SYNTHETIC BIOLOGY

Synthetic Biology, 2022, **7(1)**, 1–9
DOI: https://doi.org/10.1093/synbio/ysac012
Advance access publication date: 23 August 2022
Research Article

A toolkit for enhanced reproducibility of RNASeq analysis for synthetic biologists

Benjamin J. Garcia[1], Joshua Urrutia[2], George Zheng[3], Diveena Becker[4], Carolyn Corbet[4], Paul Maschhoff[4], Alexander Cristofaro[1,5], Niall Gaffney[2], Matthew Vaughn[2], Uma Saxena[1], Yi-Pei Chen[3], D. Benjamin Gordon[1], and Mohammed Eslami[3,*]

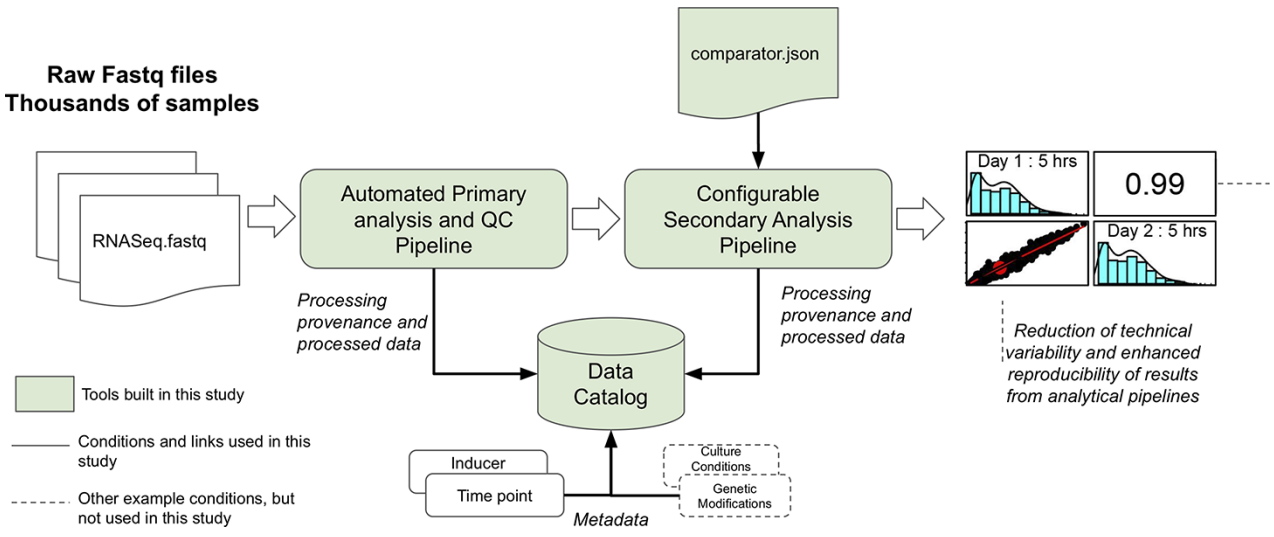
[1]Department of Biological Engineering, Synthetic Biology Center, Massachusetts Institute of Technology, Cambridge, MA, USA
[2]Texas Advanced Computing Center, University of Texas at Austin, Austin, TX, USA
[3]Netrias LLC, Annapolis, MD, USA
[4]Ginkgo Bioworks, Boston, MA, USA
[5]TScan Therapeutics, Waltham, MA, USA
*Corresponding authors: E-mail: meslami@netrias.com

Abstract

Sequencing technologies, in particular RNASeq, have become critical tools in the design, build, test and learn cycle of synthetic biology. They provide a better understanding of synthetic designs, and they help identify ways to improve and select designs. While these data are beneficial to design, their collection and analysis is a complex, multistep process that has implications on both discovery and reproducibility of experiments. Additionally, tool parameters, experimental metadata, normalization of data and standardization of file formats present challenges that are computationally intensive. This calls for high-throughput pipelines expressly designed to handle the combinatorial and longitudinal nature of synthetic biology. In this paper, we present a pipeline to maximize the analytical reproducibility of RNASeq for synthetic biologists. We also explore the impact of reproducibility on the validation of machine learning models. We present the design of a pipeline that combines traditional RNASeq data processing tools with structured metadata tracking to allow for the exploration of the combinatorial design in a high-throughput and reproducible manner. We then demonstrate utility via two different experiments: a control comparison experiment and a machine learning model experiment. The first experiment compares datasets collected from identical biological controls across multiple days for two different organisms. It shows that a reproducible experimental protocol for one organism does not guarantee reproducibility in another. The second experiment quantifies the differences in experimental runs from multiple perspectives. It shows that the lack of reproducibility from these different perspectives can place an upper bound on the validation of machine learning models trained on RNASeq data.

Key words: transcriptomics; automation; standardization; machine learning

Graphical Abstract



Submitted: 21 October 2021; **Received (in revised form):** 17 June 2022; **Accepted:** 22 August 2022

1. Introduction

Synthetic design is often an iterative process, where components are improved over time. However, despite the improvement of components, controls can often stay the same across multiple iterations (unless alterations are made to the base strain in such a way that the altered strain becomes the new control). As such, it is important to know whether or not changes are the result of design choices, experimental variability or other unintended perturbations due to inherent variability in the pipeline. Furthermore, researchers conducting experiments in synthetic biology often have combinatorial designs that test all conditions in replicates, resulting in sample sizes in the 1000s or greater. It is not always feasible to run all of these samples at the same time, so it is important to consider how experiments change across batches (i.e. differences in the execution of the pipeline, differences in experiment conditions, different technicians running the pipeline, dilutions, culturing, etc.). It then becomes important to ensure that variability is low and that controls are repeatable across different batches to ensure that experiments can be aggregated and compared.

RNASeq has become a ubiquitous tool in the design, build, test and learn cycle for synthetic biology (1, 11, 14, 17, 18, 42). Applications include debugging design failures (13, 17, 33) and optimization of genetic designs (31). Transcriptional analysis provides an opportunity to measure chassis response holistically and identify differentially expressed genes (DEGs) (17), infer transcriptional networks or co-regulated genes (22, 25, 32, 43), identify readthrough and unexpected expression in synthetic parts (13, 17, 33) and also infer the impacts on mechanistic pathways encompassing a multitude of biomolecular 'entities' ('transcripts', proteins and metabolites) (15, 19, 35, 38).

Collection and processing of RNASeq data, however, is a complex, multistep process (9, 14). Processing pipelines are highly parameterized, computationally intensive and often not comparable across experiments without significant effort in normalization and harmonization of the output measurements (1, 11). Furthermore, these tools often require a significant amount of additional biological context to explain how and why the dataset was generated, which is often disconnected from the data itself (16, 36, 45). Collection and processing is one aspect of reproducibility, which can be more broadly delineated into three categories: (i) biological, (ii) experimental and (iii) analytical (21, 44).

Biological variability results from different responses to similar stimuli, either as a mechanism for survival, due to responses not being strictly regulated or due to experimental conditions being variable enough to elicit different biological responses. Experimental variability is the result of imperfect tool precision, slight alterations in handling during cell growth, lysis, ribonucleicacid (RNA) extraction, library construction or data analysis. Experimental variability can also impact biological reproducibility. To minimize the experimental variability, consistency of conditions, reagents, instruments and protocols used to grow organisms, extract RNA and process the sample are required. Finally, analytical variability arises from either an inconsistency in the use or a lack of information about processing pipelines, tools and parameterization used to analyze the data and extract conclusions (7, 26, 28). While there are a number of consortiums that have provided guidance on methods to assess the reproducibility of RNASeq, a toolkit to enable that reproducibility for synthetic biology is seldom found (23, 47). The toolkit provided in this paper helps mitigate analytical variability by providing a clear and consistent set of methodologies that can be applied across different RNASeq experiments. Given the pipeline provides a clear set of reproducible analytical methods, we are then able to explore biological and experimental reproducibility that can then impact analytical results.

Here, we seek to present the minimal set of information required from both the experimental conditions as well as the data processing pipelines to maximize the analytical reproducibility. We first present an overview of the software infrastructure required to make analyses reproducible across RNASeq experiments. The tool outputs a set of consistently formatted datasets for downstream analyses and can be used by researchers that seek to analyze RNASeq at scale (100s to 1000s of sequencing runs). We then detail a configuration-driven pipeline that enables the reproducible analysis of data that were utilized to analyze data from a large experimental condition space (e.g. many genetic designs and inducers) (12). We use these tools to measure the variability of controls in multiple experimental runs with perturbing controls of two popular model organisms: *Escherichia coli* MG1655 and *Bacillus subtilis* Marburg 168. After quantifying the variability in controls, we examine the impact of the variability on machine learning models built using two *B. subtilis* datasets collected using nearly identical protocols surveying a large space of growth conditions.

2. Materials and methods

2.1 Infrastructure to automate and track secondary analysis stages

Secondary analysis of the sequencing data involves a set of hierarchical steps that depend on the experimental design, conditions and reference files to generate counts and normalized counts data. Given the scale of experimental conditions (e.g. genetic designs, organisms and induction conditions), software infrastructure is required to ensure the analytical reproducibility of an RNASeq pipeline to be parameterized and automated to process RNASeq data at scale. The software infrastructure we have developed has three main components: (i) applications, (ii) databases and (iii) actors.

Applications. Allow the software infrastructure to capture versions, any configuration parameters used during the alignment and quality control (QC) process and input/output data identifiers. They ensure that the specific processing or analysis pipeline that was run by the application can be run again exactly the same way, in the same environment, on any system. Tools such as Docker containers provide an opportunity for researchers to ensure that their code is self-contained, shareable and reproducible (20, 34). Applications are containerized versions for each stage in the pipeline. The main motivation behind modularization was to provide us the flexibility to rerun specific stages, if needed, without rerunning the entire pipeline. Furthermore, the modularity has the added benefit of updating specific components and tracking versions of software and configurable parameters for every execution.

Databases. Allow the software infrastructure to store locations of raw data and processed data, metadata and full parameterizations and provenance of processing pipelines that link the data together in a queryable manner. We had two databases used in our infrastructure, integrated together as part of a larger system for handling data, metadata and knowledge in the Defense Advanced Research Projects Agency SD2 program (6, 41). The first was a data

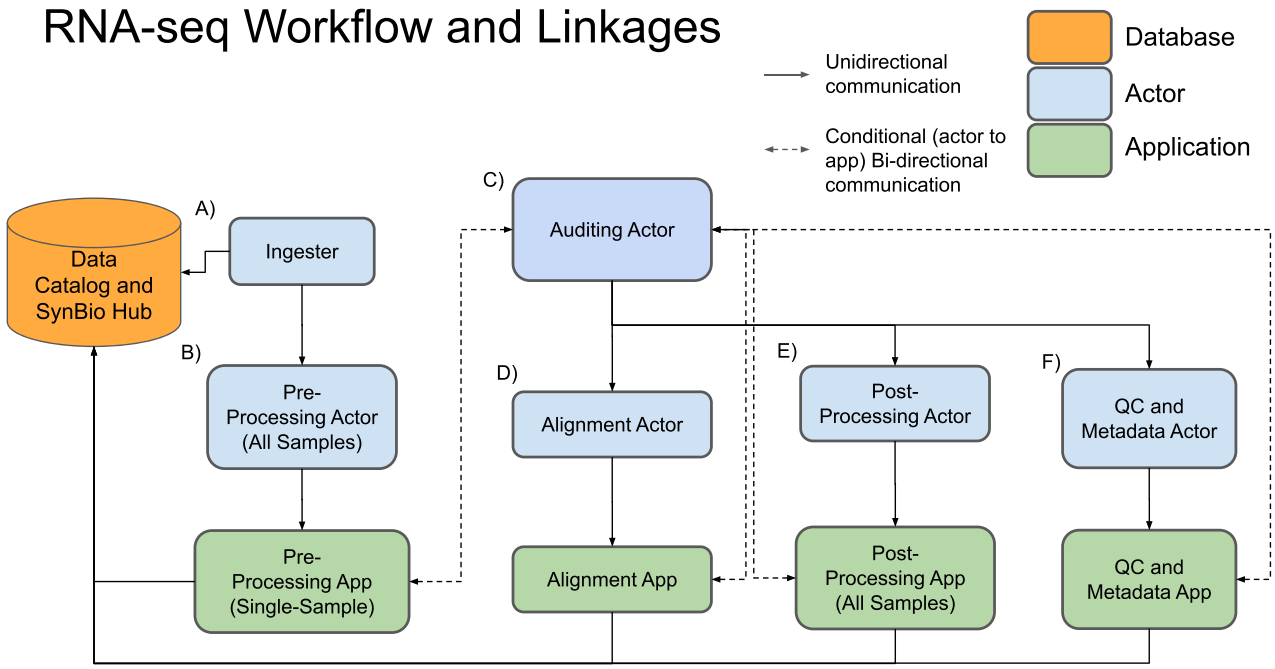


Figure 1. Diagram of the RNASeq processing pipeline. (A) The ingester monitors a metadata file uploaded to the data catalog to see if the experiment includes RNASeq data to trigger the processing pipeline. (B) The preprocessing actor triggers and stores the job ID and version of the preprocessing app to the data catalog. (C) The auditing actor receives notifications from applications to validate the outputs and triggers the next actor. The auditing actor will resubmit applications up to three times to handle stochastic failures. (D) The alignment actor queries the data catalog to determine the reference genome that is used for each preprocessed sample and triggers the alignment application. (E) The post-processing application annotates the alignments and aggregates samples into counts (raw, FPKM and TPM) dataframes. (F) The QC and metadata actor use metadata stored in the data catalog and information from the logs/outputs of each job to add experimentally relevant metadata and QC flags to the counts dataframes.

catalog that stored links to the raw data, universal identifiers and the experimental conditions. The second was a database specifically designed to store genetic parts known as SynBioHub. Some experimental conditions, such as strain and inducer names, stored links to SynBioHub (29) that led to more complete information about these materials.

Actors. Allow the software infrastructure to respond to events (e.g. file upload and completion of an intermediate stage) to automate the processing of data at scale. In our system, reads/writes to the data catalog and job submissions are managed by Abaco Actors (5). Actors are lightweight, containerized scripts that are triggered in response to events. Actors do not perform computationally intensive tasks but instead oversee and coordinate job processing throughout the pipeline. Computationally intensive tasks (trimming, alignment and aggregation) are performed by Tapis applications, a framework that provides a web-based application programming interface (API) to manage computational workloads developed by the Texas Advanced Computer Center (8). Application jobs are triggered by actors and have a unique job identification number.

Given software for each stage of the secondary analysis pipeline (preprocessing, alignment, post-processing and QC), our emphasis was to use the software infrastructure to ensure that the pipeline is manageable and reproducible for the large datasets made available over time (Figure 1).

The applications used in our RNASeq pipeline consist of trimming and quality filtering raw RNASeq data with Trimmomatic (v0.36) (3), and FastQC (2) is used to generate reports for paired, trimmed reads. Trimmomatic was chosen due to its high accuracy, although trimming methods had low overall impact on the overall RNASeq accuracy (10). Preprocessed reads are aligned to an indexed reference genome with BWA (v0.7.17) (8, 27). Burrows-Wheeler Aligner (BWA) was chosen because, for short reads, it has high coverage and alignment rates, for minimal cost in run time efficiency (30). For alignment, there is the prerequisite that the relevant reference genome has been identified, FASTA and GFF have been provided and the linkage between sample and reference genome (strain) is enumerated in the provided metadata. After alignment, the resulting Sequence Alignment Map (SAM) files are sorted by Picard tools (v2.18.15) function SortSam (36, 4) and then AddOrReplaceGroups is run on the output of SortSam. Gene-level quantification of counts is performed using the featureCounts function of Rsubread (v1.34.4) (3). Annotations are performed using the General Feature Format (GFF) provided. For RNA-sequencing, we identify any coding sequence (CDS) feature type from the GFF and annotate these in the dataframe using the 'Name=' attribute. All annotated samples are aggregated into a dataframe, and outputs include both raw counts and rudimentary normalization functions: transcripts per million (TPM) and fragment per kilobase per million mapped reads (FPKM). The following two Boolean metrics were used to measure sample quality:

(i) Number of mapped reads $\geq$ 500 kb.
(ii) Replicate correlation of TPM values of a condition >0.8.

If any of these metrics did not pass, the sample would be flagged as a low-quality sample and not used for downstream analysis.

These applications can then be run individually on a high-performance computing system. For each job that is run in the software infrastructure, an actor queries a centralized database of information (the data catalog) to identify the required metadata

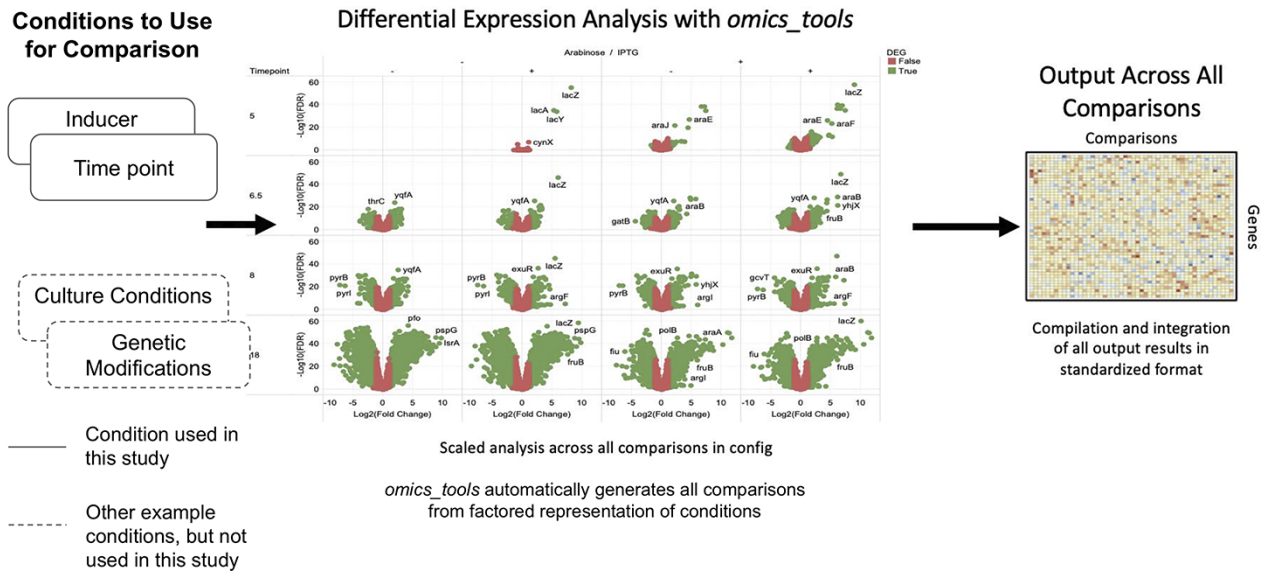


Figure 2. Omics_tools facilitates the analysis of combinatorial design RNASeq by integrating metadata (inducers, growth conditions, genetic manipulations, etc.) to automatically calculate differential expression with edgeR across all conditions of interest. The tool kit allows for parallel processing of thousands of samples and comparisons in a high-throughput manner and utilizes a standardized schema to help facilitate reproducibility in analytical analysis.

for processing. For each job submission, there is also a record created in the data catalog enumerating the inputs, references, software versions and outputs for a given job. Tapis applications stage assets (inputs, references and container images) to a temporary working directory, submit resource requirements to a Simple Linux Utility for Resource Management (46) queue on a high-performance computing system, archive outputs to a predefined location (aka storage system) and send a notification to an actor providing the exit status of a job (success/failure). By leveraging Tapis applications, Abaco Actors and the data catalog, an automated pipeline is constructed that responds to events (raw data and metadata uploads) and processes samples in parallel for trimming and alignment and serially for aggregation and annotation, without the need for human intervention. This serves both to increase efficiency and throughput and also reduce the probability for human error. Since the core components for processing are containerized and the inputs/parameters for every job are stored in a centralized database, the results are reproducible across different machines.

2.2 Configurable tertiary analysis pipeline

A challenge often faced in synthetic biology is that experimental designs are often combinatorial in nature (multiple inducers, designs, timepoints, etc.); therefore, the number of differential expression comparisons can become unwieldy and unreliable to set up and analyze in a manual fashion. Additionally, when an experiment has hundreds to thousands of different comparisons, manual processing of differential expression is often impossible, so automated solutions are necessary to analyze the data. To help facilitate the automated calculation of differential expression between the different factors, we developed a Python-based configurable toolkit, which we call 'omics_tools' (Figure 2). Instead of having to write R scripts for every desired comparison, omics_tools automatically generates each R script, runs them and then generates consistently labeled output. The automated process increases repeatability/reliability by standardizing output and by reducing the risk of producing manual errors, such as mislabeling comparisons, mislabeling groups, including the wrong samples, etc. Standardizing output across multiple different experiments also allows for development and use of tools that analyze differential expression results without having to generate formatting scripts for each different experiment, facilitating cross-experiment comparisons. The tool aggregates the outputs from the parallelized runs and combines all the data into a single unified dataset where each row represents a gene, its differential expression, statistical significance and the condition for downstream machine learning.

Omics_tools uses edgeR (39) with a generalized linear model (GLM) to conduct differential expression analysis (DEA) using trimmed mean of M-values (TMM) normalization across the set of design variables (40). EdgeR was chosen, specifically with GLM and TMM, because the combination is a well-known method that has a high accuracy across a variety of datasets (Corchete et al., 2020). Additionally, TMM normalization has a high accuracy irrespective of other components within the tool chain (such as trimmer, mapper, differential expression, etc.) (Corchete et al., 2020). Other methods can replace the aforementioned method by either replacing omics_tools with another tool of your choice (starting with the counts data generated in previous steps) or modifying the R script generation of omics_tools to replace edgeR with another method that utilizes R to function.

3. Results and discussion

3.1 Automation enhances scale and processing of RNASeq data at scale

Data used for this experiment consisted of raw, gzipped RNASeq data of 2.4 TB in total that was processed into datasets that were 35 MB in total. The data included both metadata as well as data from sequencers for 1344 samples. Raw and processed data are available via the Gene Expression Omnibus (GSE206047). The infrastructure presented in Figure 1 was not available during all of the experiments conducted with *E. coli*. During this time, it would typically take ~3 months to process the data, where most of the time was spent by a developer finding, verifying

Table 1. Percentage of the total 4097 genes assayed that were significantly differentially expressed across comparisons (sample_time, FDR < 0.05 and log2FC > 2)

Timepoint	Day 1: 5 h	Day 1: 8 h	Day 1: 18 h	Day 2: 5 h	Day 2: 8 h	Day 2: 18 h
Day 1: 5 h	0.0%	3.8%	29.4%	0.3%	4.6%	32.5%
Day 1: 8 h	3.8%	0.0%	20.2%	2.4%	0.3%	24.2%
Day 1: 18 h	29.4%	20.2%	0.0%	27.2%	20.1%	0.2%
Day 2: 5 h	0.3%	2.4%	27.2%	0.0%	2.2%	29.6%
Day 2: 8 h	4.6%	0.3%	20.1%	2.2%	0.0%	22.6%
Day 2: 18 h	32.5%	24.2%	0.2%	29.6%	22.6%	0.0%

and troubleshooting data processing and metadata integration issues. This included mismatches of number of files to metadata rows, incomplete strain references and gaps in inducer concentrations as well as timepoints. The infrastructure now checks for all of this with the auditing actor and can alert users immediately if there is a mismatch, gap or processing error in any stage. The experiments with *B. subtilis* were able to benefit from the complete infrastructure and took a ~3-month process down to 3 days, where the majority of the time was processing time. We also tested our infrastructure of Gene Expression Omnibus (GEO)-derived datasets to ascertain the extendability of our pipeline to data that were not designed for internal purposes. For example, with the manual restructuring of the metadata associated with a *Pseudomonas putida* experiment (37), we were able to run their data through our pipeline to get gene counts. While the results are not published with this paper, details of this process can be found in our documentation.

3.2 Using omics_tools to analyze the repeatability of controls

To quantify the reproducibility of controls, we ran two experiments involving *E. coli* MG1655 in media at three different timepoints (5, 8 and 18 h growth) on two different days. The timepoints were selected to represent common phases of growth in circuit characterization (17). Four replicates were used for each timepoint on each day. Our processing pipeline and omics_tools were used to analyze the data for all comparisons across all timepoints. The results from this comparison are presented in Table 1.

Samples from the same timepoint on different days were found to be most similar (average 0.3% of the genome differentially expressed), and the least similar were the 5 h versus 18 h comparison (average 29.9% differential expression). The differences in 5 h versus 8 h, 5 h versus 18 h and 8 h versus 18 h timepoints are as expected and can be largely attributed to differences in growth phase (log phase versus transition versus stationary), whereas the minor differences in the same hour comparison can be attributed to slight differences in the state of the bacteria and aggregate of error and variability across the pipeline.

We also compared different means of performing sample-to-sample normalization and identifying DEGs. The sample similarities can also be observed when using an FPKM normalization (Figure 3). We also utilized FPKMs to compute the differential expression of genes with *t*-test and Benjamini–Hochberg multiple testing correction, log2FC > 2 (fold change) and false discovery rate (FDR) < 0.05 (Supplemental Table S1) and compared those to omics_tools. Despite having a similar number of DEGs between the two methods, the resulting significant differential genes have a subset of genes that is not shared between the two. Unsurprisingly, the Jaccard coefficients (Supplemental Table S1)

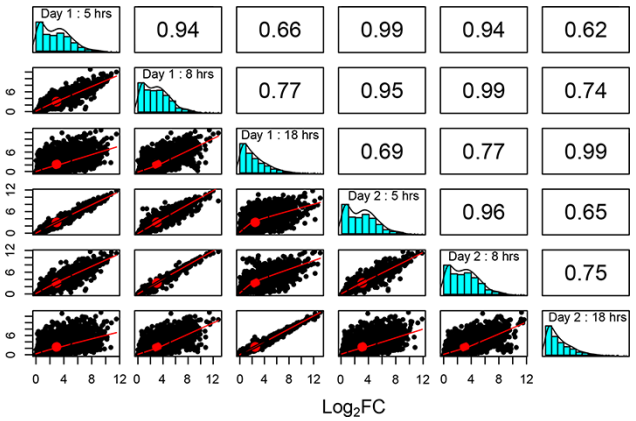


Figure 3. Pearson's correlations (upper right numbers) between log2 FPKM of *E. coli* samples at each of the measured timepoints on both days. Samples taken on different days, at the same hour, are much more similar (average 0.99) than at different hours. The 5 and 18 h have high correlations (average 0.95), suggesting that they are in similar growth states, whereas the 5 and 18 h have the least similar expression profiles (average 0.64) due to their differences in growth states. Scatterplots are gene–gene log2 FPKM comparisons, with a red linear regression line. Histogram plots are for frequencies of gene FPKMs.

for the same timepoints across the 2 days averaged only 0.214 (range, 0.083–0.308) due to the very low number of genes differentially expressed (10.8 on average with a range of 1–21). The different timepoint and day comparisons had much higher similarities in differential genes with an average Jaccard of 0.598 (range 0.468–0.762) and an average percentage of 82.6% (range 75.3–88.8%) of genes that were significant from both omics_tools and FPKM comparisons. The main difference between omics_tools and FPKM is from intersample normalization of TMM. While normalization and differential strategy do matter, the majority of genes that are significant are shared between the methods.

To better understand how different species are affected by similar experimental conditions, we also examined data collected for *B. subtilis* controls at 5 h on three different days. The data were processed through the same high-throughput sequencing pipeline and analyzed with omics_tools. Unlike the *E. coli* strain, *B. subtilis* had much higher variability with an average of 4.87% significant differential expression (range 1.1–7.16%) (Supplemental Table S2) compared to 0.269% for the 5 h *E. coli* timepoint. The FPKM correlations (Figure 4) have a similar pattern as the 5–8 h *E. coli* comparison (Figure 3), suggesting that the growth rates may be different between the different days for *B. subtilis*. It is hypothesized that *B. subtilis* was more impacted by overnight growth and recovery in wells as compared to *E. coli*. The differences in dilution potentially impact growth and sporulation as well. Fundamentally, we showed that experimental conditions that produce reproducible results for one organism do not necessarily produce reproducible results for different organisms.

3.3 The impact of reproducibility on the validation of machine learning predictions

One challenge associated with the validation of machine learning models for high-throughput experiments is regarding reproducibility of the training data, specifically, if a model is built from a set of data that lacks or has underdeveloped biological and experimental context (i.e. metadata). Despite the immediate accuracy concerns, there is no guarantee that the model will generalize to future runs of experiments with similar experimental/biological

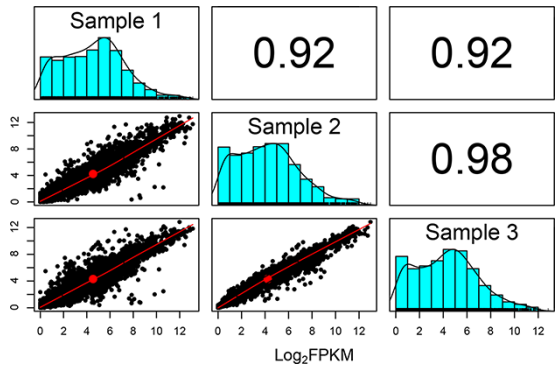


Figure 4. Pearson's correlations (upper right numbers) between log2 FPKM of 5 h timepoints for *B. subtilis* on each of the three different days. Samples 2 and 3 have much more similar patterns compared to 1–2 and 1–3. Additionally, these two comparisons have a lower correlation than the 5–8 h *E. coli* comparisons, suggesting greater variability in the growth/biological conditions for *B. subtilis*. Scatterplots are gene–gene log2 FPKM comparisons, with a red linear regression line. Histogram plots are for frequencies of gene FPKMs.

context. Furthermore, extensibility of a given model has an upper bound of the experimental reproducibility of the set of conditions being modeled.

Here, we evaluate the implications of reproducibility on a machine learning model trained with response from single inducers to predict the transcriptional response to a combination of inducers. The test data were generated twice on different days to measure the impact reproducibility has on the evaluation of the model. The samples were prepared in 96-well plates using a combination of automation and manual liquid handling methods that benefited from consistent sample volume throughout the process. We noticed that optical density (OD) values varied from 0.5 to 2.0 OD units across the replicates per condition with the first run of our experiment at 500 μl volume. Given the variability in OD, increasing the amount of culture collected allowed for more samples to meet the minimum material input criteria for concentration in the starting volume of samples. Thus, the only difference between the two experimental runs was the increase in volume of cell culture used for RNASeq from 500 to 900 μl to increase the amount of RNA available. We hypothesized that the increase would improve the quality of the samples generated, which would result in data for conditions that would drop out. Upon increasing the volume, we now observed a variability of 0.35–0.45 OD units. The decrease in variability with the increased volume was encouraging. Interestingly, cell culture volume is a factor that is often not used in downstream normalization, differential expression and pathway analyses directly, but as we will show, does have implications for sample quality that in turn impacts all other analyses. Furthermore, the increase in volume did not unanimously increase the number of replicates/samples available for analysis either, which we found surprising. This comparison can be made at multiple levels:

Sample QC. (Figure 5A) Given the set of experimental parameters, QC can filter out a set of different samples per condition. For example, a sample induced with xylose and vanillic acid measured at 18 h post-induction has lost all four replicates in the repeated test condition. This implies that the condition will not generate the data necessary to identify how reproducible the model is at that condition. A researcher could look to see if the model predicted any impact on essential genes or pathways that can be linked to dropout, especially if the sample failed in both experimental runs. This could help the researcher determine if the dropout is biological in nature and not an experimental error. The more difficult scenario is when a set of samples pass QC in one run, but not the other. In total, there are 5/18 discrepancies across the two experimental runs, which means that the day-to-day variability could cause upward of 27% difference depending on the experimental run that is used.

Gene QC. (Figure 5B) Tertiary analysis, such as DEA, uses expression levels across genes to identify outlying genes that do not fit a dispersion model. Genes that do not fit within a dispersion estimate are filtered out. Filtering out genes means that a model that is making gene-level predictions will not have predictions of those genes validated quantitatively. However, if the model predicts the gene is not differentially expressed, then its expression likely falls below a noise threshold and that could also be why the dispersion model filtered it out. Thus, it is necessary to choose a dispersion model that accounts for expression level changes across experiments. Here, we use a local regression dispersion method to determine if a gene is an outlier or not. Looking across the conditions, the impact of filtering out genes leads to a different set of genes being measured per condition. On average, there are 93 out of 4267 genes that are different between each experimental run. Of the 93, there is an average of 67 unique genes or 1.6% of the transcriptome that are different between the experiments. Namely, these genes will have different responses given the day the experiment was run and thus place an upper bound on the repeatability of the model.

Gene quantitative response. (Figure 5C) The next question to address is if the response levels of the genes (differential expression) are consistent across experimental runs. Response is measured with a quasi-likelihood negative binomial generalized log-linear model that outputs an FC as compared to a control. As mentioned in (2), the black and red lines are the genes that were filtered out as outliers by the dispersion model for a single experimental run. Beyond the different set of genes, it is clear that the largest variabilities in response between the experimental runs are genes that have lower expression in both runs (those closer to the origin). Genes with low expression are inherently unreliable through traditional processing and analysis pipelines (24). These are genes that would not pass a threshold for subsequent analysis in either experimental run and so their responses are statistically insignificant. There are, however, 44 unique genes whose response falls above the noise threshold in both experiments but are outside the 95% confidence bounds of the experiment. Repeatability of a machine learning model with common metrics such as R^2 should not take these genes into account as they fall below a noise threshold.

If a gene is differentially expressed. (Figure 5D) In some applications, such as pathway analysis, the quantitative response of a gene to a perturbation is not as important as whether or not the gene was dysregulated. In such instances, genes are labeled as DEGs if they satisfy magnitude (often in terms of log FC) and statistical significance (FDR) thresholds. DEGs are fed to downstream enrichment and pathway analysis tools to identify mechanistic and functional changes. Thus, evaluating whether the set of DEGs remains consistent is also an important aspect of reproducibility. Classification methods that are trained to predict if a gene is

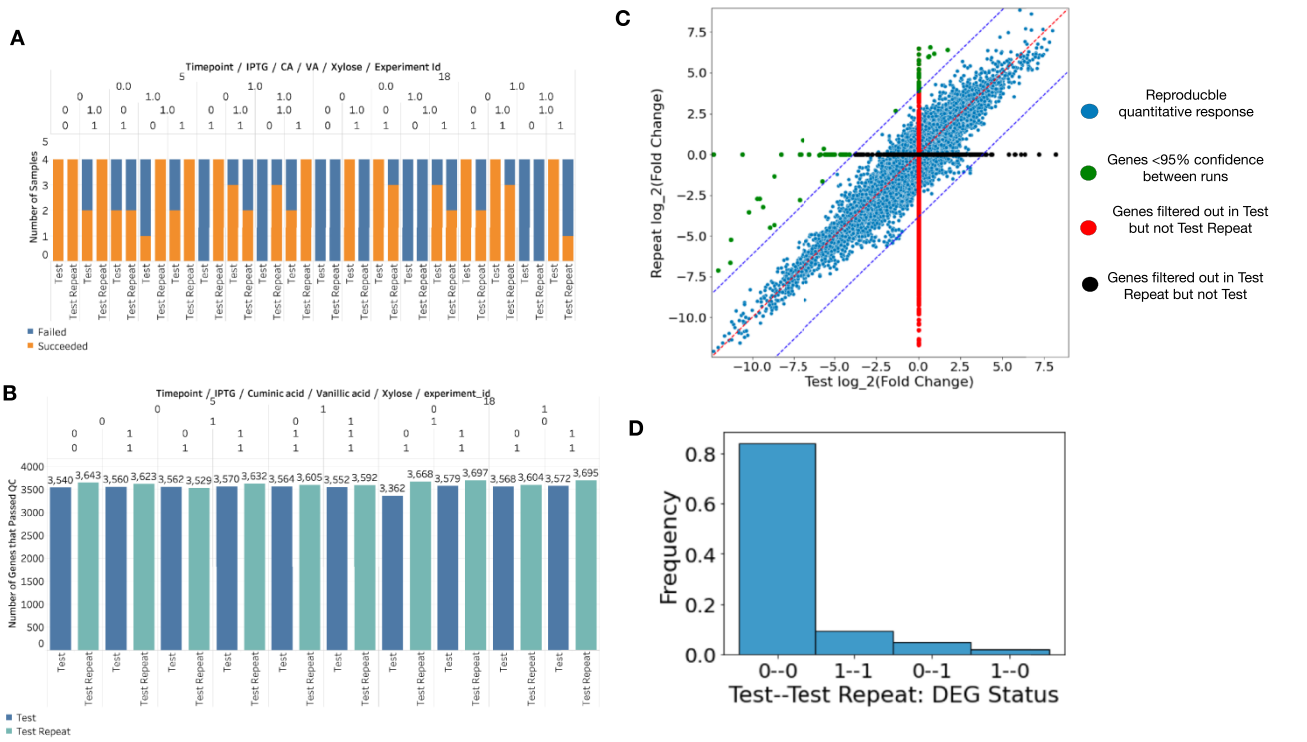


Figure 5. Comparison of experimental conditions across test and test repeat experimental runs. The 0/1 indicates the absence/presence of inducers for Panels A/B and differentially expressed status (0 not DEG and 1 is DEG for D). Each comparison will have a different impact on the experiment's ability to validate a machine learning model trained on single inducers (conditions not shown here). (A) Different sample dropouts for the two runs mean that if both experiments were not run, some predictions would not be validatable. (B) Different genes being filtered will impact the set of genes from a condition that can be validated. (C) Genes can have quantitatively different responses, which can further add complications to validation and the gene dropouts (horizontal black and vertical red lines) across the two experimental runs means different genes will be validated. (D) While the set of DEGs that are different between days are in the minority, these discrepancies can have mechanistic consequences on the inferences made.

differentially expressed should account for both class imbalance and the inconsistencies between runs. In our case, we see that the majority of genes across the experimental runs are consistently labeled as impacted (1) or not impacted (0). There are, on average, 140 genes, or 3% of the transcriptome that is labeled differently (labeled as 0–1 or 1–0 in the bar plot). If these genes are essential genes, the set of mechanistic insights that are predicted by the model can be very different based on the experimental run used for validation.

4. Conclusion

Reproducibility is a cornerstone for utilizing RNASeq data to analyze and improve synthetic designs. One important aspect of reproducible designs is computation pipelines that help standardize files, metadata, analysis techniques and protocols in a high-throughput and easy to use environment. As such, we have developed and released both a secondary analysis pipeline and omics_tools to help facilitate analysis of synthetic biology RNASeq experiments. We have also utilized these two to better understand reproducibility in two different contexts: similarity of controls run on different days and the impact that reproducibility has on the test and training data utilized in machine learning techniques. We identified that the experimental protocols utilized for *E. coli* MG1655 produced highly reproducible results. However, when those same protocols were applied to *B. subtilis*, the reproducibility decreased in a manner that suggested the growth states (or some other biological phenomena) were not the same on different days. Given that we standardized the analysis through the analytical pipeline, any differences we observe are guaranteed to be biological or experimental in nature.

Additionally, when exploring induction conditions, we found that while the majority of induction conditions were reproducible across the different days/extraction volumes, some were different. The differences have an impact on the ability to validate a model to predict response to inducers if you only have one run to choose from (i.e. you were only able to run the experiment once). While the majority of genes were consistently present and consistently differentially expressed across the two days/volumes, some genes were not. Such differences limit the upper bound of accuracy of the model, as areas where the test data are inconsistent produce a response distribution versus a single response. A standardized analytical pipeline will enable researchers to identify areas of low reproducibility (either experimental or biological) and focus their experiments on reducing the variability of those response distributions. Ultimately, computation pipelines that help facilitate reproducibility will allow for more consistent analysis of the data.

Supplementary data

Supplementary data are available at SYNBIO Online.

Data availability

The data underlying this article and all subsequent analyses are publicly available on Figshare: https://figshare.com/projects/RNASeq_Reproducibility_for_Synthetic_Biology/125005. The secondary analysis pipeline can be found at https://github.com/

SD2E/rna-seq-pipeline, and the configurable differential expression analysis tools and tutorials are available at https://github.com/SD2E/omics_tools.

Disclaimer

Any views, opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Defense Advanced Research Projects Agency, the Department of Defense or the US Government.

Funding

Defense Advanced Research Projects Agency and the Air Force Research Laboratory [FA8750-17-C-0231] (related contracts by SD2 Publication Consortium Members).

Acknowledgments

The authors would like to thank the Synergistic Discovery and Design Infrastructure team for their support in providing technology to enable large-scale execution, collection, representation and processing of data. The authors would also like to specifically thank Mark Weston as the system integrator of Synergistic Discovery and Design project, Joe Stubbs who contributed a significant amount of work for the high-performance computing infrastructure and finally, Jacob Beal and Nicholas Roehner for their support in the representation of metadata.

Conflict of interest statement. None declared.

References

1. Abbas-Aghababazadeh,F., Li,Q. and Fridley,B.L. (2018) Comparison of normalization approaches for gene expression studies completed with high-throughput sequencing. *PLoS One*, **13**, e0206312. 10.1371/journal.pone.0206312.
2. Babraham Bioinformatics – FastQC: A quality control tool for high throughput sequence data. https://www.bioinformatics.babraham.ac.uk/projects/fastqc/ (8 August 2004, date last accessed).
3. Bolger,A.M., Lohse,M. and Usadel,B. (2014) Trimmomatic: a flexible trimmer for Illumina sequence data. *Bioinformatics*, **30**, 2114–2120. 10.1093/bioinformatics/btu170.
4. Picard Toolkit. (2019) Broad Institute, GitHub Repository. https://broadinstitute.github.io/picard/.
5. Brookes,E. and Stubbs,J. (2019) GenApp, containers and Abaco: technical paper. In: *Proceedings of the Practice and Experience in Advanced Research Computing on Rise of the Machines (Learning)*. Association for Computing Machinery, New York, NY, USA, pp. 1–8.
6. Bryce,D., Goldman,R.P., DeHaven,M. *et al.* (2022) Round trip: an automated pipeline for experimental design, execution, and analysis. *ACS Synth. Biol.*, **11**, 608–622. 10.1021/acssynbio.1c00305.
7. Chavez,M., Ho,J. and Tan,C. (2017) Reproducibility of high-throughput plate-reader experiments in synthetic biology. *ACS Synth. Biol.*, **6**, 375–380. 10.1021/acssynbio.6b00198.
8. Cleveland,S.B., Jamthe,A., Padhy,S. *et al.* (2020) Tapis API Development with Python: Best Practices In Scientific REST API Implementation: experience implementing a distributed Stream API. In: *Practice and Experience in Advanced Research Computing*. Association for Computing Machinery, New York, NY, USA, pp. 181–187.
9. Conesa,A., Madrigal,P., Tarazona,S. *et al.* (2016) A survey of best practices for RNA-seq data analysis. *Genome Biol.*, **17**, 13. 10.1186/s13059-016-0881-8.
10. Corchete,L.A. *et al.* (2020) Systematic comparison and assessment of RNA-seq procedures for gene expression quantitative analysis. *Sci. Rep.*, **10**, 19737.
11. Costa-Silva,J., Domingues,D. and Lopes,F.M. (2017) RNA-Seq differential expression analysis: an extended review and a software tool. *PLoS One*, **12**, e0190152. 10.1371/journal.pone.0190152.
12. Eslami,M., Espah Borujeni,A., Eramian,H. *et al.* (2021) Prediction of whole-cell transcriptional response with machine learning. *Bioinformatics*, **38**, 404–409. 10.1101/2021.04.30.442142.
13. Espah Borujeni,A., Zhang,J., Doosthosseini,H. *et al.* (2020) Genetic circuit characterization by inferring RNA polymerase movement and ribosome usage. *Nat. Commun.*, **11**, 5001. 10.1038/s41467-020-18630-2.
14. Finotello,F. and Di Camillo,B. (2015) Measuring differential gene expression with RNA-seq: challenges and strategies for data analysis. *Brief Funct. Genomics*, **14**, 130–142. 10.1093/bfgp/elu035.
15. Garrido-Rodriguez,M., Lopez-Lopez,D., Ortuno,F.M. *et al.* (2021) A versatile workflow to integrate RNA-seq genomic and transcriptomic data into mechanistic models of signaling pathways. *PLoS Comput. Biol.*, **17**, e1008748. 10.1371/journal.pcbi.1008748.
16. Gonçalves,R.S. and Musen,M.A. (2019) The variable quality of metadata about biological samples used in biomedical experiments. *Sci Data*, **6**, 190021. 10.1038/sdata.2019.21.
17. Gorochowski,T.E., Espah Borujeni,A., Park,Y. *et al.* (2017) Genetic circuit characterization and debugging using RNA-seq. *Mol. Syst. Biol.*, **13**, 952. 10.15252/msb.20167461.
18. Hazen,T.H., Daugherty,S.C., Shetty,A. *et al.* (2015) RNA-Seq analysis of isolate- and growth phase-specific differences in the global transcriptomes of enteropathogenic *Escherichia coli* prototype isolates. *Front. Microbiol.*, **6**, 569. 10.3389/fmicb.2015.00569.
19. Intosalmi,J., Nousiainen,K., Ahlfors,H. *et al.* (2016) Data-driven mechanistic analysis method to reveal dynamically evolving regulatory networks. *Bioinformatics*, **32**, i288–i296. 10.1093/bioinformatics/btw274.
20. Jackman,S.D., Mozgacheva,T., Chen,S. *et al.* (2019) ORCA: a comprehensive bioinformatics container environment for education and research. *Bioinformatics*, **35**, 4448–4450. 10.1093/bioinformatics/btz278.
21. Jessop-Fabre,M.M. and Sonnenschein,N. (2019) Improving reproducibility in synthetic biology. *Front. Bioeng. Biotechnol.*, **7**, 18. 10.3389/fbioe.2019.00018.
22. Kc,K., Li,R., Cui,F. *et al.* (2019) GNE: a deep learning framework for gene network inference by aggregating biological information. *BMC Syst. Biol.*, **13**, 38. 10.1186/s12918-019-0694-y.
23. Łabaj,P.P. and Kreil,D.P. (2016) Sensitivity, specificity, and reproducibility of RNA-Seq differential expression calls. *Biol. Direct*, **11**, 66. 10.1186/s13062-016-0169-7.
24. Lamarre,S., Frasse,P., Zouine,M. *et al.* (2018) Optimization of an RNA-Seq differential gene expression analysis depending on biological replicate number and library size. *Front. Plant Sci.*, **9**, 108. 10.3389/fpls.2018.00108.
25. Langfelder,P. and Horvath,S. (2008) WGCNA: an R package for weighted correlation network analysis. *BMC Bioinform.*, **9**, 559. 10.1186/1471-2105-9-559.
26. Lazic,S.E. (2016) *Experimental Design for Laboratory Biologists: Maximising Information and Improving Reproducibility*. Cambridge University Press, Cambridge, UK.
27. Li,H. and Durbin,R. (2009). Fast and accurate short read alignment with Burrows-Wheeler Transform. *Bioinformatics.*, **25**, 1754–1760.

28. Li,Q., Brown,J.B., Huang,H. *et al.* (2011) Measuring reproducibility of high-throughput experiments. *Ann. Appl. Stat.*, **5**, 1752–1779. 10.1214/11-AOAS466.
29. McLaughlin,J.A., Myers,C.J., Zundel,Z. *et al.* (2018) SynBioHub: a standards-enabled design repository for synthetic biology. *ACS Synth. Biol.*, **7**, 682–688. 10.1021/acssynbio.7b00403.
30. Musich,R. *et al.* (2021) Comparison of Short-Read Sequence Aligners Indicates Strengths and Weaknesses for Biologists to Consider. *Front. Plant Sci.*, **12**, 657240.
31. Naseri,G. and Koffas,M.A.G. (2020) Application of combinatorial optimization strategies in synthetic biology. *Nat. Commun.*, **11**, 2446. 10.1038/s41467-020-16175-y.
32. Nelson,W., Zitnik,M., Wang,B. *et al.* (2019) To embed or not: network embedding as a paradigm in computational biology. *Front. Genet.*, **10**, 381. 10.3389/fgene.2019.00381.
33. Park Y. (2019) Design and debugging of ultrastable engineered genetic systems. *Ph.D. Thesis*. Massachusetts Institute of Technology.
34. Perkel,J.M. (2019) Make code accessible with these cloud services. *Nature*, **575**, 247–248. 10.1038/d41586-019-03366-x.
35. Petzold,C.J., Chan,L.J.G., Nhan,M. *et al.* (2015) Analytics for metabolic engineering. *Front. Bioeng. Biotechnol.*, **3**, 135. 10.3389/fbioe.2015.00135.
36. Pinoli,P., Ceri,S., Martinenghi,D. *et al.* (2018) Metadata management for scientific databases. *Inf. Syst.*, **81**, 1–20. 10.1016/j.is.2018.10.002.
37. Pobre,V. *et al.* (2020) Prediction of novel non-coding RNAs relevant for the growth of Pseudomonas putida in a bioreactor. *Microbiology (Reading, Engl)*, **166**, 149–156.
38. Reimand,J., Isserlin,R., Voisin,V. *et al.* (2019) Pathway enrichment analysis and visualization of omics data using g:Profiler, GSEA, Cytoscape and EnrichmentMap. *Nat. Protoc.*, **14**, 482–517. 10.1038/s41596-018-0103-9.
39. Robinson,M.D., McCarthy,D.J. and Smyth,G.K. (2010) edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, **26**, 139–140. 10.1093/bioinformatics/btp616.
40. Robinson,M.D. and Oshlack,A. (2010) A scaling normalization method for differential expression analysis of RNA-seq data. *Genome Biol.*, **11**, R25. 10.1186/gb-2010-11-3-r25.
41. Roehner,N., Beal,J., Bartley,B. *et al.* (2021) Data representation in the DARPA SD2 program. *BioRxiv*. 10.1101/2021.09.17.460644.
42. Sastry,A.V., Gao,Y., Szubin,R. *et al.* (2019) The *Escherichia coli* transcriptome mostly consists of independently regulated modules. *Nat. Commun.*, **10**, 5536. 10.1038/s41467-019-13483-w.
43. Taylor,R. and Singhal,M. (2009) Biological network inference and analysis using SEBINI and CABIN. *Methods Mol. Biol.*, **541**, 551–576. 10.1007/978-1-59745-243-4_24.
44. Tiwari,K., Kananathan,S., Roberts,M.G. *et al.* (2021) Reproducibility in systems biology modelling. *Mol. Syst. Biol.*, **17**, e9982. 10.15252/msb.20209982.
45. Wilson,S.L., Way,G.P., Bittremieux,W. *et al.* (2021) Sharing biological data: why, when, and how. *FEBS Lett.*, **595**, 847–863. 10.1002/1873-3468.14067.
46. Yoo,A.B., Jette,M.A. and Grondona,M. (2003) SLURM: simple linux utility for resource management. In: Feitelson D, Rudolph L, Schwiegelshohn U (eds). *Job Scheduling Strategies for Parallel Processing*, Vol. 2862. Springer, Berlin, Heidelberg, pp. 44–60.
47. Yu,L. (2020) RNA-Seq reproducibility assessment of the Sequencing Quality Control project. *Cancer Inform.*, **19**, 1176935120922498. 10.1177/1176935120922498.