Chapter 12

Computational Analysis of RNA–Protein Interactions via Deep Sequencing

Lei Li, Konrad U. Förstner, and Yanjie Chao

Abstract

RNA-binding proteins (RBPs) function in all aspects of RNA processes including stability, structure, export, localization and translation, and control gene expression at the posttranscriptional level. To investigate the roles of RBPs and their direct RNA ligands in vivo, recent global approaches combining RNA immunoprecipitation and deep sequencing (RIP-seq) as well as UV-cross-linking (CLIP-seq) have become instrumental in dissecting RNA–protein interactions. However, the computational analysis of these high-throughput sequencing data is still challenging. Here, we provide a computational pipeline to analyze CLIP-seq and RIP-seq datasets. This generic analytic procedure may help accelerate the identification of direct RNA–protein interactions from high-throughput RBP profiling experiments in a variety of bacterial species.

Key words RNA-seq, RIP-seq, CLIP-seq, Bioinformatics, Hfq, CsrA, ProQ, ncRNA, sRNA

1 Introduction

RNA-binding proteins (RBPs) are an important class of post-transcriptional regulators of gene expression. RBPs either directly bind to messenger RNAs (mRNAs) or act through numerous regulatory noncoding RNAs (ncRNAs), dictating the fate of the bound transcripts. In all three kingdoms of life, increasing numbers of RBPs have been identified, including many well-studied model organisms such as pathogenic bacteria [1], baker's yeast [2], and human [3]. Taking bacteria for example, a new global RBP called ProQ was recently found as a major RNA chaperone in two distantly related bacterial pathogens *Salmonella enterica* serovar Typhimurium [1] and Legionella pneumophila [4], constituting the third global RBP in bacteria besides the well-known Hfq and CsrA proteins [5, 6].

Functional understanding of RBPs requires the full account of their RNA binding partners and the exact binding sites. To identify RNAs that are bound by an RBP of interest, a classic approach is to

Yejun Wang and Ming-an Sun (eds.), *Transcriptome Data Analysis: Methods and Protocols*, Methods in Molecular Biology, vol. 1751, https://doi.org/10.1007/978-1-4939-7710-9_12,

Table 1
Recent RNAseq-based studies of RNA–protein interactions in bacteria

Technique	Organism	RNA-binding protein	Year	PMID
RIP-seq	*Salmonella enterica* serovar Typhimurium	Hfq	2008	[7]
RIP-seq	*Salmonella enterica* serovar Typhimurium	Hfq	2012	[8]
RIP-seq	*Bacillus subtilis*	Hfq	2013	[10]
RIP-seq	*Sinorhizobium meliloti*	Hfq	2014	[11]
CLIP-seq	*Escherichia coli*	Hfq	2014	[12]
RIP-seq	*Escherichia coli*	Hfq	2014	[13]
RIP-seq	*Brucella suis*	Hfq	2015	[14]
RIP-seq	*Campylobacter jejuni*	CsrA	2016	[15]
CLIP-seq	*Salmonella enterica* serovar Typhimurium	Hfq, CsrA	2016	[16]
RIP-seq	*Salmonella enterica* serovar Typhimurium	ProQ	2016	[1]
RIP-seq	*Legionella pneumophila*	CsrA	2017	[17]
CLIP-seq	*Salmonella enterica* serovar Typhimurium	RNase E	2017	[18]

immunoprecipitate the RBP using a specific antibody followed by analysis of the copurified transcripts using RNA gels or DNA arrays (RIP-chip). Thanks to the advance of high-throughput sequencing technologies, unbiased deep sequencing of the co-immunoprecipitated RNAs (RIP-seq) can now identify hundreds or even thousands of transcripts in a bacterium [7, 8]. -RIP-seq is relatively simple and experimentally straightforward, which have sparked its wide-application in the study of RNA–protein interactions in various biological systems [9] (Table 1). While RIP-seq usually identifies the full-length transcripts bound to an RBP, RIP-seq combined with UV cross-linking (CLIP-seq) can further identify the exact protein binding sites in a transcript. This approach was also referred to as HITS-CLIP, for high-throughput sequencing of RNA isolated by cross-linking immunoprecipitation [19]. The key of CLIP-seq is the in vivo cross-linking under ultraviolet (UV) light that introduces a covalent bond between RBP and the bound RNA. This covalent linkage enables the cross-linked RNA–protein complexes to survive stringent purification steps (often under denaturing conditions) and partial nuclease digestion to remove the unbound sequences. Deep sequencing of UV-cross-linked RNA fragments (CLIP-seq) informatively provides the locations of the protein-binding sites in a large number of transcripts [20]. The unique UV-cross-linking step makes CLIP-seq a powerful method to identify direct RNA–protein interactions. CLIP-seq has superior sensitivity in capturing weak or transient interactions

in vivo [21]. In addition, the cross-linked peptide on RNA often results in mutations in cDNAs during reverse transcription. These mutations help pinpoint the exact protein-binding sites at the single nucleotide resolution [22].

This chapter mainly focuses on the CLIP-seq data analysis in bacteria, owing to its higher data complexity and its recent successful applications in *Escherichia coli* [12] and *S. Typhimurium* [16] (Table 1). In these studies, CLIP-seq has demonstrated its power in identifying the direct RNA ligands and exact sequences bound by Hfq and CsrA, respectively. While CLIP-seq is becoming instrumental in studying bacterial RNA–protein interactions, the analysis of CLIP-seq data is highly demanding. A suite of bioinformatics tools and analytic procedures are required to fully reveal the information capsulated in the sequencing data, and to identify the true RNA–protein interactions. To help other bioinformaticians and RNA enthusiasts perform such sequencing data analysis, here we have outlined a computational pipeline (Fig. 1) that has been recently devised to analyze CLIP-seq data for Hfq and CsrA [23]. Because these analytical procedures are generic, the presented pipeline can be readily used for the analysis of CLIP-seq with any given RBP, as well as the analysis of RIP-seq data.

2 Materials

We use our recently published CLIP-seq dataset [24] as an example, which is hosted in NCBI GEO database (GSE74425). The *S.* Typhimurium SL1344 reference genome and annotation information can be downloaded from NCBI FTP site (ftp://ftp.ncbi.nlm.nih.gov/genomes/archive/old_refseq/Bacteria/Salmonella_enterica_serovar_Typhimurium_SL1344_uid86645/).

3 Methods

3.1 Quality Trimming

Upon completing the Illumina sequencing, the received raw sequencing reads require initial processing. A sequencing read must contain parts of the adapter sequences, which need be identified and trimmed before aligning to the reference genomes. Among many suitable tools, **Cutadapt** is a user-friendly command line interface. It can search and trim adapter sequences in an error-tolerant manner, and it is compatible with a large variety of input file formats generated by high-throughput sequencers [23] (*see* **Note 1**). The latest version can be downloaded from http://cutadapt.readthedocs.io/en/stable/index.html.

To perform adaptor trimming for paired-end reads, a typical command line employing **Cutadapt** looks like this:

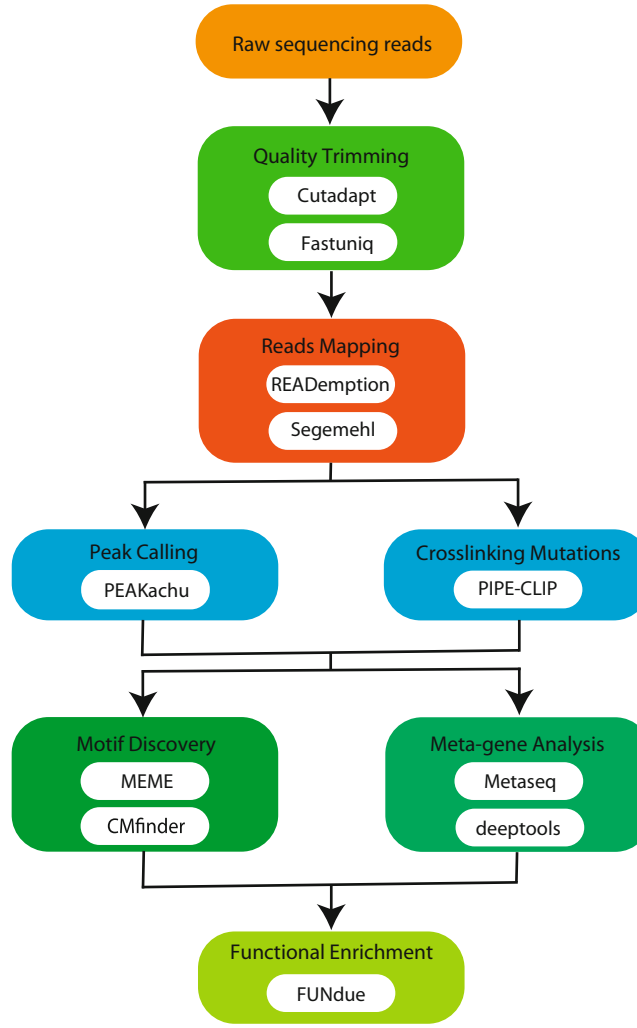

Fig. 1 Workflow for bacterial RBP profiling data analysis. Raw sequencing reads from CLIP-seq or RIP-seq are subjected to the analysis pipeline. Quality and user-defined sequence trimming removes adapter sequences, low-quality reads, and PCR duplicates using **Cutadapt** and **Fastuniq** tools. Reads are then mapped to the reference genome using **READemption** and **segemehl**. RBP-binding sites in RNA are identified using peak-calling algorithm **PEAKachu**, as well as the mutation analysis package **PIPE-seq**. The putative motifs sequences and structural properties are identified using **MEME** and **CMfinder**. Further, meta-gene analysis is performed using **Metaseq** and **deeptools** to search the global distribution of binding profiles. **FUNdue** finally reports a functional annotation including gene ontology and pathway analysis

```
cutadapt -q 20 -a "AGATCGGAAGAGCACACGTCTGAACTCCAGTCAC" -A
"GATCGTCGGACTGTAGAACTCTGAACGTGTAGATCTCGGTGGTCGCCGTATCATT"
--pair-filter=both -o [file1].out.fq -p [file2].out.fq
[file1].fq [file2].fq
```

The low-quality sequences from the end of short reads were firstly trimmed with a cutoff of 20 for the Phred quality score ($Q < 20$), then the two adapter sequences shown above were removed. This option (--pair-filter=both) removes the entire (pair-end sequenced) read pairs if at least one of the two sequences became shorter than a certain length threshold.

CLIP-seq experiments often generate numerous PCR duplicates after cDNA amplification. These duplicate reads need to be identified and removed using **Fastuniq** [24], a tool for de novo removal of duplicates in paired short reads and freely available at https://sourceforge.net/projects/fastuniq/.

3.2 Reads Mapping

The filtered and trimmed reads are then aligned against the reference genome using **READemption** [25]. **READemption** is a pipeline for the computational analysis of RNA-Seq data. It was developed initially for bacterial transcriptomic data, but now also extended to analyze eukaryotic transcriptomes as well as a mixture of both, i.e., dual RNA-Seq data [26]. The latest version can be downloaded from https://pythonhosted.org/READemption/. It requires **segemehl** [27] as the short read aligner, which can be download separately from http://www.bioinf.uni-leipzig.de/Software/segemehl/. **Segemehl** effectively handles both mismatches and short insertions and deletions. It is an ideal aligner for CLIP-seq reads, which often contain the characteristic mutations introduced by cross-linking procedures.

READemption covers most of the important mapping procedures and is organized in a command-line interface with several subcommands. These subcommands include read processing and aligning, coverage calculation, gene expression quantification, differential gene expression analysis as well as generating coverage files for visualization.

The *"create"* subcommand in **READemption** can generate the necessary folder structure. As required, transcriptome reads in FASTA format need be stored in the folder *input/reads*, and the genomes used as the reference should be in the folder *input/reference_sequences*. Also, the bacterial annotation files have to be placed into input/annotations.

After the initial folder setup, the subcommand for running the read alignment is

```
reademption align --realign, --processes 20 --segemehl_accuracy
95 --min_read_length 12 --progress [project_path]
```

Where [project_path] should be substituted by the path that was used with the *create* subsommand. Of note, reads shorter than 12 nucleotides will be removed, as well as the reads that are mapped to multiple locations. The remaining reads will then be aligned

against the reference genome with a mapping accuracy of 95% using **segemehl**. The reads mapping statistics, including the summary of uniquely aligned reads and mapped reads, will be documented in the file read_alignment_stats.csv. The read alignment and index files will be generated in BAM and BAI format, respectively.

Reads coverage information representing the numbers of mapped reads per nucleotide can be generated using the "coverage" subcommand. The command line is

```
reademption coverage --unique_only [project_path]
```

The uniquely aligned reads will be used to generate the coverage file and saved in wiggle format. **READemption** also provides other useful options such as --coverage_style first_base_only, which converts only the first base into coverage files. This option is particularly useful to identify the transcript ends, which has served the analysis of global RNase E processing sites in our recent TIER-seq data [18]. The coverage plot can be visualized in a genome browser, e.g., the Integrated Genome Browser [28].

3.3 Peak Calling

RBP-binding sites in a transcript often accumulate many sequencing reads, which form sharp peaks spanning a narrow region. Therefore, peak calling serves to identify the precise RBP-binding sites, one of the most critical steps in the CLIP-seq data analysis. A few issues may influence the binding site detection. Firstly, most of the standard CLIP-seq protocols do not include a negative background control, which makes it hard to estimate the background noise and eliminate false peaks. This is because reads falling into a given transcript can be explained by two factors: transcript abundance and RBP preference, thus a negative control is highly recommended. Secondly, reads may align to incorrect transcripts due to sequencing errors and their subsequent mapping. A robust peak-calling algorithm is crucial to distinguish the specific RBP binding from nonspecific bindings and/or background noise. Although a few computational approaches have been developed, few are optimal because of problematic null hypotheses, e.g., **Piranha** [29], which considers sites with small number of reads as noise without including a negative control. A new peak-calling algorithm [16] has been developed to address these issues. This approach first divides the consecutively mapped reads into a few genomics blocks, and the blocks, which fulfill overlapping requirements including the read coverage of each block and the distance of the blocks, are iteratively assembled into the candidate peak regions using **blockbuster** [30]. Importantly, each candidate peak is tested for significant enrichment in the cross-linked samples versus the non-cross-linked control samples using **DESeq2** [31]. This algorithm will be integrated in a peak-calling tool **PEAKachu**, which is still under development, https://github.com/tbischler/PEAKachu (T. Bischler, personal communication).

3.4 Cross-Linking-Induced Mutations

Another important step is the identification of cross-linking induced mutations, which can be used to pinpoint the direct RNA–protein interaction sites at the single-nucleotide level. However, most of the available computational tools either ignore or inadequately address this issue, because the mutations may be confounded by single nucleotide polymorphisms (SNPs) and sequencing errors. One exception is **PIPE-CLIP** [32]. This tool can statistically identify the outstanding cross-linked mutations across a background distribution. Briefly, each mutation site is described by two parameters (k_i, m_i), where k_i is the number of mapped reads covering the considered location, and m_i is the number of specific mutations at location i. Then the mutation rate is modeled in each position by the binomial distribution with size k_i and background rate, which is calculated by read coverage with a summarization of matched length of all reads divided by genome size (*see* **Note 2**). The mutations will be considered significant only if the calculated adjusted p-value is lower than a specified threshold (e.g., adjusted $p < 0.05$). The source code of **PIPE-CLIP** is freely available from https://github.com/QBRC/PIPE-CLIP.

The command line for identifying cross-linking mutations is:

```
python pipeclip.py -i [inputfile] -o [output_prefix] -c 0 -l
12 -M 0.05 -C 0.05 -s [species]
```

The -c option is to specify the CLIP-seq type, -l option is to specify minimum match length, -M option is false discovery rate for significant cross-linking mutation, -C option defines the false discovery rate for the peak clusters.

For the paired-end reads, **PIPE-CLIP** cannot be directly used for mutation calling. However, there are a few solutions. First, the Python script 'FindMutation.py' can be used to identify substitutions, deletions and insertions separately from the mapping BAM files while allowing the user to choose the specific CLIP-seq type (HITS-CLIP, PAR-CLIP). Second, to lower the bias caused by background noise, the first read of the paired-reads can be extracted using **samtools** [33] and the characteristic mutation sites need to be present in both paired reads. Thirdly, the script 'MutationFilter.py' can determine the significantly enriched mutations in each library by using the extracted first paired mapping reads in BAM format and consensus mutation sites in BED format as input.

3.5 Motif Discovery

To investigate whether any sequence preference is present near the protein binding regions, **MEME** [34], a de novo sequence motif detection tool, can be used to discover consensus sequences among peak sequences or the surrounding regions of enriched cross-linking mutations. **MEME** can be accessed via a Web interface (http://meme-suite.org/tools/meme).

In addition to sequence-specific binding, some RBPs recognize RNA partners by structural properties rather than the sequence per se. **CMfinder** [35] is a tool that performed well to search for the presence of structural motifs based on unaligned sequences with long extraneous flanking regions. It relies on an expectation maximization algorithm using covariance models for motif description, and a Bayesian framework for structure prediction combining folding energy and sequence covariation. **CMfinder** can be accessed using webserver (http://wingless.cs.washington.edu/htbin-post/unrestricted/CMfinderWeb/CMfinderInput.pl). It is also available as a stand-alone perl script, which can be downloaded from http://bio.cs.washington.edu/CMfinderWeb/CMfinderInput.pl.

The command to run **CMfinder** is

```
perl cmfinder.pl [infile]
```

The output motif files are named by using the input file name as prefix (e.g., with the input file name input_file, the file input_file.motif.* will be generated). These motif files are stored in Stockholm format, where the suffix indicates the number of stem-loops in a motif. The motif file needs be reformatted to the unblocked Stockholm format. This is done with the **HMMER** package (http://hmmer.org/).

```
sreformat --pfam stockholm [alignfile] > [infile]
```

The formatted Stockholm file can be visualized using **R2R** [36], a software that generates representations of structure-informed RNA secondary alignments. The latest version is available at http://breaker.research.yale.edu/R2R.

3.6 Meta Gene Analysis

Meta gene analysis aims to analyze the global peak distribution with respect to a specific location across all annotated genes. The peak density can be calculated by counting the number of peaks along the specified annotation features like start codons, stop codons, sRNAs, and Rho-independent terminators. For example, a meta gene analysis of Hfq peaks found that most peaks are located at 3′ of seed sequences in sRNAs, whereas in mRNAs they are found at the 5′ of sRNA base-pairing regions [37].

A few computational tools are available for meta gene analysis. **Metaseq** [38] enables integrating multiple genomic data formats and allows for customized visualization. It is freely available at https://github.com/daler/metaseq. Another tool is **deepTools2** [39], which can jointly analyze multiple signals (bigWig) and region files (BED), and visualize data in a composite image. It is freely available at https://github.com/fidelram/deepTools and can also be used with a galaxy-based platform (http://deeptools.ie-freiburg.mpg.de/).

3.7 Functional Annotation and Enrichment Analysis

After the identification of RBP-binding sites, it is of interest to understand whether there is any enrichment of functions or pathways among the RBP-bound genes. To carry out this analysis in bacteria, we have developed a computational tool named **FUNdue** (L.L., unpublished). This tool is still under development (*see* **Note 3**) and is available at https://github.com/LeiLiSysBio/FUNdue.

FUNdue covers multiple submodules for functional ontologies and pathways analysis including gene ontology and pathway retrieval, functional assignment, statistics enrichment and visualization. Briefly, the gene ontology and pathway information is automatically retrieved from UniProt and KEGG databases. The ontology of each gene is classified into three categories, the molecular function, biological process and cellular component. Enrichment analysis is performed to evaluate the significant terms compared to the background using Fisher exact test and gene set enrichment analysis [40]. The calculated *p*-values are subjected to multiple-testing analysis using the Benjamini–Hochberg method. The significant gene ontology terms will be visualized as bar plots. Furthermore, the output files can be visualized by other tools such as **REVIGO** [37], which offers an easy and interactive illustration via web interface.

The following part demonstrates the steps for a pathway enrichment analysis using **FUNdue**. To initial a project and generate the required folder structure, we use the *"create"* submodule. The call to create the folder is:

```
traplfun create [project_path]
```

Where the [project_path] is the analysis folder specified by the user. This will result in a folder structure with all the required subfolders. **FUNdue** can automatically access and retrieve the pathways stored in the KEGG database [41], if the organism code is given. The three-letter organism code for a species of choice can be found on the KEGG website http://www.genome.jp/kegg/catalog/org_list.html. For example, if you want to download all the KEGG pathway information for *S.* Typhimurium SL1344 (organism code sey), the command is:

```
traplfun retrieve_pa -c sey [project_path]
```

After a list of interesting genes is created and stored in the *input/target_ids*, we can use the subcommand *'pathway_stat'* to perform enrichment analysis with default fisher exact test. The command is:

```
traplfun pathway_stat [project_path]
```

The significantly overrepresented pathways, per default with a *p*-value lower than 0.05, are stored in the pathway folder *output/pathway/pathwy_enrichment* in plain text format.

These pathways can then be visualized using the subcommand *'path_viz'*. The command is:

```
traplfun path_viz -c [KEGG_organism_code] [project_path]
```

It creates histograms and a bar plot for the enriched pathway summary. Besides the fisher exact test, the user can choose another gene set enrichment analysis [42], which maps and renders the changes in the relevant pathway graphs.

4 Notes

1. **READemption** can perform basic quality trimming and adapter clipping; however **cutadapt** has many advanced functions such as processing of paired-end sequencing reads, which is more suitable for CLIP-seq because the size of RBP interaction regions are comparable to whole cDNA fragments, and thus more accurately defines the binding regions.
2. Installation of **FUNdue** requires a few python and R dependent packages. This included Scipy, and also a few R packages including KEGGREST, getopt, piano, optparse, gsge, and pathview.
3. **PIPE-CLIP** can identify all simple types of mutations including substitutions, deletions and insertions. To avoid sequencing or alignment errors, each different type of mutation needs to be analyzed separately. UV-cross-linking mutations such as T to C mutations should be enriched at specific sites and show high frequency compared to other mutations. In addition, integrating the enriched mutations with peaks information could further pinpoint the cross-linking induced mutations.

Acknowledgment

We thank Erik Holmqvist and Andrew Camilli for critical reading and comments on the manuscript.

References

1. Smirnov A, Förstner KU, Holmqvist E et al (2016) Grad-seq guides the discovery of ProQ as a major small RNA-binding protein. Proc Natl Acad Sci U S A 113:11591–11596
2. Tsvetanova NG, Klass DM, Salzman J, Brown PO (2010) Proteome-wide search reveals unexpected RNA-binding proteins in Saccharomyces cerevisiae. PLoS One 5:pii:e12671
3. Castello A, Fischer B, Eichelbaum K et al (2012) Insights into RNA biology from an atlas of mammalian mRNA-binding proteins. Cell 149:1393–1406
4. Attaiech L, Boughammoura A, Brochier-Armanet C et al (2016) Silencing of natural transformation by an RNA chaperone and a multitarget small RNA. Proc Natl Acad Sci U

S A 113:8813–8818. https://doi.org/10.1073/pnas.1601626113

5. Vogel J, Luisi BF (2011) Hfq and its constellation of RNA. Nat Rev Microbiol 9:578–589. https://doi.org/10.1038/nrmicro2615
6. Romeo T (1998) Global regulation by the small RNA-binding protein CsrA and the non-coding RNA molecule CsrB. Mol Microbiol 29:1321–1330
7. Sittka A, Lucchini S, Papenfort K et al (2008) Deep sequencing analysis of small noncoding RNA and mRNA targets of the global post-transcriptional regulator, Hfq. PLoS Genet 4: e1000163
8. Chao Y, Papenfort K, Reinhardt R et al (2012) An atlas of Hfq-bound transcripts reveals 3′ UTRs as a genomic reservoir of regulatory small RNAs. EMBO J 31:4005–4019. https://doi.org/10.1038/emboj.2012.229
9. Riley KJ, Steitz JA (2013) The "observer effect" in genome-wide surveys of protein-RNA interactions. Mol Cell 49:601–604. https://doi.org/10.1016/j.molcel.2013.01.030
10. Dambach M, Irnov I, Winkler WC (2013) Association of RNAs with *Bacillus subtilis* Hfq. PLoS One 8:e55156. https://doi.org/10.1371/journal.pone.0055156
11. Torres-Quesada O, Reinkensmeier J, Schlüter JP et al (2014) Genome-wide profiling of Hfq-binding RNAs uncovers extensive post-transcriptional rewiring of major stress response and symbiotic regulons in *Sinorhizobium meliloti*. RNA Biol 11(5):563–579. https://doi.org/10.4161/rna.28239
12. Tree JJ, Granneman S, McAteer SP et al (2014) Identification of bacteriophage-encoded anti-sRNAs in pathogenic *Escherichia coli*. Mol Cell 55:199–213
13. Bilusic I, Popitsch N, Rescheneder P et al (2014) Revisiting the coding potential of the *E. coli* genome through Hfq co-immunoprecipitation. RNA Biol 11(5):641–654. https://doi.org/10.4161/rna.29299
14. Saadeh B, Caswell CC, Chao Y et al (2016) Transcriptome-wide identification of Hfq-associated RNAs in *Brucella suis* by deep sequencing. J Bacteriol 198:427–435. https://doi.org/10.1128/JB.00711-15
15. Dugar G, Svensson SL, Bischler T et al (2016) The CsrA-FliW network controls polar localization of the dual-function flagellin mRNA in *Campylobacter jejuni*. Nat Commun 7:11667
16. Holmqvist E, Wright PR, Li L et al (2016) Global RNA recognition patterns of post-transcriptional regulators Hfq and CsrA revealed by UV crosslinking in vivo. EMBO J 35:991–1011. https://doi.org/10.15252/embj.201593360
17. Sahr T, Rusniok C, Impens F et al (2017) The *Legionella pneumophila* genome evolved to accommodate multiple regulatory mechanisms controlled by the CsrA-system. PLoS Genet 13:e1006629. https://doi.org/10.1371/journal.pgen.1006629
18. Chao Y, Li L, Girodat D et al (2017) In vivo cleavage map illuminates the central role of RNase E in coding and non-coding RNA pathways. Mol Cell 65:39–51
19. Licatalosi DD, Mele A, Fak JJ et al (2008) HITS-CLIP yields genome-wide insights into brain alternative RNA processing. Nature 456:464–469. https://doi.org/10.1038/nature07488
20. König J, Zarnack K, Luscombe NM, Ule J (2011) Protein-RNA interactions: new genomic technologies and perspectives. Nat Rev Genet 13:77–83. https://doi.org/10.1038/nrg3141
21. Darnell RB (2010) HITS-CLIP: panoramic views of protein-RNA regulation in living cells. Wiley Interdiscip Rev RNA 1:266–286. https://doi.org/10.1002/wrna.31
22. Zhang C, Darnell RB (2011) Mapping in vivo protein-RNA interactions at single-nucleotide resolution from HITS-CLIP data. Nat Biotechnol 29:607–614. https://doi.org/10.1038/nbt.1873
23. Martin M (2011) Cutadapt removes adapter sequences from high-throughput sequencing reads. EMBnet J 17:10. https://doi.org/10.14806/ej.17.1.200
24. Xu H, Luo X, Qian J et al (2012) FastUniq: a fast de novo duplicates removal tool for paired short reads. PLoS One 7:e52249
25. Förstner KU, Vogel J, Sharma CM (2014) READemption-a tool for the computational analysis of deep-sequencing-based transcriptome data. Bioinformatics 30:3421–3423
26. Westermann AJ, Förstner KU, Amman F et al (2016) Dual RNA-seq unveils noncoding RNA functions in host-pathogen interactions. Nature 529:496–501. https://doi.org/10.1038/nature16547
27. Hoffmann S, Otto C, Kurtz S et al (2009) Fast mapping of short sequences with mismatches, insertions and deletions using index structures. PLoS Comput Biol 5:e1000502
28. Freese NH, Norris DC, Loraine AE (2016) Integrated genome browser: visual analytics platform for genomics. Bioinformatics 32:2089–2095
29. Uren PJ, Bahrami-Samani E, Burns SC et al (2012) Site identification in high-throughput

RNA-protein interaction data. Bioinformatics 28:3013–3020. https://doi.org/10.1093/bioinformatics/bts569

30. Langenberger D, Bermudez-Santana C, Hertel J et al (2009) Evidence for human microRNA-offset RNAs in small RNA sequencing data. Bioinformatics 25:2298–2301
31. Love MI, Huber W, Anders S (2014) Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. Genome Biol 15:550. https://doi.org/10.1186/PREACCEPT-8897612761307401
32. Chen B, Yun J, Kim MS et al (2014) PIPE-CLIP: a comprehensive online tool for CLIP-seq data analysis. Genome Biol 15:R18
33. Li H, Handsaker B, Wysoker A et al (2009) The sequence alignment/map format and SAMtools. Bioinformatics 25:2078–2079. https://doi.org/10.1093/bioinformatics/btp352
34. Bailey TL, Boden M, Buske FA et al (2009) MEME SUITE: tools for motif discovery and searching. Nucleic Acids Res 37:W202–W208
35. Yao Z, Weinberg Z, Ruzzo WL (2006) CMfinder—a covariance model based RNA motif finding algorithm. Bioinformatics 22:445–452
36. Weinberg Z, Breaker RR (2011) R2R—software to speed the depiction of aesthetic consensus RNA secondary structures. BMC Bioinformatics 12:3
37. Supek F, Bošnjak M, Škunca N, Šmuc T (2011) REVIGO summarizes and visualizes long lists of gene ontology terms. PLoS One 6:e21800
38. Dale RK, Matzat LH, Lei EP (2014) metaseq: a Python package for integrative genome-wide analysis reveals relationships between chromatin insulators and associated nuclear mRNA. Nucleic Acids Res 42:9158–9170
39. Ramírez F, Ryan DP, Grüning B et al (2016) deepTools2: a next generation web server for deep-sequencing data analysis. Nucleic Acids Res 44:W160–W165
40. Subramanian A, Tamayo P, Mootha VK et al (2005) Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. Proc Natl Acad Sci 102:15545–15550
41. Kanehisa M, Sato Y, Kawashima M et al (2016) KEGG as a reference resource for gene and protein annotation. Nucleic Acids Res 44: D457–D462
42. Luo W, Brouwer C (2013) Pathview: an R/bioconductor package for pathway-based data integration and visualization. Bioinformatics 29:1830–1831