Seqpac: A New Framework for small RNA analysis in R using Sequence-Based Counts

Signe Skog^{1#}, Lovisa Örkenby^{1#}, Unn Kugelberg¹, Kanwal Tariq², Ann-Kristin Östlund Farrants², Anita Öst^{1,*} and Daniel Nätt^{1,*}

¹Department of Biomedical and Clinical Sciences, Linköping University, Cell biology building, Floor 12, SE-58185 Linköping, Sweden

²Department of Molecular Biosciences, The Wenner-Gren Institut, The Arrhenius Lab F4, Stockholm University, SE-106 91 Stockholm, Sweden

Equal contributions

* Equal contributions

Correspondence to: daniel.natt@liu.se

ORCID iD:

Daniel Nätt <https://orcid.org/0000-0001-9182-9401>

Anita Öst <https://orcid.org/0000-0003-0547-1904>

Ann-Kristin Östlund Farrants <https://orcid.org/0000-0001-9225-3264>

22

23 **ABSTRACT**

24 Small RNA sequencing (sRNA-seq) has become important for studying regulatory
 25 mechanisms in many cellular processes. Data analysis remains challenging, mainly
 26 because each class of sRNA—such as miRNA, piRNA, tRNA- and rRNA- derived
 27 fragments (tRFs/rRFs)—needs special considerations. Analysis therefore involves
 28 complex workflows across multiple programming languages, which can produce
 29 research bottlenecks and transparency issues. To make analysis of sRNA more
 30 accessible and transparent we present seqpac: a tool for advanced group-based
 31 analysis of sRNA completely integrated in R. This opens advanced sRNA analysis for
 32 Windows users—from adaptor trimming to visualization. Seqpac provides a
 33 framework of functions for analyzing a PAC object, which contains 3 standardized
 34 tables: sample phenotypic information (P), sequence annotations (A), and a counts
 35 table with unique sequences across the experiment (C). By applying a sequence-
 36 based counting strategy that maintains the integrity of the fastq sequence, seqpac
 37 increases flexibility and transparency compared to other workflows. It also contains
 38 an innovative targeting system allowing sequence counts to be summarized and
 39 visualized across sample groups and sequence classifications. Reanalyzing
 40 published data, we show that seqpac's fastq trimming performs equal to standard
 41 software outside R and demonstrate how sequence-based counting detects
 42 previously unreported bias. Applying seqpac to new experimental data, we
 43 discovered a novel rRF that was down-regulated by RNA pol I inhibition (anticancer
 44 treatment), and up-regulated in previously published data from tumor positive
 45 patients. Seqpac is available on github (<https://github.com/Danis102/seqpac>), runs
 46 on multiple platforms (Windows/Linux/Mac), and is provided with a step-by-step
 47 vignette on how to analyze sRNA-seq data.

48

49

50

51 **BACKGROUND**

52 The past decades have uncovered a diversity of small RNA (sRNA), which differs
 53 greatly in their biogenesis and biological roles. This involves miRNA that is generated
 54 from transcribed precursors and recruited by Argonaute proteins for post- and pre-
 55 transcriptional gene silencing (1-5). Having a similar mechanism, piRNA primarily
 56 silence repetitive transposable elements in the germline, and can be amplified by
 57 means of the so-called ping-pong cycle (6). Other classes involves rRNA and tRNA
 58 derived fragments (rRF/tRFs) that may interact with Argonaute proteins in a
 59 piRNA/miRNA-like fashion, but may also directly interfere with translational
 60 processes in the ribosome (7-10). Some tRFs may not even align to their genome of
 61 origin, since their parental tRNA matures post-transcriptionally by receiving additional
 62 nucleotides (11). While many sRNA classes exerts their function in the cytoplasm,
 63 some intermediately sized none-coding RNA—like the snoRNA, scaRNA and
 64 snRNA—are associated with specific organelles inside the nucleus where they play
 65 important roles in the post-transcriptional shaping (splice, fold, and modify) of other
 66 RNA molecules (12,13).

67

68 This complexity, where some sRNA may target single gene products while others
 69 target highly repetitive regions, where some are biologically active after transcription
 70 while others are post-transcriptionally modified prior to activation, where some align
 71 to the genome that they originated from while others do not, makes the analysis
 72 sRNA challenging. Today, it is also becoming increasingly popular to apply high-
 73 throughput sequencing in sRNA experiments, which makes the analysis even more
 74 complicated. Combining massively parallel sequencing with specialized library
 75 preparation protocols that select for short RNA species generate data often
 76 containing millions of unique short RNA sequences across tens-to-hundreds of
 77 samples.

78

79 Several tools and pipelines, such as Sports (14), MintMap (11), sRNAtoolbox (15),
 80 sRNAAnalyzer(16), COMPSRA (17), and iSmaRT (18) have been developed to
 81 overcome some of the analytical thresholds in sRNA analysis. As a rule, these tools
 82 wrap around multiple programs written in multiple programming languages, such as

83 cutadapt (19) for adapter trimming, bowtie (20) for genome mapping, and
 84 subread:featureCounts (21) for counting sequences across sRNA subspecies. Thus,
 85 in labs that lack strong programming skills and previous experience of sRNA
 86 analysis, troubleshooting and advanced analysis often become bottlenecks. This may
 87 result in the exclusion of ‘difficult-to-analyze’ sRNA in favor of more straight-forward
 88 sub-species, such as miRNA. Unless better, more coherent, and user-friendly tools
 89 are developed, such discrimination will result in severe literature biases.

90

91 Workflows for sRNA-seq analysis regularly build on methods from gene-centric
 92 DNA/RNA-seq approaches, such as regular mRNA-seq. This usually involves
 93 mapping individual samples against a reference genome followed by counting
 94 overlaps of genomic coordinates between sample reads and known genomic
 95 features, such as gene exons or miRNAs. Such feature-based counting (Figure 1A) is
 96 often done one read and one sample at the time. Most sRNA experiments, however,
 97 do not contain a single sample. Instead, they contain multi-sample groups. Therefore,
 98 as an alternative, read sequences across the whole experiment can be counted prior
 99 to aligning the read to a reference genome. Such sequence-based counting (Figure
 100 1A) would prevent annotating the same sequence multiple times both within and
 101 across samples. More importantly, this strategy would maintain sequence integrity.
 102 Thus, further annotation of the counted sequences would be possible at any time
 103 during the analysis. In addition, with sequence-based counting users may choose to
 104 remove sequences with low evidence, which fails to replicate across their
 105 experiment. Hypothetically, these advantages with sequence-based counting may not
 106 only have dramatic effects on computational performance. It may also increase the
 107 transparency and flexibility of the whole analysis.

108

109 Here, we present—seqpac—a novel framework for sequence-based multi-sample
 110 sRNA analysis. From adapter trimming to the visualization of group-differences,
 111 seqpac is completely integrated as an open-source package in R. This makes it
 112 accessible from multiple platforms, including Windows, Mac and Linux. Using both
 113 published and novel data, we show that sequence-based counting combined with a
 114 multi-sample approach, not only positively affects computational performance,
 115 making sRNA-seq analysis accessible on a standard computer. It also increases the
 116 flexibility and transparency throughout the analysis. We illustrate this by detecting

severe contamination in published data that was previously analyzed using a feature-based counting strategy. Finally, we use the strengths of seqpac to discover and confirm a novel rRF implicated as a diagnostic/prognostic marker in cancer.

MATERIALS AND METHODS

1. Package development

Seqpac is available for download at github (<https://github.com/Danis102/seqpac>). Procedures on how to install seqpac are explained in the vignette (<https://github.com/Danis102/seqpac/tree/master/vignettes>). Dependencies for the main seqpac functions are listed in Table 1. Seqpac was developed and tested on a Linux Mint v.19.1 computer using R 3.4.4 in RStudio 1.2.1335 and devtools 2.3.2. The computer had an Intel Core i7-9800X CPU at 3.8 GHz (8 cores with in total 16 threads) and contained 94 Gb of ram memory. All R internal functions (e.g. `make_cutadapt` excluded) were subsequently tested on multiple Windows 10 computers using R 3.6.3 and 4.0.1.

2. Testing seqpac using published datasets

Fastq files for 4 datasets were accessed through Sequence Reads Archive (SRA) and European Reads Archive (ENA). We prefer downloading these files and their metadata through ENA (<https://www.ebi.ac.uk/ena>). All code for processing and generating the results presented in Figure 2, 4, 7 and 8 are available in Supplementary text S1. A brief explanation is provided below.

2.1 Kang *et al.* 2018 – Benchmarking and reannotation using human and fruit fly multi-genome samples (Figure 2, 4)

Kang *et al.* 2018 (22) (SRA accession: PRJNA485638; ENA download: <https://www.ebi.ac.uk/ena/browser/view/PRJNA485638>) were used for benchmarking seqpac's `make_trim` function against two similar workflows. In both alternative workflows, system calls to `cutadapt` (19) and `fastq_quality_filter` (in FASTX-Toolkit; http://hannonlab.cshl.edu/fastx_toolkit/) were made from within R. The first used the `make_cutadapt` function to replicate the parallelization for `make_trim` using the `foreach` package (23), while the second used the internal parallelization option in

149 *cutadapt*. System time was monitored over 10 iterations replicated 6 times using the
150 *rbenchmark* package (24).

151

152 PAC objects with counts from trimming/filtering using the *make_trim* function and
153 *cutadapt/fasq_quality_filter* alternative, were generated using *make_counts* function
154 either with *trimming="seqpac"* or *trimming="cutadapt"*. Bar graphs from the low-level
155 evidence filtering were saved. To assure that only sRNA were include, since this
156 dataset was generated from a 75 cycle flow-cell, we removed reads that failed to
157 contain adaptor sequence and only kept reads <=45 nt. The counts lists with
158 progress reports were then applied to the standard PAC generation workflow
159 (*make_counts > make_anno > make_pheno > make_PAC*). As phenotypic input file
160 for *make_pheno* function we used metadata downloaded from SRA/ENA.

161

162 After benchmarking, only the internal (*make_trim*) PAC object was applied to the
163 reannotation workflow. Reannotation against either the human and fly reference
164 genomes or sRNA class references were applied, using either the *map_reanno*
165 *import="genome"* or *import="biotype"* options, respectively. For genome alignments
166 we downloaded Homo sapiens GRCh38.101 (hg38) and Drosophila melanogaster
167 BDGP6.28 (dm6) in fasta references at Ensembl ftp
168 (<http://www.ensembl.org/info/data/ftp/>). For the sRNA class alignment we downloaded
169 fasta references for miRNA (mirBase v.21), ncRNA (Ensembl.ncrna), tRNA
170 (GtRNadb) and piRNA (pirBase) for the human and fruit fly genomes, respectively.
171 After generating reanno objects in R using the *make_reanno* function, we added and
172 simplified the annotations using the *add_reanno* and *simplify_reanno* functions. The
173 sRNA class hierarchy in *simplify_reanno* was set to rRNA > tRNA > miRNA >
174 snoRNA > snRNA > lnc/lincRNA > piRNA. Plots in Figure 4 were generated using the
175 *PAC_pie*, *PAC_sizedist* and *PAC_nbias* functions.

176

177 **2.2 Tong et al. 2020 – Detecting contamination in cancer cell lines (Figure 7, 8)**

178 Tong et al. from 2020 (25) (SRA accession: PRJNA666144; ENA download:
179 <https://www.ebi.ac.uk/ena/browser/view/PRJNA666144>) were used for exemplifying
180 the strengths of sequence-based counting in detecting severe bias in cancer cell line
181 experiments. PAC generation and reannotation was performed similarly to the Kang
182 et al. dataset (MATERIALS AND METHODS 2.1) with a few exceptions. Since the

Tong *et al.* was generated from a 50 cycle flow-cell we did not remove reads that failed to contain adaptor sequence and did not filter by max read length. Analysis and graphs were generated using the *PAC_pca*, *PAC_sizedist* and *PAC_stackbar* functions. We also verified the 5' ETS rRF of the 45S pre-rRNA (NR_146144.1) with the *PAC_mapper* and *PAC_covplot* functions using the same fasta reference as described in Methods 2.3. When reannotating the PAC object after the initial analysis, we used the *Mycoplasma hyorhina* ATCC (ASM38351v1) genome in parallel with *Homo sapiens* (hg38).

2.3 Skog *et al.* 2021 – HeLa anti-cancer treatment dataset (current study; Figure 8)

The anti-cancer treatment dataset was generated in the current study (see Methods 3) and is available at SRA (accession: PRJNA708219). Since this dataset was generated from a 75 cycle flow-cell, an annotated PAC object was created as for the Kang *et al.* dataset (see Methods 2.1) removing reads that failed to contain adaptor sequence. To better compare with the Tong *et al.* dataset we set a max read length of 65 nt. The *PAC_deseq* function was used to initially identify BMH21 sensitive fragments comparing cells exposed to BMH21 for 60 min to those exposed to DMSO for 60 min (control). Mapping against pre-rRNA was done using the *PAC_mapper* function with a custom fasta reference (Supplementary file S2). This reference first contained the GenBank sequence NR_146144.1. After identifying 4 peaks using the *PAC_covplot* function, we added the zoomed in regions of chr 21 aligning with NR_146144.1 and containing each of the four rRF peaks (Peak 1 = chr21:8206319-8206669, Peak 2 = chr21:8212475-8212825, Peak 3 = chr21:8213765-8214115, Peak 4 = chr21:8218787-8219137). These regions were downloaded from the UCSC genome browser. Finally, we added the 47S GenBank entry U13369.1 to the fasta.

2.4 Xu *et al.* 2020 – Validation in cervical cancer patients (Figure 8)

The Xu *et al.* 2020 (26,27) (SRA accession PRJNA607023; ENA download: <https://www.ebi.ac.uk/ena/browser/view/PRJNA607023>) dataset, used for validating the 5' ETS rRF of the 45S pre-rRNA (NR_146144.1) in clinical samples. We only used the 8 fastq files obtained by sRNA size-fractions. Files were generated using a paired-end 2x150 cycle flow cell kit. Thus, we discarded the paired—second—read

216 and only kept the trimmed sequences of the first read where an adaptor was present
217 (as in see Methods 2.1 and 2.3).

218

219 **3. Generating the HeLa anti-cancer treatment dataset**

220 Adherent HeLa CLL-2 cells were obtained from ATCC and were maintained at 37°C
221 and 5% CO₂ in high glucose Dulbecco's modified Eagle's medium (DMEM),
222 supplemented with 10% fetal bovine serum and 1% Penicillin/Streptomycin cocktail.
223 Cells were treated with 1uM BMH-21 in antibiotic free media for 60min and 12h. Cells
224 treated with DMSO for 60 min were used as control. The media was removed, cells
225 were washed with PBS, collected with trypsinization, and stored at -70°C until further
226 processing.

227

228 Frozen cells were homogenized in prechilled Qiazol (Qiagen, Hilden, Germany) using
229 a Tissue Lyser LT (Qiagen) set to 2 min at 30 oscillations/second with 5 mm Stainless
230 Steel Beads (Qiagen). RNA was then extracted using miRNeasy Micro kit (Qiagen),
231 and the integrity of purified RNA was confirmed on a Bioanalyzer (Agilent
232 Technologies, Santa Clara, USA), where sample RIN values ranged between 9.3-10.
233 Library preparation was done with NEBNext Small RNA Library Prep Set for Illumina
234 (New England Biolabs, Ipswich, USA) with 100 ng of input total RNA according to
235 manufacturer instructions, except for the following minor customizations: reactions
236 were scaled-down to half the volume, adapters were diluted 1:2, amplification was
237 done for 12 cycles, and libraires were size-selected for 130 to 190 nt fragments on a
238 pre-casted 6% polyacrylamide Novex TBE gel (Invitrogen, Waltham, USA). Gel
239 extraction was done using Gel breaker tubes (IST Engineering, Milpitas, USA) in the
240 buffer provided in the NEBNext kit. After precipitation, the library concentrations were
241 estimated using QuantiFluor ONE ds DNAsystem on a Quantus fluorometer
242 (Promega, Madison, USA). Pooled libraries were sequenced on NextSeq 500 with
243 NextSeq 500/550 High Output Kit version 2.5, 75 cycles (Illumina, San Diego, USA).
244 All pooled libraries passed Illumina's default quality control.

245

246

247 **RESULTS AND DESCRIPTION**

248 **1.1 The seqpac workflow**

Seqpac comes with a vignette that contains a step-by-step in-depth guide on how to analyze sRNA data from high-throughput sequencing. All functions can be tested using a set of down-sampled fastq files, with sRNA data originating from single fruit fly embryos. A quick reference to the main functions in seqpac is available in Table 1. Scripts for generating many of the analysis presented in the figures are available in Supplementary file S1.

255

The general seqpac workflow involves three separate steps: constructing, annotating and, analyzing a PAC object (Figure 1B). A PAC object is in its simplest form an R list object, listing a phenotype (Pheno) table with sample information, an annotation (Anno) table with information about unique sequences, and a counts (Counts) table with the counts of sequences across samples (Figure 1C). While this setup reminds of many S4 class objects in packages such as limma (28), DESeq2 (29) and minfi (30) etc., we have deliberately made the PAC list a regular S3 object, holding two classifications 'PAC' and 'list'. One reason is that S4 objects are often a source of confusion for beginners in R. Another is that all basic functions for handling lists are directly applicable on the PAC object, making it easy for more advanced users to customize their workflows.

267

2.1 Constructing the PAC object

Building the PAC object starts by generating a counts table. This is primarily done by the *make_counts* function. It uses fastq formatted sequence files to generate a standardized data frame, where each row represents unique sequences in the experiment, while columns represent samples (Figure 1C). This table maintains the framework for all subsequent analysis. The phenotype and annotation tables contain further information about samples (columns in the counts table) and sequences (rows in the counts table). These tables are produced by the *make_pheno* and *make_anno* functions. The phenotype table is provided by the user and can optionally be merged with a progress report from the adaptor trimming and low-level filtering (see Results 2.2). The *make_anno* function prepares a very primitive annotation table that will expand in the reannotation workflow (see Results 3.1-3.4). Finally, *make_PAC* checks the different components and builds the PAC object.

281

2.2 Trimming fastq of adaptor sequence

283 The *make_counts* function reads raw sequence files in fastq format using the
 284 *ShortRead* package (31), trims the reads of adaptor sequence and filters low-quality
 285 and non-replicable reads, prior to counting each unique read sequence across all
 286 samples. For adaptor trimming seqpac has an internal and external alternative.
 287 Internally, *make_counts* calls the stand alone *make_trim* function that primarily uses
 288 the *Biostrings* package (32) to efficiently search and remove any adaptor sequence.
 289 In addition, sequences with low quality base scores can be filtered. For the external
 290 option, *make_counts* is dependent on system calls to externally installed *cutadapt*
 291 (19) and *fastq_quality_filter* (available in FASTX-Toolkit) (33) software.

292
 293 To test the performance of seqpac's *make_trim* function, we downloaded fastq files
 294 from the Kang *et al.* study from 2018 (22) (SRA project: SRP157338). This dataset
 295 contains 7 fastq ranging between 52.7-492.8 Mb in compressed size (mean=310.1
 296 Mb) and were generated from either human or fruit fly RNA, where some samples
 297 were generated by mixing RNA from these species in different ratios. Using the
 298 *rbenchmark* package (24) we trimmed/filtered these files over 10 iterations replicated
 299 6 times for the *make_trim* and *make_cutadapt* functions, as well as stand-alone
 300 *cutadapt/fastq_quality_filter* using near-to-identical settings. Each function was given
 301 7 parallel jobs on a Linux desktop computer (for hardware specifications, see
 302 Methods). While *make_trim* and *make_cutadapt* uses the *foreach* package (23) to
 303 parallelize jobs across processor cores/threads, the stand-alone
 304 *cutadapt/fastq_quality_filter* workflow used *cutadapt*'s internal parallelization option (-
 305 p 7). The *make_trim* function was on average 1.2 times faster than *make_cutadapt*,
 306 and 2.4 times faster than *cutadapt/fastq_quality_filter* (Figure 2A). On average,
 307 *make_trim* finished trimming/filtering all 7 fastq in 4.8 min, *make_cutadapt* in 5.8 min
 308 and the stand-alone alternative in 11.4 min. The slow performance of the stand-alone
 309 alternative was primarily due to *fastq_quality_filter* lacking the ability to run jobs in
 310 parallel.

311
 312 Seqpac's *make_trim* function generated very similar sequence counts compared to
 313 the *cutadapt/fastq_quality_filter* alternative (Figure 2B-C). We noticed, however, that
 314 *make_trim* generated slightly higher counts for some sequences (arrows in Figure
 315 2B). Manually searching for these sequences across the original and trimmed fastq
 316 files showed that one explanation was that *cutadapt* failed to identify concatemer

adaptor sequences. Concatemer (chimeric) adaptors are found in small quantity in most experiments, and are technical constructs where an incomplete adaptor associates with a complete adaptor during synthesis (34).

2.3 The low-level evidence filter

The *make_counts* function contains a low-level filtering module, here called an evidence filter. In default settings, it simply filters sequences that fails to replicate across two independent samples. Even in small experiments, such as the Kang *et al.* dataset, such filtering dramatically increases performance by reducing noise from extremely rare transcripts/degradation products (Figure 2C). Our experience is that such evidence filter often results in less than half the sequence diversity (number of unique read sequences; lower bars Figure 2C), while maintaining most of the sequencing depth (total number of reads; upper bars Figure 2C).

To illustrate this further, true sequence diversity—that can be replicated and is not due to technical bias—should expect to rise when sRNA from two species is mixed, which is also the case in the Kang *et al.* dataset (percentages in Figure 2C). Nonetheless, the evidence filter in *make_counts* can both be disabled (e.g. in single sample/replicate experiments) or intensified (e.g. to increase performance in very large datasets). In addition, confirming our initial observation that seqpac's *make_trim* function was better in identifying adaptor artifacts, such as concatemer adaptors, *make_trim* identified more replicable unique sequences passing the evidence filter than the popular *cutadapt/fastq_quality_filter* workflow (Figure 2D).

3.1 Annotating sequence with seqpac

Seqpac provides two ways to annotate a sequence in a PAC object. Firstly, the reannotation workflow (Figure 3: step 1-3) aligns the trimmed read sequences in the PAC against reference sequences, for example a reference genome, sRNA database, or sequences from another experiment, such as the results from a piwi pull-down. This is done using the reannotation family of functions: *map_reanno*, *import_reanno*, *add_reanno* and *simplify_reanno*. Seqpac also provides a 'backdoor function', *PAC_mapper*, that quickly calls the reannotation workflow for mapping the sequences in the PAC object (see Results 6.1, 7.2).

351 Secondly, after aligning a PAC object to a reference genome, the genomic
352 coordinates of PAC sequences can be overlapped with coordinates of already
353 annotated genomic features (Figure 3: step 4). This is done by the *PAC_gtf* function
354 (Figure 3; step 4). Thus, by annotating using *PAC_gtf*, users can mimic a feature-
355 based counting strategy, while saving the sequence integrity of the trimmed fastq-file
356 in the PAC object.

357

358 **3.2 Bowtie mapping using the *map_reanno* function**

359 The reannotation workflow (Figure 3: step 1-3) depends on Bowtie (20) for sequence
360 alignment, and therefore needs Bowtie indexes for the input fasta references. Similar
361 to the adaptor trimming, *map_reanno* calls Bowtie either internally or externally,
362 through the *Rbowtie* package (35) or a system call, respectively. The function can
363 parse either seqpac standard or user provided options to Bowtie. It also calls a
364 secondary function, *import_reanno*, which controls the import options from the Bowtie
365 output files. Options involve for example whether coordinates and fasta sequence
366 names should be reported, or only hit-or-no-hit. This is convenient for large repetitive
367 sRNA references that may generate massive files if everything is reported (e.g.
368 pirBase for humans and flies).

369

370 The *map_reanno* function runs multiple align/import cycles (Figure 3: step 1). After
371 each cycle, imported data are saved as Rdata files, and only sequences without an
372 alignment to any of the references will proceed to the next cycle. Each proceeding
373 cycle allows for one additional mismatch until the user-defined max mismatches (or
374 the Bowtie limit of 3 mismatches) has been reached. Reannotating only no-hit
375 sequences in proceeding cycles not only guarantees that only the best hits are
376 reported. Since system demands per sequence increases with each added
377 mismatch, it also significantly increases performance as only the minimum number of
378 sequences are aligned in each mismatch cycle. Importantly, if a sequence aligns to
379 two references, both references will be reported for that cycle. Thus, unlike feature-
380 based counting where such multimapping issues must be resolved already when
381 reads are counted, users of seqpac can decide to discriminate between annotations
382 at any stage in the analysis.

383

384 **3.3 Annotating a PAC object using the *add_reanno* and *simplify_reanno***

385 Next, using the *add_reanno* function the Rdata files from each mismatch cycle is
 386 read into R and organized into a reanno object (Figure 3: step 2). For efficient
 387 access, this object is generated as a series of tibbles available in the *tibble;tidyverse*
 388 (36) package. Using a list of search terms, *add_reanno* consolidates the fasta
 389 sequence names into short character strings, which can be used as factors in
 390 downstream analysis. Search terms are constructed using regular expressions. A
 391 match will be reported as the reference name together with the search term. For
 392 example, if two references named *mirbase* and *ensembl_ncrna* were used as input
 393 for *map_reanno*, a search term list constructed as, *list(mirbase='mir',*
 394 *ensembl=c('snoRNA', 'tRNA'))*, will result in matches being returned as '*mirbase:mir*',
 395 '*ensembl:snoRNA*' and '*ensembl:tRNA*'. The user may choose if search terms must
 396 catch all reference hits, or if failure to match a search term should be returned as
 397 'other' (e.g. '*ensembl:other*').

398

399 Neither *map_reanno* nor *add_reanno* discriminates between references. Thus, if PAC
 400 sequences align to multiple references, all alignments and search matches will be
 401 reported (e.g. '*mirbase:mir|ensembl:other*'), but only if they align in the same
 402 mismatch cycle. For better transparency and reproducibility of sRNA experiments, we
 403 recommend that analysis is performed on a class-by-class basis as far as possible.
 404 Nonetheless, hierarchical discrimination is often the only option to resolve some
 405 issues with pseudoreplication when multiple classes of sRNA are simultaneously
 406 analyzed. This is because the same sequence sometimes appears in multiple
 407 reference databases, and therefore obtains multiple classifications, such as both
 408 piRNA and miRNA. The purpose of the *simplify_reanno* function is therefore to
 409 hierarchically discriminate between search matches generated by the *add_reanno*
 410 function (Figure 3: step 3).

411

412 Importantly, since the seqpac workflow introduces simplified hierarchical
 413 classifications late in the annotation process, users can quickly set alternative
 414 hierarchies by just reapplying the *simplify_reanno* function. Unlike feature-based
 415 counting, the seqpac workflow therefore makes it easier to observe the effects of
 416 changes to the hierarchy. In addition, since seqpac maintains sequence integrity,
 417 users may at any time blast candidate sequences at their favorite genome browsers,

418 to verify that the correct classification was made and to get additional information
419 about the candidate.

420

421 **3.4 Annotating genomic coordinates using *PAC_gtf***

422 When the reannotation workflow runs using the *import='genome'* mode, the reference
423 coordinates for each PAC sequence will be imported into the reanno object and later
424 added to the PAC annotation table. These coordinates can be parsed to the *PAC_gtf*
425 function as an alternative way to obtain PAC sequence annotations (Figure 3: step 4).
426 This function uses gtf/gff formatted files that contains coordinates of genomic
427 features and are available at many popular databases, such as Ensembl (37).
428 *PAC_gtf* simply overlaps PAC genomic coordinates with the gtf/gff coordinates using
429 functions in the *GenomicRanges* package (38). It provides the user options on what
430 information in the gtf to consolidate. Two predefined tracks, specifically expecting
431 repeatMasker (39) and Ensembl (37) gtf files, are available besides a custom option.
432

433 **3.5 Example: Reannotation workflow using the Kang *et al.* dataset**

434 To exemplify seqpac's reannotation workflow and plotting functions we ran multi-
435 species mapping using the PAC object generated from the Kang *et al.* 2018 dataset
436 (22) (presented in Figure 2). This involved parallel mapping to both the human (hg38)
437 and fruit fly (dm6) genomes, as well as species specific versions of mirBase (miRNA)
438 (40), piRNA (41), GtRNAdb (tRNA) (42) and ensembl (many types of
439 ncRNA) databases (37). The hierarchy was set to rRNA > tRNA > miRNA >
440 snoRNA > snRNA > lncRNA > piRNA, indicating that rRNA was most prioritized and
441 piRNA was least prioritized. Mapping was carried out allowing for up to 3
442 mismatches.

443

444 As expected, the test clearly discriminated between human and fly samples in terms
445 of genome alignment, and correctly accounted for the expected genomic ratios when
446 samples from these two species had been mixed (Figure 4A). The human proportion
447 of the dataset was more affected by perfect matching, which is expected due to more
448 outbreeding in the population, but both species gain almost 100% 'mappability' when
449 mismatches were allowed. The fruit fly proportion of the dataset was strongly
450 enriched with an rRNA sized to 30 nt (Figure 4B). Blasting this sequence showed that
451 it was identical to the complete 2S rRNA subunit. This was expected since Kang *et al.*

did not to report of any method that depletes 2S rRNA prior to library construction, which is commonly done in fruit fly experiments (43,44). The human proportion of the dataset was instead enriched with miRNA with the expected size of 22 nt (Figure 4B). There was also a T bias in the expected range between 22-25 that may indicate piRNA (Figure 4C). Nonetheless, the proportion of piRNA classification was lower than the T bias (Figure 4B/C), which suggests that some piRNA may have been classified as miRNA given that miRNA was prioritized in the hierarchy.

4.1 Subsetting and grouping data using targeting objects

Seqpac applies an innovative strategy for extracting sample groups and sequence classifications for filtering, plotting and statistical purposes. This involves small targeting objects constructed as a list with two inputs (Figure 5). The first being a character string naming a target column in a specific table held by the PAC object, while the other is a character vector naming the target entries of the target column. Importantly, the name of the targeting object itself pinpoints to which PAC table that should be targeted. Thus, if a function has a '*pheno_target*=' input, a targeting object naming a column in the phenotype table can be used to subdivide the data. Similarly, if an '*anno_target*=' input option is available then columns in the annotation table can be targeted. The second entry of a targeting object is often order sensitive. Thus, if users want the sample groups to appear in a specific order in a graph, they only need to provide that order in the second entry of the *pheno_target* object (Figure 5).

As an example, when using the *PAC_pie* to generate the pie charts in Figure 4A, we used an *anno_target* for a column in the *Anno* table containing the four different genome classifications (second entry order: "No alignment", "Fly", "Human" and "Both fly and human"). Similarly, when generating the size distribution histograms in Figure 4B, we used an *anno_target* for a column in *Anno* holding the sRNA classifications generated by the *simplify_reanno* function.

In a few cases, seqpac functions use targeting objects for other seqpac objects, such as a PAC summary table (see Results 5.3). While the principle of these objects is similar to the *pheno_target* and *anno_target* objects, they may have differences that are carefully described in the manual to each function.

5.1 Overview preprocessing, summarization and statistical analysis

With or without advanced annotations, PAC objects can be filtered (*PAC_filter*, *PAC_filtsep*), normalized (*PAC_norm*), and summarized (*PAC_summarize*) using seqpac internal functions. More advanced statistical wrappers immediately compatible with PAC objects are also available (e.g. *PAC_deseq*, *PAC_pca*).

5.2 Filtering

PAC_filter and *PAC_filtersep* handles filtering and subsetting of PAC objects. With *PAC_filter*, users can subset the PAC object by targeting columns in the *Pheno* and *Anno* tables using the *pheno_target* and *anno_target* options (see Results 4.1). A filter that extracts sequences that have reached a percent coverage over a certain threshold is also available for both raw and normalized counts. This can for example be used for the popular ‘20 counts in 50% of samples’ filter. *PAC_filter* can also plot a graph that shows the impact on the data at different thresholds. Conveniently, seqpac provides a separate function *PAC_filtersep*, that extracts sequences reaching a coverage threshold within sample groups. The output can directly be used to construct Wenn-diagrams, for example visualizing the sequence overlap that reach 100 cpm within two sample groups. It can also be applied for more advanced filters, like removing read sequences that do not reach 20 counts across all samples within a group.

5.3 Normalize, summarize and statistical analysis

While the standard structure of a PAC list object contains three tables—*Pheno*, *Anno* and *Counts*—it may hold any number of objects as long as they do not have the same names as the standard objects, just like a regular list. There are, however, two more standard objects that are added to the PAC object later in the analysis: the *norm* list containing normalized counts tables, and the *summary* list that contains summarized tables (Figure 6). It is easy to visualize these objects as two separate ‘folders’ within a PAC object.

PAC_norm provides a few common normalization methods, like the simple reads/counts per million that standardize each sample against their total counts. It currently also maintains a wrapper for the *rlog* and *vst* functions of the *DESeq2* package (29), that automatically will prepare the PAC counts table for a

transformation blinded against experimental groups. Users are, however, encouraged to provide their own normalization tables. As long as the table contains the same sequence (row) and sample (column) names as the *Counts* table, and are stored in the *norm* list ('folder') of the PAC object, seqpac functions with a *norm* input option will automatically search the *norm* folder for a matching name.

PAC_summary generates simple group summaries, like means, standard deviations, standard errors, percent group differences and log2 fold changes. It can be applied to both raw counts, as well as normalized counts by naming a table in the PAC *norm* list/folder using the *norm* input option. The grouping of samples is controlled by a *pheno_target* object. *PAC_summary* does not maintain an *anno_target* option since summaries over annotations would result in loss of sequence integrity (= feature-based counts). Summarizing data across both phenotype and annotation is instead handled by individual functions, or by subdividing the whole PAC file using the *PAC_filter* function prior to running *PAC_summary*.

For more advanced statistical analysis seqpac provides a convenient function, *PAC_deseq*, that allow users to import a PAC object into *DESeq2* (29). This function automatically generates a report containing organized top tables, volcano-plots and p-value distribution histograms. Further, seqpac contains the *PAC_pca* function that performs a principle component analysis (PCA) with aid of the *FactoMineR* and *factoextra* packages (45,46). This function returns scatter plots of the main components annotated using either a *pheno_target* or *anno_target*. Lastly, *PAC_saturation* performs and plots the results of a sequence saturation analysis. This is often used for checking that satisfactory sequencing depths have been reached, where few new sequences are predicted given a hypothetical increase in the sequencing depth.

6.1 Advanced classification and visualization

In the quick reference presented in Table 1 a selection of visualization functions is briefly presented. In common for most of them are the option to use *pheno_target* and/or *anno_target* objects for grouping and ordering different plots (as described in Figure 5). Seqpac plots are primarily generated using the *ggplot2* package (47) and

553 outputs are often saved as lists with summarized data and graphs. As with the other
554 seqpac functions, outputs are described in detail in the functions' manuals.

555

556 Since seqpac's reannotation workflow provides a powerful and quick pipeline for
557 sequence annotation, we have also included a 'back-door' function, *PAC_mapper*.
558 This function is ideal for detailed mapping of smaller fasta references, such as a list
559 of tRNAs, the 45S pre-rRNA subunit, the mitochondrial genome, or simply a specific
560 genomic region download as a fasta from a genome browser. Conveniently, if a
561 Bowtie index is missing for a fasta reference, *PAC_mapper* will automatically
562 generate that index, making the alignment of a new fasta reference highly efficient.
563 The output of *PAC_mapper* is a map object, which is simply a list where each entry
564 refers to a specific sequence in the fasta reference and where the coordinates of all
565 PAC sequences that mapped the reference sequence is reported (e.g. the mapping
566 coordinates of PAC sequences aligning to a specific tRNA). This map object along
567 with the original PAC object can then be fed to the *PAC_covplot* function to generate
568 coverage plots across the fasta reference, as exemplified in Figure 8. As we have
569 illustrated before, such coverage plots are well suited for characterizing tRNA and
570 rRNA fragmentation (8,44,48), as well as mitochondrial RNA (48).

571

572 Lastly, using the map object the *map_rangetype* function can generate more
573 advanced classifications such as 5', i' and 3' tRFs or tRNA halves, previously best
574 demonstrated in MINTbase and MINTmap (11). Nonetheless, the
575 MINTmap/MINTbase suite is only readily available for human tRNA classification.
576 Seqpac's *PAC_mapper* and *map_rangetype* functions fills this gap and expands the
577 possibility for discovering novel tRNA fragment classes in any species. With the
578 *map_rangetype* function it is easy to classify sequences in the map object in relation
579 to where the alignment starts or ends in the reference sequence. This is done by
580 either defining different ranges (e.g. classifying a fragment as 5' if it starts within the
581 first 3 nt of a tRNA), or a percentage zone (e.g. classifying a fragment as a half if it
582 ends or starts within 45-55% of a tRNA). Even better, *map_rangetype* may use ss
583 files, which is a format commonly used for storing information about secondary
584 structures such as tRNA loops. Thus, using this option, users can classify fragments
585 in relation to for example cleavage within a specific loop. We used this strategy to
586 identify a diet-sensitive tRNA derived fragment in human sperm, that we called

587 nuclear internal T-loop tRNA derived RNA (nitRNA)(48). The *PAC_trna* function plots
588 range-classified tRNAs mimicking some graphs presented in that paper.

589

590 **7.1 Seqpac example 1: Identifying contaminants by sequence-based counting**

591 To further illustrate the strengths of seqpac, we reanalyzed a recently published
592 dataset by Tong *et al.* 2020 (25) (SRA access: SRP285629). This dataset contains 42
593 fastq files originating from 14 human cancer cell lines, where RNA was extracted
594 from cells, as well as exosomes and microvesicles of these cells. Extra-cellular
595 vesicles—such as microvesicles and exosomes—are cellular excretion particles
596 produced by cells' plasma membrane. They are found from a variety of cells—
597 including tumor cells—in peripheral body fluids (49). Therefore, characterizing the
598 sRNA content of extracellular vesicles from cancer cell lines may reveal novel
599 diagnostic/prognostic biomarkers.

600

601 We generated a PAC object from this dataset. The sequencing was done on an
602 Illumina HiSeq3000 sequencer with a flow cell kit generating read lengths of only 50
603 nt. From our experience, we do not recommend generating sRNA-seq data with read
604 lengths shorter than 75 nt. Longer reads allow for inter-adaptor length validation,
605 where detecting the opposite adaptor sequence in the read guarantees that it
606 originated from short RNA and not from long RNA. Thus, unless controlling for
607 sequence length in downstream analysis, sRNA experiments with very short reads
608 may be severely influenced by long RNA. To investigate if this was a problem in the
609 Tong *et al.* study, we therefore included all read lengths in the analysis.

610

611 Tong *et al.* (25) used a feature-based counting strategy. This strategy first aligns
612 sequences to a reference genome, often allowing for multiple mismatches, and
613 discards sequences that fails to align. Counts are then based on the overlaps
614 between the genomic coordinates of the reads and the genomic coordinates of
615 known sRNA. This poses several problems. The nucleotide sequence of some sRNA
616 may be post-transcriptionally modified, such as 3' fragments of mature tRNAs. These
617 may be discarded since they fail to align with the reference genome. Further, allowing
618 for mismatches without knowing where those mismatches occur and pool related
619 sequences with and without mismatch alignments into the same feature, can hide
620 information about sRNA subtypes and remove traces of post-transcriptional

621 modifications hidden within the reverse-transcription signature (50-52). Many sRNAs
622 are also highly repetitive, such as most rRNA derived fragments, and may thereby
623 map to multiple genomic regions. In feature-based counting strategies this is often
624 solved by randomly assigning such reads to one of the multimapping regions.
625 Together, this completely breaks sequence integrity making it difficult to interpret the
626 results.

627

628 Some of these issues with feature-based counting can be illustrated with the Tong *et*
629 *al.* dataset (25) using seqpac's workflow. It must be emphasized, however, that our
630 critic is not specifically aimed against Tong *et al.*, whose work we admire, but rather
631 against the feature-based counting strategies that hundreds of studies have been
632 using.

633

634 By applying a PCA we confirmed what the original authors reported that cells were
635 very different from extra-cellular vesicles (Figure 7A). We also observed that the
636 extra-cellular vesicles from two specific cell lines—SCC4 and SCC154—were
637 different to the other samples. Size distribution histograms immediately identified two
638 problems (Figure 7B). Firstly, most read sequences were ≥ 50 nt. Since Tong *et al.*
639 reported that the majority of sRNA from cells came from snoRNA, and sRNA from
640 extra-cellular vesicles came from rRNA, it indicates that their analysis did not account
641 for sequence length. This is because most snoRNA and rRNA are found in the ≥ 50
642 nt segment. Thus, the sRNA class proportions that was reported may involve long
643 RNA, possibly including full-length rRNA and tRNA.

644

645 Secondly, the extra-cellular vesicles from SCC4 and SCC154 failed almost
646 completely to align with known human sRNAs (Figure 7B). Since seqpac maintains
647 sequence integrity, we blasted a small selection of these non-annotating sequences
648 at NCBI (53). The result strongly indicated that most reads originated from the
649 *Mycoplasma hyorhinis* genome. Since this is a common contaminant in cell cultures
650 (54), we ran this *Mycoplasma* genome in parallel to the human genome in the
651 seqpac's reannotation workflow, thereby picking the best possible alignment from
652 either of them. This showed that all vesicle samples from SCC4 and SCC154—the
653 same samples that explained one of the main components in the PCA—suffered
654 severely from *Mycoplasma* contamination (Figure 7C).

655

656 Now, critics may argue that a feature-based counting strategy should have corrected
657 for this contamination, since reads that fail to align against the human genome will
658 automatically be removed prior to counting. Thus, Mycoplasma reads should not
659 have affected the results, since normalization of the counts was made after their
660 removal.

661

662 We tested this assumption using seqpac functions. With the output from the two-
663 genome reannotation workflow, we used the *PAC_filter* function to remove all
664 sequences that mapped to the *Mycoplasma* genome, and only kept reads that
665 mapped to the human genome. Then we re-normalized the dataset using the
666 *PAC_norm* function and made a new PCA. Removing nearly 6500 sequences, and
667 keeping only sequences exclusive to the human genome, had very limited effects on
668 the results (Figure 7D). This strongly indicates that the effect of the contamination
669 remained even after removing the contaminating sequences. Importantly, this bias
670 may have gone unnoticed if we would have used a feature-based counting strategy,
671 since contaminating sequences would have been removed prior to counting.
672 Together, this illustrates how seqpac quickly provides panoptic views of data integrity,
673 which is essential for analytical transparency and correct downstream interpretations.
674

675 **7.2 Seqpac example 2: Novel rRNA-derived sRNA affected by anticancer** 676 **treatment**

677 In cancer research, non-coding RNA has been studied not only for diagnostic and
678 prognostic purposes, but also for therapeutic purposes (55). Of particular interest,
679 rRNA synthesis is commonly exaggerated in tumor cells (56). Synthesis involves
680 transcription of 47S/45S pre-rRNA genes by RNA polymerase I at specific repetitive
681 clusters in the genome (57). Over a series of precursors, pre-rRNA is turned into the
682 active mature rRNA subunits 28S, 18S and 5.8S (58,59). Inhibiting RNA polymerase I
683 (RNA pol I) has been proposed as a possible anticancer treatment, where one of the
684 most promising candidates have been the BMH21 compound (60). However, little is
685 known about sRNA generated from the pre-rRNA and their potential role in cancer.
686

687 We, therefore, used seqpac to detect novel sRNA originating from pre-rRNA,
688 hypothesizing that inhibiting RNA pol I would result in fewer rRFs. For this we

689 conducted a small experiment by exposing HeLa cells—which originates from
690 cervical cancer cells—to BMH21. The exposure time was set to 60 min, and as
691 control we used DMSO. RNA from purified cells was then prepared for sRNA-seq,
692 which resulted in fastq files with 75 nt reads. From this raw data we generated an
693 annotated and filtered PAC object using only seqpac functions.

694

695 Using the *PAC_deseq* (see Results 5.3) function, we performed a differential
696 expression analysis only including highly expressed sRNA mapping to rRNA
697 reference sequences. This showed that only 60 min of BMH21 exposure was enough
698 to affect rRNA fragmentation (Figure 8A). Perhaps unexpectedly, not all were
699 downregulated by inhibiting RNA polymerase I. In fact, closer examination revealed
700 that most down-regulated sequences were related (Figure 8A; Supplementary table
701 S1), suggesting a single origin within an rRNA cluster on chromosome 21. We,
702 therefore, used the *PAC_mapper* and *PAC_covplot* functions (see Results 6.1) to
703 visualize the impact of BMH21 over a pre-rRNA 45S gene on chromosome 21
704 (GenBank: NR_146144.1). This revealed 4 major rRFs (Peak 1, 2, 3, 4 in Figure 8B),
705 where the related fragments from Figure 8A all aligned to Peak 1. For more detailed
706 analysis, we downloaded the sequences of the DNA immediately neighboring these
707 peaks from the UCSC genome browser and ran the sequences as a fasta reference
708 file in the *PAC_mapper* and *PAC_covplot* functions (Supplementary file S2). This
709 revealed what appeared to be a single large down-regulated fragment in Peak 1
710 (Figure 8C), an unaffected possibly degraded fragment in Peak 2 (Figure 8D), two
711 separate fragments in Peak 3 where only the shorter and less expressed fragment
712 might have been affected by BMH21 (Peak 3a in Figure 8E), and one single fragment
713 in Peak 4 that seemed slightly up-regulated following BMH21 treatment.

714

715 To better understand the relevance of these changes we summed the cpm of all
716 fragments mapping to each peak and performed a non-parametric Mann-Whitney U
717 test. For this analysis we also included a third group of samples that had been
718 exposed to BMH21 for 12 hours, to explore if any of the effects of BMH21 were
719 amplified following long-term exposure. Astonishingly, after 12 h exposure, fragments
720 of Peak 1 had almost completely disappeared (Figure 8F). This was not due to an
721 experimental failure since Peak 2 and Peak 3a were unaffected by the long-term
722 treatment (Figure 8G-H). In fact, Peak 4 fragments even showed a significant up-

723 regulation (Figure 8I). Thus, the effects observed in Peak 1 and Peak 4 were
724 amplified by long-term exposure, but in two opposite directions.

725

726 For the Peak 2 and Peak 3 rRNA fragments we have previously observed similar
727 fragments in human sperm (8), and similar fragments located to the 5' ends of the
728 5.8S and 28S subunits in fruit fly embryos (44). This is also true for the 3' fragments
729 of the 28S subunit (Peak 4), even though we never have observed such expression
730 levels as we see in the HeLa cells. To our knowledge, however, highly expressed
731 sRNA fragments from the Peak 1 region—in the 5' external transcribed spacers
732 (ETS)—have never been described. To understand the 5' ETS rRF better, we
733 performed a multi-species blast of the main sequence at NCBI to identify similar
734 GenBank entries. This showed many alignments to ribosomal precursors in humans,
735 one identical sequence in the Chimpanzee, and a few similar sequences in the
736 Gorilla (Supplementary Figure S1). Thus, this 5' ETS rRF has only evolved in our
737 closest relatives.

738

739 Confident that the 5' ETS rRF was a human sRNA, we searched for this fragment in
740 the Tong *et al.* 2020 dataset. Despite only having read lengths of 50 nt to our disposal
741 (see Results 7.1), where 5' ETS rRF of Peak 1 was 61 nt, we found clear traces of
742 this rRF (Figure 8J). Furthermore, to explore the clinical relevance of this finding we
743 downloaded the Xu *et al.* dataset (26,27). Here sRNA was extract from confirmed
744 cervical tumors and samples from normal cervix. Results indicated that the 5' ETS
745 rRF was upregulated in cancer patients (Figure 8K). Together this suggests that our
746 novel rRF—validated by the seqpac workflow in multiple unrelated datasets—may be
747 targeted for diagnostic and prognostic purposes during cancer treatment.

748

749 **DISCUSSION**

750 Here we presented a novel and innovative bioinformatic tool—seqpac—that makes
751 advanced sRNA analysis from genome-scale sequencing data more accessible and
752 transparent. The workflow is completely integrated with R, from trimming the adaptor
753 sequences to generating plots. We showed that seqpac's trimming function performs
754 as well as, or even better, than trimming using standard tools outside R. We further
755 presented the PAC object, which builds a framework of phenotypic information (P)

756 and sequence annotations (A) around a table based on sequence counts (C). Using
 757 published data we showed that a sequence-based counting strategy—in contrast to
 758 feature-based counting that is more commonly used—diminishes the risk of mistakes
 759 in downstream analysis. We demonstrated the strength of maintaining sequence
 760 integrity to enable re-annotation of sequences across species and classes of sRNA at
 761 any point in the analysis. Lastly, we showed how seqpac can be used for sRNA
 762 discovery in cancer research by the discovery of a novel rRNA derived fragment
 763 (rRF) that were down-regulated by anti-cancer treatment *in vitro* and up-regulated in
 764 tumors of cervical cancer patients.

765

766 Seqpac is available at *github* (<https://github.com/Danis102/seqpac>). As the whole
 767 workflow, from adaptor trimming to mapping and plotting, are integrated in R it runs
 768 on common computer platforms, including Windows, Mac and Linux. It comes with a
 769 complete collection of function manuals and a vignette that guides the user in how to
 770 apply the default workflow using a fastq test dataset that are included with the
 771 package. R scripts that we used to generate many of the results presented in this
 772 paper are available in Supplementary file S1.

773

774 It must be emphasized that seqpac is primarily designed for sRNA sequence
 775 analysis. This means that it does not currently supports paired-end sequencing,
 776 which is commonly applied for long RNA sequencing. Paired-end sequencing is not
 777 required for most sRNA applications where the target sequence lengths seldom
 778 exceed 75 nt. As we have demonstrated in this paper, too short reads—as those
 779 generated using the 50-cycle flow cell kits available for MiSeq, NextSeq1000 and
 780 HiSeq2500/3000/4000—should be avoided. Without some excessive sequence in
 781 which the 3' adaptor can be detected, it is difficult to reliably discriminate medium
 782 length sRNA (such our novel 5' ETS rRF) from unintentionally included longer RNA.

783

784 We see, however, many advantages to use sequence-based counting also in long
 785 RNA sequence analysis, for example to easily extract sequences annotating to a
 786 candidate mRNA and check for possible genetic variants. Coverage plots, similar to
 787 what we describe for the 45S pre-rRNA (Figure 8B) would also be applicable for
 788 mRNA coverage to visualize splice variants and intronic transcription. Even though

789 we hope to develop long RNA analysis in future updates of seqpac, there are
790 currently a few technical constraints that needs to be resolved.

791

792 As mentioned, paired-end reads are not supported either for trimming or counting. In
793 addition, while *Bowtie* (20) is still the most popular aligner for sRNA, it does not
794 support indel mapping. While this is not a great problem if sequence integrity is
795 maintained and candidate sequences subsequently can be blasted to detect any slip
796 through, this problem are slightly more announced in samples differing much from
797 their reference genomes, such as cancer cell-lines. A likely reason for Bowtie's
798 popularity in sRNA community is because it is reliable with short sequence
799 alignments. For instance, we initially tried to integrate the *Rsubreads* package (61) in
800 seqpac's workflow, which applies a highly efficient 'seed-and-vote' mapping
801 algorithm. However, for certain read lengths we consistently experienced failure to
802 correctly vote for the best alignment, possibly as a consequence that too few seeds
803 were covering the read. We will off-course explore more efficient alternatives to
804 *Bowtie* in the future.

805

806 By using the sequence-based approach of seqpac, we have discovered a novel
807 rRNA derived sRNA (rRF) in the 5' ETS of 45S pre-rRNA. This rRF responds
808 negatively to anticancer treatment and are up-regulated in tumors. The scope of our
809 study was not to dwell deep into the mechanism and clinical potential of this rRF. To
810 our knowledge, however, this fragment has not been described before, and from our
811 experience sRNA in the 5' ETS of pre-rRNAs are rare. This, together with the insight
812 that the sequence is relatively unique to humans (with only some homology in
813 Chimpanzees and Gorillas), makes it a good target for future studies on biomarkers
814 in cancer treatment and diagnosis. In our HeLa cell experiment, the main fragment
815 was 61 nt, which indicates a unique fragment given that we had a maximum read
816 length of 75 nt. Even though the methods used in Tong *et al.* (25) and Xu *et al.*
817 (26,27) were restricted to a maximum read length of 50 nt, we found traces of this
818 fragment in the pile of fragments with unverifiable length of ≥ 50 nt. It must be
819 emphasized, however, that we tried to validate the 5' ETS rRF in yet another dataset,
820 Snoek *et al.* (62) (SRA accession: PRJNA413777), but here we failed to detect
821 anything in the 5' EST region. The Snoek *et al.* dataset is so far the largest public
822 sRNA dataset from cervical cancer patients. In this study, samples were collected by

participants themselves, which may explain much higher rRF variability and excessive number of short fragments (< 20 nt), compared to the other datasets (Supplementary file S1). Importantly, in contrast to the other datasets that used the NEBNext Small RNA Library Prep kit, Snoek *et al.* used the Illumina TruSeq Small RNA Library Preparation Kit. We and others have consistently shown that these two popular kits perform differently with regard to sRNA coverage (48,63-65). Thus, future studies targeting this rRF must consider the choice of chemistry, and maybe even apply advanced protocols for better coverage (44,66).

The 5' ETS rRF should be located somewhere close to the 01 cleavage sites in the 5' EST of 47S pre-rRNA. When aligning the 61 nt main fragment against the GenBank entry U13369.1—which have been commonly used to map Human pre-rRNA cleavage sites (59)—the 5' ETS rRF only partly align. The coverage over this area in the U13369.1 pre-rRNA is also far from what we observed for the NR_146144.1 pre-rRNA, which was the GenBank sequence that we used for the chromosome 21 alignment in (Figure 8B; Supplementary File S2). Aligning the U13369.1 with NR_146144.1 reveals a “G-T” insertion in NR_146144.1, right between the C414-C416 and G420-U422 01 cleavage sites (67) (Figure 8K; Supplementary Figure S2). Since the 5' ETS rRF contains this insertion, it strongly suggests that pre-rRNA cleavage has been affected at this locus. Therefore, beside investigating possible clinical values of this 5' ETS rRF, future research may target mechanisms for how natural rRNA variants may give rise to novel sRNA, suggestively by investigating the interactions between post-transcriptional modifications, snoRNA and proteins at this locus.

In conclusion, the revolution of genome scale sequencing has not only brought enormous potential for unraveling life’s mysteries in health and disease. It has also created a gap between biology and technology. Consequently, research groups with primary biological or medical interests are often forced to rely on specialized programmers with limited understanding of the biology to handle their precious data. R has long been a platform where bridges between biology, statistics and programming are built. We have showed that building a transparent workflow in R for sRNA analysis, with the intention of making choices early in the analysis perspicuous, not only helps in detecting severe biases that would have otherwise gone

857 undiscovered. It also provides the flexibility and panoptic view needed for advanced
858 biological interpretations.

859 REFERENCES

- 860 1. Djuranovic, S., Nahvi, A. and Green, R. (2012) miRNA-Mediated Gene Silencing by
861 Translational Repression Followed by mRNA Deadenylation and Decay. *Science*, **336**,
862 237-240.
- 863 2. Iwakawa, H.-o. and Tomari, Y. (2015) The Functions of MicroRNAs: mRNA Decay and
864 Translational Repression. *Trends in Cell Biology*, **25**, 651-665.
- 865 3. Wei, Y., Li, L., Wang, D., Zhang, C.-Y. and Zen, K. (2014) Importin 8 regulates the
866 transport of mature microRNAs into the cell nucleus. *The Journal of biological*
867 *chemistry*, **289**, 10270-10275.
- 868 4. Castel, S.E. and Martienssen, R.A. (2013) RNA interference in the nucleus: roles for
869 small RNAs in transcription, epigenetics and beyond. *Nature reviews. Genetics*, **14**, 100-
870 112.
- 871 5. Liu, H., Lei, C., He, Q., Pan, Z., Xiao, D. and Tao, Y. (2018) Nuclear functions of
872 mammalian MicroRNAs in gene regulation, immunity and cancer. *Molecular Cancer*,
873 **17**, 64.
- 874 6. Ernst, C., Odom, D.T. and Kutter, C. (2017) The emergence of piRNAs against
875 transposon invasion to preserve mammalian genome integrity. *Nature Communications*,
876 **8**, 1411.
- 877 7. Lambert, M., Benmoussa, A. and Provost, P. (2019) Small Non-Coding RNAs Derived
878 from Eukaryotic Ribosomal RNA. *Non-Coding RNA*, **5**, 16.
- 879 8. Nätt, D. and Öst, A. (2020) Male reproductive health and intergenerational metabolic
880 responses from a small RNA perspective. *J Intern Med*.
- 881 9. Thompson, D.M. and Parker, R. (2009) Stressing out over tRNA cleavage. *Cell*, **138**,
882 215-219.
- 883 10. Sun, C., Fu, Z., Wang, S., Li, J., Li, Y., Zhang, Y., Yang, F., Chu, J., Wu, H., Huang, X.
884 *et al.* (2018) Roles of tRNA-derived fragments in human cancers. *Cancer Letters*, **414**,
885 16-25.
- 886 11. Loher, P., Telonis, A.G. and Rigoutsos, I. (2017) MINTmap: fast and exhaustive
887 profiling of nuclear and mitochondrial tRNA fragments from short RNA-seq data.
888 *Scientific reports*, **7**, 41184-41184.
- 889 12. Hombach, S. and Kretz, M. (2016) Non-coding RNAs: Classification, Biology and
890 Functioning. *Adv Exp Med Biol*, **937**, 3-17.
- 891 13. Esteller, M. (2011) Non-coding RNAs in human disease. *Nat Rev Genet*, **12**, 861-874.
- 892 14. Shi, J., Ko, E.-A., Sanders, K.M., Chen, Q. and Zhou, T. (2018) SPORTS1. 0: a tool for
893 annotating and profiling non-coding RNAs optimized for rRNA-and tRNA-derived
894 small RNAs. *Genomics, proteomics & bioinformatics*, **16**, 144-151.
- 895 15. Rueda, A., Barturen, G., Lebrón, R., Gómez-Martín, C., Alganza, Á., Oliver, J.L. and
896 Hackenberg, M. (2015) sRNAtoolbox: an integrated collection of small RNA research
897 tools. *Nucleic Acids Research*, **43**, W467-W473.
- 898 16. Wu, X., Kim, T.-K., Baxter, D., Scherler, K., Gordon, A., Fong, O., Etheridge, A., Galas,
899 D.J. and Wang, K. (2017) sRNAAnalyzer—a flexible and customizable small RNA
900 sequencing data analysis pipeline. *Nucleic Acids Research*, **45**, 12140-12151.
- 901 17. Li, J., Kho, A.T., Chase, R.P., Pantano, L., Farnam, L., Amr, S.S. and Tantisira, K.G.
902 (2020) COMPSRA: a COMprehensive Platform for Small RNA-Seq data Analysis.
903 *Scientific Reports*, **10**, 4552.
- 904 18. Panero, R., Rinaldi, A., Memoli, D., Nassa, G., Ravo, M., Rizzo, F., Tarallo, R.,
905 Milanesi, L., Weisz, A. and Giurato, G. (2017) iSmaRT: a toolkit for a comprehensive
906 analysis of small RNA-Seq data. *Bioinformatics*, **33**, 938-940.

- 907 19. Martin, M. (2011) Cutadapt removes adapter sequences from high-throughput
908 sequencing reads. *2011*, **17**, 3.
- 909 20. Langmead, B., Trapnell, C., Pop, M. and Salzberg, S.L. (2009) Ultrafast and memory-
910 efficient alignment of short DNA sequences to the human genome. *Genome biology*, **10**,
911 R25.
- 912 21. Liao, Y., Smyth, G.K. and Shi, W. (2014) featureCounts: an efficient general purpose
913 program for assigning sequence reads to genomic features. *Bioinformatics*, **30**, 923-930.
- 914 22. Kang, W., Eldfjell, Y., Fromm, B., Estivill, X., Biryukova, I. and Friedländer, M.R.
915 (2018) miRTrace reveals the organismal origins of microRNA sequencing data. *Genome*
916 *biology*, **19**, 1-15.
- 917 23. Ooi, H., Weston, S. and Microsoft. (2020), Vol. R package version 1.5.0.
- 918 24. Kusnierczyk, W. (2012). 1.0.0.
919 ed.
- 920 25. Tong, F., Andress, A., Tang, G., Liu, P. and Wang, X. (2020) Comprehensive profiling
921 of extracellular RNA in HPV-induced cancers using an improved pipeline for small
922 RNA-seq analysis. *Sci Rep*, **10**, 19450.
- 923 26. Xu, J., Zou, J., Wu, L. and Lu, W. (2020) Transcriptome analysis uncovers the diagnostic
924 value of miR-192-5p/HNF1A-AS1/VIL1 panel in cervical adenocarcinoma. *Sci Rep*,
925 **10**, 16584.
- 926 27. Xu, J., Zhang, Y., Huang, Y., Dong, X., Xiang, Z., Zou, J., Wu, L. and Lu, W. (2020)
927 circEYA1 Functions as a Sponge of miR-582-3p to Suppress Cervical Adenocarcinoma
928 Tumorigenesis via Upregulating CXCL14. *Mol Ther Nucleic Acids*, **22**, 1176-1190.
- 929 28. Ritchie, M.E., Phipson, B., Wu, D., Hu, Y., Law, C.W., Shi, W. and Smyth, G.K. (2015)
930 limma powers differential expression analyses for RNA-sequencing and microarray
931 studies. *Nucleic Acids Research*, **43**, e47-e47.
- 932 29. Love, M.I., Huber, W. and Anders, S. (2014) Moderated estimation of fold change and
933 dispersion for RNA-seq data with DESeq2. *Genome Biology*, **15**, 550.
- 934 30. Aryee, M.J., Jaffe, A.E., Corrada-Bravo, H., Ladd-Acosta, C., Feinberg, A.P., Hansen,
935 K.D. and Irizarry, R.A. (2014) Minfi: a flexible and comprehensive Bioconductor
936 package for the analysis of Infinium DNA methylation microarrays. *Bioinformatics*, **30**,
937 1363-1369.
- 938 31. Morgan, M., Anders, S., Lawrence, M., Aboyoun, P., Pagès, H. and Gentleman, R.
939 (2009) ShortRead: a bioconductor package for input, quality assessment and exploration
940 of high-throughput sequence data. *Bioinformatics*, **25**, 2607-2608.
- 941 32. Pages, H., Aboyoun, P., Gentleman, R. and DebRoy, S. (2018) Biostrings: String objects
942 representing biological sequences, and matching algorithms v2.48.0. *R package*.
- 943 33. Hannon, G., Gordon, A. and etc. (2010). 0.0.13 ed.
- 944 34. Kircher, M., Heyn, P. and Kelso, J. (2011) Addressing challenges in the production and
945 analysis of illumina sequencing data. *BMC genomics*, **12**, 1-14.
- 946 35. Hahne, F., Lerch, A. and Stadler, M. (2012).
- 947 36. Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L.D.A., François, R.,
948 Grolemond, G., Hayes, A., Henry, L. and Hester, J. (2019) Welcome to the Tidyverse.
949 *Journal of Open Source Software*, **4**, 1686.
- 950 37. Yates, A.D., Achuthan, P., Akanni, W., Allen, J., Allen, J., Alvarez-Jarreta, J., Amode,
951 M.R., Armean, I.M., Azov, A.G., Bennett, R. *et al.* (2019) Ensembl 2020. *Nucleic Acids*
952 *Research*, **48**, D682-D688.
- 953 38. Lawrence, M., Huber, W., Pages, H., Aboyoun, P., Carlson, M., Gentleman, R., Morgan,
954 M.T. and Carey, V.J. (2013) Software for computing and annotating genomic ranges.
955 *PLoS computational biology*, **9**, e1003118.
- 956 39. Smit, A.F.A., Hubley, R. and Green, P. (2013-2015).

- 957 40. Kozomara, A., Birgaoanu, M. and Griffiths-Jones, S. (2018) miRBase: from microRNA
958 sequences to function. *Nucleic Acids Research*, **47**, D155-D162.
- 959 41. Wang, J., Zhang, P., Lu, Y., Li, Y., Zheng, Y., Kan, Y., Chen, R. and He, S. (2018)
960 piRBase: a comprehensive database of piRNA sequences. *Nucleic Acids Research*, **47**,
961 D175-D180.
- 962 42. Chan, P.P. and Lowe, T.M. (2015) GtRNAdb 2.0: an expanded database of transfer RNA
963 genes identified in complete and draft genomes. *Nucleic Acids Research*, **44**, D184-
964 D189.
- 965 43. Wickersheim, M.L. and Blumenstiel, J.P. (2013) Terminator oligo blocking efficiently
966 eliminates rRNA from Drosophila small RNA sequencing libraries. *BioTechniques*, **55**,
967 269-272.
- 968 44. Kugelberg, U., Nätt, D., Skog, S., Kutter, C. and Öst, A. (2021) 5' XP sRNA-seq:
969 efficient identification of transcripts with and without 5' phosphorylation reveals
970 evolutionary conserved small RNA. *RNA biology*, 1-12.
- 971 45. Lê, S., Josse, J. and Husson, F. (2008) FactoMineR: an R package for multivariate
972 analysis. *Journal of statistical software*, **25**, 1-18.
- 973 46. Kassambara, A. and Mundt, F. (2017) Package 'factoextra'. *Extract and visualize the*
974 *results of multivariate data analyses*, **76**.
- 975 47. Wickham, H. (2016) *ggplot2: elegant graphics for data analysis*. Springer.
- 976 48. Nätt, D., Kugelberg, U., Casas, E., Nedstrand, E., Zalavary, S., Henriksson, P., Nijm,
977 C., Jäderquist, J., Sandborg, J., Flincke, E. *et al.* (2019) Human sperm displays rapid
978 responses to diet. *PLOS Biology*, **17**, e3000559.
- 979 49. Xu, R., Rai, A., Chen, M., Suwakulsiri, W., Greening, D.W. and Simpson, R.J. (2018)
980 Extracellular vesicles in cancer—implications for future improvements in cancer care.
981 *Nature reviews Clinical oncology*, **15**, 617-638.
- 982 50. Motorin, Y. and Helm, M. (2019) Methods for RNA Modification Mapping Using Deep
983 Sequencing: Established and New Emerging Technologies. *Genes (Basel)*, **10**, 35.
- 984 51. Hauenschild, R., Tserovski, L., Schmid, K., Thüring, K., Winz, M.-L., Sharma, S.,
985 Entian, K.-D., Wacheul, L., Lafontaine, D.L. and Anderson, J. (2015) The reverse
986 transcription signature of N-1-methyladenosine in RNA-Seq is sequence dependent.
987 *Nucleic acids research*, **43**, 9950-9964.
- 988 52. Kuksa, P.P., Leung, Y.Y., Vandivier, L.E., Anderson, Z., Gregory, B.D. and Wang, L.-S.
989 (2017), *RNA Methylation*. Springer, pp. 211-229.
- 990 53. (2018) Database resources of the National Center for Biotechnology Information.
991 *Nucleic Acids Res*, **46**, D8-d13.
- 992 54. Drexler, H.G. and Uphoff, C.C. (2002) Mycoplasma contamination of cell cultures:
993 Incidence, sources, effects, detection, elimination, prevention. *Cytotechnology*, **39**, 75-
994 90.
- 995 55. Wang, W.-T., Han, C., Sun, Y.-M., Chen, T.-Q. and Chen, Y.-Q. (2019) Noncoding
996 RNAs in cancer therapy resistance and targeted drug development. *Journal of*
997 *Hematology & Oncology*, **12**, 55.
- 998 56. Montanaro, L., Treré, D. and Derenzini, M. (2008) Nucleolus, Ribosomes, and Cancer.
999 *The American Journal of Pathology*, **173**, 301-310.
- 1000 57. Pederson, T. (2011) The nucleolus. *Cold Spring Harb Perspect Biol*, **3**.
- 1001 58. Haag, J.R. and Pikaard, C.S. (2007) RNA polymerase I: a multifunctional molecular
1002 machine. *Cell*, **131**, 1224-1225.
- 1003 59. Mullineux, S.T. and Lafontaine, D.L. (2012) Mapping the cleavage sites on mammalian
1004 pre-rRNAs: where do we stand? *Biochimie*, **94**, 1521-1532.
- 1005 60. Peltonen, K., Colis, L., Liu, H., Trivedi, R., Moubarek, Michael S., Moore, Henna M.,
1006 Bai, B., Rudek, Michelle A., Bieberich, Charles J. and Laiho, M. (2014) A Targeting

- Modality for Destruction of RNA Polymerase I that Possesses Anticancer Activity. *Cancer Cell*, **25**, 77-90.
61. Liao, Y., Smyth, G.K. and Shi, W. (2019) The R package Rsubread is easier, faster, cheaper and better for alignment and quantification of RNA sequencing reads. *Nucleic Acids Res*, **47**, e47.
62. Snoek, B.C., Verlaet, W., Babion, I., Novianti, P.W., van de Wiel, M.A., Wilting, S.M., van Trommel, N.E., Bleeker, M.C.G., Massuger, L., Melchers, W.J.G. *et al.* (2019) Genome-wide microRNA analysis of HPV-positive self-samples yields novel triage markers for early detection of cervical cancer. *Int J Cancer*, **144**, 372-379.
63. Giraldez, M.D., Spengler, R.M., Etheridge, A., Godoy, P.M., Barczak, A.J., Srinivasan, S., De Hoff, P.L., Tanriverdi, K., Courtright, A., Lu, S. *et al.* (2018) Comprehensive multi-center assessment of small RNA-seq methods for quantitative miRNA profiling. *Nat Biotechnol*, **36**, 746-757.
64. Dard-Dascot, C., Naquin, D., d'Aubenton-Carafa, Y., Alix, K., Thermes, C. and van Dijk, E. (2018) Systematic comparison of small RNA library preparation protocols for next-generation sequencing. *BMC Genomics*, **19**, 118.
65. Yeri, A., Courtright, A., Danielson, K., Hutchins, E., Alsop, E., Carlson, E., Hsieh, M., Ziegler, O., Das, A., Shah, R.V. *et al.* (2018) Evaluation of commercially available small RNASeq library preparation kits using low input RNA. *BMC Genomics*, **19**, 331.
66. Mohr, S., Ghanem, E., Smith, W., Sheeter, D., Qin, Y., King, O., Polioudakis, D., Iyer, V.R., Hunicke-Smith, S., Swamy, S. *et al.* (2013) Thermostable group II intron reverse transcriptase fusion proteins and their use in cDNA synthesis and next-generation RNA sequencing. *Rna*, **19**, 958-970.
67. Kass, S., Craig, N. and Sollner-Webb, B. (1987) Primary processing of mammalian rRNA involves two adjacent cleavages and is not species specific. *Mol Cell Biol*, **7**, 2891-2898.

1034 TABLES AND FIGURES

Table 1. Quick reference for seqpac

Family	Description	Function name	Dependence	Function description
PAC generation	Prepares and builds a PAC object from fastq files.	<i>make_counts</i>	<i>foreach, ShortReads</i>	Reads fastq, calls <i>make_trim</i> or <i>make_cutadapt</i> , performs low-level filter, and counts sequences.
		<i>make_trim</i>	<i>foreach, Biostrings</i>	Adaptor trimming and fastq quality filter using R internal packages
		<i>make_cutadapt</i>	<i>foreach, cutadapt, fastq_quality_filter</i>	Adaptor trimming and fastq quality filter using external installed software
		<i>make_anno</i>	-	Generates a simple annotation table from a count table.
		<i>make_pheno</i>	-	Prepares user provided phenotype table containing sample information.
		<i>make_PAC</i>	-	Builds PAC object with Pheno, Anno and Counts tables.
		<i>PAC_check</i>	-	Checks if a PAC object is compatible with seqpac functions
PAC annotation	Sequence annotations by aligning against fasta references or overlap with genomic coordinates.	<i>map_reanno</i>	<i>Rbowtie/bowtie</i>	Reannotates PAC sequences through progressive mismatch cycles.
		<i>import_reanno</i>	<i>foreach, tibble</i>	Called by <i>map_reanno</i> to import bowtie output into R.
		<i>add_reanno</i>	<i>foreach, tibble</i>	Builds a reannotation object with genome coordinates and/or classified fasta sequence names.
		<i>simplify_reanno</i>	-	Makes hierarchical classifications from classified fasta sequence names.
		<i>PAC_gtf</i>	<i>tibble, rtracklayer, GenomicRanges</i>	Overlaps genomic coordinates of PAC sequences with features of a gtf file.
		<i>PAC_mapper</i>		Backdoor to the reanno workflow for fast mapping of PAC resulting in a map object.
		<i>map_rangetype</i>		Classifies sequences in map objects according to ranges or secondary structures (e.g. 5', i', 3')
PAC analysis (preprocessing)	Filtering and normalization.	<i>PAC_filter</i>	-	Subsets data by target objects or coverage thresholds.
		<i>PAC_filtersep</i>	-	Extracts sequences reaching a threshold within groups of a pheno_target object.
		<i>PAC_norm</i>	<i>DESeq2</i>	Normalize a raw counts table and saves it in the PAC\$norm 'folder'.
PAC analysis (statistics)	Performs statistical analyses and visualizations.	<i>PAC_summary</i>	-	Simple summaries using pheno_targets (means, sd, se, %diff, log2fc) saved in PAC\$summary.
		<i>PAC_deseq</i>	<i>foreach, DESeq2</i>	Prepares, performs and plots DESeq2 analysis from PAC object.
		<i>PAC_pca</i>	<i>FactoMineR, factoextra</i>	Performs principal component analysis and plots the results.
		<i>PAC_saturation</i>	<i>foreach, ggplot2</i>	Performs a sequence saturation analysis and plots the results.
PAC analysis (visualization)	Generates graphs and saves processed data summarized over both phenotype and annotations.	<i>PAC_pie</i>	<i>ggplot2, cowplot, grDevices</i>	Pie-plots using pheno_target and anno_target.
		<i>PAC_stackbar</i>	<i>ggplot2, ggthemes, reshape2, grDevices</i>	Stacked bar diagrams using pheno_target and anno_target.
		<i>PAC_jitter</i>	<i>ggplot2</i>	Jitter plots using pheno_target and anno_target.
		<i>PAC_nbias</i>	<i>ggplot2, ggthemes, grDevices</i>	Size distributed histogram stacked by nucleotide at a defined position (e.g. 1st nucleotide bias).
		<i>PAC_sizedist</i>	<i>ggplot2, ggthemes, grDevices</i>	Size distributed bars stacked by an anno_target column (e.g. miRNA/piRNA size distributions).
		<i>PAC_covplot</i>	<i>ggplot2, reshape2, grDevices, GenomicRanges</i>	Plots PAC sequence coverage over a reference sequence such as a tRNA or rRNA.
		<i>PAC_trna</i>	<i>ggplot2, reshape2, grDevices</i>	tRNA fragment analysis using information from range-classified map object.

All functions are described in detail in the manual for each function (e.g. '?make_counts' in the R terminal) and are exemplified in the seqpac vignette; 'vignette("seqpac")' in R terminal).

Figure 1. Small RNA sequence analysis using seqpac. (A) The difference between sequence-based counting and feature-based counting in sRNA analysis. With sequence-based counting an experiment-wide count table can be created before genome alignment. This allows for efficient mapping and for sequence integrity to be maintained through the analysis. In contrast, feature-based counting strategies counts overlaps between reads' genome alignment and coordinates of genomic features. This is less efficient and disrupts sequence integrity. (B) Sequence-based counting is central in the seqpac workflow, which is completely integrated in R, from fastq adaptor trimming and preprocessing to group-based visualization and statistical analysis. (C) Seqpac builds a framework of functions that processes and analyzes a standardized list: the PAC object. In its simplest form PAC contains three tables: the *Pheno* table with sample information; the *Anno* table with sequence information, and the *Counts* table containing the counts of sequences across samples.

Figure 2. Performance of seqpac's internal trimming and counting functions. Seqpac contains multiple options for trimming and filtering adaptor sequences prior to generating a count table. The *make_counts* function counts sequences of already trimmed fastq files or calls *make_trim* (R internal) or *make_cutadapt* (R external) functions prior to counting. Using the Kang *et al.* 2018 dataset (SRA access: PRJNA485638), (A) shows side-by-side the preprocessing time for seqpac's trimming functions and a popular alternative workflow based on the *cutadapt* and *fastq_quality_filter* functions. The test involved 7 fastq files iterated 10 times over 6 batches per function using 7 parallel processes. (B-D) Further evaluated performance of *make_trim* in terms of the output dataset. (B) While sequence counts strongly correlated with the alternative workflow, *make_trim* more often generated higher counts (arrows). This was primarily a result of concatemer trimming in *make_trim* (see main text). (C) The *make_counts* function also contains an evidence filter, which in default mode discard sequences that fails to replicate in at least two independent fastq files. Normally, this low-level filter maintains most reads (top bars), while limiting the sequence diversity (bottom bars). While *make_trim* and the alternative generated very similar datasets after evidence filters, *make_trim* generated slightly more unique sequences, which was confirmed by Venn-diagram (D) showing higher ratio of sequences unique to the *make_trim* workflow.

Figure 3. Annotating read sequences in PAC objects. For annotation of read sequences, seqpac mainly relies on the re-annotation workflow [1-3]. The *map_reanno* and *import_reanno* functions use *Bowtie* to align PAC sequences against references sequences, e.g. species genome or sRNA database [1]. This is done over cycles where each cycle introduces 1 additional mismatch in the mapping, and where only read sequences with no alignment proceed to the next cycle. After the mismatch cycles, *add_reanno* reads the resulting .Rdata files and organize the output into a reannotation list object [2]. Tables in this list can either directly be merged with a PAC annotation table or can be simplified hierarchically using the *simplify_reanno* function [3]. The *PAC_mapper* function is a convenient wrapper for smaller reference sequences (e.g. tRNAs or rRNAs) that will automatically generate *Bowtie* indexes. [4] After a PAC object has been aligned to a genome, the *PAC_gtf* can be used to overlap genomic coordinates of PAC sequences with known coordinates for genomic features, e.g. repeats and protein coding exons.

Figure 4. Multi-genome sRNA analysis demonstrating the strength in seqpac reannotation workflow. Graphs were plotted using a PAC object generated from the Kang *et al.* 2018 dataset (SRA access: PRJNA485638) primarily developed for studying interspecies contaminated samples where RNA from fly (S2) and human (HEK-293T) cells was mixed in different ratios. Seqpac functions used for generating the graphs were: (A) *PAC_pie* for genome proportion pie charts, (B) *PAC_sizedist* for size distribution histograms with sRNA class annotation, and (C) *PAC_nbias* for the frequency of the first nucleotide stratified over the sequence size distribution.

Pheno

	Bio_sample	Type	Replicate
S1_Ov1	S1	Ovaries	1
S1_Ov2	S1	Ovaries	2
S1_Ov3	S1	Ovaries	3
S5_Sp1	S5	Sperm	1
S5_Sp2	S5	Sperm	2
S5_Sp3	S5	Sperm	3

pheno_target=list("Type") → All Types as they appear in column
 pheno_target=list("Replicate", c(1, 3)) → Extract only replicate 1 and 3
 pheno_target=list("Type", c("Sperm", "Ovaries")) → Change order of Ovaries and Sperm

Anno

	Length	Biotypes	Genome
GGCTCGTTGGTCTAGAGGTATGATT	25	tRNA	X:18565764:+[3L:18618390:+[3L:17339956:-
CGATATTTCTCCTCGGACC	20	rRNA	X:23454910:- X:23535571:-
CATCACAGTCTGAGTTCTTGCT	22	miRNA	3R:21622503:-
TGACCTGATGAAGTGTATT	19	snoRNA	3L:9510787:-
CGAACTGTAGAGTTTG	16	NA	no_anno
CGTTCAAATTTAAGTCTGT	20	rRNA	MT:12879:-

anno_target=list("Biotypes") → All Biotypes as they appear in column
 anno_target=list("Biotypes", "miRNA") → Extracts only miRNA
 anno_target=list("Length", 20:22) → Length in the range 20-22 nucleotides

1084

1085 **Figure 5.** The principles for working with target objects. Many seqpac functions applies a novel system
 1086 for grouping and sub-dividing samples (*Pheno*) and sequences (*Anno*) in a PAC object. This system
 1087 relies on small target objects, which targets information either in the *Pheno* (*pheno_target*) or *Anno*
 1088 (*anno_target*) tables. A target object is a list with two-character inputs. The first pointing to a column in
 1089 the target table, and the second to the entries of that column.

1090

1091

1092 **Figure 6.** Advanced PAC objects. All PAC objects must contain *Pheno*, *Anno* and *Counts* tables. Many
 1093 seqpac functions may optionally use data stored in two additional PAC objects: the *norm* and *summary*
 1094 lists. The *norm* contains tables of normalized counts having identical row and column names as *Counts*.
 1095 The *summary* contains tables with identical row names as *Counts*, but column names based on
 1096 aggregates over the *Counts* columns (e.g. group means). These PAC 'folders' can be generated by the
 1097 *PAC_norm* or *PAC_summary* functions, but users are encouraged to provide their own tables.
 1098

Figure 7. Seqpac example 1: Identifying contaminants by sequence-based counting. We generated a PAC object from a public dataset (SRA access: PRJNA666144). This study used a feature-based counting strategy to examine the sRNA in cells, and their extracellular vesicles, of 14 cervical and head/neck cancer cell lines. (A) Scatter plot generated by the *PAC_pca* function after vst normalization using *PAC_norm*. The two first principle components identify extracellular vesicles from SCC4 and SCC154 cells as outliers. (B) Size distribution histograms generated by the *PAC_sizedist* function. Most samples show high content of sRNA ≥ 50 nt, except for the SCC4 and SCC154 outliers, which are enriched with 29 nt fragments with no annotation in humans. (C) Bar graphs generated by the *PAC_stackbar* function after reannotating the PAC object against the human and *Mycoplasma hyorhinis* (contaminant) genomes show high content of Mycoplasma in outliers. (D) Scatter plot generated by *PAC_pca* after removing all non-human sequences with the *PAC_filter* function, only including sRNA between 16-45 nt, and re-normalizing the data with *PAC_norm*. No big differences from the original dataset (A).

Figure 8. Seqpac's sRNA coverage functions reveal novel rRNA derived fragment (rRF) in cancer. RNA polymerase I (Pol I) transcribes the 47/45S pre-rRNA and has been targeted in anti-cancer treatment. Small RNA analysis of HeLa cells either exposed to an RNA polymerase I inhibitor (BMH21) or DMSO (control) for 60 min (A-J)(SRA access: PRJNA708219). Volcano plot (A) from a differential expression analysis using the *PAC_deseq* function showing down-regulated rRFs. Red indicates related sequences that primarily originates from an rRNA cluster on chr 21. Coverage plot (B) using the *PAC_mapper* and

1122 *PAC_covplot* functions of an 45S pre-rRNA on chr 21 (GenBank: NR_146144.1). Shows 4 major peaks
 1123 affected differently by BMH21. Zoomed in coverage plots of Peak 1-4 (C-F). Peak 1 (C) shows a novel
 1124 5' ETS rRF downregulated by treatment. Peak 2 (D) shows a plausibly degraded fragment. Peak 3 (E)
 1125 contains two overlapping rRFs where only the small (Peak 3a) varies by treatment. Peak 4 (F) shows a
 1126 rRF possibly upregulated by treatment. Bar graphs (G-I) showing that 12h exposure to BMH21 amplifies
 1127 the effects in Peak 1 and Peak 4, but not in Peak 2 and Peak 3. Coverage plots (K) validating the 5' ETS
 1128 rRF of Peak 1 in cancer cells, and to a lesser degree in extra-cellular vesicles, using the Tong *et al.*
 1129 dataset from Figure 7 (without the contaminated SCC4 and SCC154 cell lines). The zoomed in genomic
 1130 sequence (bottom) shows the main fragment from (C) where the longest fragment from Tong *et al.* is
 1131 presented as bold letters. Dotted grey box indicate GT-insertion compared to a commonly studied 47S
 1132 pre-rRNA (GenBank: U13369.1). Bar graphs (L) showing an up-regulation of 5' ETS rRF related
 1133 fragments in cervical tumor samples published by Xu *et al.* (SRA access: PRJNA607023). Number of
 1134 summed sequences that were found in the target regions are indicated by n_{seqs} . Mann-Whitney U tests
 1135 indicated by * $p < 0.05$ and # $p < 0.1$ significance levels. SEM = Standard error of the mean.

1136

1137

Figure 1.

Figure 2.

The PAC list object

Anno

```
GGCTCGTTGGTCTAGAGGTATGATT
CGATAATTTCTCTCGGACC
CATCACAGTCTGAGTTCTTGCT
TGACCTGATGAAGTGTATT
TTAGCTCGTTCTGCAACTCGTTGC
CGTTCAAATTTAAGTCTGT
```

Length
25
20
22
19
16
20

Biotypes	Genome
tRNA	X:18565764:+ 3L:18618390:+
rRNA	X:23454910:- X:23535571:-
miRNA	3R:21622503:-
snoRNA	3L:9510787:-
NA	3L:26579843:+ 3R:3166087:+ 3L:27028388:+ 2R:6423733:-
rRNA	MT:12879:-

simplify_reanno
(hierarchy)

[3]

Reannotation
list object

add_reanno

[2]

.Rdata output
0 mismatches
1 mismatch
2 mismatches
etc...

import_reanno

(PAC_mapper)

map_reanno

BOW TIE
An ultrafast memory-efficient
short read aligner

Fasta references

Ref genome
Mature tRNA
piRNA
etc...

PAC_gtf

[4]

Gtf features
repeatMasker
Ensembl gtf
etc...

repeatClass	Ensembl	Gene
-	Bx tRNA:Pro-CGG-2-1/2/3	
rRNA	-	E2f1 mir-11
-	asRNA:CR45874 RpS9 snoRNA:Me28S-G3253	
LTR	Pzl	
-	mt:lrRNA	

Figure 3.

Figure 4.

Pheno

	<i>Bio_sample</i>	<i>Type</i>	<i>Replicate</i>
S1_Ov1	S1	Ovaries	1
S1_Ov2	S1	Ovaries	2
S1_Ov3	S1	Ovaries	3
S5_Sp1	S5	Sperm	1
S5_Sp2	S5	Sperm	2
S5_Sp3	S5	Sperm	3

`pheno_target=list("Type")` → All Types as they appear in column
`pheno_target=list("Replicate", c(1, 3))` → Extract replicate 1 and 3 in that order
`pheno_target=list("Type", c("Sperm", "Ovaries"))` → Change order of Ovaries and Sperm

Anno

	<i>Length</i>	<i>Biotypes</i>	<i>Genome</i>
GGCTCGTTGGTCTAGAGGTATGATT	25	tRNA	X:18565764:+ 3L:18618390:+ 3L:17339956:-
CGATATTTTCTCCTCGGACC	20	rRNA	X:23454910:- X:23535571:-
CATCACAGTCTGAGTTCTTGCT	22	miRNA	3R:21622503:-
TGACCTGATGAACTGTATT	19	snoRNA	3L:9510787:-
CGAACTGTAGAGTTTG	16	NA	no_anno
CGTTCAAATTTTAAGTCTGT	20	rRNA	MT:12879:-

`anno_target=list("Biotypes")` → All Biotypes as they appear in column
`anno_target=list("Biotypes", c("rRNA", "tRNA"))` → Extracts rRNA and tRNA in that order
`anno_target=list("Length", 20:22)` → Length in the range 20-22 nucleotides

Figure 5.

Figure 6.

Figure 7.

Figure 8.