ject.org/rampage/](http://www.encodeproject.org/rampage/)

632 24. N. Merchant, E. Lyons, S. Goff, M. Vaughn, D. Ware, D. Micklos, and P. Antin, 633 “The iPlant Collaborative: Cyberinfrastructure for Enabling Data to Discovery for 634 the Life Sciences.” *PLoS Biology*, vol. 14, no. 1, p. e1002342, Jan. 2016.

635 25. C. A. Stewart, T. M. Cockerill, I. Foster, D. Hancock, N. Merchant, 636 E. Skidmore, D. Stanzione, J. Taylor, S. Tuecke, G. Turner, M. Vaughn, 637 and N. I. Gaffney, “Jetstream: A self-provisioned, scalable science and 638 engineering cloud environment,” in *Proceedings of the 2015 XSEDE Conference:* 639 *Scientific Advancements Enabled by Enhanced Cyberinfrastructure*, ser. XSEDE 640 ’15. New York, NY, USA: ACM, 2015, pp. 29:1–29:8. [Online]. Available:

641 <http://doi.acm.org/10.1145/2792745.2792774>

642 26. R. Leinonen, H. Sugawara, M. Shumway, and International Nucleotide Sequence 643 Database Collaboration, “The sequence read archive.” *Nucleic Acids Research*, 644 vol. 39, no. Database issue, pp. D19–21, Jan. 2011.

645 27. E. Aronesty, “Comparison of Sequencing Utility Programs,” *The Open Bioinfor-*

646 *matics Journal*, vol. 7, no. 1, pp. 1–8, Jan. 2013.

647 28. H. Lab, “FASTX Toolkit.” [Online]. Available:

|  |  |  |
| --- | --- | --- |
| 648 |  | <http://hannonlab.cshl.edu/fastx_toolkit/> |
| 649 | 29. | T. Lassmann, “TagDust2: a generic method to extract reads from sequencing data,” |
| 650 |  | *BMC Bioinformatics*, vol. 16, no. 1, p. 1, Jan. 2015. |
| 651 | 30. | H. Li, B. Handsaker, A. Wysoker, T. Fennell, J. Ruan, N. Homer, G. Marth, G. R. |
| 652 |  | Abecasis, R. Durbin, and 1000 Genome Project Data Processing Subgroup, “The |
| 653 |  | Sequence Alignment/Map format and SAMtools,” *Bioinformatics (Oxford, Eng-* |
| 654 |  | *land)*, vol. 25, no. 16, pp. 2078–2079, Aug. 2009. |
| 655 | 31. | A. Dobin and T. R. Gingeras, “Optimizing RNA-Seq Mapping with STAR,” in |
| 656 |  | *Transcription Factor Regulatory Networks*. New York, NY: Springer New York, |
| 657 |  | Apr. 2016, pp. 245–262. |
| 658 | 32. | R Core Team, *R: A Language and Environment for Statistical Computing*, R |
| 659 |  | Foundation for Statistical Computing, Vienna, Austria, 2017. [Online]. Available: |
| 660 |  | https:/[/www.R-project.org](http://www.R-project.org/) |
| 661 | 33. | M. Lawrence and M. Morgan, “Scalable Genomics with R and Bioconductor,” |
| 662 |  | *Statistical Science*, vol. 29, no. 2, pp. 214–226, May 2014. |
| 663 | 34. | A. Yates, W. Akanni, M. R. Amode, D. Barrell, K. Billis, D. Carvalho-Silva, |
| 664 |  | C. Cummins, P. Clapham, S. Fitzgerald, L. Gil, C. G. GirÃşn, L. Gordon, |
| 665 |  | T. Hourlier, S. E. Hunt, S. H. Janacek, N. Johnson, T. Juettemann, S. Keenan, |
| 666 |  | I. Lavidas, F. J. Martin, T. Maurel, W. McLaren, D. N. Murphy, R. Nag, |
| 667 |  | M. Nuhn, A. Parker, M. Patricio, M. Pignatelli, M. Rahtz, H. S. Riat, |
| 668 |  | D. Sheppard, K. Taylor, A. Thormann, A. Vullo, S. P. Wilder, A. Zadissa, |
| 669 |  | E. Birney, J. Harrow, M. Muffato, E. Perry, M. Ruffier, G. Spudich, S. J. |
| 670 |  | Trevanion, F. Cunningham, B. L. Aken, D. R. Zerbino, and P. Flicek, “Ensembl |
| 671 |  | 2016,” *Nucleic Acids Research*, vol. 44, no. D1, pp. D710–D716, 2016. [Online]. |
| 672 |  | Available: + <http://dx.doi.org/10.1093/nar/gkv1157> |
| 673 | 35. | V. Haberle, A. R. R. Forrest, Y. Hayashizaki, P. Carninci, and B. Lenhard, |
| 674 |  | “CAGEr: precise TSS data retrieval and high-resolution promoterome mining for |
| 675 |  | integrative analyses.” *Nucleic Acids Research*, vol. 43, no. 8, pp. gkv054–e51, Feb. |
| 676 |  | 2015. |
| 677 | 36. | H. PagÃĺs, *BSgenome: Infrastructure for Biostrings-based genome data packages* |
| 678 |  | *and support for efficient SNP representation*, 2016, r package version 1.42.0. |
| 679 | 37. | H. Thorvaldsdottir, J. T. Robinson, and J. P. Mesirov, “Integrative Genomics |
| 680 |  | Viewer (IGV): high-performance genomics data visualization and exploration,” |
| 681 |  | *Briefings in Bioinformatics ()*, vol. 14, no. 2, pp. 178–192, Mar. 2013. |

|  |  |  |
| --- | --- | --- |
| 682 | 38. | E. W. E. Sayers, T. T. Barrett, D. A. D. Benson, E. E. Bolton, S. H. S. Bryant, |
| 683 |  | K. K. Canese, V. V. Chetvernin, D. M. D. Church, M. M. Dicuccio, S. S. Federhen, |
| 684 |  | M. M. Feolo, I. M. I. Fingerman, L. Y. L. Geer, W. W. Helmberg, Y. Y. Kapustin, |
| 685 |  | S. S. Krasnov, D. D. Landsman, D. J. D. Lipman, Z. Z. Lu, T. L. T. Madden, |
| 686 |  | T. T. Madej, D. R. D. Maglott, A. A. Marchler-Bauer, V. V. Miller, I. I. Karsch- |
| 687 |  | Mizrachi, J. J. Ostell, A. A. Panchenko, L. L. Phan, K. D. K. Pruitt, G. D. G. |
| 688 |  | Schuler, E. E. Sequeira, S. T. S. Sherry, M. M. Shumway, K. K. Sirotkin, D. D. |
| 689 |  | Slotta, A. A. Souvorov, G. G. Starchenko, T. A. T. Tatusova, L. L. Wagner, Y. Y. |
| 690 |  | Wang, W. J. W. Wilbur, E. E. Yaschenko, and J. J. Ye, “Database resources of the |
| 691 |  | National Center for Biotechnology Information.” *Nucleic Acids Research*, vol. 40, |
| 692 |  | no. Database issue, pp. D13–D25, Jan. 2012. |
| 693 | 39. | O. Tange, “Gnu parallel - the command-line power tool,” *;login: The* |
| 694 |  | *USENIX Magazine*, vol. 36, no. 1, pp. 42–47, Feb 2011. [Online]. Available: |
| 695 |  | <http://www.gnu.org/s/parallel> |
| 696 | **7** | **Checklist of Items to be Sent to Volume Editors** |

697

698

699

700

701

702

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

D The final LATEX source files

D A final PDF file

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

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