

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 and TSSrchitect, providing details instructions with the aim of taking the user from raw reads to processed results.

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

1 Introduction

1.1 TSS Profiling Identifies Promoters at Genome-Scale

The promoter, defined in eukaryotes as the genomic region bound by RNA Polymerase II immediately prior to transcription initiation [1], 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) [2], 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 [3], 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 [4] 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 [4] 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 [5][6]. The authors also showed a strong asso-
 34 ciation between promoter class and motif composition (consistent with previous
 35 findings [5, 7]). 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* [8]. 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 [9], fish [10] and mammals [11, 6].

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*) [12]. Using TSS-seq (specifically Oligo-capping; for details on this method

see [13]), 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 [12]. CpG island promoters (CPIs) form the largest class of promoter in mammals [14]; by contrast, CPIs are not known to exist as a class in invertebrates.

1.4 Paired-end TSS Profiling with RAMPAGE

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

2 Materials

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

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

2.1 Hardware Requirements

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

2.2 Software Requirements

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

2.3 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()
```

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

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

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

```

133 source("https://bioconductor.org/biocLite.R")
134 biocLite("TSRchitect")

```

```

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

```

```

137

```

```

138 library(TSRchitect)

```

139 3 Methods

140 3.1 Retrieving the RAMPAGE sequence data from NCBI's 141 Gene Expression Omnibus (GEO)

```

142 To begin our analysis, we must download the RAMPAGE data to our work-
143 station. We will utilize tools provided by the SRA Toolkit, which should
144 already be installed on your machine (see Materials). The command fastq-
145 dump allows one to directly retrieve data from the GEO database using
146 the appropriate identifier(s). While there are 36 RAMPAGE libraries in the
147 Batut et al. dataset, we will select a subset of these to analyze here. We
148 will compare samples from selected embryonic (E01h-E03h) and larval (L1-
149 L3) tissues, representing the beginning and end of embryonic development.
150 For more information about the experiment and the available RAMPAGE li-
151 braries, please see the following link: https://www.ncbi.nlm.nih.gov/Traces/study/?acc=SRP011193
152 First, let's proceed with the libraries from early embryonic tissues. Note
153 that since these fastq files are paired-end, we use the argument -split-files
154 to generate separate files for each read pair.

```

```

155 mkdir fastq_files #creating a new folder to house the downloaded files
156 cd fastq_files #moving into this directory
157 fastq-dump --split-files SRR424683
158 fastq-dump --split-files SRR424684
159 fastq-dump --split-files SRR424685

```

```

160 We continue by downloading the RAMPAGE libraries from late embryonic
161 tissues:

```

```

162 fastq-dump --split-files SRR424707
163 fastq-dump --split-files SRR424708
164 fastq-dump --split-files SRR424709

```

```

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

```

```

167 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:

```
mkdir symlinks

#embryonic libraries
ln -s SRR424683_1.fastq symlinks/E01h.R1.clipped.fq
ln -s SRR424683_2.fastq symlinks/E01h.R2.clipped.fq
ln -s SRR424684_1.fastq symlinks/E02h.R1.clipped.fq
ln -s SRR424684_2.fastq symlinks/E02h.R2.clipped.fq
ln -s SRR424685_1.fastq symlinks/E03h.R1.clipped.fq
ln -s SRR424685_2.fastq symlinks/E03h.R2.clipped.fq

#larval libraries
ln -s SRR424707_1.fastq symlinks/L1.R1.clipped.fq
ln -s SRR424707_2.fastq symlinks/L1.R2.clipped.fq
ln -s SRR424708_1.fastq symlinks/L2.R1.clipped.fq
ln -s SRR424708_2.fastq symlinks/L2.R2.clipped.fq
ln -s SRR424709_1.fastq symlinks/L3.R1.clipped.fq
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:

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

The rRNA sequences are found at the following link: <https://iu.box.com/s/3a5lqbo58qlykhmqxw00h2uo>. You should see a file entitled "Dmel_rRNA.fasta" in your current directory.

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

3.4 Filtering and alignment of RAMPAGE reads using GoRAMPAGE

At this stage we are ready to commence with the rRNA filtering and alignment of the RAMPAGE libraries. We will use GoRAMPAGE, a tool we developed, to perform these tasks in a concerted workflow. GoRAMPAGE runs TagDust [15] to remove rRNA and low-complexity reads, and uses STAR [16] to align RAMPAGE (or other paired-end) reads to a given genome assembly.

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

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

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

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

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

```
less -S errScript
```

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

Inspecting the rRNA filtering results To evaluate the results from Step 3 (rRNA filtering), please navigate to the top level of the "output" directory and open the file "LOGFILES". You'll see the recorded progress of the program Tagdust and a record of the results. We notice that (for the L3h library)

1046448 of reads (78.1%) were "extracted", meaning that slightly more than 20% of reads were removed because of matches with ribosomal sequences. The removed reads from all libraries are found in the "dusted_discard" directory, and the extracted reads are found in the current directory. Due to their sheer abundance within cells, ribosomal RNA sequences are an inevitable contaminant within TSS profiling libraries. For analysis purposes, it is important that these sequences be removed, which is what has been completed here. Since this step was conducted appropriately, we can proceed to the next step.

Evaluating the alignments The folder "alignments/" in your 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 [17], 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 *BSGenome* 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 [18]. It makes use of existing packages and data structures within this environment, where available, to identify promoters from sequence alignments. Since you have already installed *TSRchitect* and its dependencies (see section 2.3), we are set to proceed.

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

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

304 R CMD BATCH MMB_chapter_TSRchitect.R #assumes variables BAMDIR and DmAnnot have already
 305 bg #puts the job in the background

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

310 Before we evaluate the results (which will have been written to your working
 311 directory after running the batch script), there are some important param-
 312 eters to review. First, we must initialize the *tssObject* (which stores the
 313 information about the experiment) appropriately.
 314

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

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

324 A critical step in our analysis is identifying TSRs from the aligned TSS
 325 data; to do this we use the function *determineTSR*. We have selected the
 326 job to run on 4 cores in this example (*n.cores*=4). Please enter the number
 327 of cores appropriate for your system. Because we want to identify TSRs
 328 from every one of the selected RAMPAGE libraries, we specify *tssSet*="all".

329 The parameter *tagCountThreshold* was set to 25, meaning that only TSSs
 330 supported by 25 or more 5' RAMPAGE reads will be included within a TSR.
 331 Setting *writeTable* to "TRUE" means that the identified TSRs from each set
 332 will be written to the working directory.

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

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

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

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

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

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

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

```
361 #merging the sample data into two groups
362 DmRAMPAGE <- mergeSampleData(DmRAMPAGE)
```

```
363
364 # ... identifying TSRs from the merged samples:
365 DmRAMPAGE <- determineTSR(experimentName=DmRAMPAGE, \
366 n.cores=4, tsrSetType="merged", \
367 tssSet="all", tagCountThreshold=40, \
368 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.* [4].

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 FBgn0
2L.102386.102386.+		2L	102386	102386	+	284 1	2 FBgn0

3.6 Summary

The workflow provided here is intended to serve as a useful entry point for the analysis of TSS profiling data in insects. On the computational side, we

have provided an open source set of tools so that the uninitiated genome scientist can begin to analyze RAMPAGE (or other forms of TSS profiling data) quickly. While the analysis centered on *D. melanogaster* via the use of public datasets, it is anticipated that this will assist groups who may be interested in performing TSS profiling in their preferred insect model system. The application of TSS profiling technology across a more representative sample of insect diversity will improve our understanding of the positions and general structure *cis*-regulatory regions in this phylum.

Acknowledgments

The authors would like to thank Philippe Batut for generous technical assistance with the RAMPAGE protocol, and to Nathan Keith for his help establishing the protocol in our laboratory.

Disclosure Declaration

The authors declare that they have no competing interests.

4 Figures

For L^AT_EX users, we recommend using the *graphics* or *graphicx* package and the `\includegraphics` command.

Please check that the lines in line drawings are not interrupted and are of a constant width. Grids and details within the figures must be clearly legible and may not be written one on top of the other. Line drawings should have a resolution of at least 800 dpi (preferably 1200 dpi). The lettering in figures should have a height of 2 mm (10-point type). Figures should be numbered and should have a caption which should always be positioned *under* the figures, in contrast to the caption belonging to a table, which should always appear *above* the table; this is simply achieved as matter of sequence in your source.

Please center the figures or your tabular material by using the `\centering` declaration. Short captions are centered by default between the margins and typeset in 9-point type (Fig. 1 shows an example). The distance between text and figure is preset to be about 8 mm, the distance between figure and caption about 6 mm.

To ensure that the reproduction of your illustrations is of a reasonable quality, we advise against the use of shading. The contrast should be as pronounced as possible.

If screenshots are necessary, please make sure that you are happy with the print quality before you send the files.

Please define figures (and tables) as floating objects. Please avoid using optional location parameters like “[h]” for “here”.

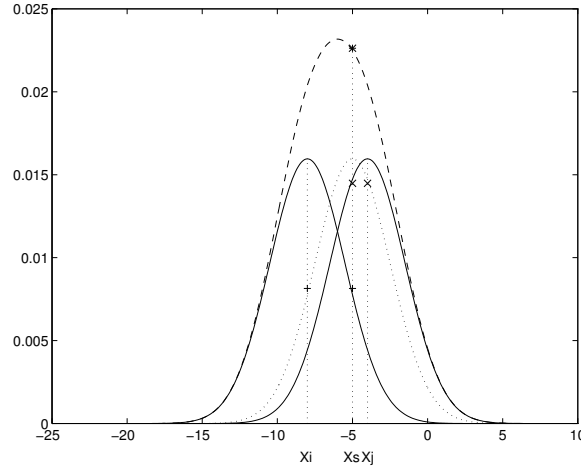


Fig. 1. One kernel at x_s (*dotted kernel*) or two kernels at x_i and x_j (*left and right*) lead to the same summed estimate at x_s . This shows a figure consisting of different types of lines. Elements of the figure described in the caption should be set in *italics*, in parentheses, as shown in this sample caption.

4.1 Formulas

Displayed equations or formulas are centered and set on a separate line (with an extra line or halflin space above and below). Displayed expressions should be numbered for reference. The numbers should be consecutive within each section or within the contribution, with numbers enclosed in parentheses and set on the right margin – which is the default if you use the *equation* environment, e.g.,

$$\psi(u) = \int_o^T \left[\frac{1}{2} (\Lambda_o^{-1} u, u) + N^*(-u) \right] dt . \quad (1)$$

Equations should be punctuated in the same way as ordinary text but with a small space before the end punctuation mark.

4.2 Footnotes

The superscript numeral used to refer to a footnote appears in the text either directly after the word to be discussed or – in relation to a phrase or a sentence – following the punctuation sign (comma, semicolon, or period). Footnotes should appear at the bottom of the normal text area, with a line of about 2 cm set immediately above them.¹

¹ The footnote numeral is set flush left and the text follows with the usual word spacing.

5 References

References

1. J. T. Kadonaga, "Perspectives on the RNA polymerase II core promoter." *Wiley Interdisciplinary Reviews: Developmental Biology*, vol. 1, no. 1, pp. 40–51, Jan. 2012.
2. R. Kodzius, M. Kojima, H. Nishiyori, M. Nakamura, S. Fukuda, M. Tagami, D. Sasaki, K. Imamura, C. Kai, M. Harbers, Y. Hayashizaki, and P. Carninci, "CAGE: cap analysis of gene expression." *Nature Methods*, vol. 3, no. 3, pp. 211–222, Mar. 2006.
3. P. Carninci, T. Kasukawa, S. Katayama, J. Gough, M. C. Frith, N. Maeda, R. Oyama, T. Ravasi, B. Lenhard, C. Wells, R. Kodzius, K. Shimokawa, V. B. Bajic, S. E. Brenner, S. Batalov, A. R. R. Forrest, M. Zavolan, M. J. Davis, L. G. Wilming, V. Aidinis, J. E. Allen, A. Ambesi-Impimbato, R. Apweiler, R. N. Aturaliya, T. L. Bailey, M. Bansal, L. Baxter, K. W. Beisel, T. Bersano, H. Bono, A. M. Chalk, K. P. Chiu, V. Choudhary, A. Christoffels, D. R. Clutterbuck, M. L. Crowe, E. Dalla, B. P. Dalrymple, B. de Bono, G. Della Gatta, D. di Bernardo, T. Down, P. Engstrom, M. Fagiolini, G. Faulkner, C. F. Fletcher, T. Fukushima, M. Furuno, S. Futaki, M. Gariboldi, P. Georgii-Hemming, T. R. Gingeras, T. Gojobori, R. E. Green, S. Gustincich, M. Harbers, Y. Hayashi, T. K. Hensch, N. Hirokawa, D. Hill, L. Huminiecki, M. Iacono, K. Ikeo, A. Iwama, T. Ishikawa, M. Jakt, A. Kanapin, M. Katoh, Y. Kawasawa, J. Kelso, H. Kitamura, H. Kitano, G. Kollias, S. P. T. Krishnan, A. Kruger, S. K. Kummerfeld, I. V. Kurochkin, L. F. Lareau, D. Lazarevic, L. Lipovich, J. Liu, S. Liuni, S. McWilliam, M. Madan Babu, M. Madera, L. Marchionni, H. Matsuda, S. Matsuzawa, H. Miki, F. Mignone, S. Miyake, K. Morris, S. Mottagui-Tabar, N. Mulder, N. Nakano, H. Nakauchi, P. Ng, R. Nilsson, S. Nishiguchi, S. Nishikawa, F. Nori, O. Ohara, Y. Okazaki, V. Orlando, K. C. Pang, W. J. Pavan, G. Pavesi, G. Pesole, N. Petrovsky, S. Piazza, J. Reed, J. F. Reid, B. Z. Ring, M. Ringwald, B. Rost, Y. Ruan, S. L. Salzberg, A. Sandelin, C. Schneider, C. Schönbach, K. Sekiguchi, C. A. M. Semple, S. Seno, L. Sessa, Y. Sheng, Y. Shibata, H. Shimada, K. Shimada, D. Silva, B. Sinclair, S. Sperling, E. Stupka, K. Sugiura, R. Sultana, Y. Takenaka, K. Taki, K. Tammoja, S. L. Tan, S. Tang, M. S. Taylor, J. Tegner, S. A. Teichmann, H. R. Ueda, E. van Nimwegen, R. Verardo, C. L. Wei, K. Yagi, H. Yamanishi, E. Zabarovsky, S. Zhu, A. Zimmer, W. Hide, C. Bult, S. M. Grimmond, R. D. Teasdale, E. T. Liu, V. Brusic, J. Quackenbush, C. Wahlestedt, J. S. Mattick, D. A. Hume, C. Kai, D. Sasaki, Y. Tomaru, S. Fukuda, M. Kanamori-Katayama, M. Suzuki, J. Aoki, T. Arakawa, J. Iida, K. Imamura, M. Itoh, T. Kato, H. Kawaji, N. Kawagashira, T. Kawashima, M. Kojima, S. Kondo, H. Konno, K. Nakano, N. Ninomiya, T. Nishio, M. Okada, C. Plessy, K. Shibata, T. Shiraki, S. Suzuki, M. Tagami, K. Waki, A. Watahiki, Y. Okamura-Oho, H. Suzuki, J. Kawai, Y. Hayashizaki, F. Consortium, R. G. E. R. Group, and G. S. G. G. N. P. C. Group, "The transcriptional landscape of the mammalian genome," *Science (New York, NY)*, vol. 309, no. 5740, pp. 1559–1563, Sep. 2005.
4. R. A. Hoskins, R. A. Hoskins, J. M. Landolin, J. M. Landolin, J. B. Brown, J. B. Brown, J. E. Sandler, J. E. Sandler, H. Takahashi, H. Takahashi, T. Lassmann, T. Lassmann, C. Yu, C. Yu, B. W. Booth, B. W. Booth, D. Zhang, D. Zhang, K. H. Wan, K. H. Wan, L. Yang, L. Yang, N. Boley, N. Boley, J. Andrews, J. Andrews, T. C. Kaufman, T. C. Kaufman, B. R. Graveley, B. R. Graveley, P. J.

- 512 Bickel, P. J. Bickel, P. Carninci, J. W. Carlson, J. W. Carlson, S. E. Celniker,
513 and S. E. Celniker, "Genome-wide analysis of promoter architecture in *Drosophila*
514 *melanogaster*." *Genome Research*, vol. 21, no. 2, pp. 182–192, Feb. 2011.
- 515 5. E. A. Rach, H.-Y. Yuan, W. H. Majoros, P. Tomancak, and U. Ohler, "Motif
516 composition, conservation and condition-specificity of single and alternative tran-
517 scription start sites in the *Drosophila* genome." *Genome Biology*, vol. 10, no. 7, p.
518 R73, 2009.
- 519 6. B. Lenhard, A. Sandelin, and P. Carninci, "Metazoan promoters: emerging char-
520 acteristics and insights into transcriptional regulation." *Nature Reviews Genetics*,
521 vol. 13, no. 4, pp. 233–245, Apr. 2012.
- 522 7. T. Ni, D. L. Corcoran, E. A. Rach, S. Song, E. P. Spana, Y. Gao, U. Ohler,
523 and J. Zhu, "A paired-end sequencing strategy to map the complex landscape of
524 transcription initiation." *Nature Methods*, vol. 7, no. 7, pp. 521–527, Jul. 2010.
- 525 8. U. Ohler, G.-c. Liao, H. Niemann, and G. M. Rubin, "Computational analysis of
526 core promoters in the *Drosophila* genome." *Genome Biology*, vol. 3, no. 12, pp.
527 research0087.1–0087.12, 2002.
- 528 9. R. T. Raborn, K. Spitze, V. P. Brendel, and M. Lynch, "Promoter Architecture
529 and Sex-Specific Gene Expression in *Daphnia pulex*." *Genetics*, vol. 204, no. 2, pp.
530 593–612, Aug. 2016.
- 531 10. C. Nepal, Y. Hadzhiev, C. Previti, V. Haberle, N. Li, H. Takahashi, A. M. M.
532 Suzuki, Y. Sheng, R. F. Abdelhamid, S. Anand, J. Gehrig, A. Akalin, C. E. M.
533 Kockx, A. A. J. van der Sloot, W. F. J. van IJcken, O. Armant, S. Rastegar,
534 C. Watson, U. Strahle, E. Stupka, P. Carninci, B. Lenhard, and F. Muller, "Dy-
535 namic regulation of the transcription initiation landscape at single nucleotide res-
536 olution during vertebrate embryogenesis," *Genome Research*, vol. 23, no. 11, pp.
537 1938–1950, Nov. 2013.
- 538 11. P. Carninci, A. Sandelin, B. Lenhard, S. Katayama, K. Shimokawa, J. Ponjavic,
539 C. A. M. Semple, M. S. Taylor, P. G. Engström, M. C. Frith, A. R. R. For-
540 rest, W. B. Alkema, S. L. Tan, C. Plessy, R. Kodzius, T. Ravasi, T. Kasukawa,
541 S. Fukuda, M. Kanamori-Katayama, Y. Kitazume, H. Kawaji, C. Kai, M. Naka-
542 mura, H. Konno, K. Nakano, S. Mottagui-Tabar, P. Arner, A. Chesi, S. Gustincich,
543 F. Persichetti, H. Suzuki, S. M. Grimmond, C. A. Wells, V. Orlando, C. Wahlest-
544 edt, E. T. Liu, M. Harbers, J. Kawai, V. B. Bajic, D. A. Hume, and Y. Hayashizaki,
545 "Genome-wide analysis of mammalian promoter architecture and evolution," *Na-
546 ture Genetics*, vol. 38, no. 6, pp. 626–635, Apr. 2006.
- 547 12. S. Mwangi, G. Attardo, Y. Suzuki, S. Aksoy, and A. Christoffels, "TSS seq based
548 core promoter architecture in blood feeding Tsetse fly (*Glossina morsitans mor-
549 sitans*) vector of Trypanosomiasis," *BMC Genomics*, vol. 16, no. 1, p. 722, Sep.
550 2015.
- 551 13. K. Tsuchihara, Y. Suzuki, H. Wakaguri, T. Irie, K. Tanimoto, S.-i. Hashimoto,
552 K. Matsushima, J. Mizushima-Sugano, R. Yamashita, K. Nakai, D. Bentley, H. Es-
553 umi, and S. Sugano, "Massive transcriptional start site analysis of human genes in
554 hypoxia cells," *Nucleic Acids Research*, vol. 37, no. 7, pp. 2249–2263, Apr. 2009.
- 555 14. N. Cvetesic and B. Lenhard, "Core promoters across the genome," *Nature Biotech-
556 nology*, vol. 35, no. 2, pp. 123–124, Feb. 2017.
- 557 15. T. Lassmann, "TagDust2: a generic method to extract reads from sequencing data,"
558 *BMC Bioinformatics*, vol. 16, no. 1, p. 1, Jan. 2015.
- 559 16. A. Dobin and T. R. Gingeras, "Optimizing RNA-Seq Mapping with STAR," in
560 *Transcription Factor Regulatory Networks*. New York, NY: Springer New York,
561 Apr. 2016, pp. 245–262.

- 562 17. V. Haberle, “CAGEr: an R package for CAGE (Cap Analysis of Gene Expression)
563 data analysis and promoterome mining,” 2013.
- 564 18. M. Lawrence and M. Morgan, “Scalable Genomics with R and Bioconductor,”
565 *Statistical Science*, vol. 29, no. 2, pp. 214–226, May 2014.

566 6 Checklist of Items to be Sent to Volume Editors

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

- 568 ☐ The final L^AT_EX source files
- 569 ☐ A final PDF file
- 570 ☐ A copyright form, signed by one author on behalf of all of the authors of the
571 paper.
- 572 ☐ A readme giving the name and email address of the corresponding author.