Gene expression changes in response to diverse environmental stimuli to regulate numerous cellular functions. Genes are expressed into their functional products with the help of messenger RNA (mRNA). Thus, measuring levels of mRNA in cells is important to understand cellular functions. With advances in next-generation sequencing (NGS), the abundance of cellular mRNA has been elucidated via transcriptome sequencing. However, several studies have found a discrepancy between mRNA abundance and protein levels induced by translational regulation, including different rates of ribosome entry and translational pausing. As such, the levels of mRNA are not necessarily a direct representation of the protein levels found in a cell. To determine a more precise way to measure protein expression in cells, the analysis of the levels of mRNA associated with ribosomes is being adopted. With an aid of NGS techniques, a single nucleotide resolution footprint of the ribosome was determined using a method known as RiboSeq or ribosome profiling. This method allows for the high-throughput measurement of translation in vivo, which was further analyzed to determine the protein synthesis rate, translational pausing, and cellular responses toward a variety of environmental changes. Here, we describe a simple analysis pipeline for Ribo-Seq in bacteria, denoted as simple translatome analysis tool (STATR). STATR performs the primary processing of Ribo-Seq data, subsequently allowing for multiple levels of translatome study, from experimental validation to in-depth analyses. A command-by-command explanation is provided here to allow a broad spectrum of biologists to easily reproduce the analysis.

Keywords: ribosome profiling, Ribo-Seq, NGS analysis
such that all the required files (example files, adapter sequences, and codes) are available at a public repository (https://github.com/robinald/STATR) under the GNU General Public License v.3.0 (https://www.gnu.org/licenses/gpl-3.0.en.html). In addition, the command-by-command explanation provided here can be used by a variety of biologists to easily reproduce our analysis.

Applications

Ribo-Seq provides a comprehensive understanding of translation in bacteria. It has been widely used to measure translation levels, as well as to investigate translational changes in response to diverse environmental stresses (Gerashchenko et al., 2012; Shalgi et al., 2013). Since Ribo-Seq accurately resolves the nucleotide-level profile of the ribosome, it provides a means to analyze ribosome pausing/stalling (Li et al., 2012) and codon usage (Paulet et al., 2017). In bacteria, it is believed that gene expression is regulated only transcriptionally since, unlike eukaryotes, bacteria lack complex post-transcriptional processes, including splicing, localization, and modification. In addition, translation and transcription are coupled in the cytosol. However, the presence and significance of translation have been highlighted recently, with regulatory processes being elucidated by Ribo-Seq (Van Assche et al., 2015; Jeong et al., 2016; Potts et al., 2017). Novel proteins and upstream open reading frames (uORFs) have also been recently elucidated (Pelechano et al., 2013).

Methods

Ribo-Seq and recommended sequencing parameters

With respect to ribosomes remaining for different durations in different positions, the positional density of ribosomes varies within a transcript. This cellular variation can be analyzed experimentally using nuclease footprinting. First, to capture and isolate the ribosome-protected mRNA fragments, protein synthesis inhibitor antibiotics are used to efficiently freeze the ribosomes. Several methods can be used to isolate ribosome-protected mRNAs, including polyacrylamide gel electrophoresis (PAGE), followed by gel elution and size exclusion chromatography. Lastly, the isolated ribosome-protected mRNA fragments are subjected to sequencing library construction. Typically, the size of ribosome-protected mRNA fragments ranges between 26 and 34 nt. Thus, sequencing with a longer read cycle is not recommended. In addition, pair-ended sequencing is also unnecessary. Ribo-Seq is performed routinely using a single-ended 50 bp recipe.
on the Illumina platform, as in previous reports (Pelechano et al., 2013; Jeong et al., 2016; Choe et al., 2019).

Read trimming

The STATR pipeline comprises multiple steps of data processing (Fig. 1). The analysis starts with the trimming of raw reads. The raw sequencing output file contains reads with quality scores for each base. The file contains reads with a variety of quality scores and may contain sequencing adapter sequences that are not removed during primary processing by the sequencer. The first step of the analysis involves removing the low-quality reads (or bases) and the adapter sequence that may not be removed from sequencer, so that only the high quality ribosome-protected sequence remains. Trimmomatic is a read processing tool designed for Illumina sequence data (Bolger et al., 2014). Both the removal of low-quality reads and the adapter sequence can be carried out using Trimmomatic, generating trimmed reads ready for analysis.

Mapping reads on reference

STATR uses Bowtie2 software to map trimmed reads to a reference genome sequence. Bowtie2 is a fast and sensitive read alignment tool. In Bowtie2, the long reference genome sequence is indexed into multiple partitions for memory-efficiency (Langmead and Salzberg, 2012). One output of Bowtie2 is the alignment information of all the reads in SAM file format. STATR includes the conversion step of SAM file into BAM file using Samtools (Li et al., 2009), since the following pipeline (BEDTools) accepts only sorted binary format.

Decompiling the sequence alignment

The mapping file is a compressed binary coded format of a SAM file containing all the information of quality and alignment for each of the reads to the reference. The file is a tab-delimited text that is readable, however, several codes (e.g. CIGAR string) are required to be further decompiled for downstream analysis. BEDTools is a robust tool for handling genomic file-formats including BAM and SAM files (Quinlan and Hall, 2010). In the STATR pipeline, two output files are generated by using BEDTools. One contains the number of reads aligned on each gene, which represents the translation level. Reads aligned only in the anti-sense strand of a gene are counted, since Ribo-Seq has strand-specificity. This translation level can be normalized and analyzed to determine the genome-wide translation level and differential translation in cells. The other output file contains the alignment information of individual reads throughout the reference genome. The result will be processed further to verify the integrity of Ribo-Seq and in-depth per base meta-analysis to gain biological insights.

Experiment validation via meta-profile analysis

Ribo-seq provides a nucleotide-resolution footprint of the ribosome. Since the ribosome elongates the peptide chain discretely as decoding three-nucleotide code of mRNA, its footprint shows repeats of alternating dense-open peaks (Ingolia et al., 2009; Mohammad et al., 2019). Although alternative decoding events occur, such as frameshifting, hopping, and read-throughs (Chen et al., 2014; Lang et al., 2014; Fan et al., 2017), the average ribosome density maintains a three-nucleotide periodicity. Thus, we exploited this characteristic to determine the quality of Ribo-Seq. However, Ribo-Seq in bacterial cells tends to show less clear periodicity than that of eukaryotes (Mohammad et al., 2019). In addition, the periodicity profile differs when the ribosome position is assigned to the 5' or 3' end of the reads, depending on the experimental conditions (Woolstenhulme et al., 2015). The STATR pipeline calculates the average ribosome density over the CDSs to monitor the end that represents the ribosome position more precisely.

Differential translation

In the previous processing step, the translation level (i.e. the number of mapped reads on every gene) was counted using BEDTools. However, this measure is a relative value, and needs to be normalized by sequencing depth in order to be compared sample-by-sample. DESeq2 is a model-based analysis tool that is widely used for transcriptome analysis (Love et al., 2014). R scripts are included in this pipeline to examine the reproducibility of the experiment by accessing the hierarchical clustering of the samples and principal component analysis (PCA). After normalization and statistical analysis, data can be further analyzed to elucidate the specific responses of the experimental group with differentially translated genes.

Materials

Equipment, software dependencies, and requirements

STATR runs on a Linux system. If a workstation or a computer cluster is not available, cloud computing is a cost-effective alternative. STATR requires a number of widely adapted tools, including Trimmomatic (v.0.39) for sequence quality control, Bowtie2 (v.2.2.6) for sequence alignment, Samtools (v.0.1.19) for convert/sort SAM file to binary coded BAM file, and BEDTools (v.2.25.0) for compiling individual reads from the alignment (BAM file). Downstream scripts were intended to run on R (v.3.6.1) and Python (v.3.5.2). In addition, the following R packages are also required: DESeq2 (v.1.24.0), RColorBrewer (v.1.1-2), ggplots (v.3.0.1), data.frame (v.1.12.6), optparse (v.1.6.4), and getopt (v.1.20.3).

Protocols

A. Conduct Ribo-Seq

The STATR pipeline analyzes Ribo-Seq data, wherein the example dataset used in this manuscript has been reported in previous studies (Latif et al., 2015; Choe et al., 2019). The dataset was generated following the standard experimental procedure. Briefly, 50 ml of culture was collected via centrifugation at 4,000 × g, 4°C for 10 min after 5 min of treatment with chloramphenicol (34 mg/ml) at an exponential growth phase. The pellet was resuspended with 0.5 ml of lysis buffer...
(1% Triton X-100, 34 μg/ml chloramphenicol, 133 mM of NaCl, 4.75 mM of MgCl₂, and 19 mM of pH 7.5 Tris-HCl). The resuspension was flash-frozen with liquid nitrogen and lysed by pestle and mortar. Then, the cell lysate containing 10 μg of RNA was treated with 2,000 gel units of Micrococcal Nuclease (MNase; NEB) at 37°C for 30 min. Polysomes were recovered from the MNase-digested sample by size-exclusion chromatography using Illustra MicroSpin S-400 HR Columns (GE Healthcare) followed by phenol : chloroform : isoamyl alcohol extraction. The ribosomal RNA was removed from 5 μg of polysome-protected RNA using a RiboZero rRNA Removal Kit (Illumina) according to the manufacturer’s instructions. rRNA-subtracted RNA samples were phosphorylated by treating 10 U of T4 Polynucleotide Kinase (NEB) at 37°C for an hour and purifying with RNasy MinElute columns (Qiagen). Sequencing libraries were prepared from the phosphorylated RNA samples using the NEBNext Small RNA Library Prep Set for Illumina (NEB) according to the manufacturer’s protocol. NGS was performed on an Illumina HiSeq 2500 instrument with a single-ended 50 cycles recipe.

B. Prepare software and directory before analysis

The STATR command line is intended to be run on a directory structure, as shown in Fig. 2. It can be modified freely by the user. However, the path should be modified accordingly.

1. Prepare Trimmomatic (v.0.39) by downloading and unzipping the binary file (available at http://www.usadellab.org/cms/index.php?page=trimmomatic). Trimmomatic runs on a Linux system with a working Java Development Kit (tested on v.1.8.0_222).

2. Install Bowtie2 (v.2.2.6) according to the instructions provided at http://bowtie-bio.sourceforge.net/bowtie2/index.shtml. One can install package to Linux as following.

   1. Download Bowtie2 source file (for example, bowtie2-2.2.6-source.zip) at http://bowtie-bio.sourceforge.net/bowtie2/index.shtml.
   2. Unzip the source file and use make command to build Bowtie2 as following.
   
   User@Linux> unzip bowtie2-2.2.6-source.zip
   User@Linux> cd bowtie2-2.2.6/
   User@Linux> make

   ▶ By default, Bowtie2 builds with Threading Building Blocks (TBB) library. However, Bowtie2 can build without TBB if make fails. Use make NO_TBB=1 instead.

   2.3. Add executable files that have been made from Step 2.2 to environment variable PATH. The simplest way of doing so is just copying all the executable files to the PATH as following.

   User@Linux> sudo chmod +x /path to bowtie2/bowtie2
   User@Linux> sudo cp /bowtie2-2.2.6/bowtie2 /usr/bin/
   User@Linux> sudo cp /bowtie2-2.2.6/bowtie2-align-s /usr/bin/
   User@Linux> sudo cp /bowtie2-2.2.6/bowtie2-build /usr/bin/
   User@Linux> sudo cp /bowtie2-2.2.6/bowtie2-build-s /usr/bin/
   User@Linux> sudo cp /bowtie2-2.2.6/bowtie2-build-l /usr/bin/
   User@Linux> sudo cp /bowtie2-2.2.6/bowtie2-inspect /usr/bin/
   User@Linux> sudo cp /bowtie2-2.2.6/bowtie2-inspect-s /usr/bin/
   User@Linux> sudo cp /bowtie2-2.2.6/bowtie2-inspect-l /usr/bin/

   2.4. Check installation by calling bowtie2. Make sure that printed version matches the version installed.

   User@Linux> bowtie2 --version

3. Install Samtools (v.0.1.19) according to the instructions provided in http://www.htslib.org/ and Step B.2.

4. Install BEDTools (v.2.25.0) according to the instructions provided in https://bedtools.readthedocs.io/en/latest/ and Step B.2.

5. Install Python (v.3.5) according to the instructions provided in https://www.python.org/ and Step B.2.

6. Install R (v.3.6.0) according to the instructions provided in https://cran.r-project.org/doc/manuals/R-devel/R-admin.html/ and Step B.2.

C. Trim low quality reads and adapter sequences

In this manuscript, we will demonstrate the pipeline using example dataset that is subsampled from previously reported Ribo-Seq result of E. coli, available through the Github repository (Choe et al., 2019).

1. Locate the raw sequencing file (fastq.gz) in a directory named 1.Raw_data. Example files are available through the Github repository of STATR (https://github.com/robinald/STATR). Full files of previous studies are avail-

Fig. 2. Default directory structure to run STATR. Default directory structure that supports running of STATR with a single Linux shell script. This structure is not mandatory.
able via the European Nucleotide Archive (https://www.ebi.ac.uk/ena/data/view/PRJEB21199) (Choe et al., 2019).

2. Locate the adapter sequence file (RiboSeq_adapter.fa) in Trimmomatic/adapters/directory. If using a different RiboSeq adapter than in previous studies (Latif et al., 2015; Choe et al., 2019), make a custom FASTA file containing the adapter sequence.

3. Trim the sequences using the following command:

   ```bash
   User@Linux>mkdir -p ./ReferenceDB
   User@Linux>java -jar Trimmomatic-0.39.jar SE -phred33 /1.Raw_reads/MG1655_RiboSeq_Sample1.fastq.gz /2.Trimmed_reads/MG1655_RiboSeq_Sample1.trimmed.fastq.gz
   ``

   The Trimmomatic command above can be used if required. Three additional parameters follow in this option. The first parameter 2 is the maximum mismatches allowed in 16 nt seed sequence comparison. The second parameter 30 is for pair-ended reads and is omitted. The last parameter 6 is the minimum alignment score between the adapter and read for trimming. Ten matches are needed to score 6.

   The second parameter 30 is for pair-ended reads and is omitted. The last parameter 6 is the minimum alignment score between the adapter and read for trimming. Ten matches are needed to score 6.

   ```bash
   ILLUMINACLIP:/Trimmomatic/adapters/RiboSeq_adapter.as.fa:2:30:6 SLIDINGWINDOW:4:15 MINLEN:12
   ``

   E. Decompile alignment file

   1. Convert the .SAM alignment file into a .BAM file format as follows:

   ```bash
   User@Linux>mkdir -p ./4.Decompiled
   User@Linux>java -jar Trimmomatic-0.39.jar SE -phred33 /1.Raw_reads/MG1655_RiboSeq_Sample1.fastq.gz /2.Trimmed_reads/MG1655_RiboSeq_Sample1.trimmed.fastq.gz
   ```

   2. Sort the .BAM file for further analysis:

   ```bash
   User@Linux>samtools sort -o ./4.Decompiled/MG1655_RiboSeq_Sample1_mapping.sorted.bam ./4.Decompiled/MG1655_RiboSeq_Sample1_mapping.bam
   ```

   3. Decompile the sorted alignment file (.BAM) using the following command:

   ```bash
   User@Linux>bedtools bamtobed -i ./4.Decompiled/MG1655_RiboSeq_Sample1_mapping.sorted.bam > /4.Decompiled/MG1655_RiboSeq_Sample1.bed
   ```

   Note that the second > mark denotes the redirection operator of Linux and in the command line, and not that a new line is started.

3. Align the trimmed reads on the reference genome as follows:

   ```bash
   User@Linux>mkdir -p ./3.Aligned
   User@Linux>bowtie2 -q -U ./2.Trimmed_reads/MG1655_RiboSeq_Sample1_trimmed.fastq.gz -x ./ReferenceDB/MG1655_genome -S ./3.Aligned/MG1655_RiboSeq_Sample1_mapping.sam --local
   ```

   ```bash
   -q: Declare the input file is FASTQ format.
   -U: Declare the input file is single-ended file and provide the file path to the input file.
   -x: Provide the file path to the reference sequence built at Step D.2.
   -S: Write the output file in SAM file format with the followed file path.
   --local: It is important to use the local alignment mode rather than end-to-end, since some additional bases can be attached at the end of the sequence. This option allows for read alignment with dangling ends.

   F. Validate experiment and visualize genome-wide ribosome profile

   1. To validate the Ribo-Seq experiment, three-nucleotide codon periodicity will be tested. Genome annotation should be provided as a GFF file. STATR contains a Python script that converts/flushes GFF3 annotation file downloaded from the NCBI database to a readable GFF3 file. The following steps should be performed:

   1.1. Find the reference genome from the NCBI Nucleotide Database (https://www.ncbi.nlm.nih.gov/nuccore).
1.2. Download the genome annotation from the menu ‘Send to-Complete Record-File-GFF3’.
1.3. Convert the information-rich GFF3 file into an essential GFF file as follows:

```bash
User@Linux> python ./Python_scripts/ParseGenomeAnnotation.py -i ./ReferenceDB/NC_000913.3.gff3 -o ./ReferenceDB/NC_000913.3_CDS.gff
```

- **-i:** This parameter specifies the input GFF3 file.
- **-o:** This parameter specifies the output file to generate.

This python script extracts only CDSs from the input GFF3 and retrieves “locus_tag” as an identifier of CDS among many attributes in the input GFF3.

- **Note that this step is optional if using custom genome annotations. Also, the GFF files can be freely customizable by the user in the text or in a spreadsheet editor software, such as Microsoft Excel.**

2. Conduct meta-analysis using Python script, CheckPeriodicity.py, as follows:

```bash
User@Linux> mkdir -p ./5.MetaAnalysis/
User@Linux> python ./Python_scripts/CheckPeriodicity.py -i ./4.Decompiled/MG1655_RiboSeq_Sample1.bed -a ./ReferenceDB/NC_000913.3_CDS.gff -o ./5.Meta_analysis/Meta_positional_density.txt
```

- **-i:** This parameter specifies the decompiled BED file from the previous section, Step E.3.
- **-a:** This parameter specifies the CDS annotation file in Step F.1.
- **-o:** This parameter specifies the output file to generate.

3. The output file is comprised of tab-separated information of meta-ribosome density. Check the three-nucleotide periodicity, as in Fig. 3.

- **Note that the position of adenine of the start codon is 0 rather +1, unlike general biological convention.**

4. Generate a genome-wide Ribo-Seq profile using one of the two ends of the reads. The following python script generates a genome-wide Ribo-Seq profile with a user-selected end. The profile is a read count on each position of the genome:

```bash
User@Linux> python ./Python_scripts/GenerateProfile.py -i ./4.Decompiled/MG1655_RiboSeq_Sample1.bed -e 5 -a ./ReferenceDB/NC_000913.3_CDS.gff -n 1000000 -o ./5.Meta_analysis/MG1655_RiboSeq_Sample1_Profile.gff
```

- **-i:** This parameter specifies the input decompiled BED file in Step E.3.
- **-e:** This option defines which end of the read is used for analysis. The user must select either the 5' or 3' end of the read to use, based on the resolution determined in Step F.3.
- **-a:** This parameter specifies the annotation file containing CDSs generated in Step F.1.
- **-n:** This parameter represents the normalization factor. If N is given, the script does not normalize the data. If a number is given, the script calculates how many reads may be mapped if there were a given number of reads mapped in total, similar to the TPM metric in RNA-Seq analysis.

**Fig. 3.** Average ribosome density on relative position from the start codon. Position of an adenine base of the start codon is 0. To assign ribosome position, 5’ or 3’ end position of reads shows clearer resolution than 3’ end assignment.

**Fig. 4.** Genome-wide Ribo-Seq profile generated by the STATR pipeline and visualized by SignalMap (v2.0.0.5).
A simple bacterial translatome analysis tool for biologists (Li et al., 2010).

- o: This parameter specifies the output file to generate.

5. To visualize the profile multiple free software available, for example, SignalMap (https://sequencing.roche.com/en/products-solutions/by-category/target-enrichment/software/signal-map-software.html) and Integrative Genomic Viewer (http://software.broad institute.org/software/igv/) on the user’s own liability. By doing so, the user can freely inspect the genome-wide Ribo-Seq profile (Fig. 4).

G. Differential expression

1. To analyze the differential translation, count the mapped reads on every CDSs from the alignment file. Genome annotation should be provided as a GFF file.

User@Linux> mkdir -p ./6.Differential_expression

User@Linux> bedtools coverage -a ./ReferenceDB/NC_000913.3_CDS.gff -b ./4.Decompiled/MG1655_RiboSeq_Sample1.bed -s > ./6.Differential_expression/MG1655_RiboSeq_Sample1.cov

- a: This parameter specifies the annotation file containing CDSs generated in Step F.1.3.
- b: This parameter specifies the input decompiled BED file in Step E.3.
- s: Ribo-Seq is a strand-specific RNA-Seq method. Sequenced reads are on the same strand as the gene. Thus, only reads with the same strand of CDS are counted by this option. Careful not to use -S (upper S) option.
- Note that the second > mark denotes the redirection operator of Linux and in the command line, and not that a new line is started.

2. Next, merge and re-format the read count into a single file that can be analyzed by the DESeq2 package. Use the python script, FormatDESeqInput.py, as follows:

User@Linux> python ./Python_scripts/FormatDESeqInput.py -i ./6.Differential_expression/MG1655_RiboSeq_Ctrl1.cov:./6.Differential_expression/MG1655_RiboSeq_Ctrl2.cov:./6.Differential_expression/MG1655_RiboSeq_Exp1.cov:./6.Differential_expression/MG1655_RiboSeq_Exp2.cov -o./6.Differential_expression/DESeq_Input.txt

- i: This parameter specifies the coverage files calculated in Step G.1. Provide multiple files separated by a colon (:)..
- o: This parameter specifies the output file to generate.

3. Prepare an experiment design sheet. The experimental design sheet couples the given sequencing data files with each experimental condition, so that DESeq2 can identify which samples are replicates and which samples are required for comparison. The design sheet is a tab-separated text file with the following format:

File name: Design_Sheet.txt

File content:

| Sample             | Condition     |
|--------------------|---------------|
| MG1655_RiboSeq_Ctrl1 | Control group |
| MG1655_RiboSeq_Ctrl2 | Control group |
| MG1655_RiboSeq_Exp1  | Experimental group |
| MG1655_RiboSeq_Exp2  | Experimental group |

4. Install the required R packages manually or by using InstallPackages.R as follows:

User@Linux> sudo Rscript ./R_scripts/InstallPackages.R

5. Run DESeq2 analysis with R script included in the STATR pipeline:

User@Linux> sudo Rscript ./R_scripts/RunDESeq.R -i ./6.Differential_expression/DESeq_Input.txt -d ./6.Differential_expression/Design_sheet.txt -o ./6.Differential_expression/

6. The RunDESeq.R script generates multiple pdf files. First, Dendrogram_and_heatmap.pdf contains a dendrogram and a heatmap showing the hierarchical clustering of samples based on pairwise Euclidean distances between samples (Fig. 5A). Also, the two-dimensional principal component analysis (PCA) plot is printed as PCA.pdf (Fig. 5B). Based on the two graphs, the reproducibility of experiments can be examined. Then, a tab-separated text file comprising the DESeq2 normalized expression levels of each gene is printed as Normalized_expression.txt. Lastly, pairwise comparisons between given conditions are printed as Contrast_<Condition1>_vs_<Condition2>.txt files. The number of files is the same as the number of pairwise combinations of conditions (NC2; where N equals the number of experimental conditions including control). The comparison files contain the mean expression, log2 fold-change, and p-value statistics of the differential expression.

**Expected Results**

Methicillin-resistance *Staphylococcus aureus* (MRSA) is mul-
tidrug resistant pathogenic bacteria that has become a major healthcare burden (Highlander et al., 2007). To determine the virulence and resistance mechanism, we analyzed the translatomic response of the *S. aureus* USA300-HOU-MR strain under antibiotic treatment using the STATR pipeline. One of the oxazolidinone linezolid was treated with a sub-lethal dosage to reproduce the antibiotic treatment carried out in clinics. In the Ribo-Seq dataset, the 3' end of the reads showed a clearer three-nucleotide periodicity than the 5' end (Fig. 6A). Interestingly, the ribosome density of the linezolid-treated *S. aureus* near start codon was significantly higher than the control sample (Fig. 6B). Linezolid is a translation initiation inhibitor that binds to the A site of the ribosome, preventing cognate-charged tRNA entering as a result (Renslo, 2010; Zhanel et al., 2015). Considering that the A site of the ribosome lies -14 nt upstream of the 3' end of the RPF (Mohammad et al., 2019), the increased RPF profile indicates ribosome stalling at the start codon (Fig. 6B). As this example shows, Ribo-Seq analysis with STATR provides a nucleotide-resolution mechanistic understanding of translation *in vivo*. Next, we analyzed the differential translation of genes involved in the response toward linezolid treatment. Before analysis, the reproducibility between biological replicates and between the different treatment groups was examined. The two biological replicates were clustered well and the different experimental groups were separated distinctly (Fig. 6C and D). By analyzing the normalized expression levels, we discovered 342 differentially expressed genes (DEGs) are colored red.
(translated) genes (DEGs) in which the expression level was significantly changed (adjusted p-value lower than 0.05) when treated with linezolid (Fig. 6E). It has been reported that the transcription of the accessory gene regulator (Agr) is induced upon exposure to linezolid (Choe et al., 2018). Agr proteins are the major regulators of the virulence and pathogenesis of *S. aureus* (Queck et al., 2008), whose translation in response to linezolid has never been reported. DEG analysis revealed that the translational levels of Agr regulators are activated similar to the transcriptional levels (Fig. 6E). Overall, STATR provides a simple and robust analysis pipeline for translational studies, including high-resolution ribosome footprinting, allowing for a molecule-level study of translation, as well as a genome-wide understanding of translation.

**Acknowledgments**

This work was supported by the Korea Bio Grand Challenge (2018M3A9H3024759 to B-KC) and the Basic Core Technology Development Program for the Oceans and the Polar Regions (2016M1A5A1027458 to B-KC) through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT. This work was also funded by National Institutes of Health/National Institute of General Medical Sciences Grant (1R01GM098105 and 1-U01-AI124316-01 to BOP).

**Data and Material Availability**

All the scripts and example files of the STATR pipeline are freely available through the Github repository (https://github.com/robinald/STATR) under the GNU General Public License v 3.0 (https://www.gnu.org/licenses/gpl-3.0.en.html) and may be reused by Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/).

**Conflicts of Interest**

The authors declare that there are no conflicts of interest.

**References**

Bolger, A.M., Lohse, M., and Usadel, B. 2014. Trimmomatic: a flexible trimmer for Illumina sequence data. *Bioinformatics* 30, 2144–2142.

Chen, J., Petrov, A., Johansson, M., Tsai, A., O’Leary, S.E., and Puglisi, J.D. 2014. Dynamic pathways of -1 translational frameshifting. *Nature* 512, 328–332.

Choe, D., Lee, J.H., Yoo, M., Hwang, S., Sung, B.H., Cho, S., Palsson, B., Kim, S.C., and Cho, B.K. 2019. Adaptive laboratory evolution of a genome-reduced *Escherichia coli*. *Nat. Commun.* 10, 935.

Choe, D., Szubin, R., Daresh, S., Cho, S., Nizet, V., Palsson, B., and Cho, B.K. 2018. Genome-scale analysis of methicillin-resistant *Staphylococcus aureus* USA300 reveals a tradeoff between pathogenesis and drug resistance. *Sci. Rep.* 8, 2215.

Dar, D., Shamir, M., Mellin, J.R., Koutero, M., Stern-Ginossar, N., Cossart, P., and Sorek, R. 2016. Term-seq reveals abundant ribosome regulation of antibiotics resistance in bacteria. *Science* 352, aad9822.

Davies, E. and Larkins, B.A. 1973. Polyribosomes from Peas: II. Polyribosomal metabolism during normal and hormone-induced growth. *Plant Physiol.* 52, 339–345.

Fan, Y., Evans, C.R., Barber, K.W., Banerjee, K., Weiss, K.J., Margolin, W., Igoshin, O.A., Rinehart, J., and Ling, J. 2017. Heterogeneity of stop codon readthrough in single bacterial cells and implications for population fitness. *Mol. Cell* 67, 826–836.e825.

Gerashchenko, M.V., Lobanov, A.V., and Gladyshev, V.N. 2012. Genome-wide ribosome profiling reveals complex translational regulation in response to oxidative stress. *Proc. Natl. Acad. Sci. USA* 109, 17394–17399.

Highlander, S.K., Hultén, K.G., Qin, X., Jiang, H., Yerrapragada, S., Mason, E.O.Jr., Shang, Y., Williams, T.M., Fortunov, R.M., Liu, Y., et al. 2007. Subtle genetic changes enhance virulence of methicillin resistant and sensitive *Staphylococcus aureus*. *BMC Microbiol.* 7, 99.

Ingolia, N.T., Ghaemmaghami, S., Newman, J.R., and Weissman, J.S. 2009. Genome-wide analysis in vivo of translation with nucleotide resolution using ribosome profiling. *Science* 324, 218–223.

Jeong, Y., Kim, J.N., Kim, M.W., Bucca, G., Cho, S., Yoon, Y.J., Kim, B.G., Roe, J.H., Kim, S.C., Smith, C.P., et al. 2016. The dynamic transcriptional and translational landscape of the model antibiotic producer *Streptomyces coelicolor* A3(2). *Nat. Commun.* 7, 11605.

Lang, B.F., Jakubkova, M., Hegedusova, E., Daoud, R., Forget, L., Brejova, B., Vinar, T., Kosa, P., Fricova, D., Nebohacova, M., et al. 2014. Massive programmed ribosome profiling protocol for the characterization of microorganisms. *Biotechniques* 58, 329–332.

Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Abecasis, G., Durbin, R., and 1000 Genome Project Data Processing S. 2009. The sequence alignment/map format and SAMtools. *Bioinformatics* 25, 2078–2079.

Li, G.W., Oh, E., and Weissman, J.S. 2012. A systematic-ly revised ribosome profiling method for bacteria reveals pauses at single-codon resolution. *Nucleic Acids Res.* 40, 538–541.

Li, B., Ruotti, V., Stewart, R.M., Thomson, J.A., and Dewey, C.N. 2010. RNA-Seq gene expression estimation with read mapping uncertainty. *Bioinformatics* 26, 493–500.

Love, M.I., Huber, W., and Anders, S. 2014. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biol.* 15, 550.

Mohammad, F., Green, R., and Buskirk, A.R. 2019. A systematically-revised ribosome profiling method for bacteria reveals pauses at single-codon resolution. *Elife* 8, e42591.

Nagalakshmi, U., Wang, Z., Waern, K., Shou, C., Raha, D., Gerstein, M., and Snyder, M. 2008. The transcriptional landscape of the yeast genome defined by RNA sequencing. *Science* 320, 1344–1349.

Paulet, D., David, A., and Rivals, E. 2017. Ribo-seq enlightens codon usage bias. *DNA Res.* 24, 303–310.

Pelechano, V., Wei, W., and Steinmetz, L.M. 2013. Extensive transcriptional heterogeneity revealed by isoform profiling. *Nature* 497, 127–131.

Potts, A.H., Vakulskas, C.A., Pannuri, A., Yakhnin, H., Babitzke, P., and Romeo, T. 2017. Global role of the bacterial post-transcriptional regulator CrsA revealed by integrated transcriptomics. *Nat. Commun.* 8, 1596.

Queck, S.Y., Jameson-Lee, M., Villaruz, A.E., Bach, T.H., Khan, B.A., Choe, D., Lee, J.H., Yoo, M., Hwang, S., Sung, B.H., Cho, S., Palsson, B., Kim, S.C., and Cho, B.K. 2018. Adaptive laboratory evolution of a genome-reduced *Escherichia coli*. *Nat. Commun.* 10, 935.
Sturdevant, D.E., Rickles, S.M., Li, M., and Otto, M. 2008. RNAIII-independent target gene control by the agr quorum-sensing system: insight into the evolution of virulence regulation in *Staphylococcus aureus*. *Mol. Cell* 32, 150–158.

Quinlan, A.R. and Hall, I.M. 2010. BEDTools: a flexible suite of utilities for comparing genomic features. *Bioinformatics* 26, 841–842.

Renslo, A.R. 2010. Antibacterial oxazolidinones: emerging structure-toxicity relationships. *Expert Rev. Anti Infect. Ther.* 8, 565–574.

Shalgi, R., Hurt, J.A., Krykbaeva, I., Taipale, M., Lindquist, S., and Burge, C.B. 2013. Widespread regulation of translation by elongation pausing in heat shock. *Mol. Cell* 49, 439–452.

Sharma, C.M., Hoffmann, S., Darfeuille, F., Reignier, J., Findeiss, S., Sittka, A., Chabas, S., Reiche, K., Hackermuller, J., Reinhardt, R., et al. 2010. The primary transcriptome of the major human pathogen *Helicobacter pylori*. *Nature* 464, 250–255.

Van Assche, E., Van Puyvelde, S., Vanderleyden, J., and Steenackers, H.P. 2015. RNA-binding proteins involved in post-transcriptional regulation in bacteria. *Front. Microbiol.* 6, 141.

Woolstenhulme, C.J., Guydosh, N.R., Green, R., and Buskirk, A.R. 2015. High-precision analysis of translational pausing by ribosome profiling in bacteria lacking EFP. *Cell Rep.* 11, 13–21.

Zhanel, G.G., Love, R., Adam, H., Golden, A., Zelenitsky, S., Schweizer, F., Gorityala, B., Lagace-Wiens, P.R., Rubinstein, E., Walkty, A., et al. 2015. Tedizolid: a novel oxazolidinone with potent activity against multidrug-resistant gram-positive pathogens. *Drugs* 75, 253–270.