RNA-seq transcript quantification from reduced-representation data in `recount2`

January 12, 2018

Jack M. Fu[1,2], Kai Kammers[2,3], Abhinav Nellore[4,5], Leonardo Collado-Torres[6,2], Jeffrey T. Leek[1,2,*], Margaret A. Taub[1,2,*]

[1] Department of Biostatistics, Johns Hopkins University, Baltimore MD, USA
[2] Center for Computational Biology, Johns Hopkins University, Baltimore, MD, USA
[3] Division of Biostatistics and Bioinformatics, Department of Oncology, Sidney Kimmel Comprehensive Cancer Center, Johns Hopkins University School of Medicine, Baltimore, MD, USA
[4] Department of Biomedical Engineering, Oregon Health and Science University, Portland, OR, USA
[5] Department of Surgery, Oregon Health and Science University, Portland, OR, USA
[6] Lieber Institute for Brain Development, Baltimore, MD, USA
[*] To whom correspondence should be addressed: mtaub@jhsph.edu, jtleek@gmail.com

Abstract

More than 70,000 short-read RNA-sequencing samples are publicly available through the `recount2` project, a curated database of summary coverage data. However, no current methods can estimate transcript-level abundances using the reduced-representation information stored in this database. Here we present a linear model utilizing coverage of junctions and subdivided exons to generate transcript abundance estimates of comparable accuracy to those obtained from methods requiring read-level data. Our approach flexibly models bias, produces standard errors, and is easy to refresh given updated annotation. We illustrate our method on simulated and real data and release transcript abundance estimates for the samples in `recount2`.

Keywords: RNA-seq, non-negative least squares, transcript expression

Background

RNA sequencing (RNA-seq) can be used to measure gene (and transcript) expression levels genome-wide. Large-scale RNA-seq datasets have been produced by studies such as the GTEx (Genotype-Tissue Expression) consortium [1], which comprises 9,662 samples from 551 individuals and 54 body sites (under version 6), and the Cancer Genome Atlas (TCGA) study [2], which comprises 11,350 samples from 10,340 individuals and 33 cancer types. Furthermore, public data repositories such as the Sequence Read Archive (SRA) host tens of thousands of human RNA-seq samples [3]. These data collectively provide a rich resource which researchers can use for discovery, validation, replication, or methods development.

These data are even more valuable when processed in a consistent manner and presented in an accessible format. The recently published `recount2` project [4] is the result of such an undertaking. All raw data from the thousands of sequencing studies were aligned to a common reference genome using a scalable and reproducible aligner Rail-RNA [5]. Summary measures (gene, exon, junction, and base-pair level coverage) were derived from the Rail-RNA output and made available in a R package and through a web portal (`https://jhubiostatistics.shinyapps.io/recount/`).

Currently, `recount2` provides summary measures that directly allow for analyses like annotation-agnostic base-pair level and annotation-specific gene/exon/junction differential expression. However, transcript-level abundance estimates are missing from `recount2`, preventing subsequent transcript-level analyses. Despite the existence of many successful transcript quantification programs (such as Cufflinks [6], StringTie [7], Kallisto [8], Salmon [9], and RSEM [10]), this deficiency persists because methods capable of estimating transcript abundances using the summarized output collected in `recount2` do not exist.

Inspired by previous linear models for transcript quantification (such as IsoformEx [11] and MultiSplice [12]), we present our model `recountNNLS`, which does not require the raw sequencing read data, but instead requires only the base-pair and junction level coverages stored in `recount2`. Our approach is useful both for creating transcript abundances specifically for the `recount2` project and as a general purpose method for estimating transcript abundances based on RNA-seq summaries that are smaller and easier to distribute than raw alignment files. Our goal in developing this method was not to be faster or more accurate than existing methods operating on raw sequencing data, but to provide a way to more fully utilize the wealth of data in the `recount2`

project by performing transcript-level abundance estimates.

Results

Overview of method

`recount2` includes a repository of coverage summary measures, including coverage of exon-exon splice junctions, produced by a uniform application of the aligner Rail-RNA to more than 70,000 publicly available RNA-seq samples. For a given read length and a reference transcriptome, we determine a set of sufficient **features** comprised of subdivided exonic segments and exon-exon junctions, such that the coverage of these features adequately summarizes the transcript quantification encoded in the raw reads. Those counts of reads overlapping the features are the sufficient statistics of our linear model, which we denote as **feature counts** (see **Figure 1**). By storing and using only these summary coverages, `recount2` can be viewed as a lossy compression of the raw alignment files, which can achieve greater than 1000x compression for storage and transfer (see **Figure. 2**).

Using these feature counts as the dependent variable, we fit a non-negative least squares regression model to estimate the underlying transcript abundances. The independent variables in our model are transcriptome annotation-specific, denoted as **feature probabilities**. A feature probability encodes the chance that a random read from a transcript will contribute an observed count to the corresponding feature. Our regression weights each feature by its uniqueness within the set of transcripts at a specific locus, in order to emphasize features that distinguish transcripts. Standard error estimates are also reported that reflect our model's confidence in abundance assignment. Lastly, updates to transcriptome annotations can be efficiently incorporated by our model without requiring re-alignment. Further details about our methods are described in Methods and are implemented in the R package `recountNNLS`.

Performance on simulated data

Using simulated data, we evaluated the performance of our model compared to the popular pipelines HISAT2-Cufflinks [13]-[6], HISAT2-StringTie [7], Kallisto [8], Salmon [9], and RSEM [10]. We simulated 10 scenarios of varying read-length and paired-end status using the `polyester`

R package [14]. Our method was run on the reduced-representation output from applying the aligner Rail-RNA [5] to the simulated FASTA files. All other methods extracted information from the full simulated FASTA files. For our main metric of performance, we calculated the Root Mean Squared Error (RMSE) of the estimated versus true number of reads on the transcript and gene levels. **Table 1** presents the RMSE of each simulated scenario with all 6 methods evaluated. **Figure 3** is a collection of MA (mean-difference) plots of estimate vs truth for each of the 6 methods in the 150bp single-end simulation. It is representative of the visualization of the other scenarios, the rest of which are included in the supplement (**Supplementary Figures 1-9**).

Visual inspection reveals that all methods appear to have minimal errors in quantifying gene level expression, while transcript abundance estimation appears to be more variable. Although our linear model is not the best as judged by RMSE, we approximate other methods closely, considering we do not use the raw reads/alignments as input but limit ourselves to a reduced-representation summary as would be found in `recount2`. Furthermore, although the RMSEs of the other methods remain relatively stable as read length of the simulation scenarios increases, our linear model improves steadily with increasing read length. Our method's RMSE decreased by almost 30% as the read length of the scenarios increased from 37bp to 150bp. The likely driver of these improvements is the increased identifiability of transcripts with increasing read length. As read lengths increase, our method's performance should continue to improve.

Performance on GEUVADIS Consortium data

Using `recount2` feature counts of the 667 samples from the GEUVADIS dataset [15], we ran `recountNNLS` to estimate transcript abundance levels. We also downloaded the FASTQ files for the 667 samples and applied Kallisto [8], HISAT2-StringTie [7], HISAT2-Cufflinks [13]-[6], RSEM [10], and Salmon [9] to estimate transcript abundances. Pair-wise comparisons of the estimates were carried out, with the comparisons for sample ERR188412 of the consortium presented in **Table 2**.

For sample ERR188412, the Spearman correlations between Salmon, Kallisto, and HISAT2-Cufflinks are high at greater than 0.85. `recountNNLS` and HISAT2-StringTie both achieve correlation of above 0.7 with other methods, but achieve only 0.62 correlation with each other. The lower triangle of **Table 2** presents the number of transcripts that the corresponding methods both

assigned non-zero expression to. RSEM assigned non-zero expression to the lowest number of transcripts at 68,084 transcripts while our method assigned it to the most at 81,733.

We were also able to indirectly compare our method to IsoformEx [11], as the authors of IsoformEx had evaluated pair-wise correlations between IsoformEx, Cufflinks, RSEM, Kallisto, and Salmon. Using the testing data in IsoformEx, the authors reported almost identical pairwise correlations between RSEM, Salmon, and Cufflinks to the correlation we observe using the GEUVADIS dataset. IsoformEx achieved correlations of between 0.62 and 0.67 with Cufflinks, RSEM, and Kallisto, while our method achieved correlations between 0.71 and 0.74 for the same set of methods. The correlation of Salmon with the other methods tested in IsoformEx increased substantially in our evaluation compared to their publication [11]. This is likely due to the refinements in the algorithm made to Salmon between 2011 and now.

Discussion

We have presented here the first method to provide transcript-level abundance estimates on the reduced-representation expression data available in `recount2`. While our goal in developing this method was not to produce one that is "better" than existing transcript quantification methods that operate on raw data, our method produces results that are comparable, in terms of accuracy, to the other leading methods. In addition, our method has some additional features that do distinguish it from existing methods. Firstly, our model produces standard errors of our quantified estimates. Secondly, we provide the transcript quantification for more than 70,000 samplefgys processed in a uniform manner that is easily accessible for downstream analyses. Lastly, our method's estimates are easy to update in light of evolving transcriptome annotations.

Many loci have annotated transcripts that are structurally very similar. Unsurprisingly, expression levels for transcripts that are highly similar are difficult to tease apart. Our linear model's standard error estimates are able to systematically reflect the identifiability of the transcripts. In **Figure 4**, we present the structure and estimates of 2 selected genes from sample ERR188412 of the GEUVADIS dataset. For the gene KLHL17, all 5 transcripts have unique features that make the transcripts highly distinct. Our method shows that the standard errors are well controlled (see **Figure 4** A). In the gene B4GALT2, there are strong structural similarities between some of the 8 annotated transcripts. The difference in the identifiability of the transcripts is clearly

reflected in our reported standard errors: transcripts that are difficult to distinguish from others are assigned significantly higher standard errors (see **Figure 4** B).

Working with the transcript abundances produced by our method is very straightforward. For a given SRA project id (`x`) currently curated in `recount2`, one can access the transcript quantification stored as a RSE object by installing the `recountNNLS` R package and calling a single function, `getRseTx(x)`. We also include an example differential transcript expression analysis of healthy versus cancer TCGA breast samples in the Supplementary Materials. We input our model estimates into a popular differential expression pipeline with R packages `limma` [16] and `edgeR` [17] to produce estimates of transcript-level differential expression between these groups of samples.

Our model is readily able to adjust for factors that might affect quantification (such as GC content, mappability, and transcript location bias) by adjusting the feature probability matrices. For example, to adjust for GC content, we could learn the GC content bias of the sample by selecting for the subset of 1-transcript genes and assessing GC bias using their sequence composition and expression levels. The selected transcripts can be broken down into the set of features and feature counts that they are comprised of. Using a loess smoother, one could model the relationship between the GC content of those features and the feature counts. This relationship could then be used on multi-transcript genes to up-weight or down-weight the feature probability matrices. Substituting the adjusted matrices into NNLS estimation would yield GC-adjusted estimates. Similar processes can be carried out for any kind of adjustment for which one could attain feature-level characteristics, such as mappability, positional biases, etc.

Similarly, our model is able to rapidly and efficiently respond to the constant evolution of transcriptome annotations. By modifying the set of features, feature counts, and feature probability matrices to match the new annotation, our model simply reuses the annotation-agnostic Rail-RNA alignment summaries in `recount2` without requiring re-alignment. Furthermore, should a new annotation affect only certain loci, only the summary measures pertaining to those loci need to be re-quantified.

Conclusion

We have presented a method capable of estimating transcript-level abundances and their standard errors on a set of reduced representation data on more than 70,000 RNA-seq samples in

recount2. Our method performs comparably to methods that access raw read data as input. The benefits of our method are: (1) maximal availability of samples for analysis and comparison; (2) re-computation in response to changing annotation without re-alignment; (3) standard errors informing which transcripts are hard to distinguish; and (4) reduced storage and re-computation burden. The quantified transcript abundances for the more than 70,000 samples of recount2 are now available for direct access and download.

Methods

recount2 summary measures

recount2 includes a repository of coverage summary measures produced by a uniform application of the aligner rail to more than 70,000 publicly available RNA-seq samples. In particular for each sample, recount2 contains two primary files necessary for our linear modeling approach. First, each sample has a BigWig-format file [18] containing the number of reads that overlapped each genomic position of the entire GRCh38 assembly. Secondly, each sample has a file containing the number of reads spanning observed exon-exon splice junctions. Other useful summarizations are also available directly from recount2, like precomputed exon-level and gene-level coverages based upon the GencodeV25 reference transcriptome using the above mentioned BigWig files. As mentioned above, by storing only these summary coverages, recount2 achieves greater than 1000x compression compared to raw FASTQ and aligned BAM files (see **Figure 2**).

Sufficient statistics for transcript quantification

Given the read length of a particular experiment and a reference transcriptome, we determine a set of sufficient **features** such that the coverage of these features adequately summarizes the transcript quantification encoded in the raw reads. For simplicity, we illustrate our definition of features with an example gene containing two transcripts, and with an example data-generating experiment with read lengths of 100 base pairs, but our method generalizes to arbitrary transcript structure and different read lengths.

Consider the gene portrayed in **Figure 1**, which is composed of 2 transcripts, 3 distinct exons, and 1 exon-exon junction, and suppose that the experiment produces reads of length 100bp.

We first disjoin the annotation into unique, non-overlapping sub-exonic segments, similar to the scheme that IsoformEx [11] employs. However, in a process unique to our model, any bins longer than 200bp (twice the experiment read length) are then further evenly subdivided so that the largest resulting piece is less than 100bp. This process further increases the identifiability of the transcripts. For our example, the final product is a set of 7 features, of which 6 are sub-exonic segments while 1 is an exon-exon splice junction.

The sufficient statistics for our linear model are the counts of reads that overlap each feature, which we will denote as **feature counts**. To extract the feature counts given a set of features, we query the BigWig files for the coverage of sub-exonic sections, and the junction file for the junction equivalent. The values in the BigWig files are stored as the number of overlapping base pairs and must be divided by the read-length of the experiment to determine the equivalent number of reads. No such normalization is necessary for the values from the junction coverage file.

Deriving model inputs

Our goal is to derive transcript abundances given the known gene structure and feature counts summarized above. Continuing with the example gene from **Figure 1**, Transcript 1 is composed of all 7 features, while Transcript 2 is composed of features 1 and 2 only. Based on this structure, reads aligning to features 3-7 should have originated from Transcript 1 and not Transcript 2. We use this structure to set up the design matrix for our linear model.

We name our independent variables **feature probability** vectors, and denote them as X^1 and X^2 respectively for each transcript. Each vector is of length 7, corresponding to the number of features. Each element X^k_j encodes the probability a random read from transcript k overlaps feature j of our gene, where $k = 1 \dots 2$ and $j = 1 \dots 7$. The column-wise collection of feature probability vectors for our example gene is denoted as $\mathbf{X}$ with dimension [7x2] and is referred to as the **feature probability matrix** for this gene. Note that the values in this matrix depend on both the calculated features and the length of the reads of the sequencing experiment (100bp in this example).

The true $\mathbf{X}$ is not known, but it can be estimated based on sequence content of the transcripts and the read length of the experiment. To estimate X^1, sliding segments of 100bp from Transcript 1 are aligned to the GRCh38 reference using the aligner HISAT2 [13]. The number of aligned

segments overlapping each feature is summed and divided by the number of total 100bp segments to produce the estimate of X^1, denoted by $\hat{X}^1$. The estimated feature probability matrix $\hat{\mathbf{X}}$ is the column-wise collection of such estimated feature probability vectors for all transcripts in the gene. More complex implementations can readily include adjustments for GC content, 5' bias, and mappability differences by weighting each row of $\hat{\mathbf{X}}$ appropriately.

Non-negative linear model

For our gene, we denote the observed feature counts vector Y. The underlying assumed data generation process is illustrated in **Figure 5**. We are interested in estimating the true transcript abundances β using our estimated $\hat{\mathbf{X}}$ by solving for:

$$\arg\min_{\hat{\beta}\geq 0} \left\| \mathbf{P}(Y - \hat{\mathbf{X}}\hat{\beta}') \right\|_2$$

subject to the constraint that each element of $\hat{\beta}$ is non-negative. $\mathbf{P}$ is a diagonal weight matrix where the (k,k)-th element is the inverse of the number of entries of the k-th row of $\hat{\mathbf{X}}$ that are non-zero. $\mathbf{P}$ helps weight each feature's contribution to the norm by its uniqueness amongst the transcripts. A feature that has non-zero feature probabilities from many transcripts is down-weighted in comparison to a feature that has only 1 transcript with a non-zero feature probability entry.

Existing non-negative least squares (NNLS) algorithms can quickly optimize the solution using $\mathbf{P}Y$ as the dependent variable and $\mathbf{P}\hat{\mathbf{X}}$ as the independent variables:

$$\hat{\beta} = NLLS(\mathbf{P}Y, \mathbf{P}\hat{\mathbf{X}})$$

In particular for `recountNNLS`, we used the function found in the R package `nnls` [19]

Standard error calculation

Our model also produces an estimate of the standard error of quantification based upon previous work in constrained least-squares estimation by Liew [20]. NNLS is a special case of the generalized framework presented by Liew. Adapting Liew's results to our annotation for a given gene yields:

$$\mathbf{P}Y = \mathbf{P}\hat{\mathbf{X}}\hat{\beta} + \epsilon$$

subject to the constraint:

$$\mathbf{A}_f\hat{\beta} \geq 0$$

where $\mathbf{A}_f$ is an fxf identity matrix where f is the number of the features of the gene.

Let $\tilde{\beta} = (\hat{\mathbf{X}}'\hat{\mathbf{X}})^{-1}\hat{\mathbf{X}}'Y$, the ordinary least-squares solution to the same input. Liew's results show:

$$V(\hat{\beta}) = \mathbf{M}V(\tilde{\beta})\mathbf{M}'$$

where $\mathbf{M}$ simplifies to the following under our $\mathbf{A}_f$:

$$\mathbf{M} = (\mathbf{I} + (\hat{\beta} - \tilde{\beta})\mathbf{A}_f)$$

and yields for each transcript k of our gene:

$$V(\hat{\beta}_k) = (1 + (\hat{\beta}_k - \tilde{\beta}_k))^2 V(\tilde{\beta}_k)$$

Quantification compilation for `recount2`

To quantify the entire transcriptome for a given sample, we execute the feature probability matrix estimation and linear modeling one locus at a time. We define a locus as all genes that are overlapping, irrespective of the strandedness in the annotation. As an addition to the `recountNNLS` package, we offer precomputed features and feature probability matrices reflecting read lengths of 37, 50, 75, 100, and 150bp in the package `recountNNLSdata`.

Our method produces a Ranged Summarized Experiment (RSE) object per project - mirroring the structure of `recount2`. For each sample of a project, we utilize the set of feature probability matrices that match the read length of the sample most closely. If the match is not exact, we adjust the estimated abundances and standard errors by the ratio of feature probability matrix read length over actual sample read length.

For each project, the RSE contains the estimated quantification and the standard errors under the accessor function `assays` as `counts` and `se` respectively. Each row of the `counts` and `se` matrices represents a transcript, and each column represents a sample. The corresponding transcripts are stored as a `GRangesList` accessible via the `rowRanges` function. Transcripts that introduce

colinearity (either perfect or computational) in the model matrix $\hat{\mathbf{X}}$ are reported as `NA` in both `counts` and `se`.

Performance Evaluation

Using our linear model on real and simulated data, we compare our estimates to those from the established methods Kallisto [8], StringTie [7], Cufflinks [13], RSEM [10], and Salmon [9].

Simulation scenarios

We simulated RNA-seq data using the R package `polyester` [14] under 10 scenarios: read lengths of 37, 50, 75, 100 and 150bp with either single-end or paired-end FASTA reads. For the sake of simulation expediency, we selected all coding transcripts from chr1 and chr14 from the GencodeV25 transcriptome annotation, which comprises 12.5% of the entire annotation. The number of simulated reads for the same transcript across scenarios is identical and generated via `polyester` [14] with fragment length distribution Gaussian with mean 250 and standard deviation 25. The number of reads to simulate was determined on a gene-by-gene basis, with most genes having a dominant transcript that produces over 50% of the sequencing reads. The relative abundances of the transcripts are chosen via a Dirichlet distribution with $\alpha = 1/f$, where f is the number of transcripts in the gene. The total number of reads at a gene is chosen as a Negative Binomial with size=4 and p=0.01. The number of reads of each transcript is the product of the outcomes of the Dirichlet and the Negative Binomial.

We created alignment indices for our subset of the transcriptome for use with Kallisto [8], HISAT2 [13], and Salmon [9] using default recommended procedures. Subsequently, the simulated FASTA files were fed to Kallisto [8], HISAT2-StringTie [7], HISAT2-Cufflinks [13]-[6], RSEM [10], and Salmon [9] with default parameters where applicable. Methods were only asked to quantify the abundances of the selected set of transcripts. For single end simulations, Cufflinks [6] and Kallisto [8] require input of the fragment length distribution, for which the true parameters of (250, 25) were used. For our linear model, we utilized Rail-RNA [5] to process the FASTA files in the same manner as in `recount2` [4]. For evaluation, each method's abundance estimates were compared to the true number of reads on the $\log_2$ scale at both the transcript and the gene-level using Root Mean Squared Error (RMSE). The output of StringTie was transformed from the

FPKM to the $\log_2$ read count scale.

GEUVADIS Consortium

For 667 experiments of the GEUVADIS consortium sequencing project, we downloaded the raw paired-end FASTQ files of each of the experiments. The FASTQ files were used directly as input for Kallisto [8], HISAT2-StringTie [7], HISAT2-Cufflinks [13]-[6], RSEM [10], and Salmon [9] using default parameters. The `recount2` summary measures for the GEUVADIS project samples were used as inputs for our linear model. We were only interested in estimating the abundances of the coding transcripts of the GencodeV25 annotation.

Abundance estimations on the transcript and gene-level were compared pair-wise between methods using Spearman's correlation. We also examine pair-wise the number of transcripts that were assigned non-zero expression by both methods.

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Availability of data and material

The following code will reproduce the analyses presented in this project (if R has access to sufficient resources) for a given project id. An example case is demonstrated below for project DRP000366, and additional commands are located in the supplement.

```
install_github('JMF47/recountNNLSdata', ref='5d5283e')
install_github('JMF47/recountNNLS', ref='56901b3')


library(recountNNLS)
```

```
pheno = processPheno('DRP000366')
rse_tx = recountNNLS(pheno)
```

Competing interests

The authors have no competing interests.

Funding

This work was supported by the National Institutes of Health [Grant number R01 GM105705 05].

Authors' contributions

JTL, JMF, and MAT envisioned the objectives of the study. JMF derived the model with input from all other authors. JMF implemented the R package with contribution from LC. JMF, JTL, and MAT wrote the manuscript. All authors read and approved the manuscript.

Acknowledgements

We would like to thank SciServer for hosting the recount2 files. SciServer is being developed at, and administered by, the Institute for Data Intensive Engineering and Science at Johns Hopkins University and is funded by the National Science Foundation Award ACI-1261715. For more information about SciServer, visit http://www.sciserver.org/.

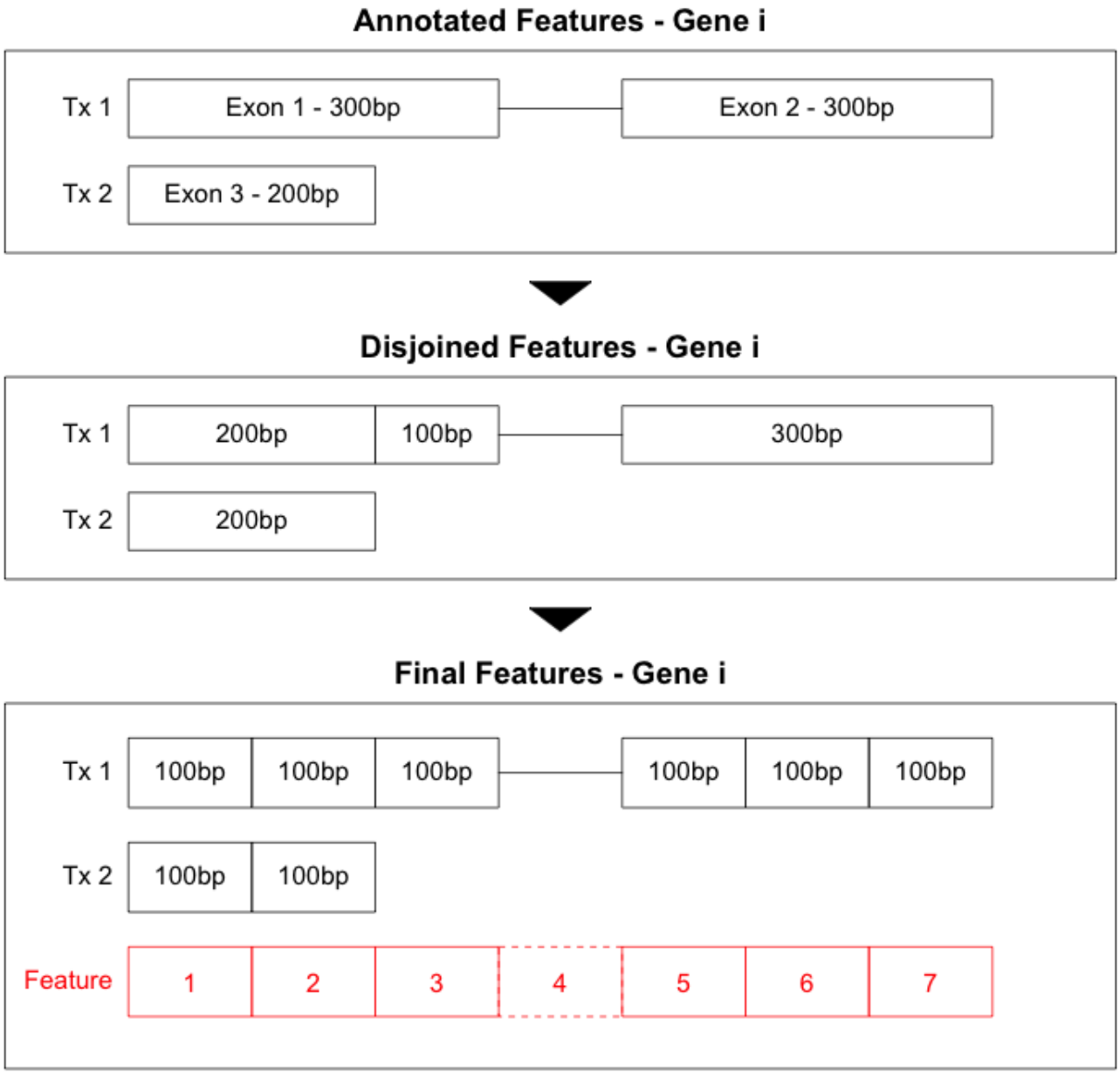

Figure 1: Given the read length of a particular experiment and a reference transcriptome, we determine a set of sufficient **features** such that the coverage of these features adequately summarizes the transcript quantification encoded in the raw reads. This figure illustrates the process to determine the set of features for a mock gene with 2 transcripts and 100bp reads. We first disjoin the annotated exons into non-overlapping bins. Any remaining exonic segments longer than twice the read length are further evenly subdivided to be below 100bp. Each unique splice junction is included without modification as a feature. The number of reads overlapping the final set of features are the sufficient statistics for our linear model and serve as a compression of the raw read-level data.

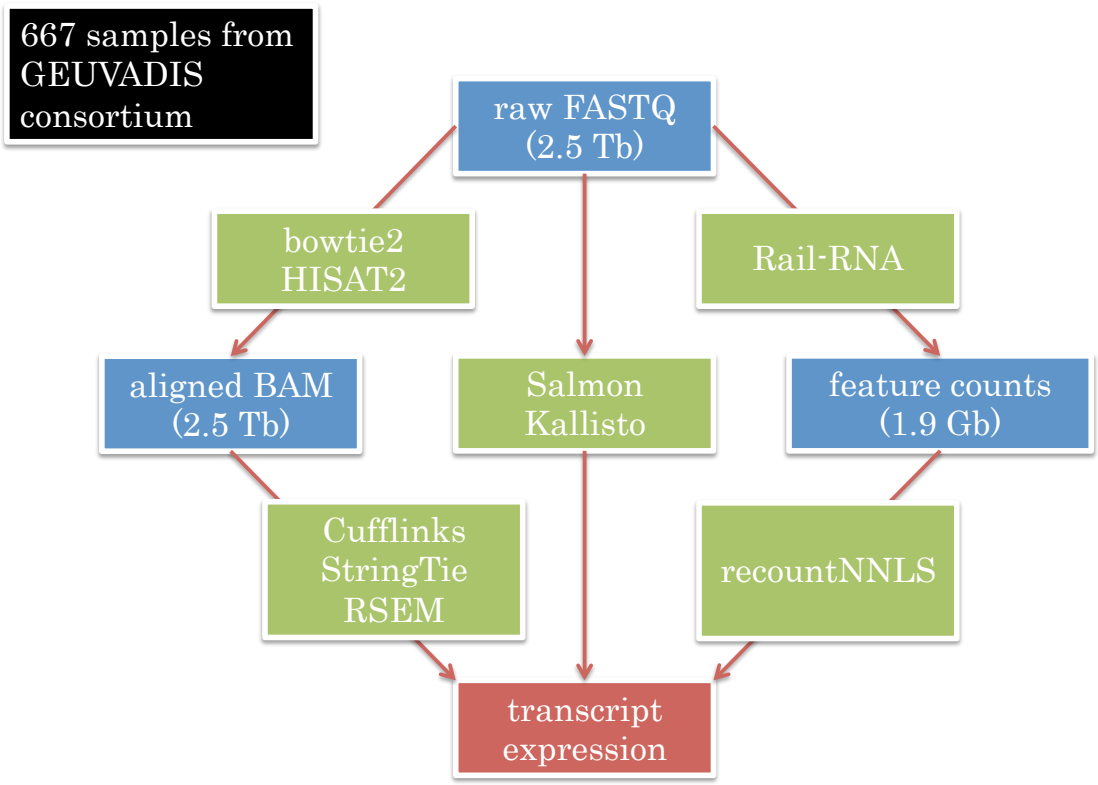

Figure 2: A flowchart illustrating the transcript quantification process of the 667 samples from the GEUVADIS Consortium dataset we used to evaluate the methods in this analysis. Methods are highlighted in green, and the final product in red. Data formats are highlighted in blue, with size statistics. The raw FASTQ files require 2.5Tb of storage and are used directly by k-mer counting methods like Salmon and Kallisto. Cufflinks, StringTie, and RSEM utilize output from splice-read aligners requiring roughly 2.2Tb of space in the BAM format. Our linear model uses the feature counts refined from the Rail-RNA aligner outputs and require only 1.9Gb of space.

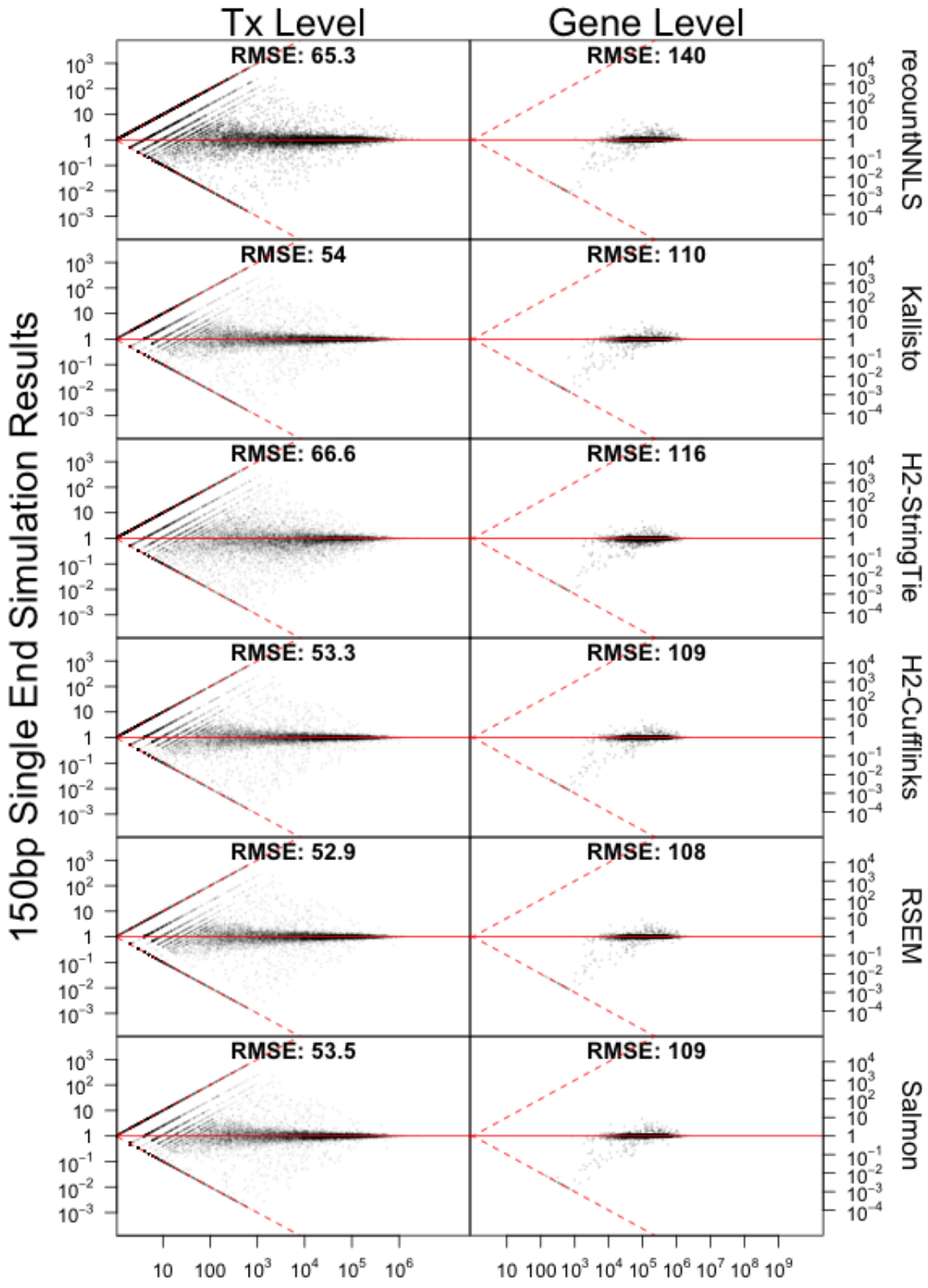

Figure 3: A representative figure of the performance of our method compared to Kallisto, HISAT2-StringTie, HISAT2-Cufflinks, RSEM and Salmon. The left column shows MA plots of the transcript (Tx) level quantification under the scenario of 150bp single end simulated reads. The Y axis represents the difference between estimated and true counts on the $\log_2$ reads scale, while the X axis represents the average of the estimated and true counts on the same scale. The right column shows MA plots of the same at the gene level.

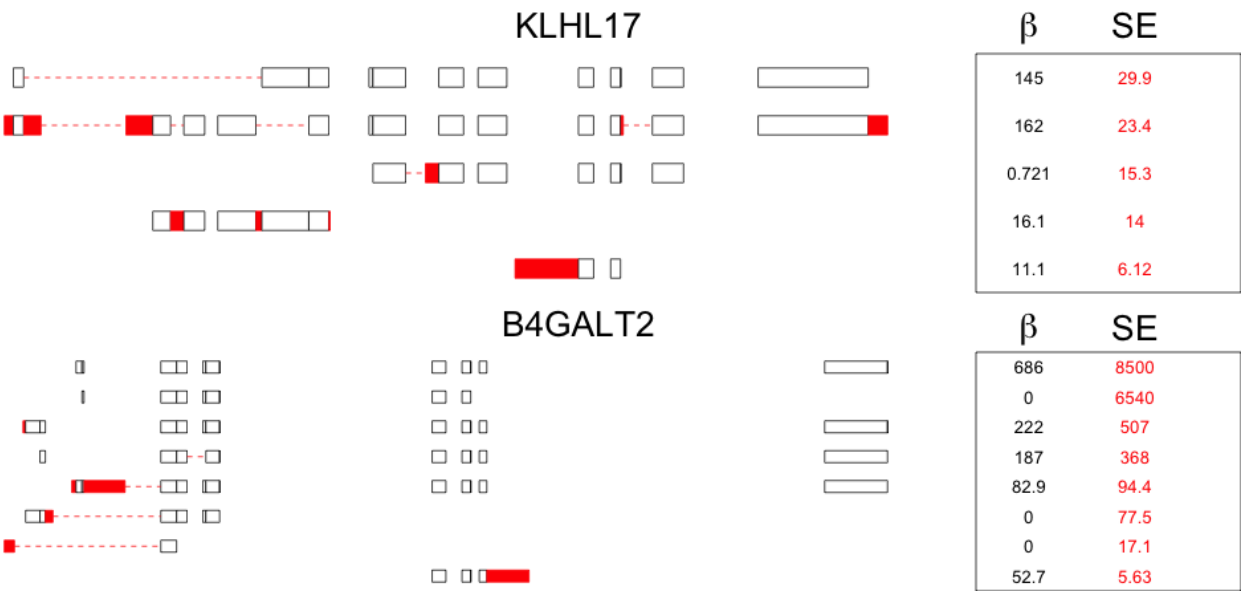

Figure 4: The gene on top (KLHL17) is comprised of 5 transcripts, while the gene on the bottom (B4GALT2) consists of 8 transcripts. For each panel, the transcript structure is shown on the left, with unique features highlighted in red. On the right side of each subplot are the estimated abundances in black and the standard errors of these estimates in red, on the scale of number of reads. The transcripts are ordered by decreasing standard error from top to bottom. The higher estimated standard errors for B4GALT2 reflect how much similar sequence its transcript variants have.

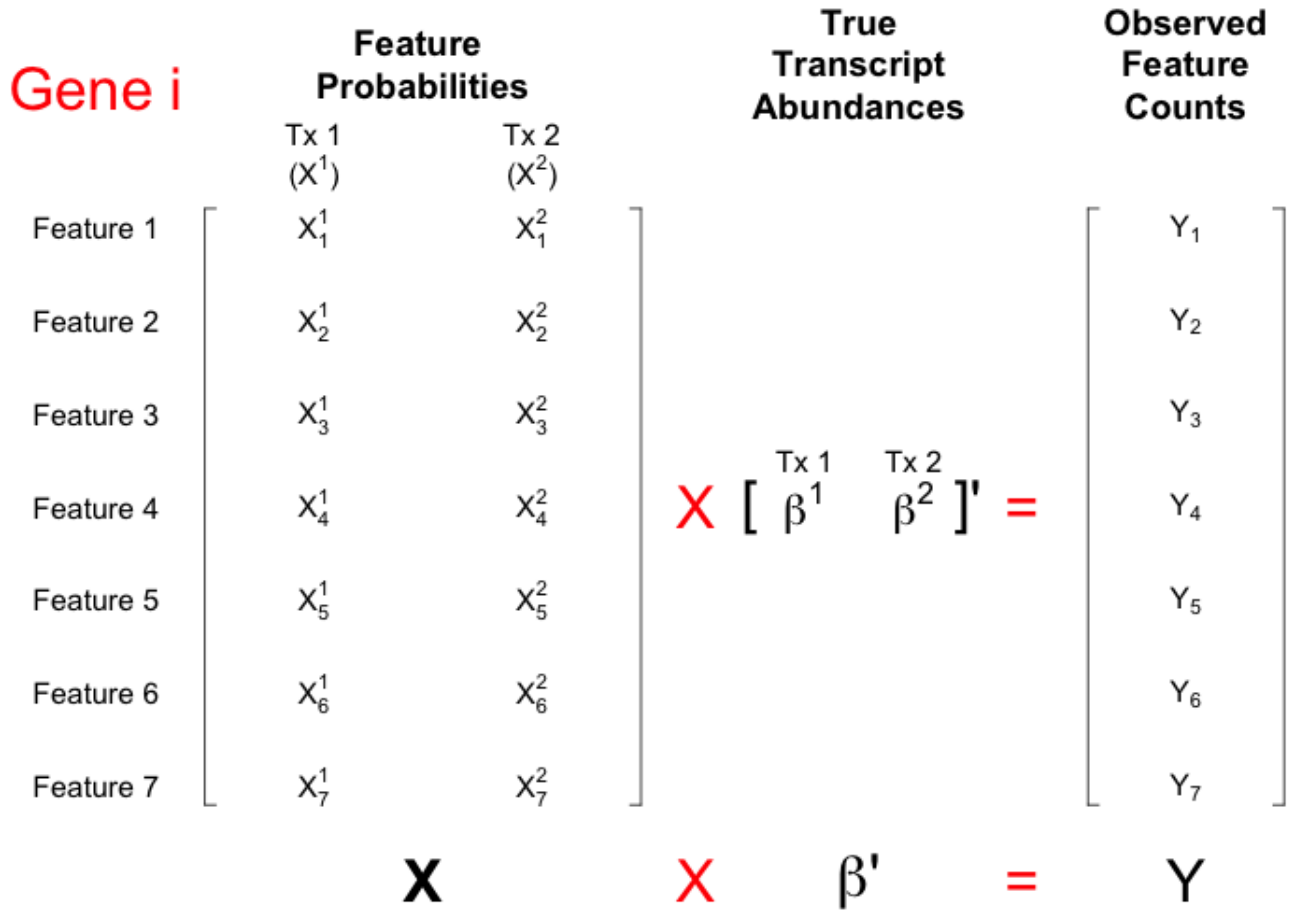

Figure 5: For one example gene, an illustration of our model formulation of the relationship between transcript abundances and the observed feature counts. A column of the feature probability matrix represents the expected contribution to the observed feature counts by a random read from the corresponding transcript. Our model estimates β using a weighted non-negative linear model. Furthermore, since the true feature probability matrix is unknown, we estimate it by applying the aligner HISAT2 to possible reads from each matrix.

Scenario	`recount2` NNLS	Kallisto	HISAT2 StringTie	HISAT2 Cufflinks	RSEM	Salmon
37bp Single End	85.2 (205)	55 (112)	70.5 (123)	54 (110)	54.2 (109)	58 (112)
37bp Paired End	86.5 (217)	52.6 (107)	68.8 (121)	53.3 (111)	52.5 (108)	52.5 (107)
50bp Single End	81.7 (200)	53.9 (107)	68.1 (119)	53.8 (110)	53.6 (108)	55.8 (107)
50bp Paired End	82.2 (203)	52.9 (107)	66.8 (119)	52.7 (110)	52.5 (108)	52.9 (108)
75bp Single End	72.2 (167)	54.6 (108)	67.4 (119)	53.7 (108)	54.5 (111)	56.4 (112)
75bp Paired End	71.8 (167)	52.7 (107)	65.2 (117)	52.9 (110)	52.9 (109)	52.7 (108)
100bp Single End	77.6 (175)	54 (110)	67 (115)	53.6 (109)	53.6 (107)	54.8 (109)
100bp Paired End	76.2 (172)	52.4 (108)	63.9 (116)	52.6 (109)	52.8 (109)	52.3 (107)
150bp Single End	65.3 (140)	54 (110)	66.6 (116)	53.3 (109)	52.9 (108)	53.5 (109)
150bp Paired End	64.2 (140)	52.3 (106)	65.2 (116)	52.5 (107)	52.9 (107)	52.3 (106)

Table 1: Summary of the Root Mean Squared Error (RMSE) of each method's transcript quantification against the known truth for 10 different simulation scenarios, for transcripts and (genes). Simulations were carried out via `polyester`, with fragment lengths generated from a Gaussian distribution with mean 250 and standard deviation 25. All coding transcripts on chr1 and chr14 from GencodeV25 annotation were considered. Reads were simulated on a gene-by-gene basis. The total number of reads at a gene is simulated by Negative Binomial with size=4 and p=0.01. The distribution of reads to each transcript at a gene is by Dirichlet(1/k), where k is the number of transcripts in a gene.

Quantifier	recount2 NNLS	Kallisto	HISAT2 StringTie	HISAT2 Cufflinks	RSEM	Salmon
recount2 NNLS	81,733	0.73	0.62	0.74	0.71	0.73
Kallisto	62,429	76,778	0.72	0.86	0.91	0.99
HISAT2 StringTie	61,818	64,264	83,638	0.76	0.71	0.73
HISAT2 Cufflinks	64,738	67,529	66,861	80,846	0.90	0.87
RSEM	56,931	65,158	58,459	65,095	68,084	0.91
Salmon	63,010	76,228	64,761	68,364	65,782	77,683

Table 2: Pair-wise comparison of the evaluated methods on example ERR188412 of the GEUVADIS consortium samples. The upper half shows pair-wise Pearson's correlation. The lower half consists of the number of transcripts that both methods assigned non-zero expression to. The diagonal consists of the number of transcripts that the corresponding method assigned non-zero expression to.

References

[1] Consortium, T.G.: The genotype-tissue expression (gtex) project. Nature genetics **45**(6), 580–585 (2013)

[2] Network, T.C.G.A.R., Weinstein, J.N., Collisson, E.A., Mills, G.B., Shaw, K.R.M., Ozenberger, B.A., Ellrott, K., Shmulevich, I., Sander, C., Stuart, J.M.: The cancer genome atlas pan-cancer analysis project. Nat Genet **45**(10), 1113–1120 (2013)

[3] Leinonen, R., Sugawara, H., Shumway, M., : The sequence read archive. Nucleic Acids Research **39**(suppl_1), 19–21 (2011)

[4] Collado-Torres, L., Nellore, A., Kammers, K., Ellis, S.E., Taub, M.A., Hansen, K.D., Jaffe, A.E., Langmead, B., Leek, J.T.: Reproducible rna-seq analysis using recount2. Nat Biotech **35**(4), 319–321 (2017)

[5] Nellore, A., Collado-Torres, L., Jaffe, A.E., Alquicira-Hernández, J., Wilks, C., Pritt, J., Morton, J., Leek, J.T., Langmead, B.: Rail-rna: scalable analysis of rna-seq splicing and coverage. Bioinformatics **33**(24), 4033–4040 (2017)

[6] Trapnell, C., Williams, B.A., Pertea, G., Mortazavi, A., Kwan, G., van Baren, M.J., Salzberg, S.L., Wold, B.J., Pachter, L.: Transcript assembly and quantification by rna-seq reveals unannotated transcripts and isoform switching during cell differentiation. Nature Biotechnology **28**, 511 (2010)

[7] Pertea, M., Kim, D., Pertea, G.M., Leek, J.T., Salzberg, S.L.: Transcript-level expression analysis of rna-seq experiments with hisat, stringtie and ballgown. Nat. Protocols **11**(9), 1650–1667 (2016)

[8] Bray, N.L., Pimentel, H., Melsted, P., Pachter, L.: Near-optimal probabilistic rna-seq quantification. Nat Biotech **34**(5), 525–527 (2016)

[9] Patro, R., Duggal, G., Love, M.I., Irizarry, R.A., Kingsford, C.: Salmon provides fast and bias-aware quantification of transcript expression. Nat Meth **14**(4), 417–419 (2017)

[10] Li, B., Dewey, C.N.: Rsem: accurate transcript quantification from rna-seq data with or without a reference genome. BMC Bioinformatics **12**, 323–323 (2011)

[11] Kim, H., Bi, Y., Pal, S., Gupta, R., Davuluri, R.V.: Isoformex: isoform level gene expression estimation using weighted non-negative least squares from mrna-seq data. BMC Bioinformatics **12**, 305–305 (2011)

[12] Huang, Y., Hu, Y., Jones, C.D., MacLeod, J.N., Chiang, D.Y., Liu, Y., Prins, J.F., Liu, J.: A robust method for transcript quantification with rna-seq data. Journal of Computational Biology **20**(3), 167–187 (2013)

[13] Kim, D., Langmead, B., Salzberg, S.L.: Hisat: a fast spliced aligner with low memory requirements. Nature Methods **12**, 357 (2015)

[14] Frazee, A.C., Jaffe, A.E., Langmead, B., Leek, J.T.: Polyester: simulating rna-seq datasets with differential transcript expression. Bioinformatics **31**(17), 2778–2784 (2015)

[15] Lappalainen, T., Sammeth, M., Friedländer, M.R., 't Hoen, P.A., Monlong, J., Rivas, M.A., Gonzàlez-Porta, M., Kurbatova, N., Griebel, T., Ferreira, P.G., Barann, M., Wieland, T., Greger, L., van Iterson, M., Almlöf, J., Ribeca, P., Pulyakhina, I., Esser, D., Giger, T., Tikhonov, A., Sultan, M., Bertier, G., MacArthur, D.G., Lek, M., Lizano, E., Buermans, H.P., Padioleau, I., Schwarzmayr, T., Karlberg, O., Ongen, H., Kilpinen, H., Beltran, S., Gut, M., Kahlem, K., Amstislavskiy, V., Stegle, O., Pirinen, M., Montgomery, S.B., Donnelly, P., McCarthy, M.I., Flicek, P., Strom, T.M., Consortium, T.G., Lehrach, H., Schreiber, S., Sudbrak, R., Carracedo, Á., Antonarakis, S.E., Häsler, R., Syvänen, A.-C., van Ommen, G.-J., Brazma, A., Meitinger, T., Rosenstiel, P., Guigó, R., Gut, I.G., Estivill, X., Dermitzakis, E.T.: Transcriptome and genome sequencing uncovers functional variation in humans. Nature **501**(7468), 506–511 (2013)

[16] Ritchie, M.E., Phipson, B., Wu, D., Hu, Y., Law, C.W., Shi, W., Smyth, G.K.: limma powers differential expression analyses for rna-sequencing and microarray studies. Nucleic Acids Research **43**(7), 47 (2015)

[17] Robinson, M.D., McCarthy, D.J., Smyth, G.K.: edger: a bioconductor package for differential expression analysis of digital gene expression data. Bioinformatics **26**(1), 139–140 (2010)

[18] Kent, W.J., Zweig, A.S., Barber, G., Hinrichs, A.S., Karolchik, D.: Bigwig and bigbed: enabling browsing of large distributed datasets. Bioinformatics **26**(17), 2204–2207 (2010)

[19] Mullen, K.M., van Stokkum, I.H.M.: nnls: The lawson-hanson algorithm for non-negative least squares (nnls)

[20] Liew, C.K.: Inequality constrained least-squares estimation **71**(355), 746–751 (1976)