

# Using RAMPAGE to identify and annotate promoters in insect genomes

R. Taylor Raborn<sup>\*1,2</sup> and Volker P. Brendel<sup>1,2</sup>

<sup>1</sup>Department of Biology, Indiana University

<sup>2</sup>School of Informatics and Computing, Indiana University

Department of Biology and School of Informatics and Computing,  
Indiana University

212 S. Hawthorne Drive 205 Simon Hall, Bloomington, IN 47401, USA

<http://www.brendelgroup.org>

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

In this chapter we highlight the utility of one TSS profiling method, RAMPAGE (RNA annotation and mapping of promoters for analysis of gene expression), for the precise, quantitative identification of promoters in insect genomes. We demonstrate this using our tools GoRAMPAGE [3] and TSRchitect [4], providing details instructions with the aim of taking the user from raw reads to processed results.

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

## 1 Introduction

### 1.1 TSS Profiling Identifies Promoters at Genome-Scale

The promoter, defined in eukaryotes as the genomic region bound by RNA Polymerase II immediately prior to transcription initiation [5], is the site where regulatory signals unite to direct gene expression. The identification of promoter regions is a valuable step for understanding the *cis*-regulatory signals that are present in an organism, and is also important for genome annotation. However, despite the rapid accumulation of genome sequences across metazoan and arthropod diversity, accurate annotation of promoter regions remains sparse. This is because—absent empirically-defined information—precisely identifying

---

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

sequence motifs that demarcate the promoter is unreliable. In contrast with current *in silico* approaches, direct mapping of TSSs identifies the location of the core promoter. Cap Analysis of Gene Expression (CAGE) [6], one of the first methods devised to identify 5'-ends of mRNAs at large-scale, involves selective capture of 5'-capped transcripts, first-strand reverse-transcription and ligation of a short oligonucleotide (CAGE tag). CAGE was initially utilized by the FANTOM (Functional Annotation of the Mammalian Genome) consortium to identify promoter architecture in human and mouse [7], providing the first glimpse of the global landscape of transcription initiation. At the onset of the NGS era, CAGE was coupled with massively-parallel sequencing to generate 5'-ends of mRNAs at substantially higher scale. This advance provided more extensive coverage of the expressed transcriptome, and provided increased sensitivity for quantitative measurements *i.e.* measurement of promoter activity.

## 1.2 Promoter Architecture of *Drosophila melanogaster*

Hoskins and colleagues [1] performed CAGE in *D. melanogaster* as part of the modENCODE consortium, identifying promoters at large-scale and characterizing the promoter architecture of an insect genome for the first time. Hoskins [1] indicated that TSS distributions at *Drosophila* promoters exhibit a range of shapes that can be generally grouped into two major classifications: *peaked* and *broad*. Peaked promoters have a single, major TSS position occupying a narrow genomic region, whereas broad promoters lack a single, major TSS and contain TSSs across a wider region [8, 9]. The authors also showed a strong association between promoter class and motif composition (consistent with previous findings [8, 10]). Peaked promoters were associated with positionally-enriched *cis*-regulatory motifs including TATA, Initiator (Inr) and DPE, while broad promoters contained an enrichment of less-well characterized motifs, including *Ohler6* and *Ohler7* [11]. The existence of two promoter classes appears to be conserved among metazoans, and has been reported (using TSS profiling methodologies) in insects, cladocerans [12], fish [13] and mammals [14, 9].

## 1.3 Promoter Structure of Insects

Beyond *D. melanogaster*, few investigations have utilized TSS profiling in insect genomes. As a consequence, what is known about promoter architecture in insects is largely restricted to the *Drosophila* genus. As part of the modENCODE effort, CAGE was performed in multiple tissues and developmental stages of the *Drosophila pseudoobscura*. TSSs were found to be highly similar between species: more than 80% of TSSs (81%) of aligned, CAGE-identified TSSs from *D. pseudoobscura* were positioned within 20nt of their counterparts in *D. melanogaster*. An enrichment of the CA dinucleotide was detected at the TSS ( $[-1, +1]$ ), and the motifs corresponding to TATA, Inr and DPE were positioned at the same locations relative to the TSS in both species. The one other insect species for

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

#### 1.4 Paired-end TSS Profiling with RAMPAGE

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

## 2 Materials

The analyses described herein require a workstation capable of doing modern bioinformatics, including a reasonably-appointed laptop. An intermediate understanding of the Linux/Unix command line will be extremely useful, although

we make efforts to explain the procedures with clarity. In addition, it will likely be necessary for the participant to have superuser privileges on the machine. If you do not have a machine (or have access to one) that meets these requirements, it is recommended that you consider cloud-based cyberinfrastructure, including Amazon Web Services (AWS; <https://aws.amazon.com/>) or CyVerse (<http://www.cyverse.org/>) [19]. The former is a well-known pay-per-use solution, while the latter is an NSF-funded resource that makes compute allocations freely available to the public.

## 2.1 Hardware

1. x86-64 compatible processors
2. At least 8GB RAM
3. 30GB+ hard disk space

## 2.2 Operating System

- 64 bit Linux (preferred) or Mac OS X (with Command Line Tools from XCode)

## 2.3 Software

Below is a list of the software packages required for this demonstration (*see Note 1*).

### Sequence retrieval

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

### GoRAMPAGE

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

### TSRchitect

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

## 2.4 Online Appendix

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

## 130 2.5 Installation of R packages

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

```
135 source("https://bioconductor.org/biocLite.R")
136 biocLite()
```

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

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

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

```
148 source("https://bioconductor.org/biocLite.R")
149 biocLite("TSRchitect")
```

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

```
152 library(TSRchitect) #installing TSRchitect
```

## 153 3 Methods

### 154 3.1 Retrieving the RAMPAGE sequence data from NCBI

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

165  
 166 First, let's proceed with the libraries from early embryonic tissues (see **See**  
 167 **Note 3**).

```

168 mkdir fastq_files #creating a new folder to house the downloaded files
169 cd fastq_files #moving into this directory
170 fastq-dump --split-files SRR424683
171 fastq-dump --split-files SRR424684
172 fastq-dump --split-files SRR424685

```

173 We continue by downloading the RAMPAGE libraries from late embryonic tis-  
 174 sues:

```

175 fastq-dump --split-files SRR424707
176 fastq-dump --split-files SRR424708
177 fastq-dump --split-files SRR424709

```

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

```

180 ls -l *.fastq | wc -l

```

### 181 3.2 Creating symlinks to the files

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

```

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

```

### 202 3.3 Downloading genomic data from *D. melanogaster*

203 Now that we have the fastq files from the RAMPAGE libraries downloaded and  
 204 named appropriately, we now must retrieve the genome assembly and rRNA  
 205 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.

211

212 Please download the assembly from the ENSEMBL database as follows:

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

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

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

### 221 3.4 Filtering and alignment of RAMPAGE reads using 222 GoRAMPAGE

223 At this stage we are ready to commence with the rRNA filtering and alignment  
224 of the RAMPAGE libraries. We will use GoRAMPAGE, a tool we developed,  
225 to perform these tasks in a concerted workflow. GoRAMPAGE runs TagDust  
226 [23] to remove rRNA and low-complexity reads, and uses STAR [25] to align  
227 RAMPAGE (or other paired-end) reads to a given genome assembly.

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

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

235 **Setting up the GoRAMPAGE job.** Now, once this is complete, please  
236 copy the contents of the "symlinks" directory that you created earlier (*i.e.*  
237 all of the \*.fq files) into the "clipped/" directory. Please refer to the script  
238 "GoRAMPAGE\_script\_MMB.sh" and (using a text editor) provide the appropriate  
239 paths to the genome assembly, output directory (see above) and rRNA sequences  
240 (see **Note 4**). GoRAMPAGE jobs can optionally be run in parallel (see **Note**  
241 **5**). The script can be executed as follows:

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

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

247 `less -S errScript`

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

251 **Inspecting the rRNA filtering results.** To evaluate the results from Step  
 252 3 (rRNA filtering), please navigate to the top level of the "output" directory  
 253 and open the file "LOGFILES". You'll see the recorded progress of the program  
 254 Tagdust and a record of the results. We notice that (for the L3h library) 1046448  
 255 of reads (78.1%) were "extracted", meaning that slightly more than 20% of  
 256 reads were removed because of matches with ribosomal sequences. The removed  
 257 reads from all libraries are found in the "dusted\_discard" directory, and the  
 258 extracted reads are found in the current directory. Due to their sheer abundance  
 259 within cells, ribosomal RNA sequences are an inevitable contaminant within TSS  
 260 profiling libraries. For analysis purposes, it is important that these sequences be  
 261 removed, which is what has been completed here.  
 262 Since this step was conducted appropriately, we can proceed to the next step.

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

275  
 276 Now that our RAMPAGE libraries are filtered and aligned, we can commence  
 277 with the second half of our analysis.

### 278 3.5 Promoter identification from aligned RAMPAGE libraries

279 We can now use the prepared alignment files to identify TSSs and promoters from  
 280 the selected RAMPAGE libraries. There are currently several tools available  
 281 for this purpose. *CAGEr*, developed by Haberle [28], was utilized to perform  
 282 TSS identification as part of the FANTOM5 efforts. We will use *TSRchitect* in



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

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

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

304 **Running the Analysis.** To run *TSRchitect* using the batch script, provide  
 305 full paths for the variables "BAMDIR" and "DmAnnot" in the script provided  
 306 (see **Note 6**). *BAMDIR* should be a path to the subdirectory "alignments/" in  
 307 RAMPAGE output directory you specified earlier, and *DmAnnot* should be a  
 308 full path to the *D. melanogaster* gene annotation listed above.

309 Once this is complete, we can run the batch script from the Linux command-line  
 310 as follows:

```
311 R CMD BATCH MMB_chapter_TSRchitect.R
312 #assumes variables BAMDIR and DmAnnot have already been set
313 bg #puts this job in the background
```

314 Once the job is underway, you can monitor its progress by looking at the con-  
 315 tents of the .Rout file (in this case, "MMB\_chapter\_TSRchitect.Rout"). The job  
 316 should complete within an hour on most systems.

317

318 **Reviewing the *TSRchitect* script.** Before we evaluate the results (which  
 319 will have been written to your working directory after running the batch script),  
 320 there are some important aspects of the analysis to review. We discuss these for  
 321 informational purposes only; it will not necessary to perform these commands  
 322 separate from the batch script provided. First, we must initialize the *tssObject*

(which stores the information about the experiment) appropriately (*see Note 7*).

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

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

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

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

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

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

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

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

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

Now we have generated a set of identified TSSs, TSRs from all 6 RAMPAGE libraries, and have associated the identified TSRs with annotated genes. Next, we will merge the libraries into two samples according to condition: early embryonic

364 (E1h, E2h, E3h) and late larval (L1, L2, L3) using the information we provided  
 365 when we initialized the *tssObject* at the start of this section. After merging, we  
 366 identify promoters i) within the merged samples and ii) within the entire dataset  
 367 combined, and associate with the *D. melanogaster* gene annotation as described  
 368 previously (not shown).

```
369 #merging the sample data into two groups
370 DmRAMPAGE <- mergeSampleData(DmRAMPAGE)
371
372 # ... identifying TSRs from the merged samples:
373 DmRAMPAGE <- determineTSR(experimentName=DmRAMPAGE, \
374   n.cores=4, tsrSetType="merged", \
375   tssSet="all", tagCountThreshold=40, \
376   clustDist=20, writeTable=TRUE)
```

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

- 379 – TSSs from each sample *e.g.* TSSset-1.txt: (6)
- 380 – TSRs from each sample (in both .txt and .tab formats): (12)
- 381 – TSRs from each merged group (in both .txt and .tab formats): *e.g.* TSRsetMerged-  
 382 1.txt: (4)
- 383 – TSRs from the combined set of TSSs: TSRsetCombined.tab: (1)

384 Let's briefly review the files. We can quickly obtain the counts on the command  
 385 line, as follows:

```
386 wc -l *.tab
387 8377 TSRset-1.tab
388 6159 TSRset-2.tab
389 4814 TSRset-3.tab
390 17924 TSRset-4.tab
391 11851 TSRset-5.tab
392 3242 TSRset-6.tab
393 13986 TSRsetCombined.tab
394 7344 TSRsetMerged-1.tab
395 12126 TSRsetMerged-2.tab
396 85823 total
```

397 We will see that we have identified between roughly 3,200 and 18,000 TSRs  
 398 within the individual RAMPAGE samples, which is attributable to the dif-  
 399 ferences in library sizes. We detect 7,344 TSRs within the early embryonic  
 400 samples ("TSRsetMerged-1.tab") and 12,126 TSRs in the late larval samples  
 401 ("TSRsetMerged-2.tab"). Within the combined samples ("TSRsetCombined.tab")  
 402 we find 13,986 TSRs, which is similar to the number reported by Hoskins *et. al.*  
 403 [1].  
 404

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

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

### 3.6 Summary

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

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

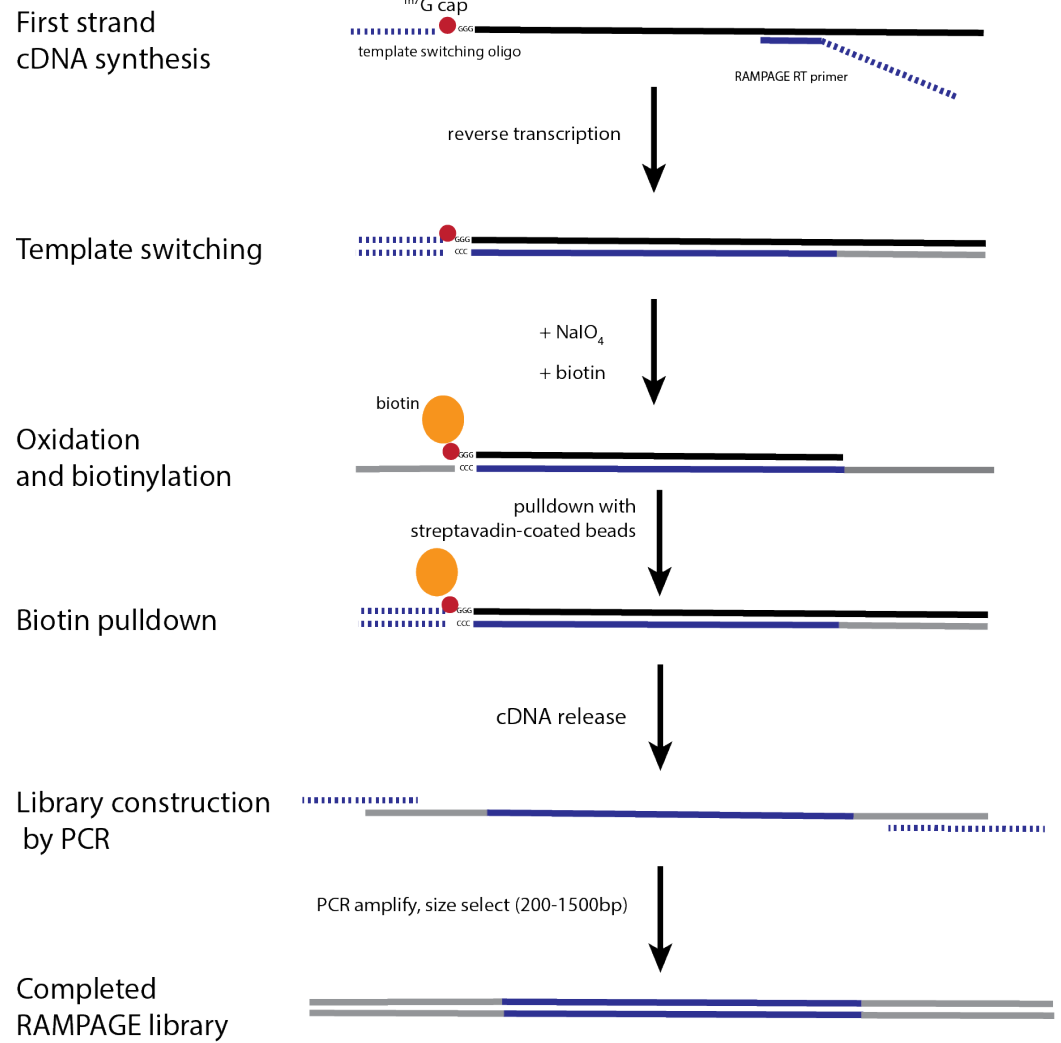
### 3.7 Figures

## 4 Notes

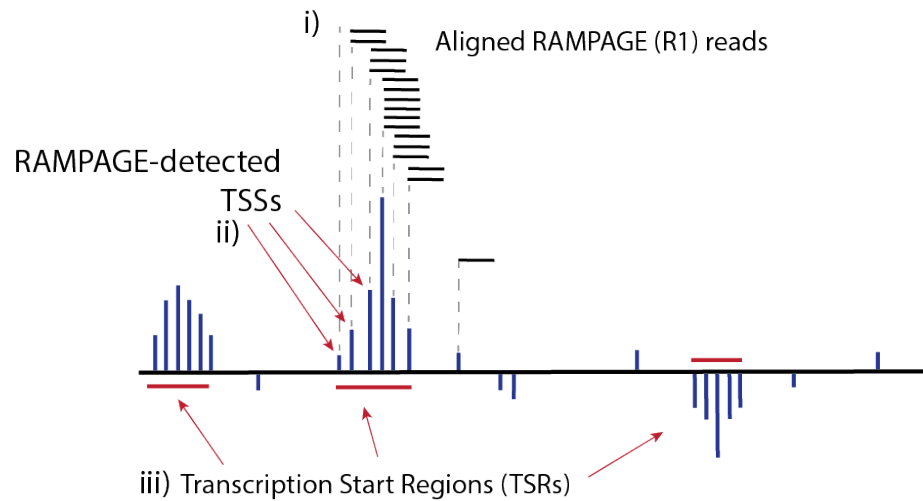
1. Please consult the GoRAMPAGE documentation found here:  
<https://github.com/BrendelGroup/GoRAMPAGE>. Installation instructions for the prerequisites of GoRAMPAGE (which includes some of the items listed) are found at the following link:  
<https://github.com/BrendelGroup/GoRAMPAGE/tree/master/src>.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

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

- The "scripts/" folder in the Appendix contains code for you to run the two major workflows described in this chapter. The "additional\_files/" folder contains the following files which are necessary for the analysis: i) a fasta file containing ribosomal RNA sequences for *D. melanogaster* (*Dmel\_rRNA.fasta*) and ii) a gene annotation for *D. melanogaster* (*Drosophila\_melanogaster.BDGP5.78.gff*).
3. Since these fastq files are paired-end, we use the argument *-split-files* to generate separate files for each read pair.



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



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

4. If you are running this on a cluster with a job scheduler you'll need to add the necessary headers to the top of the script and submit the job in the appropriate manner.
5. For parallel execution, GoRAMPAGE uses the Linux package *GNU parallel* [29]. Please see the GoRAMPAGE documentation for more information.
6. To do this, please edit the batch script `TSRchitect_script_MMB.R` with a text editor of your choice.
7. Because the samples provided derive from related developmental stages, we will merge them for annotation purposes using the argument *replicateIDs*, (though it must be emphasized that they are not replicates).

### Acknowledgments

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

### Disclosure Declaration

The authors declare that they have no competing interests.

## 5 References

### References

1. R. A. Hoskins, R. A. Hoskins, J. M. Landolin, J. M. Landolin, J. B. Brown, J. B. Brown, J. E. Sandler, J. E. Sandler, H. Takahashi, H. Takahashi, T. Lassmann,

- 463 T. Lassmann, C. Yu, C. Yu, B. W. Booth, B. W. Booth, D. Zhang, D. Zhang,  
464 K. H. Wan, K. H. Wan, L. Yang, L. Yang, N. Boley, N. Boley, J. Andrews, J. An-  
465 drews, T. C. Kaufman, T. C. Kaufman, B. R. Graveley, B. R. Graveley, P. J.  
466 Bickel, P. J. Bickel, P. Carninci, J. W. Carlson, J. W. Carlson, S. E. Celniker,  
467 and S. E. Celniker, "Genome-wide analysis of promoter architecture in *Drosophila*  
468 *melanogaster*." *Genome Research*, vol. 21, no. 2, pp. 182–192, Feb. 2011.
- 469 2. P. J. Batut, A. Dobin, C. Plessy, P. Carninci, and T. R. Gingeras, "High-fidelity  
470 promoter profiling reveals widespread alternative promoter usage and transposon-  
471 driven developmental gene expression." *Genome Research*, Aug. 2012.
- 472 3. V. P. Brendel and R. T. Raborn, "Gorampage- a workflow for promoter detection  
473 by 5'-read mapping," <https://github.com/brendelGroup/GoRAMPAGE>, 2016.
- 474 4. R. T. Raborn and V. Brendel, *TSRchitect: Promoter identification from large-scale*  
475 *TSS profiling data*, 2017, r Bioconductor package version 1.0.0. [Online]. Available:  
476 <http://bioconductor.org/packages/release/bioc/html/TSRchitect.html>
- 477 5. J. T. Kadonaga, "Perspectives on the RNA polymerase II core promoter." *Wiley*  
478 *Interdisciplinary Reviews: Developmental Biology*, vol. 1, no. 1, pp. 40–51, Jan.  
479 2012.
- 480 6. R. Kodzius, M. Kojima, H. Nishiyori, M. Nakamura, S. Fukuda, M. Tagami,  
481 D. Sasaki, K. Imamura, C. Kai, M. Harbers, Y. Hayashizaki, and P. Carninci,  
482 "CAGE: cap analysis of gene expression." *Nature Methods*, vol. 3, no. 3, pp. 211–  
483 222, Mar. 2006.
- 484 7. P. Carninci, T. Kasukawa, S. Katayama, J. Gough, M. C. Frith, N. Maeda,  
485 R. Oyama, T. Ravasi, B. Lenhard, C. Wells, R. Kodzius, K. Shimokawa, V. B.  
486 Bajic, S. E. Brenner, S. Batalov, A. R. R. Forrest, M. Zavolan, M. J. Davis, L. G.  
487 Wilming, V. Aidinis, J. E. Allen, A. Ambesi-Impimbato, R. Apweiler, R. N. Atu-  
488 raliya, T. L. Bailey, M. Bansal, L. Baxter, K. W. Beisel, T. Bersano, H. Bono, A. M.  
489 Chalk, K. P. Chiu, V. Choudhary, A. Christoffels, D. R. Clutterbuck, M. L. Crowe,  
490 E. Dalla, B. P. Dalrymple, B. de Bono, G. Della Gatta, D. di Bernardo, T. Down,  
491 P. Engstrom, M. Fagiolini, G. Faulkner, C. F. Fletcher, T. Fukushima, M. Furuno,  
492 S. Futaki, M. Gariboldi, P. Georgii-Hemming, T. R. Gingeras, T. Gojobori, R. E.  
493 Green, S. Gustincich, M. Harbers, Y. Hayashi, T. K. Hensch, N. Hirokawa, D. Hill,  
494 L. Huminiecki, M. Iacono, K. Ikeo, A. Iwama, T. Ishikawa, M. Jakt, A. Kanapin,  
495 M. Katoh, Y. Kawasawa, J. Kelso, H. Kitamura, H. Kitano, G. Kollias, S. P. T. Kr-  
496 ishnan, A. Kruger, S. K. Kummerfeld, I. V. Kurochkin, L. F. Lareau, D. Lazarevic,  
497 L. Lipovich, J. Liu, S. Liuni, S. McWilliam, M. Madan Babu, M. Madera, L. Mar-  
498 chionni, H. Matsuda, S. Matsuzawa, H. Miki, F. Mignone, S. Miyake, K. Mor-  
499 ris, S. Mottagui-Tabar, N. Mulder, N. Nakano, H. Nakauchi, P. Ng, R. Nilsson,  
500 S. Nishiguchi, S. Nishikawa, F. Nori, O. Ohara, Y. Okazaki, V. Orlando, K. C.  
501 Pang, W. J. Pavan, G. Pavesi, G. Pesole, N. Petrovsky, S. Piazza, J. Reed, J. F.  
502 Reid, B. Z. Ring, M. Ringwald, B. Rost, Y. Ruan, S. L. Salzberg, A. Sandelin,  
503 C. Schneider, C. Schönbach, K. Sekiguchi, C. A. M. Semple, S. Seno, L. Sessa,  
504 Y. Sheng, Y. Shibata, H. Shimada, K. Shimada, D. Silva, B. Sinclair, S. Sperling,  
505 E. Stupka, K. Sugiura, R. Sultana, Y. Takenaka, K. Taki, K. Tammoja, S. L. Tan,  
506 S. Tang, M. S. Taylor, J. Tegner, S. A. Teichmann, H. R. Ueda, E. van Nimwegen,  
507 R. Verardo, C. L. Wei, K. Yagi, H. Yamanishi, E. Zabarovsky, S. Zhu, A. Zim-  
508 mer, W. Hide, C. Bult, S. M. Grimmond, R. D. Teasdale, E. T. Liu, V. Brusic,  
509 J. Quackenbush, C. Wahlestedt, J. S. Mattick, D. A. Hume, C. Kai, D. Sasaki,  
510 Y. Tomaru, S. Fukuda, M. Kanamori-Katayama, M. Suzuki, J. Aoki, T. Arakawa,  
511 J. Iida, K. Imamura, M. Itoh, T. Kato, H. Kawaji, N. Kawagashira, T. Kawashima,  
512 M. Kojima, S. Kondo, H. Konno, K. Nakano, N. Ninomiya, T. Nishio, M. Okada,

- 513 C. Plessy, K. Shibata, T. Shiraki, S. Suzuki, M. Tagami, K. Waki, A. Watahiki,  
514 Y. Okamura-Oho, H. Suzuki, J. Kawai, Y. Hayashizaki, F. Consortium, R. G. E. R.  
515 Group, and G. S. G. G. N. P. C. Group, "The transcriptional landscape of the mam-  
516 malian genome," *Science (New York, NY)*, vol. 309, no. 5740, pp. 1559–1563, Sep.  
517 2005.
- 518 8. E. A. Rach, H.-Y. Yuan, W. H. Majoros, P. Tomancak, and U. Ohler, "Motif  
519 composition, conservation and condition-specificity of single and alternative tran-  
520 scription start sites in the *Drosophila* genome." *Genome Biology*, vol. 10, no. 7, p.  
521 R73, 2009.
- 522 9. B. Lenhard, A. Sandelin, and P. Carninci, "Metazoan promoters: emerging char-  
523 acteristics and insights into transcriptional regulation." *Nature Reviews Genetics*,  
524 vol. 13, no. 4, pp. 233–245, Apr. 2012.
- 525 10. T. Ni, D. L. Corcoran, E. A. Rach, S. Song, E. P. Spana, Y. Gao, U. Ohler,  
526 and J. Zhu, "A paired-end sequencing strategy to map the complex landscape of  
527 transcription initiation." *Nature Methods*, vol. 7, no. 7, pp. 521–527, Jul. 2010.
- 528 11. U. Ohler, G.-c. Liao, H. Niemann, and G. M. Rubin, "Computational analysis of  
529 core promoters in the *Drosophila* genome." *Genome Biology*, vol. 3, no. 12, pp.  
530 research0087.1–0087.12, 2002.
- 531 12. R. T. Raborn, K. Spitze, V. P. Brendel, and M. Lynch, "Promoter Architecture  
532 and Sex-Specific Gene Expression in *Daphnia pulex*." *Genetics*, vol. 204, no. 2, pp.  
533 593–612, Aug. 2016.
- 534 13. C. Nepal, Y. Hadzhiev, C. Previti, V. Haberle, N. Li, H. Takahashi, A. M. M.  
535 Suzuki, Y. Sheng, R. F. Abdelhamid, S. Anand, J. Gehrig, A. Akalin, C. E. M.  
536 Kockx, A. A. J. van der Sloot, W. F. J. van IJcken, O. Armant, S. Rastegar,  
537 C. Watson, U. Strahle, E. Stupka, P. Carninci, B. Lenhard, and F. Muller, "Dy-  
538 namic regulation of the transcription initiation landscape at single nucleotide res-  
539 olution during vertebrate embryogenesis," *Genome Research*, vol. 23, no. 11, pp.  
540 1938–1950, Nov. 2013.
- 541 14. P. Carninci, A. Sandelin, B. Lenhard, S. Katayama, K. Shimokawa, J. Ponjavic,  
542 C. A. M. Semple, M. S. Taylor, P. G. Engström, M. C. Frith, A. R. R. For-  
543 rest, W. B. Alkema, S. L. Tan, C. Plessy, R. Kodzius, T. Ravasi, T. Kasukawa,  
544 S. Fukuda, M. Kanamori-Katayama, Y. Kitazume, H. Kawaji, C. Kai, M. Naka-  
545 mura, H. Konno, K. Nakano, S. Mottagui-Tabar, P. Arner, A. Chesi, S. Gustincich,  
546 F. Persichetti, H. Suzuki, S. M. Grimmond, C. A. Wells, V. Orlando, C. Wahlest-  
547 edt, E. T. Liu, M. Harbers, J. Kawai, V. B. Bajic, D. A. Hume, and Y. Hayashizaki,  
548 "Genome-wide analysis of mammalian promoter architecture and evolution," *Nat-  
549 ure Genetics*, vol. 38, no. 6, pp. 626–635, Apr. 2006.
- 550 15. S. Mwangi, G. Attardo, Y. Suzuki, S. Aksoy, and A. Christoffels, "TSS seq based  
551 core promoter architecture in blood feeding Tsetse fly (*Glossina morsitans mor-  
552 sitans*) vector of Trypanosomiasis," *BMC Genomics*, vol. 16, no. 1, p. 722, Sep.  
553 2015.
- 554 16. K. Tsuchihara, Y. Suzuki, H. Wakaguri, T. Irie, K. Tanimoto, S.-i. Hashimoto,  
555 K. Matsushima, J. Mizushima-Sugano, R. Yamashita, K. Nakai, D. Bentley, H. Es-  
556 umi, and S. Sugano, "Massive transcriptional start site analysis of human genes in  
557 hypoxia cells," *Nucleic Acids Research*, vol. 37, no. 7, pp. 2249–2263, Apr. 2009.
- 558 17. N. Cvetesic and B. Lenhard, "Core promoters across the genome," *Nature Biotech-  
559 nology*, vol. 35, no. 2, pp. 123–124, Feb. 2017.
- 560 18. P. J. Batut and T. R. Gingeras, "RAMPAGE: Promoter Activity Profiling by  
561 Paired-End Sequencing of 5'-Complete cDNAs." in *Current Protocols in Molecular  
562 Biology*. Current protocols in molecular biology / edited by Frederick M Ausubel  
563 [et al], 2013, pp. 25B.11.1–25B.11.16.



- 564 19. N. Merchant, E. Lyons, S. Goff, M. Vaughn, D. Ware, D. Micklos, and P. Antin,  
565 “The iPlant Collaborative: Cyberinfrastructure for Enabling Data to Discovery for  
566 the Life Sciences.” *PLoS Biology*, vol. 14, no. 1, p. e1002342, Jan. 2016.
- 567 20. R. Leinonen, H. Sugawara, M. Shumway, and International Nucleotide Sequence  
568 Database Collaboration, “The sequence read archive.” *Nucleic Acids Research*,  
569 vol. 39, no. Database issue, pp. D19–21, Jan. 2011.
- 570 21. E. Aronesty, “Comparison of Sequencing Utility Programs,” *The Open Bioinform-*  
571 *atics Journal*, vol. 7, no. 1, pp. 1–8, Jan. 2013.
- 572 22. H. Lab, “FASTX Toolkit.” [Online]. Available:  
573 [http://hannonlab.cshl.edu/fastx\\_toolkit/](http://hannonlab.cshl.edu/fastx_toolkit/)
- 574 23. T. Lassmann, “TagDust2: a generic method to extract reads from sequencing data,”  
575 *BMC Bioinformatics*, vol. 16, no. 1, p. 1, Jan. 2015.
- 576 24. H. Li, B. Handsaker, A. Wysoker, T. Fennell, J. Ruan, N. Homer, G. Marth, G. R.  
577 Abecasis, R. Durbin, and 1000 Genome Project Data Processing Subgroup, “The  
578 Sequence Alignment/Map format and SAMtools,” *Bioinformatics (Oxford, Eng-*  
579 *land)*, vol. 25, no. 16, pp. 2078–2079, Aug. 2009.
- 580 25. A. Dobin and T. R. Gingeras, “Optimizing RNA-Seq Mapping with STAR,” in  
581 *Transcription Factor Regulatory Networks*. New York, NY: Springer New York,  
582 Apr. 2016, pp. 245–262.
- 583 26. R Core Team, *R: A Language and Environment for Statistical Computing*, R  
584 Foundation for Statistical Computing, Vienna, Austria, 2017. [Online]. Available:  
585 <https://www.R-project.org>
- 586 27. M. Lawrence and M. Morgan, “Scalable Genomics with R and Bioconductor,”  
587 *Statistical Science*, vol. 29, no. 2, pp. 214–226, May 2014.
- 588 28. V. Hablerle, A. R. R. Forrest, Y. Hayashizaki, P. Carninci, and B. Lenhard,  
589 “CAGER: precise TSS data retrieval and high-resolution promoterome mining for  
590 integrative analyses.” *Nucleic Acids Research*, vol. 43, no. 8, pp. gkv054–e51, Feb.  
591 2015.
- 592 29. O. Tange, “Gnu parallel - the command-line power tool,” *login: The*  
593 *USENIX Magazine*, vol. 36, no. 1, pp. 42–47, Feb 2011. [Online]. Available:  
594 <http://www.gnu.org/s/parallel>

## 595 6 Checklist of Items to be Sent to Volume Editors

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

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