steps, such as variant calling (e.g., using SAMTools or GATK), can be carried out in the usual way. Our optimized and parallelized implementation of Quartz is available, along with a high-quality human genome *k*-mer dictionary (**Supplementary Software**). We also provide here preliminary results on how Quartz changes compression levels and variant-calling accuracy on an *Escherichia coli* genome, indicating Quartz's utility beyond human genomics (**Supplementary Fig. 12**).

With improvements in sequencing technologies increasing the pace at which genomic data is generated, quality scores will require ever greater amounts of storage space; compressive quality scores will become crucial to fully realizing the potential of large-scale genomics. Our data here, in contrast to previous results[21], indicate that the twin goals of compression and accuracy do not have to be at odds with each other. Although total compression comes at the cost of accuracy (**Supplementary Fig. 6**), and quality score recalibration generally decreases compressibility (**Supplementary Table 1**), there is a happy medium at which we can get good compression and improved accuracy. The Quartz software (available at http://quartz.csail.mit.edu and in **Supplementary Software**) will greatly benefit any researchers who are generating, storing, mapping or analyzing large amounts of DNA, RNA, CHIP-seq or exome sequencing data.

Note: Any Supplementary Information and Source Data files are available in the online version of the paper.

ACKNOWLEDGMENTS

We thank L. Cowen and N. Daniels for helpful discussions and comments. Y.W.Y. gratefully acknowledges support from the Fannie and John Hertz Foundation. D.Y., J.P. and B.B. are partially supported by US National Institutes of Health grant GM108348.

COMPETING FINANCIAL INTERESTS

The authors declare no competing financial interests.

Y William Yu[1,2], Deniz Yorukoglu[1], Jian Peng[1,2] & Bonnie Berger[1,2]

[1]Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA.
[2]Department of Mathematics, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA.
e-mail: bab@mit.edu

1. Berger, B., Peng, J. & Singh, M. *Nat. Rev. Genet.* **14**, 333–346 (2013).
2. Kahn, S.D. *Science* **331**, 728–729 (2011).
3. The 1000 Genomes Project Consortium. *Nature* **491**, 56–65 (2012).
4. Veeramah, K.R. & Hammer, M.F. *Nat. Rev. Genet.* **15**, 149–162 (2014).
5. Shapiro, E., Biezuner, T. & Linnarsson, L. *Nat. Rev. Genet.* **14**, 618–630 (2013).
6. Bonfield, J.K. & Mahoney, M.V. *PLoS ONE* **8**, e59190 (2013).
7. Apostolico, A. & Lonardi, S. in *Proceedings of the IEEE Data Compression Conference 2000 (DCC'00)* 143–152 (IEEE Computer Society, 2000).
8. Kozanitis, C., Saunders, C., Kruglyak, S., Bafna, V. & Varghese, G. *J. Comput. Biol.* **18**, 401–413 (2011).
9. Jones, D.C., Ruzzo, W.L., Peng, X. & Katze, M.G. *Nucleic Acids Res.* **40**, e171 (2012).
10. Fritz, M.H.Y., Leinonen, R., Cochrane, G. & Birney, E. *Genome Res.* **21**, 734–740 (2011).
11. Deorowicz, S. & Grabowski, S. *Bioinformatics* **27**, 860–862 (2011).
12. Loh, P.R., Baym, M. & Berger, B. *Nat. Biotechnol.* **30**, 627–630 (2012).
13. Ochoa, I. *et al. BMC Bioinformatics* **14**, 187 (2013).
14. Hach, F., Numanagic, I., Alkan, C. & Sahinalp, S.C. *Bioinformatics* **28**, 3051–3057 (2012).
15. Christley, S., Lu, Y., Li, C. & Xie, X. *Bioinformatics* **25**, 274–275 (2009).
16. Janin, L., Rosone, G. & Cox, A.J. *Bioinformatics* **30**, 24–30 (2014).
17. DePristo, M.A. *et al. Nat. Genet.* **43**, 491–498 (2011).
18. Yu, Y.W., Yorukoglu, D. & Berger, B. in *Research in Computational Molecular Biology: 18th Annual International Conference, RECOMB 2014—Proceedings* (ed. Sharan, R.) 385–399 (Springer, 2014).
19. Kelley, D.R., Schatz, M.C. & Salzberg, S.L. *Genome Biol.* **11**, R116 (2010).
20. Grabherr, M.G. *et al. Nat. Biotechnol.* **29**, 644–652 (2011).
21. Cánovas, R., Moffat, A. & Turpin, A. *Bioinformatics* **30**, 2130–2136 (2014).
22. Li, H. *et al. Bioinformatics* **25**, 2078–2079 (2009).
23. Li, H. & Durbin, R. *Bioinformatics* **26**, 589–595 (2010).
24. Langmead, B. & Salzberg, S.L. *Nat. Methods* **9**, 357–359 (2012).

Ballgown bridges the gap between transcriptome assembly and expression analysis

To the Editor:

Analysis of raw reads from RNA sequencing (RNA-seq) makes it possible to reconstruct complete gene structures, including multiple splice variants, without relying on previously established annotations[1–3]. Downstream statistical modeling of summarized gene or transcript expression data output from these pipelines is facilitated by the Bioconductor project, which provides open-source tools for analysis of high-throughput genomics data[4]. However, the outputs of upstream processing tools often are aggregated across samples or are not in a format that is readily compatible with downstream Bioconductor packages. This gap has slowed rigorous statistical analysis of expression quantitative trait locus (eQTL), time-course, continuous covariates or of confounded experimental designs at the transcript level and has led to considerable controversy in the analysis of population-level RNA-seq data[5]. In this Correspondence, we report the development of two pieces of software, Tablemaker and Ballgown, that bridge the gap between transcriptome assembly and fast, flexible differential expression analysis (**Supplementary Fig. 1**).

Tablemaker uses a GTF file (the standard output from any transcriptome assembler) and spliced read alignments to produce files that explicitly specify the structure of assembled transcripts, mappings from exons and splice junctions to transcripts, and several measures of feature expression, including fragments per kilobase of transcript per million reads sequenced (FPKM) and average per-base coverage (**Supplementary Note 1**). Tablemaker wraps Cufflinks to estimate FPKM for each assembled transcript. After the transcriptome assembly is processed using Tablemaker, the output files (**Supplementary Note 1**) can be explored interactively in R using the Ballgown package. Ballgown converts Tablemaker's assembly structure and expression estimates into an easy-to-access R object (**Supplementary Fig. 2**) for downstream analyses. Alternatively, the Tablemaker step can be skipped: the R object can be created based on an assembly created with StringTie[6], a new, efficient assembler, or from a transcriptome whose expression estimates have been calculated with RSEM's 'rsem-calculate-expression'[7]. Ballgown can be used to visualize the transcript assembly on a gene-by-gene basis, extract abundance estimates for exons, introns, transcripts or genes, and perform linear model–based differential expression analyses (**Supplementary Note 2**).

The basic linear modeling strategy for differential expression testing implemented in Ballgown allows analysis of eQTL, time-course, continuous covariates or confounded experimental designs at the exon, gene or transcript level. This approach is similar to the linear modeling strategy implemented in limma[8], without empirical Bayes shrinkage, and can be applied to exon or gene counts available through the Ballgown object after appropriately transforming the count data[9]. Alternatively, users may choose to apply the widely used Bioconductor packages for sequence count data[10,11]. There is no other existing statistical software that allows this level of flexibility for modeling transcript-

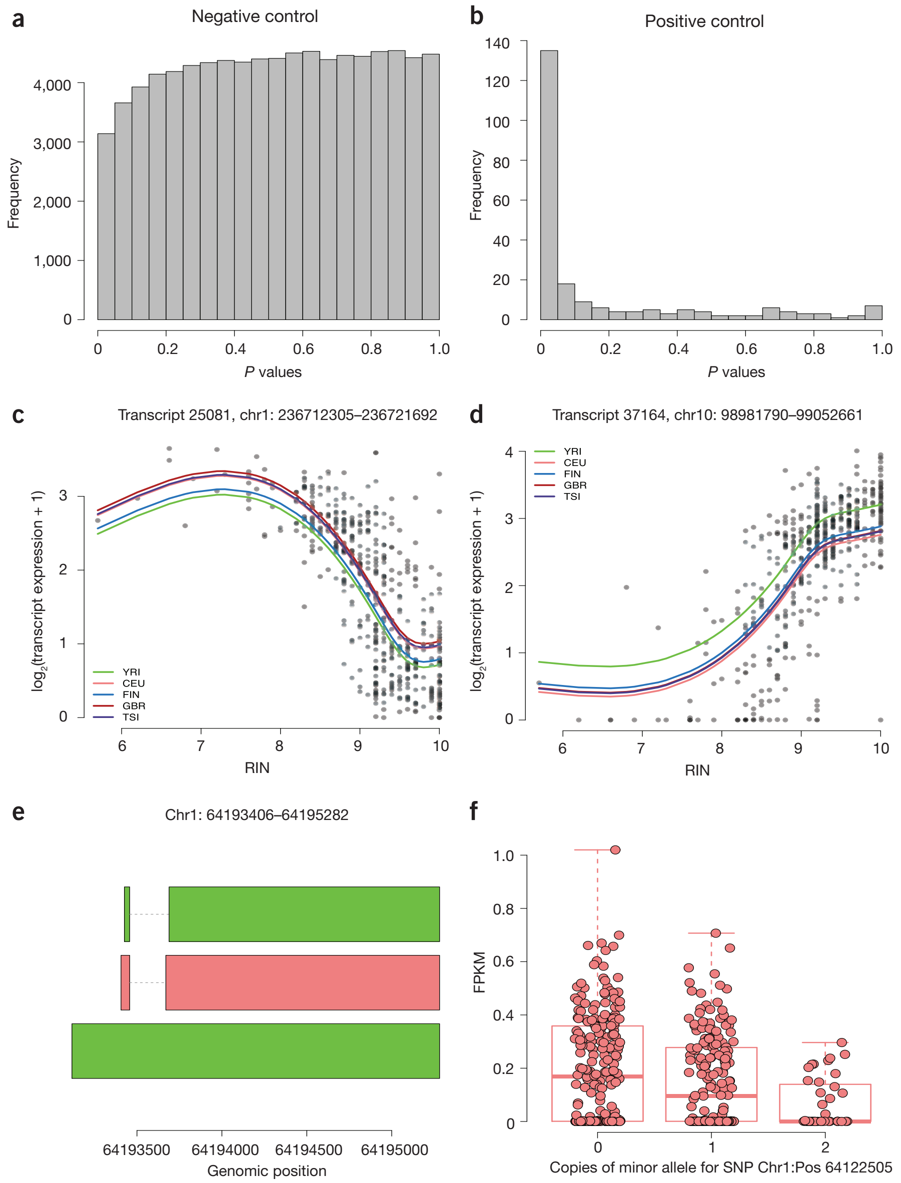

Figure 1 Experimental results obtained with the Ballgown framework. (**a**) Distribution of transcript-level *P* values obtained with Ballgown's *F*-tests in an experiment without signal (negative control). (**b**) Transcript-level *P*-value distribution from Ballgown's *F*-tests for expression differences in Y-chromosome transcripts between males and females (positive control). (**c**,**d**) Nonlinear effects of RNA quality are shown for expression of two representative transcripts: (**c**) transcript 25081 on chromosome 1; (**d**) transcript 37164 on chromosome 10. These two transcripts (FDR < 0.001) and 1,497 others showed a relationship with RNA quality (RIN) that was significantly better captured by a nonlinear trend with three degrees of freedom than a standard linear model. Colored lines shown are predicted values from a natural cubic spline fit and represent predictions for the specified population, assuming average library size. (**e**) Structures for an assembled transcript that does not overlap any annotated transcripts but shows a significant eQTL. (**f**) Boxplot of the FPKM transcripts for the middle (red) transcript from **e**, which shows a consistent and statistically significant eQTL.

level expression data. Count-based modeling strategies are not applicable to transcript-level data[12], and Cuffdiff2 can only be applied to two-group transcript-level differential expression analysis[13]. EBSeq could be used in combination with RSEM as a pipeline for transcript-level differential expression analysis, but it is less efficient than linear modeling and does not handle experimental designs beyond multigroup comparison[14].

Here we illustrate how to use Tablemaker and Ballgown with the Tuxedo suite, a widely used pipeline for transcript assembly, quantification and flexible differential expression analysis at transcript resolution. The Tuxedo suite process consists of aligning reads using Bowtie[15] and Tophat2 (ref. 16), assembling transcripts using Cufflinks[2] and carrying out differential expression analysis using Cuffdiff2 (ref. 17). This suite has been used in many projects[18–20], including the ENCODE[21] and modENCODE[22] consortium projects. However, statistical analysis through Cuffdiff2 can only be applied to two-group differential expression analyses, is computationally demanding and produces strongly conservative estimates of statistical significance. Although several other fast and accurate tools for differential expression analysis, such as EdgeR[10], DESeq[11] and Voom[9], are present in Bioconductor[4], no software connects these tools to the estimated transcript structures and abundances that are output by such tools as the Tuxedo suite. Furthermore, per-feature read counts are not appropriate for isoform-level analysis. The reason is that isoforms from the same gene may have a high degree of overlap that would lead to ambiguous read counts. Here we integrate the Tuxedo suite with Tablemaker, Ballgown and downstream Bioconductor packages to improve the statistical accuracy, flexibility in experimental design and computational speed of RNA-seq analyses.

To show that the default methods in Ballgown can work in the absence of a differential expression signal, we downloaded and processed data from the GEUVADIS RNA sequencing project[23,24] (**Supplementary Note 3**) and compared them with Cuffdiff2 and EdgeR[10]. After aligning RNA-seq reads, assembling the transcriptome and processing the results with Tablemaker, we used Ballgown to load the data into R, where we extracted a single-population subset of data to study. The populations included in the GEUVADIS study were Utah residents with Northern and Western European ancestry (CEU), Yoruba in Ibadan, Nigeria (YRI), Toscani in Italy (TSI), British in England and Scotland (GBR), and Finnish in Finland (FIN). Considering only individuals in the FIN population ($n = 95$), we randomly assigned subjects to one of two groups and tested all assembled transcripts for differential expression between those two groups. We compared the results from using linear models (Ballgown), Cuffdiff2 and EdgeR[10] (at the exon level). We used transcript FPKM as the transcript expression measurement in Ballgown, and per-exon read counts for EdgeR. In this type of experiment, the distribution of the *P* values from all the transcripts should be approximately uniformly distributed, and *q* values[25] should be large.

As expected, the transcript-level *P* values from the linear model tests implemented in Ballgown were approximately uniformly distributed (**Fig. 1a**), and all transcripts had *q* values of ~1, indicating that these models do not generate excess false discoveries. We compared this result to the statistical results from Cuffdiff2 (version 2.2.1, the newest release available in August 2014) on the same data set and found that the *P* values obtained using Cuffdiff2 were not uniformly distributed: the distribution had more mass near 1 than near 0 (**Supplementary Fig. 3a**). This indicates that Cuffdiff2 may be somewhat conservatively biased and calls into question the use of the *q* value as a multiple testing adjustment because it assumes uniformly distributed *P* values. When we compared Ballgown results to EdgeR, the latter called two exons differentially expressed ($q < 0.05$), and the exon-level *P*-value distribution was not uniform, having a bit of extra mass around 0.1 (**Supplementary Fig. 3b**).

These results show that using a well-established, count-based method gives a slightly too liberal result, that Cuffdiff2 is likely conservatively biased and that using a linear model test like the one implemented in Ballgown gives a reasonable *P*-value distribution without calling any transcripts differentially expressed. The linear models from Ballgown took 18 seconds to run on a standard laptop (MacBook Pro, 8 GB memory). For comparison, Cuffdiff2 took 69 hours and 148 GB of memory using four cores on a cluster node. EdgeR was also run on the laptop and took 2.5 minutes. The negative control experiment showed that Ballgown's default statistical tests are appropriately conservative when there is no signal present in the data.

We then carried out a second experiment to investigate whether default statistical tests are capable of making discoveries when differential expression is present. For this experiment, we analyzed differential expression of Y-chromosome transcripts between males and females, a test data set in which all transcripts should be differentially expressed. We used a data set consisting of the 95 FIN individuals in the GEUVADIS RNA-seq data set (58 females, 37 males). The *P*-value histogram from this experiment using the linear model framework implemented in Ballgown shows a very strong signal (**Fig. 1b**). Of the 433 assembled transcripts on the Y chromosome, 225 had a mean FPKM >0.01 in the males. Of these 225 transcripts, 58% were called differentially expressed with a *q* value <0.05 and 72% with a *q* value <0.2. This result shows that the models in Ballgown are capable of discovering true signal in the data set. The *P*-value histogram for the latest Cuffdiff2 version (2.2.1) also revealed a signal (**Supplementary Fig. 4**), which is an improvement compared with earlier versions of Cuffdiff2 on similar Y-chromosome tests[26]. However, only 29 of the 433 assembled transcripts were tested using the inclusion criteria implemented in Cuffdiff2. Of these 29, 24 had $q < 0.05$ and 26 had $q < 0.02$. This suggests that Cuffdiff2 is too conservative to detect appropriate levels of differential expression in this experiment. The Y chromosome linear models from Ballgown took less than 0.1 seconds to run after Tablemaker, whereas Cuffdiff2 took 58 hours and 178 GB of memory on four cores. Note, though, that this footprint could have been substantially reduced by subsetting all BAM files and the merged assembly to only the Y chromosome, but this would have necessitated extra processing and was not required for analysis in Ballgown.

Next, we carried out experiments designed to represent realistic differential expression scenarios; usually some, but not all, transcripts are truly differentially expressed between populations. We evaluated differential expression results from Ballgown and Cuffdiff2 (versions 2.0.2 and 2.2.1) on two publicly available clinical data sets (**Supplementary Note 4**). The first clinical experiment[27] compared lung adenocarcinoma ($n = 12$) and normal control samples ($n = 12$) from nonsmoking female patients. The second experiment[28] compared cells at five developmental stages; we analyzed the data from two stages: embryonic stem cells ($n = 34$) and preimplantation blastomeres ($n = 78$). On these data sets, the *P*-value distributions from the linear model tests implemented in Ballgown were reasonable, as were the *P*-value distributions from Cuffdiff2 version 2.2.1, though Cuffdiff2 2.2.1 was more conservative: it did not identify as many transcripts as differentially expressed as Ballgown. Results from Cuffdiff2 version 2.0.2 (downloaded from the InSilico DB database[29]) showed noticeable conservative bias (**Supplementary Fig. 5**). We also carried out two simulation studies using data simulated with the Polyester package[30] that demonstrated improved sensitivity and specificity estimates for Ballgown compared with Cuffdiff2 (**Supplementary Note 5**; **Supplementary Fig. 6**).

Ballgown offers researchers the flexibility to explore the effects of using alternative expression measurements for analysis. There are two major classes of statistical methods for differential expression analysis of RNA-seq: those based on RPKMs or FPKMs, as exemplified by Cufflinks; and those based on counting the reads overlapping specific regions, as exemplified by DESeq[11] and edgeR[11]. Tablemaker outputs both FPKM estimates from Cufflinks and average coverage of each exon, intron and transcript. We investigated the effect of expression measurement using both simulated data and the GEUVADIS data set, and we confirmed that, as expected, differential expression results obtained using average coverage and using FPKM were strongly correlated (**Supplementary Note 6**; **Supplementary Fig. 7**). This suggests that coverage—an expression measurement that is easier to estimate the FPKM—is potentially a viable alternative expression metric for use in differential expression analyses.

One advantage of the Ballgown framework over Cuffdiff2 is the option either to compare any nested set of models for differential expression or to apply standard differential expression tools in Bioconductor, such as the limma package[8]. To demonstrate the flexibility of linear models like those in Ballgown or limma, we carried out two popular analyses that have not been possible with standard transcriptome assembly and differential expression tools: modeling of continuous covariates and eQTL analysis.

In the first analysis, we treated RNA integrity number (RIN)[31] as a continuous covariate (Storey *et al.*, 2005, ref. 32) and used Ballgown's modeling framework to discover transcripts in the GEUVADIS data set[24] whose expression levels were significantly associated with RIN (**Supplementary Note 7**). Of 43,622 assembled transcripts with average FPKM above 0.1, 19,203 showed a significant effect ($q < 0.05$) of RIN on expression, as determined using a natural cubic spline model for RIN and adjusting for population and library size[33].

A previous analysis of the GEUVADIS data modeled variation in RNA-quality as a linear effect[23]. We fit a model with a linear RIN effect and population and library size adjustments to each transcript and identified an enrichment of transcripts with a positive correlation between FPKM values and RNA quality (**Supplementary Fig. 8**). To investigate the impact of using a more flexible statistical model to detect RIN artifacts, we tested whether applying a cubic polynomial fit for RIN on transcript expression was significantly better than simply including a linear term for RIN after adjusting for population. We compared the cubic and linear fits on 43,622 transcripts with average FPKM > 0.1 across all samples. We found that the cubic fit was significantly better than the linear fit ($q < 0.05$) for 1,499 of the 43,622 transcripts (**Fig. 1c**,**d**), suggesting that flexible nonlinear models may be helpful when measuring the relationship between quantitative covariates and transcript abundance levels.

To further illustrate the flexibility of using the postprocessed Ballgown data for differential expression analysis, we next carried out an eQTL analysis of the 464 non-duplicated GEUVADIS samples across all populations (**Supplementary Note 8**). We removed transcripts with an average FPKM across samples <0.1 and removed single-nucleotide polymorphisms (SNPs) with a minor allele frequency <5%, resulting in 7,072,917 SNPs and 44,140 transcripts. We constrained our analysis to search for *cis*-eQTLs, where the genotype and transcript pairs were within 1,000 kb of each other, resulting in 218,360,149 SNP-transcript pairs. We adjusted for the first three principal components of the genotype data[34] and the first three principal components of the observed transcript FPKM data[35] in the eQTL model fits. The analysis was performed in 2 hours and 3 minutes on a standard desktop computer using the MatrixEQTL package[36]. Visual inspection of the distribution of statistically significant results and corresponding QQ plot indicated that our confounder adjustment was sufficient to remove major sources of bias (**Supplementary Fig. 9**). We identified significant eQTL at the false discovery rate (FDR) 1% level for 17,276 transcripts overlapping 10,524 unique Ensembl-annotated genes. We calculated a global estimate of the number of null hypotheses and estimated that 5.8% of SNP-transcript pairs showed differential expression. In our list of significant transcript eQTLs, 57% and 78% of the SNP-transcript pairs were called significant in the original analysis of the EUR and YRI populations[24], respectively. 14% of eQTL pairs were identified for transcripts that did not overlap Ensembl annotated transcripts (**Fig. 1e**,**f**).

On the basis of the above results, we conclude that the linear model differential expression testing framework built into Ballgown or limma provides computational benefits over Cuffdiff2 and EBSeq. Our timing results (**Supplementary Note 9**; **Supplementary Fig. 10**) also suggest that Ballgown is less computationally intensive than either Cuffdiff2 or EBSeq in addition to providing the flexibility and accuracy advantages detailed above. Ballgown reduces the computational burden of differential expression analysis of assembled transcriptomes without imposing a price in terms of accuracy.

Ballgown can function as a bridge between upstream assembly tools, such as Cufflinks, and downstream statistical modeling tools in Bioconductor. The Ballgown suite includes functions for interactive exploration of the transcriptome assembly, visualization of transcript structures and feature-specific abundances for each locus and *post hoc* annotation of assembled features to annotated features. Direct availability of feature-by-sample expression tables makes it easy to apply alternative differential expression tests or to evaluate other statistical properties of the assembly, such as dispersion of expression values across replicates or genes. The Tablemaker preprocessor writes the tables directly to disk, and they can be loaded into R with a single function call. The Ballgown and Tablemaker software packages are available from Bioconductor and GitHub (**Supplementary Note 10**), and code and data from the analyses presented here are available on GitHub (**Supplementary Note 11**).

Note: Any Supplementary Information and Source Data files are available in the online version of the paper.

ACKNOWLEDGMENTS
The authors would like to thank P.A.C. 't Hoen and T. Lappalainen for providing QC data and assistance with contacting ArrayExpress. They would also like to acknowledge C. Trapnell for helpful conversations about Cufflinks architecture and results. J.T.L., G.P. and B.L. were partially supported by US National Institutes of Health 1R01GM105705, and A.C.F. is supported by a Hopkins Sommer Scholarship.

COMPETING FINANCIAL INTERESTS
The authors declare no competing financial interests.

***Alyssa C Frazee*[1,2]*, Geo Pertea*[2,3]*, Andrew E Jaffe*[1,2,4]*, Ben Langmead*[1,2,5]*, Steven L Salzberg*[1–3,5] *& Jeffrey T Leek*[1,2]**

[1]Department of Biostatistics, Johns Hopkins Bloomberg School of Public Health, Baltimore, Maryland, USA. [2]Center for Computational Biology, Johns Hopkins University, Baltimore, Maryland, USA. [3]McKusick-Nathans Institute of Genetic Medicine, Johns Hopkins University School of Medicine, Baltimore, Maryland, USA. [4]Lieber Institute for Brain Development, Johns Hopkins Medical Campus, Baltimore, Maryland, USA. [5]Department of Computer Science, Johns Hopkins University, Baltimore, Maryland, USA. email: jtleek@gmail.com.

1. Mortazavi, A., Williams, B.A., McCue, K., Schaeffer, L. & Wold, B. *Nat. Methods* **5**, 621–628 (2008).
2. Trapnell, C. *et al. Nat. Biotechnol.* **28**, 511–515 (2010).
3. Grabherr, M.G. *et al. Nat. Biotechnol.* **29**, 644–652 (2011).
4. Gentleman, R.C. *et al. Genome Biol.* **5**, R80 (2004).
5. Dermitzakis, M., Getz, G., Ardlie, K., Guigo, R. & GTEx Consortium. *Bits of DNA* https://liorpachter.wordpress.com/author/edermitzakis/. (2013).
6. Pertea, M. *et al. Nat. Biotechnol.* **33**, 290–295 (2015).
7. Li, B. & Dewey, C.N. *BMC Bioinformatics* **12**, 323 (2011).
8. Smyth, G.K. in *Bioinformatics and Computational Biology Solutions Using R and Bioconductor* (eds. Gentleman, R., Carey, V., Huber, W., Irizarry, R. & Dudoit, S.) 397–420 (Springer, 2005).
9. Law, C.W., Chen, Y., Shi, W. & Smyth, G.K. *Genome Biol.* **15**, R29 (2014).
10. Robinson, M.D., McCarthy, D.J. & Smyth, G.K. *Bioinformatics* **26**, 139–140 (2010).
11. Anders, S. & Huber, W. *Genome Biol.* **11**, R106 (2010).
12. Chen, Y., McCarthy, D., Robinson, M. & Smyth, G.K. *edgeR: differential expression analysis of digital gene expression data User's Guide.* http://www.bioconductor.org/packages/release/bioc/vignettes/edgeR/inst/doc/edgeRUsersGuide.pdf (2011).
13. Trapnell, C. *et al. Nat. Protoc.* **7**, 562–578 (2012).
14. Leng, N. *et al. Bioinformatics* **29**, 1035–1043 (2013).
15. Langmead, B., Trapnell, C., Pop, M. & Salzberg, S.L. *Genome Biol.* **10**, R25 (2009).
16. Kim, D. *et al. Genome Biol.* **14**, R36 (2013).
17. Trapnell, C. *et al. Nat. Biotechnol.* **31**, 46–53 (2013).
18. Lister, R. *et al. Science* **341**, 1237905 (2013).
19. Young, R.S. *et al. Genome Biol. Evol.* **4**, 427–442 (2012).
20. Lister, R. *et al. Nature* **471**, 68–73 (2011).
21. Djebali, S. *et al. Nature* **489**, 101–108 (2012).
22. Graveley, B.R. *et al. Nature* **471**, 473–479 (2011).
23. C't Hoen, A.P., Friedländer, M.R. & Almlöf, J. *Nature* **31**, 1015–1022 (2013).
24. Lappalainen, T. *et al. Nature* **501**, 506–511 (2013).
25. Storey, J.D. & Tibshirani, R. *Proc. Natl. Acad. Sci. USA* **100**, 9440–9445 (2003).
26. Frazee, A.C., Sabunciyan, S., Hansen, K.D., Irizarry, R.A. & Leek, J.T. *Biostatistics* **15**, 413–426 (2014).
27. Kim, S.C. *et al. PLoS ONE* **8**, e55596 (2013).
28. Yan, L. *et al. Nat. Struct. Mol. Biol.* **20**, 1131–1139 (2013).
29. Coletta, A., Molter, C., Duqué, R. & Steenhoff, D. *Genome Biol.* **13**, R104 (2012).
30. Frazee, A.C., Jaffe, A.E., Langmead, B. & Leek, J. *bioRxiv* http://biorxiv.org/content/biorxiv/early/2014/06/06/006015.full.pdf (2014)
31. Schroeder, A. *et al. BMC Mol. Biol.* **7**, 3 (2006).
32. Storey, J.D. *et al. PNAS* **102**, 12837–12842 (2005).
33. Paulson, J.N., Stine, O.C., Bravo, H.C. & Pop, M. *Nat. Methods* **10**, 1200–1202 (2013).
34. Price, A.L. *et al. Nat. Genet.* **38**, 904–909 (2006).
35. Leek, J.T. & Storey, J.D. *PLoS Genet.* **3**, 1724–1735 (2007).
36. Shabalin, A.A. *Bioinformatics* **28**, 1353–1358 (2012).

Supplementary Information: Ballgown bridges the gap between transcriptome assembly and expression analysis

Alyssa C. Frazee[1,3], Geo Pertea[2,3],Andrew E. Jaffe[1,3,4], Ben Langmead[1,2,3,5], Steven L. Salzberg[1,2,3,5], & Jeffrey T. Leek[1,3,*]

December 2014

1. Department of Biostatistics, Johns Hopkins Bloomberg School of Public Health

2. McKusick-Nathans Institute of Genetic Medicine, Johns Hopkins University School of Medicine

3. Center for Computational Biology, Johns Hopkins University

4. Lieber Institute for Brain Development, Johns Hopkins Medical Campus

5. Department of Computer Science, Johns Hopkins University

∗ *Correspondence to jtleek@gmail.com*

Supplementary Figure 1

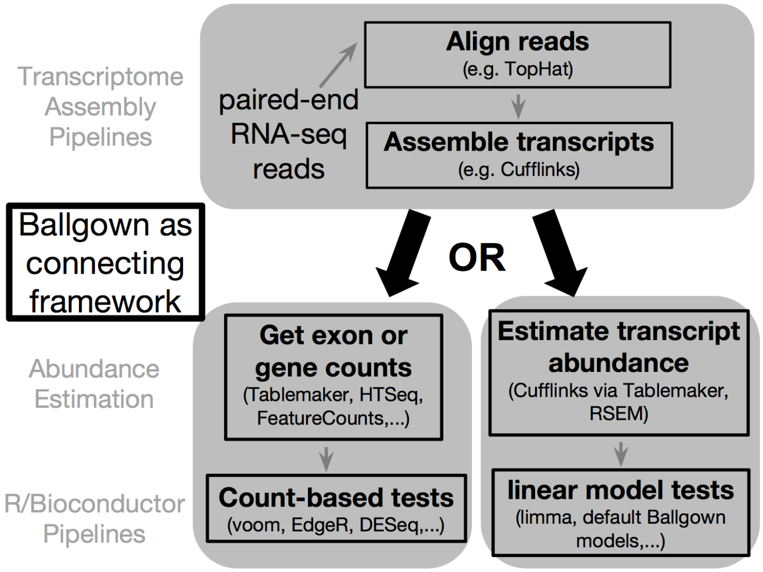

Figure 1: **The Ballgown pipeline**. *Ballgown* is designed to be a tool-agnostic bridge between transcriptome assemblers and abundance estimation tools, and fast, flexible differential expression analysis pipelines in R and Bioconductor. Ballgown as a bridge between transcriptome assembly and fast, flexible differential expression analysis. For example, the *Ballgown* workflow connects transcript assembly tools like *Tophat2* and *Cufflinks* to Bioconductor tools like *EdgeR* and *DESeq* for downstream analysis, but it is not specific to these particular tools. The software can be used with any assembly whose structure is specified in GTF format, coupled with a set of spliced read alignments in BAM format. RSEM and *StringTie* (in addition to *Cufflinks*) are currently officially supported, and we plan to add support for more tools.

Supplementary Note 1: *Tablemaker* output files

Tablemaker outputs the following set of related tab-delimited text files. *Tablemaker* is designed to be run on the output of *Cufflinks* and *Cuffmerge* but *Ballgown* can be used with any assembly output that can be converted into the following sets of tab-delimited files.

- *e_data.ctab*: exon-level expression measurements. One row per exon. Columns are *e_id* (numeric exon id), *chr*, *strand*, *start*, *end* (genomic location of the exon), and the following expression measurements for each sample:
  - *rcount*: reads overlapping the exon
  - *ucount*: uniquely mapped reads overlapping the exon
  - *mrcount*: multi-map-corrected number of reads overlapping the exon

  - *cov*: average per-base read coverage
  - *cov_sd*: standard deviation of per-base read coverage
  - *mcov*: multi-map-corrected average per-base read coverage
  - *mcov_sd*: standard deviation of multi-map-corrected per-base coverage

- *i_data.ctab*: intron- (i.e., junction-) level expression measurements. One row per intron. Columns are *i_id* (numeric intron id), *chr*, *strand*, *start*, *end* (genomic location of the intron), and the following expression measurements for each sample:
  - *rcount*: number of reads supporting the intron
  - *ucount*: number of uniquely mapped reads supporting the intron
  - *mrcount*: multi-map-corrected number of reads supporting the intron

- *t_data.ctab*: transcript-level expression measurements. One row per transcript. Columns are:
  - *t_id*: numeric transcript id
  - *chr*, *strand*, *start*, *end*: genomic location of the transcript
  - *t_name*: Cufflinks-generated transcript id
  - *num_exons*: number of exons comprising the transcript
  - *length*: transcript length, including both exons and introns
  - *gene_id*: gene the transcript belongs to
  - *gene_name*: HUGO gene name for the transcript, if known
  - *cov*: per-base coverage for the transcript (available for each sample)
  - *FPKM*: Cufflinks-estimated FPKM for the transcript (available for each sample)

- *e2t.ctab*: table with two columns, *e_id* and *t_id*, denoting which exons belong to which transcripts. These ids match the ids in the *e_data* and *t_data* tables.

- *i2t.ctab*: table with two columns, *i_id* and *t_id*, denoting which introns belong to which transcripts. These ids match the ids in the *i_data* and *t_data* tables.

Supplementary Figure 2

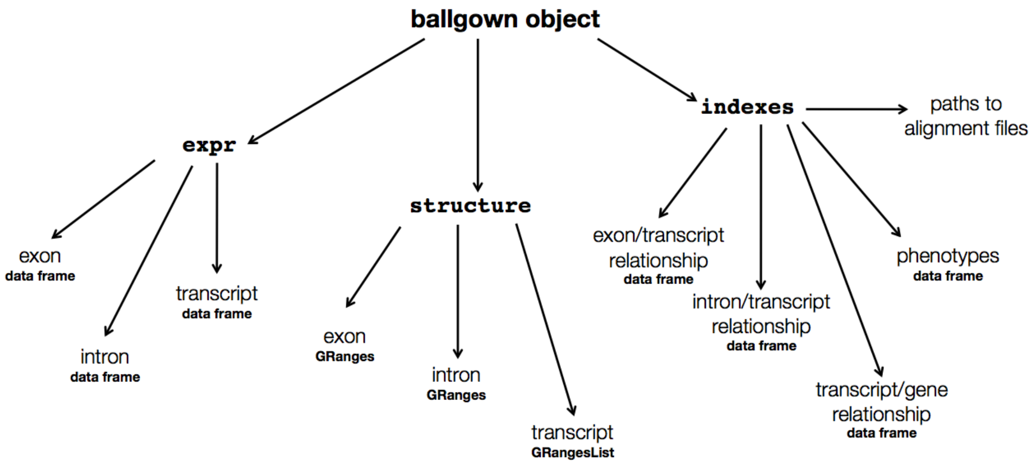

Figure 2: **The `ballgown` data structure.** The *Ballgown* R provides a comprehensive data structure for transcriptome assemblies. The package loads assembly data into an object with linked data frames of expression measurements (`expr`) for exons, introns, and transcripts. The object also loads information about exon, intron, and transcript structures (`structure`), utilizing the efficient GenomicRanges [16] data structures for storage. Finally, the object contains other relevant assembly data (`indexes`), including phenotype data, relationships between exons, introns, and transcripts, and paths to alignment files on disk for easy connection with the assembly.

Supplementary Note 2: Data, notation, and statistical models

There are two distinct components to the data that *Ballgown* is equipped to analyze: the actual structure of the assembled transcriptome: (1) genomic locations of features and the relationships between exons, introns, transcripts and (2) genes and the expression measurements for the features in the transcriptome. Here we precisely define both the assembly structure and the associated data.

Assembly structure

The transcriptome is assembled based on a set R of aligned RNA-seq reads. We denote the yth read from the zth sample with r_{yz}, where $y = 1, ..., N_z$ and $z = 1, ..., n$, so there are n samples in the study, and sample z has N_z aligned reads.

The transcriptome assembled from the reads consists of four types of features: transcripts, genes, exons, and introns. These features all have start and finishing positions on the genome, which represent using the functions $s()$ and $f()$, e.g., $s(x)$ represents the start position of feature x. The K assembled transcripts are denoted by t_k, where $k = 1, ..., K$. These transcripts can be organized into G genes, denoted by g_l, $l = 1, ..., G$. Each gene can be represented by a set of transcripts falling within its boundaries:

$$g_l = \{t_k : s(t_k) > s(g_l) \text{ and } f(t_k) < f(g_l)\}$$

The assembly also contains M exons, each of which we represent as a closed interval of genomic locations:

$$e_m = [s(e_m), f(e_m)], m = 1, ..., M$$

With this notation, we can then represent transcript k as a subset of the M exons comprising the assembly:

$$t_k = \{e_m : m \in I_k\}, I_k \subset \{1, ..., M\}$$

Here, I_k represents the indices of the exons that make up transcript k. Note that the exon e_m can belong to several different transcripts. We can then easily define $s(t_k)$ and $f(t_k)$ in terms of exon boundaries:

$$s(t_k) = \min\{s(e_m) : m \in I_k\}$$

$$f(t_k) = \max\{f(e_m) : m \in I_k\}$$

Finally, let w_k represent the wth element of I_k. Then we can denote the wth intron in transcript k with an open interval:

$$i_{kw} = (f(e_{w_k}), s(e_{(w+1)_k}))$$

In other words, i_{kw} is simply the genomic interval between the wth and $w + 1$th exons of transcript k.

With these definitions in place, we can now precisely define the reads r_{yz}. An RNA-seq read is simply a subsequence of an RNA transcript. Using set notation, we can define each read using the form:

$$r_{yz} = \left\{ x \in [E, E'] : E < E' \text{ and } x, E, E' \in \bigcup_{m \in I_k} e_m \text{ for some } k \right\}$$

An assembly algorithm applied to the set of reads r_{yz} produces estimates of the exons: $\hat{e}_m, m = 1, \ldots, M$, transcripts: $\hat{t}_k, k = 1, \ldots K$ of the transcripts and genes: $\hat{g}_l, l = 1, \ldots, G$. Most current statistical models treat this assembly as fixed and correct when performing analyses. But as we will demonstrate in the methods section, assembled transcripts are subject to error and may be improved through statistical analysis [20, 27].

Expression data

Next we can define expression measurements for each type of feature given a particular assembled set of transcripts. Here we define sensible expression measurements that are currently implemented in the *Ballgown* package, but the statistical models are flexible enough to handle other types of measures as well.

For each sample z, each transcript $\hat{t}_k$ has two measurements that are calculated by our upstream *Ballgown* preprocessing software: average per-base read coverage: $cov(t_k, z)$ and FPKM (fragments per kilobase of transcript per million mapped reads): $FPKM(t_k, z)$. Currently, these transcript-level measurements are estimated in *Cufflinks* via maximum likelihood; the procedure is described in detail by [28].

Each gene g_l has one expression measurement for each sample, $FPKM(g_l, z)$. This measurement is reconstructed from the transcripts in g_l as follows: first, the number of fragments per million mapped reads for sample z for each $t_k \in g_l$ is calculated by multiplying $FPKM(t_k, z)$ by the length of transcript t_k in kilobases. The gene's total fragments per million mapped reads is the sum of the transcript-level fragments per million mapped reads for all the transcripts in the gene. Finally, the gene-level FPKM is calculated by dividing the gene's total fragments per million mapped reads by the gene's length.

The *Ballgown* preprocessor also calculates average per-base read coverage for each exon in the assembly, given the assembly structure and the aligned reads R. For sample z, we have:

$$cov(e_m, z) = \frac{\sum_{r_{yz} \in R} \sum_{bp \in [s(e_m), f(e_m)]} \mathbb{1}\{bp \in r_{yz}\}}{f(e_m) - s(e_m) + 1}$$

Each exon also has a raw read count, defined as the number of reads whose alignments overlap that exon:

$$rcount(e_m, z) = \sum_{r_{yz} \in R} \mathbb{1}\{r_{yz} \cap e_m \neq \emptyset\}$$

The main expression measurement for introns is also raw read count, defined as the number of reads whose alignments support the intron in the sense that their alignments are split across that intron's neighboring exons:

$$rcount(i_{kw}, z) = \sum_{r_{yz} \in R} \mathbb{1}\{s(r_{yz}) \in e_m \text{ and } f(r_{yz}) \in e_{m'}\}$$

where $m \leq w_k$ and $m' \geq (w+1)_k$.

Statistical methods for detecting differential expression

After exploring the structure of the assembled transcriptome and performing any necessary transcript post processing, the next step is to identify transcripts or genes that are differentially expressed across groups. Here we outline a framework for statistical analysis of transcript and gene abundances. To make the ideas concrete we use FPKM as the expression measurement and transcripts as the feature of interest, but these can be replaced in the following model definitions with any of the expression measurements and any of the available genomic features in the assembly (genes, transcripts, exons, or introns).

Differential expression tests are implemented as follows: for each transcript $\hat{t}_k$, the following model is fit:

$$h(FPKM(\hat{t}_k, z)) = \alpha_k + \sum_{p=1}^{P} \beta_{pk} X_{zp} + \varepsilon_{zk} \tag{1}$$

where:

- $FPKM(\hat{t}_k, z)$ is the FPKM expression measurement for transcript k for sample z
- h is a transformation [3] to reduce the impact of mean-variance relationships observed in the counts [2]. For example, the transformation $h(\cdot) = \log_2(\cdot + 1)$ is commonly applied in the analysis of sequence-count data [14].
- α_k represents the baseline expression for transcript k
- X_{zp} represents covariate p for sample z. These covariates differ by experiment type. X_{z1} generally represents a library size adjustment for sample z. Assuming c_k represents the 75th percentile of all log FPKM values for transcript k, `ballgown`'s default the covariate X_{1z} is:

  $$\sum_k FPKM(\hat{t}_k, z)\mathbb{1}[FPKM(\hat{t}_k, z) <= c_k]$$

  This normalization term is derived from the "cumulative sum scaling" (CSS) normalization approach [21].
- β_{pk} quantifies the association of covariate p on the expression of transcript k
- ε represents residual measurement error

A flexible approach to differential expression is to compare nested sub models of model (1) using parametric F-tests [24]. The null hypothesis can be as flexible as any linear contrast of the coefficients β_{pk} but for simplicity we focus on null hypotheses of the form: $H_0 : \beta_{pk} = 0, p \in \mathcal{S}$ versus the alternative that all β_{pk} are nonzero. The general principle is that a model including any potential confounders plus the covariate(s) of interest – a 0/1 indicator for group in the two-group comparison, several indicator variables for the multi-group comparison, or a generalized additive model [11] for a time variable for timecourse experiments – is compared with a model that includes only the potential confounders. For the two models fit for each transcript k, *Ballgown* calculates the statistic

$$F = \frac{\frac{RSS_0 - RSS_1}{P_1 - P_0}}{\frac{RSS_1}{n - P_1}}$$

where RSS_0 represents the residual sum of squares from the model without group or time covariates, RSS_1 represents the residual sum of squares from the model including the covariates of interest, P_0 is the number of covariates in the smaller model, P_1 is the number of covariates in the larger model, and n is the total number of samples. Under the null hypothesis that the larger model does not fit the data significantly better than the smaller model, this statistic follows an F distribution with $(P_1 - P_0, n - P_1)$ degrees of freedom, so p-values can be generated by comparing the two models for each transcript k [17]. We control for multiple testing using standard FDR controlling procedures [25].

Supplementary Note 3: Processing the GEUVADIS data

We downloaded the FASTQ files from the GEUVADIS project [15, 1] from `http://www.ebi.ac.uk/ena/data/view/ERP001942`. With this data, we:

- Aligned reads with TopHat 2.0.9, using the `-G` option to align reads to the transcriptome first. We used the hg19 genome reference available from the Illumina iGenomes project.

- Assembled sample-specific transcriptomes with *Cufflinks* 2.1.1, using default options and no annotation

- Merged sample-specific assemblies into an experiment-wide assembly with *Cuffmerge* 2.1.1

- Estimated feature expression and organized the assembly with *Tablemaker* so that all files described in Supplmentary Section 1 were available.

- Created several *Ballgown* objects using the *Ballgown* R package

The resulting *Ballgown* objects include phenotype data available from several sources, including `http://www.ebi.ac.uk/ena/data/view/ERP001942`, the 1000 Genomes Project [5], and additional quality control data from GEUVADIS researchers (available at `https://github.com/alyssafrazee/ballgown_code/blob/master/GEUVADIS_preprocessing/GD667.QCstats.masterfile.txt`). The *Ballgown* R objects are available for download at `http://figshare.com/articles/GEUVADIS_Processed_Data/1130849`. So the objects can be feasibly loaded into memory and stored on disk, a separate object is available for each expression measurement.

Supplementary Figure 3

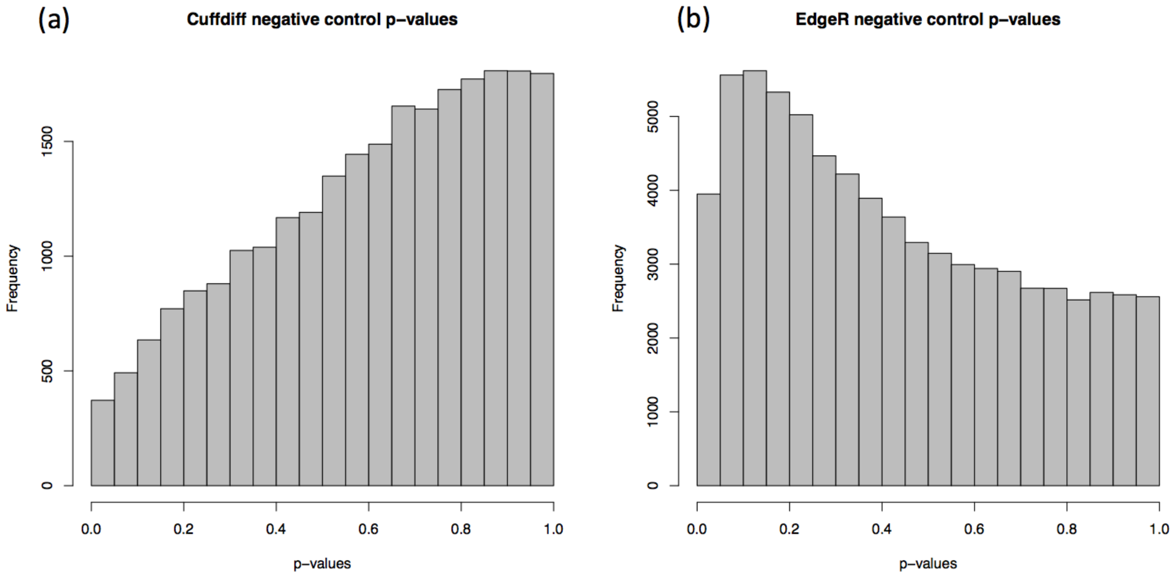

Figure 3: **P-value histograms of results of differential expression analyses between two randomly selected groups: Cuffdiff and EdgeR**. The main manuscript describes a negative control experiment, performed to demonstrate that the default methods in *Ballgown* perform appropriately in a scenario where there is no differential expression signal. Subjects in the FIN population group in the processed GEUVADIS dataset were randomly assigned to one of two groups, and all assembled transcripts for differential expression between those two groups. Linear models as implemented in *Ballgown* gave uniformly distributed null p-values, as expected (Figure 1a, main manuscript). However, the statistical results from *Cuffdiff2* (version 2.2.1, the newest release available as of August 2014) on the same dataset, gave p-values that were not uniformly distributed but instead were biased toward 1 (Panel a). At the exon level, the p-value distribution from *EdgeR* was also not uniform, having a bit of extra mass around 0.1 (Panel b). These results show that a well-established, count-based methods gives a slightly too-liberal result on this kind of experiment and illustrates a potential conservative bias still present in *Cuffdiff2* version 2.2.1.

Supplementary Figure 4

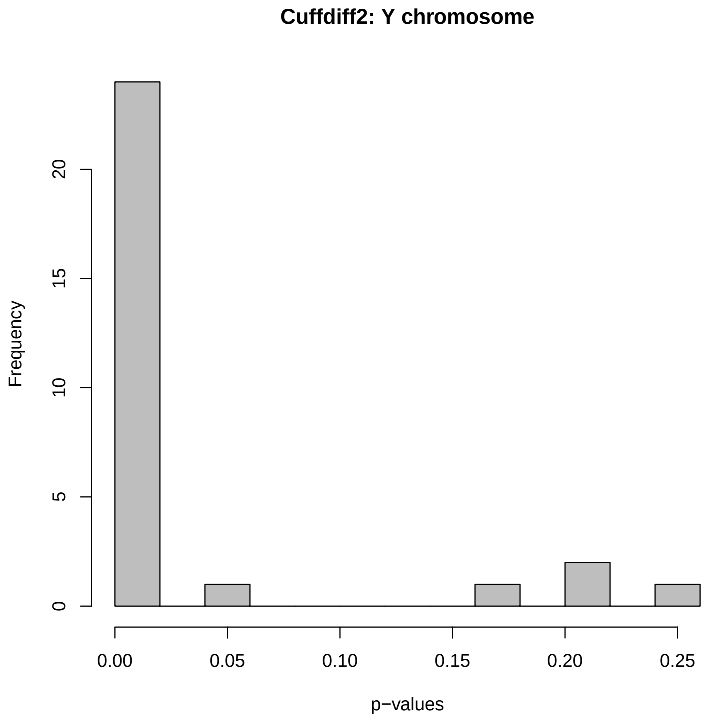

Figure 4: **P-value histogram of Cuffdiff results of a differential expression analysis of Y-chromosome transcripts between males and females**. The positive control experiment in the main manuscript tested transcripts on the Y chromosome for differential expression between males and females in the FIN population in the processed GEUVADIS dataset. Previous research [9] has shown earlier versions of *Cuffdiff2* performing poorly on this type of experiment, but version 2.2.1 discovers some statistically significant differences in expression on the Y chromosome between males and females. Most of the p-values from the 29 tested transcripts were low (Figure 4). However, 433 transcripts were assembled, so *Cuffdiff2* 2.2.1 seems to be a bit conservative in this regard about when it should perform a test: 29 of 433 assembled transcripts were tested.

Supplementary Note 4: InSilico DB Analyses

Methods

InSilico DB [4] includes processed data from public experiments on the Sequence Read Archive. We downloaded the *Cuffdiff2* output from the cancer versus normal and developmental data sets from InSilico DB on March 5th, 2014. We extracted the p-values for differential expression for the cancer versus normal comparison[13] and the embryonic stem cells versus preimplantation blastomeres data. We also reformatted the FPKM values from this analysis and applied the linear models included in the *Ballgown* package to perform the comparison. The versions and parameters for the software used by *InSilico DB* were *cufflinks, cuffmerge, cuffdiff: v 2.0.2*, *cufflinks -p 6 -q*, *tophat: v 2.0.4 –mate_inner_dist 80 –no-coverage-search* (personal communication Alain Coletta from the InSilico DB).

In order to run the latest versions of *TopHat*, *Cufflinks*, and *Cuffdiff2*, we downloaded the raw sequencing reads from both experiments from the NCBI Sequence Read Archive [18]. The analysis steps were the same as the steps outlined for processing the GEUVADIS dataset in the previous subsection, except *Tophat2* version 2.0.11 and *Cufflinks* version 2.2.1 was used. In addition, there was a small change at the *Cufflinks* step: because the data sets in InSilico DB were created by estimating transcript abundances for pre-annotated isoforms, we did the same when we processed the data ourselves. This means we ran *Cufflinks* with the `-G` option and estimated FPKM values for Illumina's iGenomes annotated genes for hg19. These are the isoforms considered in the analysis results. All code for this analysis is available at `https://github.com/alyssafrazee/ballgown_code/tree/master/InSilicoDB`.

We analyzed data from an experiment comparing lung adenocarcinoma ($n = 12$) and normal control samples ($n = 12$) in nonsmoking female patients [13] and from an experiment comparing cells at five developmental stages [30]. Since *Cuffdiff2* was designed for two-class comparisons, we only compared expression between two developmental stages: embryonic stem cells ($n = 34$) and pre-implantation blastomeres ($n = 78$). We compared results from *Cuffdiff2* (versions 2.0.2 and 2.2.1), the linear models from *Ballgown*, the empirical Bayes linear modeling framework implemented in *limma* [24], and EBSeq [19], a Bayesian framework designed for isoform-level differential expression.

Results

Transcript-level differential expression analysis comparing lung adenocarcinoma ($n = 12$) and normal cells ($n = 12$), and comparing embryonic stem cells ($n = 34$) to pre-implantation blastomeres ($n = 78$), should show a strong differential expression signal, especially considering the sample sizes for these experiments. In the cancer vs. normal comparison, there were 19,748 transcripts with average FPKM greater than 1. *Cuffdiff2* (version 2.2.1) identified 4608 of these transcripts as differentially expressed ($q < 0.05$). F-tests comparing nested linear models, as implemented in *Ballgown*, flagged 8875 of these highly-expressed transcripts as differentially expressed. Of 27,058 transcripts tested, EBSeq called 8736 differentially expressed (posterior probability of differential expression of at least 0.95). Similarly, in the cell type dataset, there were 16,430 transcripts with mean FPKM greater than 1. *Cuffdiff2* (2.2.1) calls 6816 of these differentially expressed ($q < 0.05$) while *Ballgown* calls 9701 of them differentially expressed. And of 15,462 transcripts tested, EBSeq identifies 10,307 with posterior probabilities of differential expression of at least 0.95.

While both linear modeling and *Cuffdiff2* produced reasonable p-value distributions for these experiments (Supplementary Figure 5a,c), the relative numbers of differentially expressed transcripts discovered and the p-value distribution shapes show that *Cuffdiff2* is more conservative than the linear models. On its own, this result does not necessarily mean that *Cuffdiff2* (2.2.1) is too conservative, but *Cuffdiff2* also produced conservative p-value distributions in the negative and positive control experiments (main manuscript), we have prior knowledge that the differential expression signal should be quite strong in a tumor/normal or a cell type comparison, and the numbers of discoveries made by another published differential expression method (EBSeq) align more closely with the results from the linear model comparisons. Together these results indicate that conservative bias persists in *Cuffdiff2* (2.2.1). Past versions of *Cuffdiff2*, particularly 2.0.2, produced extremely conservative results on these datasets (Supplementary Figure 5b,d), calling 1 of 4454 highly-expressed transcripts in the tumor/normal dataset differentially expressed, while *Ballgown*'s linear models identified 774. Similarly, in the cell type dataset, *Cuffdiff2* 2.0.2 found 0 of 12,469 highly-expressed transcripts to be differentially expressed ($q < 0.05$) between embryonic stem cells and preimplantation blastomeres, while the linear model tests in *Ballgown* found 6964.

These results in large-scale studies suggest that *Cuffdiff2*'s statistical significance estimates were strongly conservatively biased in version 2.0.2. While version 2.2.1 is better, *Cuffdiff2* is still not producing uniformly-distributed null p-values and is more conservative than other differential expression methods.

Supplementary Figure 5

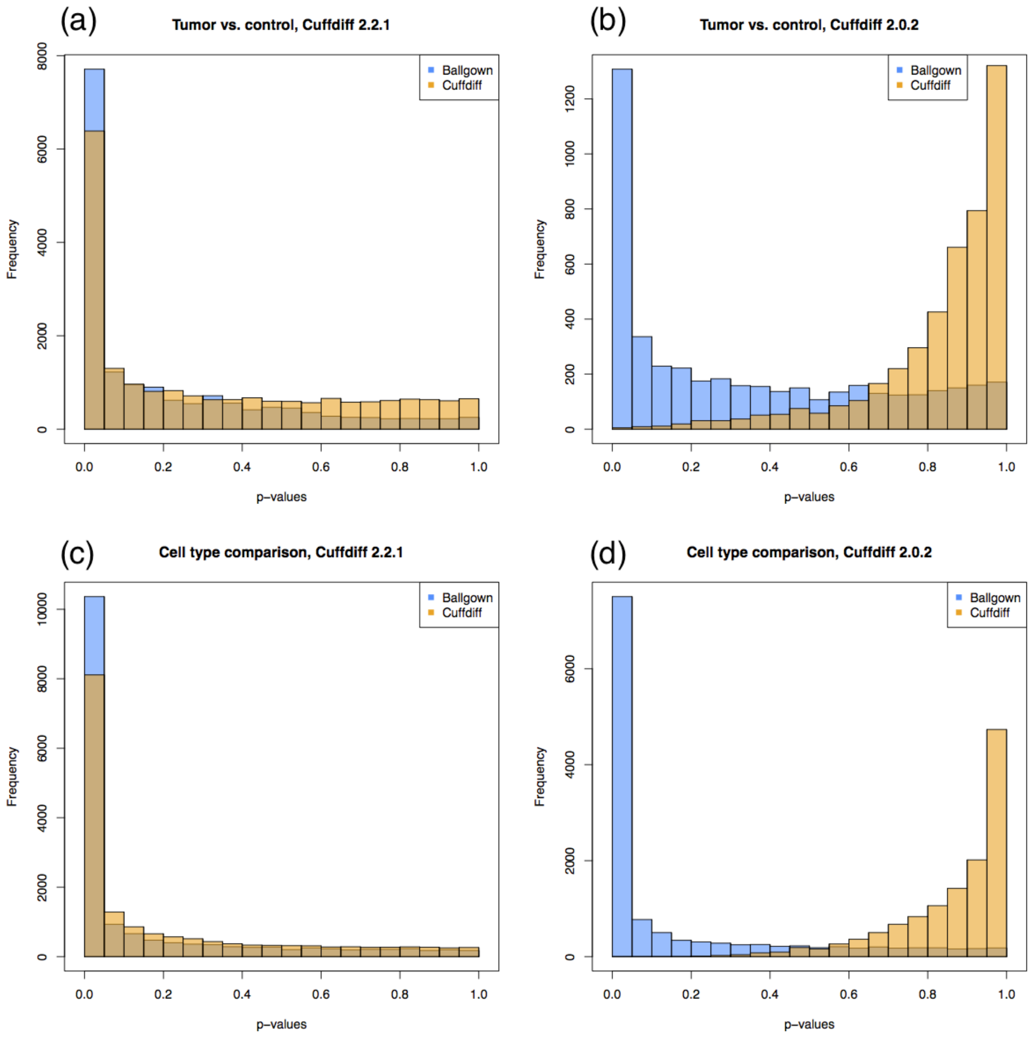

Figure 5: **Comparison of statistical significance estimates between** *Cuffdiff2* **and linear models in real datasets a.** Histograms of p-values from a comparison of 12 lung adenocarcinomas and 12 normal controls from female patients who never smoked. *Ballgown* in blue, *Cuffdiff2* (2.2.1) in orange. **b.** Same comparison as in panel (a), but using the *Cuffdiff2* version 2.0.2 results available from InSilico DB. *Cuffdiff2* version 2.0.2 had a strong conservative bias. Linear model results from *Ballgown* differ from panel (a) because the FPKM estimates used were from an older version of *Cufflinks*, though the linear model results do not demonstrate conservative bias. **c.** Histograms of p-values from the comparison of 78 pre-implantation blastomere samples and 34 embryonic stem cell samples (*Ballgown* in blue, *Cuffdiff2* (2.2.1) in orange). **d.** Same comparison as in panel (c), but using the *Cuffdiff2* version 2.0.2 results available from InSilico DB. As in panel (b), *Cuffdiff2* 2.0.2 showed a strong conservative bias.

Supplementary Note 5: Simulation studies

Methods

To ensure that the linear models implemented in *Ballgown* perform accurately, we performed two separate simulation studies. For both studies, reads were generated from 2745 annotated transcripts on Chromosome 22 from Ensembl [8], using genome build GRCh37 and Ensembl version 74. Data was generated for 20 biological replicates, divided into two groups of 10, where 274 transcripts were randomly chosen to be differentially expressed (at a 6x increase in expression level) in one of the two groups, randomly chosen.

The first simulation study was set up as follows:

- Expression was measured in FPKM. Each transcript's baseline mean FPKM value was determined based on the distribution of mean FPKM values for highly-expressed transcripts in the GEUVADIS dataset. Specifically, the mean of all nonzero FPKM values was calculated for each transcript in the GEUVADIS dataset with mean FPKM larger than 100, and each isoform in the simulated dataset was assigned a randomly selected baseline mean FPKM from this distribution.

- We defined a log-log relationship between a transcript's mean expression level and the variance of its expression levels:

  $$\log \text{variance} = 2.23 \log \text{mean} - 3.08$$

  This relationship was estimated empirically from the assembled GEUVADIS transcriptome (transcripts with mean FPKM values greater than 10) using simple linear regression. The GEUVADIS dataset includes both biological and technical replicates, so this model should encompass both biological and technical variability.

- Then, for each transcript, we randomly drew FPKM expression values from a log-normal distribution with the pre-set mean and variance. For the differentially expressed transcripts, the pre-set mean FPKM was 6 times larger in one group than in the other.

- For each transcript, we also set a sample's expression level to 0 with probability p_0, which was estimated from the GEUVADIS data: for each simulated transcript, p_0 was randomly drawn from the empirical distribution of the proportion of samples with zero expression, over transcripts in the GEUVADIS dataset with mean FPKM larger than 100.

- To translate the pre-set FPKM value into a number of reads to be generated from a transcript for a given sample, we used the definition of FPKM and calculated the number of "fragments" (reads) that should be generated from a transcript by multiplying the set FPKM value by the transcript's length over 1000, then multiplying by an approximate library size of 150,000 reads, over 1 million. The decision to use a mixture of two distributions (log-normal and point mass at 0) was informed by exploratory analysis of the FPKM distributions among several transcripts in the GEUVADIS dataset.

The exploratory analysis is available at `http://htmlpreview.github.io/?https://github.com/alyssafrazee/ballgown_code/blob/master/simulations/mean_var_relationship.html`.

This simulation setup made it such that more reads were generated from longer transcripts, as is expected with RNA-seq protocols.

A second simulation was also conducted with a slightly simpler setup:

- Expression was defined directly by the number of reads being generated from each transcript (instead of using FPKM).
- The mean number of reads generated from each transcript was set to be 300, unless the transcript was randomly selected to be overexpressed in one group, in which case, that group's mean read number for that transcript was 1800.
- The actual number of reads to be simulated from a transcript was drawn from a negative binomial distribution with mean $\mu = 300$ or 1800, and size equal to 0.005μ (so, 1.5 for $\mu = 300$ and 9 for $\mu = 1800$). Note that in the negative binomial distribution, the variance is equal to $\mu + \mu^2/\text{size}$.
- Each sample's read counts were scaled and rounded such that approximately 600,000 reads were generated per sample.

For both these scenarios, the specified number of reads was then generated from transcripts using the *Polyester* Bioconductor package [12]. These simulated reads were then aligned to the genome using TopHat 2.0.11 (aligning to the annotated transcriptome first with the `-G` option), and the resulting alignments were used to assemble transcripts with *Cufflinks* 2.2.1. *Cuffdiff2* (2.2.1) was then run on the simulated datasets. For the *Ballgown* results, we used *Tablemaker* to organize the output, but because *Tablemaker* calls *Cufflinks* version 2.1.1 to estimate per-transcript FPKMs, we updated the `ballgown` object to use the FPKMs written in the `isoforms.read_group_tracking` file by *Cuffdiff2*.

The following models were fit for each transcript in each simulation scenario:

$$\begin{aligned} H_A &: \log_2(FPKM_i + 1) = \beta_0^* + \beta_1^* grp_i + \eta^* q75_i + \epsilon_i^* \\ H_0 &: \log_2(FPKM_i + 1) = \beta_0 + \eta q75_i + \epsilon_i \end{aligned}$$

where grp_i is the value of the group indicator for sample i and $q75$ is a library-size normalizing constant equal to the sum of the log of the nonzero FPKM values to the 75th percentile (known as "cumulative sum scaling" normalization; [21]). We then tested the hypothesis $H_0 : \beta_1^* = 0$ versus the alternative that the coefficient was non-zero. For the analysis with average coverage we replaced $FPKM_i$ with $acov_i$ in the above equations.

We performed simulation studies to precisely assess the accuracy of the differential expression methods. However, assessing the accuracy of transcript-level differential expression is complicated because the annotated transcripts from which reads were generated do not exactly match the assembled transcripts which were tested for differential. This means there is no standard way to define which assembled transcripts should be called differentially expressed. In our accuracy assessments (Supplementary Figure 6), we chose to identify the three closest assembled "neighbors" for each of the 274 truly DE annotated transcripts. Distance was measured by percent overlap, so each annotated transcript's 3 closest assembled neighbors were the 3 transcripts overlapping it the most. All of these selected "neighbors" were considered as part of the sensitivity and specificity calculations: sensitivity was defined as the ratio of the number of truly differentially expressed annotated transcripts with at least one of its three closest assembled neighbors called differentially expressed to the total number of truly differentially expressed annotated transcripts. Specificity was defined as percentage of "non-neighbor" assembled transcripts that were correctly called not differentially expressed, where "non-neighbor" means the assembled transcript was not one of the three closest to an annotated transcript set to be differentially expressed.

Results

The results for the first simulation, where the differential expression was set at the FPKM level, were that *Cuffdiff* (2.2.1) showed the same conservative bias we observed in the negative control experiment and possibly in the InSilico DB experiments. Using the q-value as a significance cutoff, *Cuffdiff2* called 1 transcript differentially expressed (controlling FDR at the 5% level), compared to 56 using *Ballgown* 's F-test (Supplementary Note 2). Accordingly, the p-value distributions showed similar patterns to those we observed in the adenocarcinoma and developmental cell datasets (Supplementary Figure 6a). While the accuracy of the transcript rankings was comparable for both methods – for the linear models in *Ballgown*, 81 of the top 100 transcripts called differentially expressed were truly differentially expressed for *Ballgown* versus 85 for *Cuffdiff* 2.2.1 – an ROC curve based on q-value cutoff shows *Ballgown* outperforming *Cuffdiff2* in terms of sensitivity and specificity (Supplementary Figure 6b).

We hypothesize that transcript length normalization may have something to do with the problems observed in *Cuffdiff2*'s statistical significance estimation, because in the second simulation scenario, where the number of reads sampled from each transcript was independent of its length, *Cuffdiff2* performed comparably to the linear model framework included in *Ballgown*, and both seemed to be performing accurately. The p-value histograms for both methods showed uniformly-distributed p-values at the high end and some signal at the low end, as expected (Supplementary Figure 6c), and the ROC curves are approximately equivalent; both display high sensitivity and specificity (Supplementary Figure 6d). In this scenario, of the top 100 transcripts ranked by each method, 96 are truly differentially expressed for *Cuffdiff2* and 91 are for the linear models implemented in *Ballgown.* This shows that the models implemented in *Cuffdiff2* are accurate under some conditions – e.g., when the number of sequencing reads from a transcript is unrelated to its length – that may be somewhat unrealistic.

Supplementary Figure 6

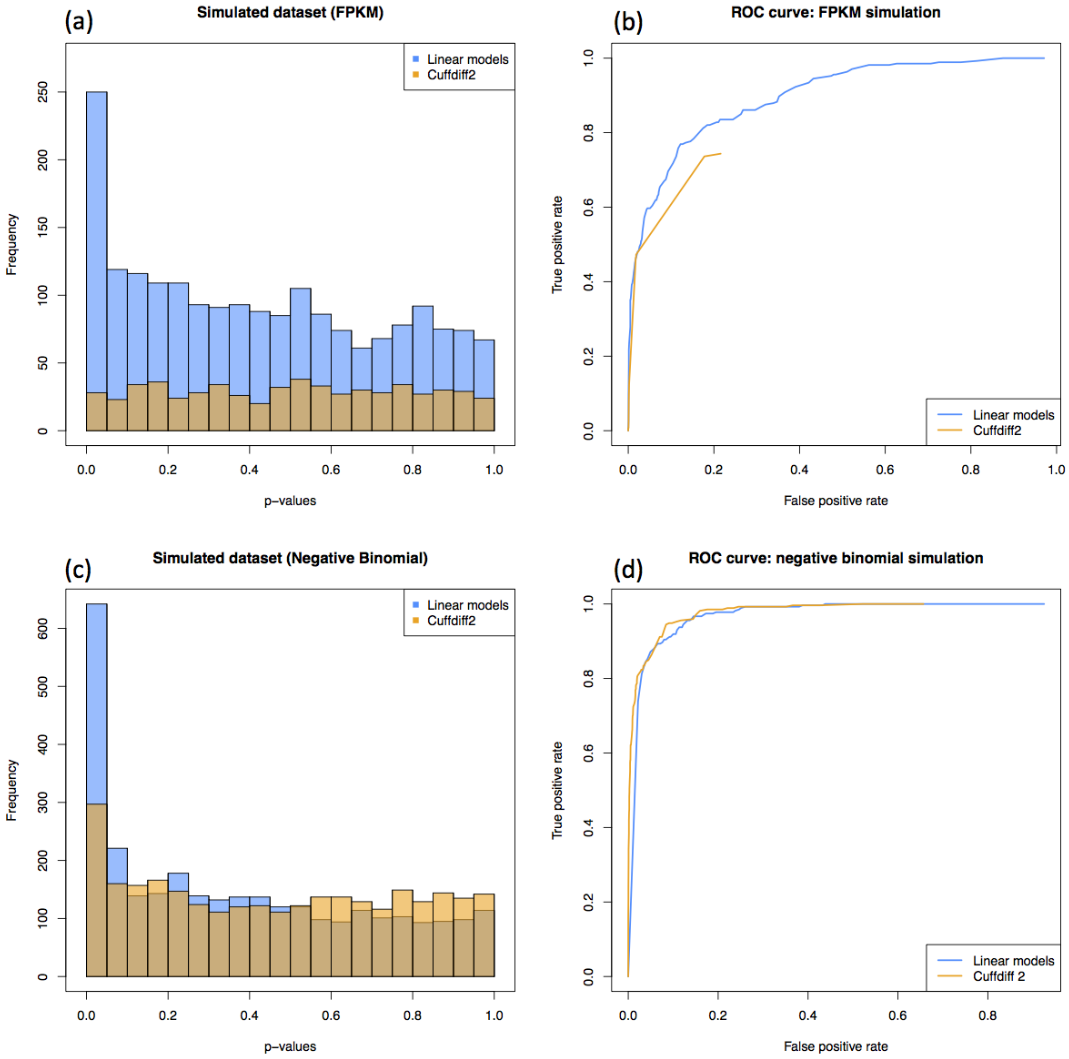

Figure 6: **Comparison of statistical significance between** *Cuffdiff2* **and linear models in** *Ballgown* **in simulated datasets a.** Histograms of p-values from a simulated data set of 2,745 transcripts where differential expression was induced between 10 cases and 10 controls in 10% of transcripts at the FPKM level (*Ballgown* in blue, *Cuffdiff2* in orange). **b.** ROC curve comparing the abilities of *Cuffdiff2* and linear modeling to identify differentially expressed transcripts in the FPKM simulation based on q-value. **c.** Histograms of p-values from a simulated data set of 2,745 transcripts in 10 cases and 10 controls, where 10% of transcripts were simulated to be differentially expressed, but the number of reads generated from each transcript was independent of transcript length. **d.** ROC curve comparing the abilities of *Cuffdiff2* and linear modeling to identify differentially expressed transcripts in the transcript-length-independent simulation study.

Supplementary Note 6: Comparison of average coverage and FPKM as expression measurements in differential expression studies

We compared the previous differential expression results, which were based on measuring transcript abundance using FPKM, with analyses using average per-base read coverage as the transcript expression measurement instead. Doing this comparison was straightforward, since *Tablemaker* outputs FPKM estimates from *Cufflinks* along with a variety of other expression measurements for the features of a transcriptome. We first did a comparison between FPKM and average coverage using First, we used our simulated dataset to investigate the impact of using average coverage as the transcript expression measurement, compared to using FPKM, as was done in our previous analyses. To do this comparison, we re-analyzed the data we simulated in the scenario where differential expression occurred at the FPKM level (Supplementary Note 5, first simulation; Supplementary Figure 6a-b) but used average coverage as the transcript-level expression measurement. The differential expression rankings measuring transcript expression with FPKM and with average coverage were highly correlated (Supplementary Figure 7a), with a correlation coefficient of 0.66. The p-value distribution using average coverage (Supplementary Figure 7b) was similar to the p-value distribution using FPKM (Supplementary Figure 6a), though only 25 transcripts were found to be differentially expressed ($q < 0.05$), compared to 56 using FPKM. We also observed correlated ranks ($r = 0.57$) between the differential expression results testing whether RIN value affected expression in the GEUVADIS dataset (Supplementary Figure 7c). These results confirm that in differential expression analyses, count-based and FPKM-based (length-normalized) expression measurements perform similarly. *Ballgown* allows users to perform analyses with whatever expression measurements are available for their transcriptome, so for example, RSEM [20] users can use such as transcripts per million (TPM) [29] as an expression measurement. The framework also facilitates easy exploration of the different measurement options.

Supplementary Figure 7

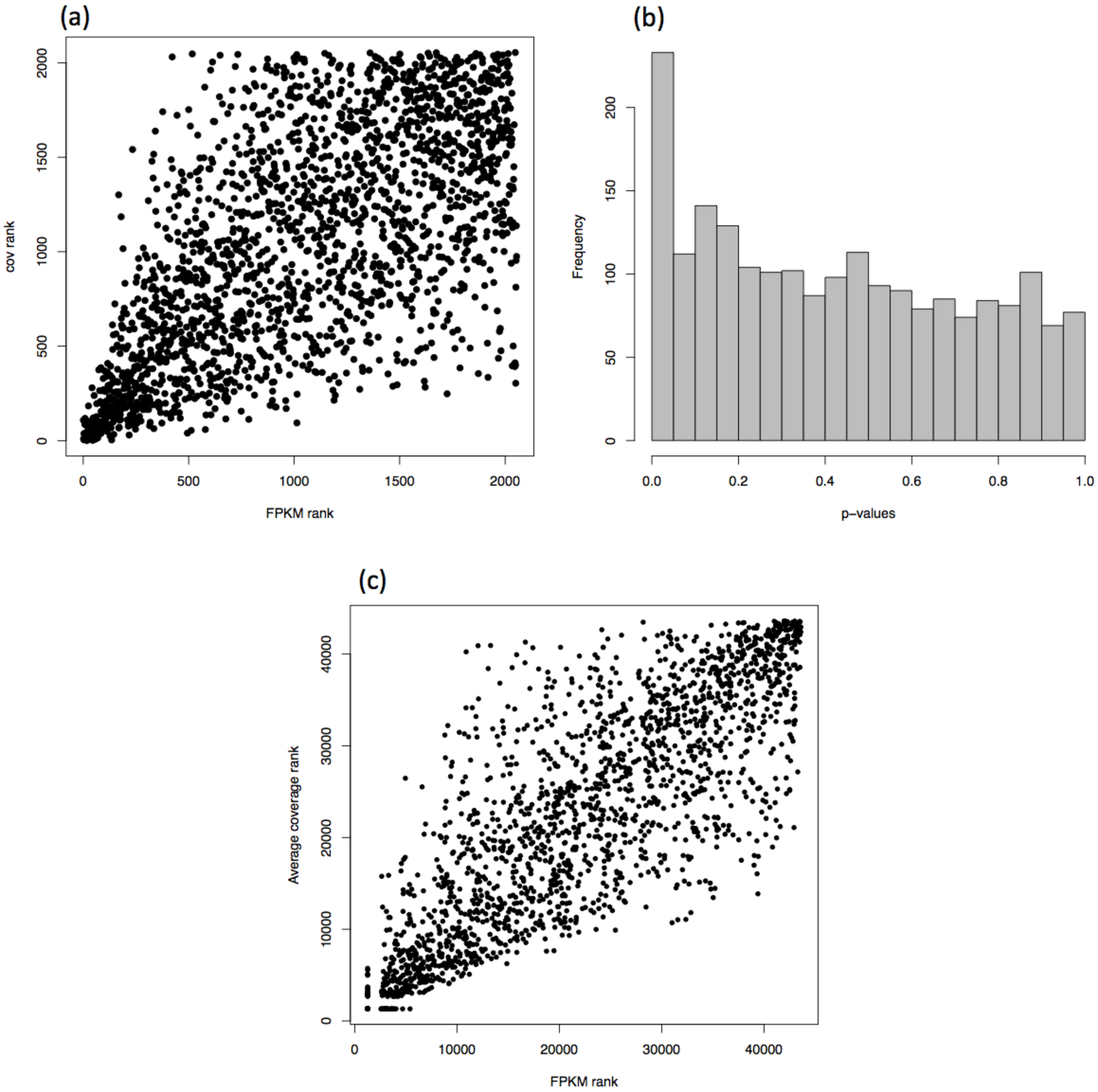

Figure 7: **Using average per-base coverage as transcript expression measurement instead of FPKM. a.** Differential expression ranks for transcripts in a case/control simulation ($n = 10$ per group), using FPKM as the expression measurement (x-axis) vs. using average coverage (y-axis). **b.** Distribution of p-values from differential expression tests between the 10 cases and 10 controls, using average coverage as the expression measurement. This distribution is very similar to the distribution observed when using FPKM as the expression measurement (Figure 6a). **c.** Rankings of the effect of RIN on transcript expression in the GEUVADIS dataset, using FPKM as the transcript expression measurement (x-axis) vs. using average coverage (y-axis). For visibility, 2000 transcripts were randomly sampled from the dataset for the plot.

Supplementary Note 7: RIN Analysis

We filtered to the 464 unique replicates in the GEUVADIS study [15, 1] as indicated in the quality control data, and we analyzed only transcripts with FPKM > 0.1. We first searched for differential expression with respect to RNA quality (RIN) using the following set of nested linear models to each transcript.

$$\begin{aligned} H_A &: \log_2(FPKM_i + 1) = \beta_0^* + \sum_{t=1}^{4} \beta_t^* \text{spline}_t(RIN_i) + \sum_{p=1}^{5} \gamma_p^* 1(Pop_i = p) + \eta^* q75_i + \epsilon_i^* \\ H_0 &: \log_2(FPKM_i + 1) = \beta_0 + \sum_{p=1}^{5} \gamma_p 1(Pop_i = p) + \eta q75_i + \epsilon_i \end{aligned}$$

Here i indicates sample and the subscript for transcript has been suppressed for clarity. H_0 denotes the null model and H_A denotes the alternative. The first set of terms encode a natural cubic spline fit with 4 degrees of freedom between the RIN values and the FPKM levels; the term $\text{spline}_t(RIN_i)$ refers to the tth B-spline basis term for sample i. The second set of terms encode a factor model for the relationship between population and FPKM and the last term is a library size normalization term that consists of the sum of log of the the non-zero FPKMs up to the 75th percentile for that sample (”cumulative sum scaling” normalization; [21]). We then tested the hypothesis that $H_0 : \beta_1 = \beta_2 = \beta_3 = 0$ versus the alternative that at least one coefficient was non-zero. All transcripts with a q-value [25] less than 0.05 were called significant.

Next we attempted to identify transcripts that were significantly better explained by a non-linear polynomial fit, rather than a linear trend. We fit the following nested set of models:

$$H_A : \log_2(FPKM_i + 1) = \beta_0^* + \sum_{t=1}^{3} \beta_t^* RIN_i^t + \sum_{p=1}^{5} \gamma_p^* 1(Pop_i = p) + \eta^* q75_i + \epsilon_i^* \quad (2)$$

$$H_0 : \log_2(FPKM_i + 1) = \beta_0 + \beta_1 RIN_i + \sum_{p=1}^{5} \gamma_p 1(Pop_i = p) + \eta q75_i + \epsilon_i \quad (3)$$

and tested the hypothesis that $H_0 : \beta_2 = \beta_3 = 0$ versus the alternative that at least one of the higher order polynomial coefficients was nonzero. Again, all transcripts with a q-value [25] less than 0.05 were called significant.

The transcripts in Figure 1c and 1d in the main manuscript were statistically significant at the FDR 5% level for this second analysis. In Figures 1c and 1d, the curves represent the fitted values for the average library size within each population. We show one example each of a positive and negative relationship between expression and RIN. While there were several examples of associations in both directions, there were more positive associations, as expected (Supplementary Figure 8).

Supplementary Figure 8

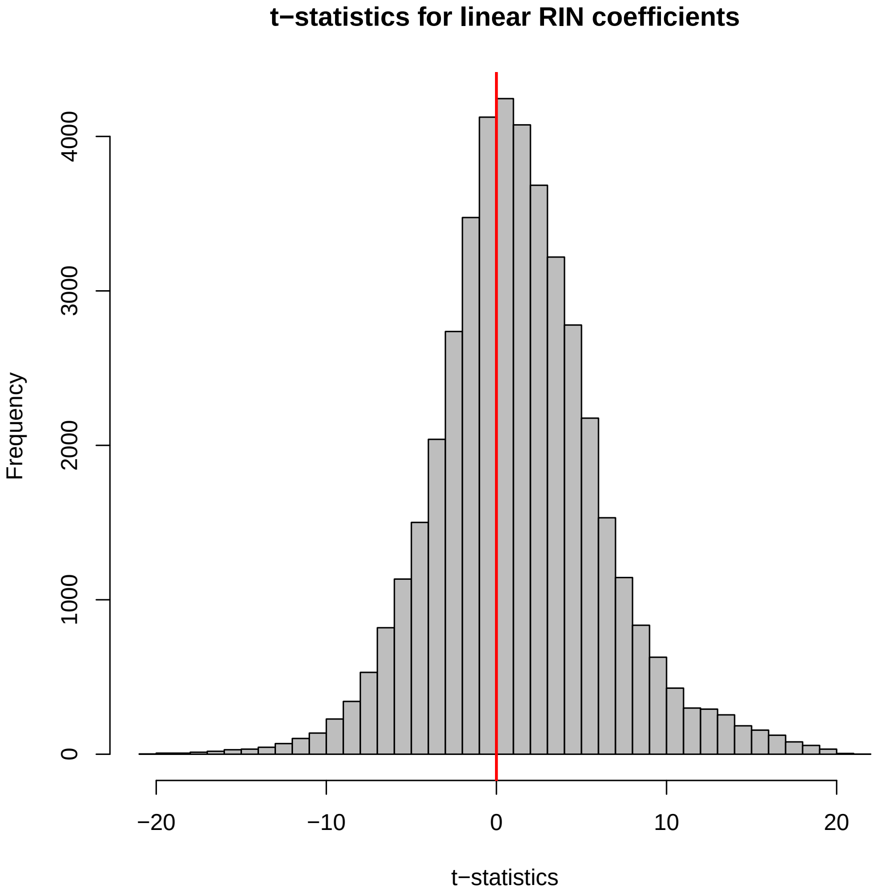

Figure 8: **Distribution of t-statistics for the linear RIN term for GEUVADIS transcripts.** These are moderated t-statistics calculated with *limma* for the β_1 coefficient in model (3), indicating directionality of the RIN-FPKM relationship. We observe associations in both directions, but as expected, there are more positive associations.

Supplementary Note 8: eQTL Analysis

We downloaded genotype information for the GEUVADIS cohort from `ftp://ftp.ebi.ac.uk/pub/databases/microarray/data/experiment/GEUV/E-GEUV-1/genotypes/`. We filtered to only SNPs with a minor allele frequency greater than 5%. We used the processed transcriptome data from *Tablemaker* as described above. We removed samples that were sequenced multiple times according to the protocol described by GEUVADIS [1]. We calculated the first three principal components of the genotype data using the Plink software [22]. We filtered to transcripts with an average FPKM $>$ 0.1 and took the log2 transform of the FPKM values. We then used the MatrixEQTL package [23] to perform the eQTL analysis testing an additive linear regression model for the SNPs adjusting for three expression principal components and three genotype principal components. We filtered to only transcript-SNP pairs that were no more than 1000Kb apart.

We recorded the histogram of p-values from all transcript-SNP pairs. We calculated an estimate of the fraction of null hypotheses based on the distribution of observed p-values [25] and obtained an estimate of $\hat{\pi}_0 = 0.942$. The p-value histogram (Supplementary Figure 9a) and QQ-plot of -log10(p-values) (Supplementary Figure 9b) versus their theoretical distribution under the null do not show any gross deviation suggesting unmodeled confounding [7].

For the transcript overlap analysis, we downloaded the list of significant cis-eQTL from `ftp://ftp.ebi.ac.uk/pub/databases/microarray/data/experiment/GEUV/E-GEUV-1/genotypes/` for the EUR and YRI populations. We identified all Ensembl genes overlapped to any degree by each assembled transcript. We then calculated the number of gene-SNP pairs in common between the GEUVADIS EUR and YRI analyses and our eQTL analysis.

Supplementary Figure 9

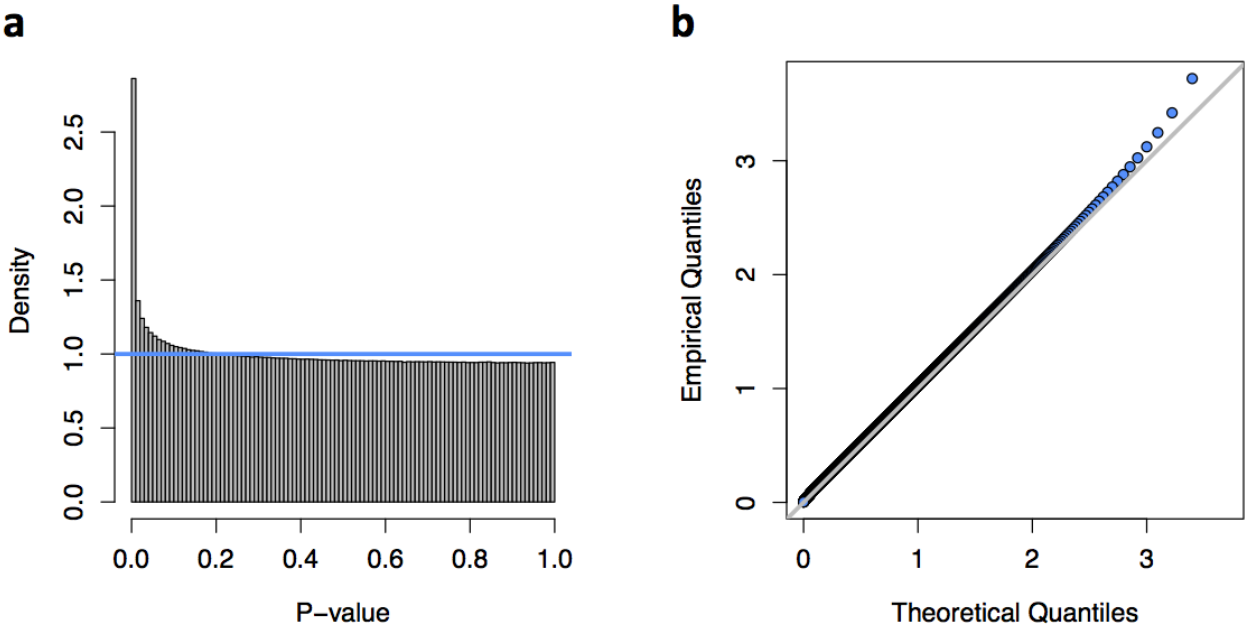

Figure 9: **Distribution of statistical significance scores for all cis-eQTL tests a.** P-value histogram for all p-values from cis-eQTL tests, the estimated fraction of null hypotheses is 94.2%. **b.** QQ-plot of -log10(p-values) versus theoretical quantiles shows no gross deviation from expected behavior.

Supplementary Note 9: Computational efficiency and timing results

Next we investigated the computational efficiency of our approach compared to the standard *Cufflinks* pipeline. *Tophat* and *Cufflinks* can be run on each sample separately, but *Cuffdiff2* must be run on all samples simultaneously. While *Cuffdiff2* can make use of many cores on a single computer, is not parallelizable across computers. It has been noted that *Cuffdiff2* can take weeks or longer to run on experiments with a few hundred samples. This issue has led consortia and other groups to rely on unpublished software for transcript abundance estimation[1, 6].

We compared each component of the pipeline in terms of computational time on one of our simulated dataset (the second, simpler scenario) with 20 samples and 2,745 transcripts. The *Tophat2* - *Cufflinks* - *Tablemaker*-*Ballgown* pipeline was fastest, taking about 5.4 minutes per sample for *Tablemaker*, 2.3 seconds to load transcript data into R and less than 0.1 seconds for differential expression analysis. This is faster than the recently published *Tophat2* - *Cufflinks* - *Cuffquant* - *Cuffdiff2* pipeline [26], which required about 3 minutes per sample for *Cuffquant* and 19 minutes for differential expression analysis with *Cuffdiff2*. The *Ballgown* - *Tablemaker* pipeline was also substantially faster than directly running *Cufflinks* - *Cuffdiff2* , where the *Cuffdiff2* step took about 68 minutes. For all these pipelines, *Tophat2* took about 1 hour per sample and *Cufflinks* about 2 minutes per sample. All possible multicore processes (*TopHat*, *Cufflinks*, *Cuffdiff2*, *Cuffquant*, *Tablemaker*) were run on 4 cores.

We also calculated the per-sample distribution of processing times for each step in the *Tophat2* - *Cufflinks* - *Tablemaker* pipeline for all 667 samples in the GEUVADIS study [15] (Supplementary Figure 10). *Tablemaker* took a median of 0.97 hours per sample (IQR 0.24 hours) on a standard 4 core computer; this calculation can be parallelized across samples. By contrast, *Cuffdiff2* would take months to perform this analysis on a standard 4 core computer. *Ballgown* multiclass differential expression analysis between the CEU ($n = 162$), YRI ($n = 163$), FIN ($n = 114$), GBR ($n = 115$) and TSI ($n = 93$) samples for 334,206 transcripts took 42 minutes on a single core Desktop computer.

Supplementary Figure 10

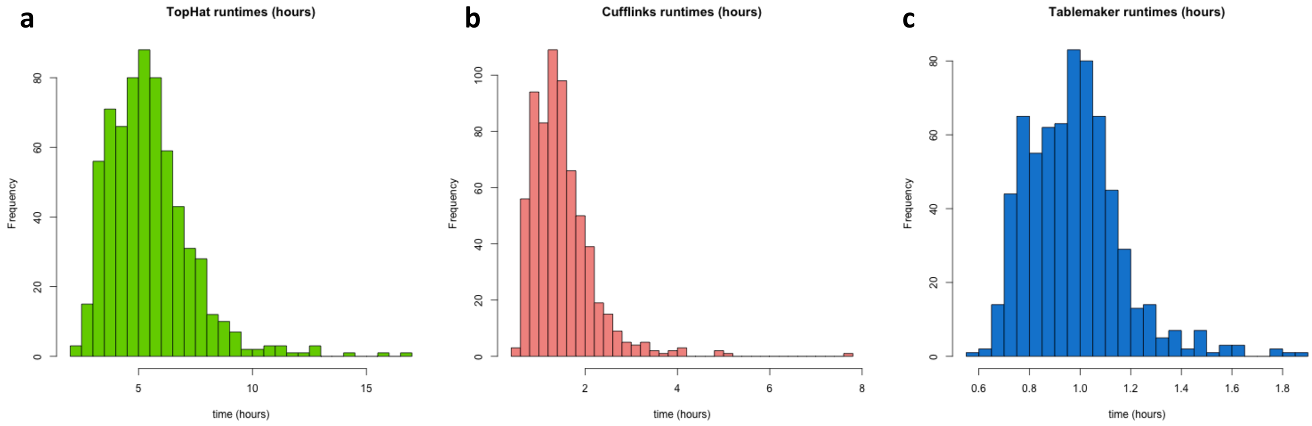

Figure 10: **Timing results for the 667 GEUVADIS samples at each stage of the pipeline. a.** Timing (in hours) for each sample to run through *TopHat2*. **b.** Timing (in hours) for each sample to run through *Cufflinks*. **c.** Timing (in hours) for each sample to run through *Tablemaker*.

Supplementary Note 10: Software

1. *Ballgown* - Available from Bioconductor [10]: `http://www.bioconductor.org/packages/release/bioc/html/ballgown.html` Installation instructions and tutorial for use are available in the package vignette, and at `https://github.com/alyssafrazee/ballgown`

2. *Tablemaker* - Linux binary: `http://figshare.com/articles/Tablemaker_Linux_Binary/1053137`; Mac OS X binary: `http://figshare.com/articles/Tablemaker_OS_X_Binary/1053136`; source code/installation instructions: `https://github.com/alyssafrazee/tablemaker`

3. *polyester* - Available from Bioconductor [10]: `http://www.bioconductor.org/packages/release/bioc/html/polyester.html`. Expanded development version available at `https://github.com/alyssafrazee/polyester`

Supplementary Note 11: Scripts and Data

Processing scripts and links to all data are available at: `https://github.com/alyssafrazee/ballgown_code/`

s

References

[1] Peter AC't Hoen, Marc R Friedländer, Jonas Almlöf, Michael Sammeth, Irina Pulyakhina, Seyed Yahya Anvar, Jeroen FJ Laros, Henk PJ Buermans, Olof Karlberg, Mathias Brännvall, et al. Reproducibility of high-throughput mrna and small rna sequencing across laboratories. *Nature biotechnology*, 2013.

[2] Simon Anders and Wolfgang Huber. Differential expression analysis for sequence count data. *Genome biol*, 11(10):R106, 2010.

[3] George EP Box and David R Cox. An analysis of transformations. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 211–252, 1964.

[4] Alain Coletta, Colin Molter, Robin Duqué, David Steenhoff, Jonatan Taminau, Virginie De Schaetzen, Stijn Meganck, Cosmin Lazar, David Venet, Vincent Detours, et al. Insilico db genomic datasets hub: an efficient starting point for analyzing genome-wide studies in genepattern, integrative genomics viewer, and r/bioconductor. 2012.

[5] 1000 Genomes Project Consortium et al. An integrated map of genetic variation from 1,092 human genomes. *Nature*, 491(7422):56–65, 2012.

[6] Manolis Dermitzakis, Gad Getz, Kristin Ardle, Roderic Guigo, and for the GTEx consortium. Response to: "gtex is throwing away 90% of their data". `http://liorpachter.wordpress.com/2013/10/31/response-to-gtex-is-throwing-away-90-of-their-data/`, 2013.

[7] B Devlin and Kathryn Roeder. Genomic control for association studies. *Biometrics*, 55(4):997–1004, 1999.

[8] Paul Flicek, M Ridwan Amode, Daniel Barrell, Kathryn Beal, Konstantinos Billis, Simon Brent, Denise Carvalho-Silva, Peter Clapham, Guy Coates, Stephen Fitzgerald, et al. Ensembl 2014. *Nucleic acids research*, 42(D1):D749–D755, 2014.

[9] Alyssa C Frazee, Sarven Sabunciyan, Kasper D Hansen, Rafael A Irizarry, and Jeffrey T Leek. Differential expression analysis of rna-seq data at single-base resolution. *Biostatistics*, page kxt053, 2014.

[10] Robert C Gentleman, Vincent J. Carey, Douglas M. Bates, and others. Bioconductor: Open software development for computational biology and bioinformatics. *Genome Biology*, 5:R80, 2004.

[11] Trevor Hastie and Robert Tibshirani. Generalized additive models. *Statistical science*, pages 297–310, 1986.

[12] Andrew E. Jaffe, Alyssa C. Frazee, and Jeffrey T. Leek. *Polyester: Simulate RNA-seq reads.* R package version 1.0.0.

[13] Sang Cheol Kim, Yeonjoo Jung, Jinah Park, Sooyoung Cho, Chaehwa Seo, Jaesang Kim, Pora Kim, Jehwan Park, Jihae Seo, Jiwoong Kim, et al. A high-dimensional, deep-sequencing study of lung adenocarcinoma in female never-smokers. *PloS one*, 8(2):e55596, 2013.

[14] Ben Langmead, Kasper D Hansen, Jeffrey T Leek, et al. Cloud-scale rna-sequencing differential expression analysis with myrna. *Genome Biol*, 11(8):R83, 2010.

[15] Tuuli Lappalainen, Michael Sammeth, Marc R Friedländer, Peter AC't Hoen, Jean Monlong, Manuel A Rivas, Mar Gonzàlez-Porta, Natalja Kurbatova, Thasso Griebel, Pedro G Ferreira, et al. Transcriptome and genome sequencing uncovers functional variation in humans. *Nature*, 2013.

[16] Michael Lawrence, Wolfgang Huber, Hervé Pagès, Patrick Aboyoun, Marc Carlson, Robert Gentleman, Martin T Morgan, and Vincent J Carey. Software for computing and annotating genomic ranges. *PLoS computational biology*, 9(8):e1003118, 2013.

[17] Jeffrey T Leek and John D Storey. Capturing heterogeneity in gene expression studies by surrogate variable analysis. *PLoS Genetics*, 3(9):e161, 2007.

[18] Rasko Leinonen, Hideaki Sugawara, and Martin Shumway. The sequence read archive. *Nucleic acids research*, page gkq1019, 2010.

[19] Ning Leng, John A Dawson, James A Thomson, Victor Ruotti, Anna I Rissman, Bart MG Smits, Jill D Haag, Michael N Gould, Ron M Stewart, and Christina Kendziorski. Ebseq: an empirical Bayes hierarchical model for inference in RNA-seq experiments. *Bioinformatics*, 29(8):1035–1043, 2013.

[20] Bo Li and Colin Dewey. Rsem: accurate transcript quantification from rna-seq data with or without a reference genome. *BMC bioinformatics*, 12(1):323, 2011.

[21] Joseph N Paulson, O Colin Stine, Héctor Corrada Bravo, and Mihai Pop. Differential abundance analysis for microbial marker-gene surveys. *Nature methods*, 2013.

[22] Shaun Purcell, Benjamin Neale, Kathe Todd-Brown, Lori Thomas, Manuel AR Ferreira, David Bender, Julian Maller, Pamela Sklar, Paul IW De Bakker, Mark J Daly, et al. Plink: a tool set for whole-genome association and population-based linkage analyses. *The American Journal of Human Genetics*, 81(3):559–575, 2007.

[23] Andrey A Shabalin. Matrix eqtl: ultra fast eqtl analysis via large matrix operations. *Bioinformatics*, 28(10):1353–1358, 2012.

[24] Gordon K Smyth et al. Linear models and empirical bayes methods for assessing differential expression in microarray experiments. *Stat Appl Genet Mol Biol*, 3(1):3, 2004.

[25] John D Storey and Robert Tibshirani. Statistical significance for genomewide studies. *Proceedings of the National Academy of Sciences*, 100(16):9440–9445, 2003.

[26] C. Trapnell, D. Cacchiarelli, J. Grimsby, P. Pokharel, S. Li, M. Morse, N. J. Lennon, K. J. Livak, T. S. Mikkelsen, and J. L. Rinn. The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nat. Biotechnol.*, Mar 2014.

[27] Cole Trapnell, David G Hendrickson, Martin Sauvageau, Loyal Goff, John L Rinn, and Lior Pachter. Differential analysis of gene regulation at transcript resolution with rna-seq. *Nature biotechnology*, 31(1):46–53, 2012.

[28] Cole Trapnell, Brian A Williams, Geo Pertea, Ali Mortazavi, Gordon Kwan, Marijke J van Baren, Steven L Salzberg, Barbara J Wold, and Lior Pachter. Transcript assembly and quantification by rna-seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nature biotechnology*, 28(5):511–515, 2010.

[29] Günter P Wagner, Koryu Kin, and Vincent J Lynch. Measurement of mRNA abundance using RNA-seq data: RPKM measure is inconsistent among samples. *Theory in Biosciences*, 131(4):281–285, 2012.

[30] Liying Yan, Mingyu Yang, Hongshan Guo, Lu Yang, Jun Wu, Rong Li, Ping Liu, Ying Lian, Xiaoying Zheng, Jie Yan, et al. Single-cell rna-seq profiling of human preimplantation embryos and embryonic stem cells. *Nature structural & molecular biology*, 2013.