Artemis: Rapid and Reproducible RNAseq Analysis for End Users

Anthony Colombo[1], Timothy J. Triche Jr.[1], and Giridharan Ramsingh[1]

[1] Jane Anne Nohl Division of Hematology, Keck School of Medicine of USC, Los Angeles, California, USA

The recently introduced Kallisto[1] pseudoaligner has radically simplified the quantification of transcripts in RNA-sequencing experiments. However, as with all computational advances, reproducibility across experiments requires attention to detail. The elegant approach of Kallisto reduces dependencies, but we noted differences in quantification between versions of Kallisto, and both upstream preparation and downstream interpretation benefit from an environment that enforces a requirement for equivalent processing when comparing groups of samples. Therefore, we created the Artemis[3] and TxDbLite[4] R packages to meet these needs and to ease cloud-scale deployment of the above. TxDbLite extracts structured information directly from source FASTA files with per-contig metadata, while Artemis enforces versioning of the derived indices and annotations, to ensure tight coupling of inputs and outputs while minimizing external dependencies. The two packages are combined in Illumina's BaseSpace cloud computing environment to offer a massively parallel and distributed quantification step for power users, loosely coupled to biologically informative downstream analyses via gene set analysis (with special focus on Reactome annotations for ENSEMBL transcriptomes). Previous work (e.g. Soneson et al., 2016[33]) has revealed that filtering transcriptomes to exclude lowly-expressed isoforms can improve statistical power, while more-complete transcriptome assemblies improve sensitivity in detecting differential transcript usage. Based on earlier work by Bourgon et al., 2010[10], we included this type of filtering for both gene- and transcript-level analyses within Artemis. For reproducible and versioned downstream analysis of results, we focused our efforts on ENSEMBL and Reactome[2] integration within the qusage[18] framework, adapted to take advantage of the parallel and distributed environment in Illumina's BaseSpace cloud platform. We show that quantification and interpretation of repetitive sequence element transcription is eased in both basic and clinical studies by just-in-time annotation and visualization. The option to retain pseudoBAM output for structural variant detection and annotation, while not insignificant in its demand for computation and storage, nonetheless provides a middle ground between *de novo* transcriptome assembly and routine quantification, while consuming a fraction of the resources used by popular fusion detection pipelines and providing options to quantify gene fusions with known breakpoints without reassembly. Finally, we describe common use cases where investigators are better served by cloud-based computing platforms such as BaseSpace due to inherent efficiencies of scale and enlightened common self-interest. Our experiences suggest a common reference point for methods development, evaluation, and experimental interpretation.

correspondence: tim.triche@usc.edu
keywords:transcription, sequencing, RNAseq, reproducibility, automation, cloud computing

1 Introduction

The scale and complexity of sequencing data in molecular biology has exploded in the 15 years following completion of the Human Genome Project[**?**]. Furthermore, as a dizzying array of *-Seq protocols have been developed to open new avenues of investigation, a much broader cross-section of biologists,

physicians, and computer scientists have come to work with biological sequence data. The nature of gene regulation (or, perhaps more appropriately, transcription regulation), along with its relevance to development and disease, has undergone massive shifts propelled by novel approaches, such as the discovery of evolutionarily conserved non-coding RNA by enrichment analysis of DNA[34] and isoform-dependent switching of protein interactions. Sometimes lost within this excitement, however, is the reality that biological interpretation these results can be highly dependent upon both their extraction and annotation. A rapid, memory-efficient approach to estimate abundance of both known and putative transcripts substantially broadens the scope of experiments feasible for a non-specialized laboratory and of methodological work deemed worthwhile given the pace of innovations. Recent work on the Kallisto pseudoaligner, among other k-mer based approaches, has resulted in exactly such an approach.

In order to leverage this advance for our own needs, which have included the quantification of repetitive element transcription in clinical trials, the comparison of hundreds of pediatric malignancies with their adult counterparts, and the analysis of single-cell RNA sequencing benchmarks, we first created an R package (Artemis) which automates the construction of composite transcriptomes from multiple sources, along with their digestion into a lightweight analysis environment. Further work led us to automate the extraction of genomic and functional annotations directly from FASTA contig comments, eliding sometimes-unreliable dependencies on services such as Biomart. Finally, in a nod to enlightened self-interest and mutual benefit, we collaborated with Illumina to deploy the resulting pipeline within their BaseSpace cloud computing platform. The underlying packages are freely available with extensive use-case vignettes on GitHub, are in preparation for Bioconductor submission and review, and available in beta for review upon request, pending full public BaseSpace deployment.

Current genomic experiments are increasing in complexity returning data reaching up to Terabytes and Petabytes in size over a field of high-performance computing (HPC) environments including clouds, grids, and graphics processing unit with varying parallelism techniques. HPC bioinformatic workflows have three main subfamilies: in-house computational packages, virtual-machines, and cloud based computational environments. The in-house computational packages may limit reproducible research because some HPC library versions are controlled by the HPC staff making it difficult to replicate experiments. Virtual-machines are defined as small allocations of hard drive space. Although virtual-machines are useful for transporting preserved development environments, transferring data files to small hard drive partitions presents technical challenges. Cloud based computational environments are remote computer clusters that are networked to storage databases. Recent advances in cloud computing from Amazon Elastic Compute Cloud (EC2) provided by Amazon Web Services (AWS) support computational environments and have attracted bioinformaticians and private businesses to develop portable work-flows over AWS high performance machine instances. AWS EC2 could be more cost effective compared to local clustered computing or grids [26]. Several recent high impact publications have used advanced cloud computing work flows such as CloudBio-linux, CloudMap, and Mercury over AWS EC2 [26]. CloudBio-linux software is centered around comparative genomics, phylogenomics, transcriptomics, proteomics, and evolutionary genomics studies using Perl scripts [26]. Although offered with limited scalability, the CloudMap software allows scientists to detect genetic variations over a field of virtual machines operating in parallel [26]. For comparative genomic analysis, the Mercury [27] workflow can be deployed within Amazon EC2 through instantiated virtual machines but is limited to BWA and produces a variant call file (VCF) without considerations of pathway analysis or comparative gene set enrichment analyses. The effectiveness for conducting genomic research is greatly influenced by the choice of computational environment.

We've encountered several bottlenecks for processing biological data such as read archive importing, work-flow reproducibility, and computational speed. Many high impact publications used in-house HPC nodes, yet in-house workflows limit reproducible research because they sit in as isolated computer instances, un-connected to sequencing archives hindering the ability to reconstruct identically versioned computational environments. Further, in-house computational workflows are not as portable as virtualized operating systems and require data transfers to the in-house software which is

cumbersome for Terabytes of biological data. Some cloud HPC software may connect with sequence reads archives (SRA), but their virtualized work flows are not up to date with leading quantification algorithms.

Current RNA-Seq analysis pipelines consist of read preparation steps followed by computationally expensive alignment algorithms which create performance bottlenecks. Software for calculating transcript abundance and assembly can surpass 30 hours of computational time [1]. One computational bottleneck can be reduced by using novel software Kallisto[1] which implements a pseudoalignment, for near-optimal RNAseq transcript quantification. Kallisto [1] can quantify transcripts from 30 million unaligned paired-end RNA-Seq reads in less than 5 minutes on a standard laptop. We designed a virtualized operating system using Docker that performs transcript abundance quantification using Kallisto into an integrative work-flow, Artemis, that includes pathway analysis and rapid set enrichment analysis over Illumina's BaseSpace Platform.

Illumina's BaseSpace Platform is an HPC cloud environment that utilizes AWS. Illumina's BaseSpace Platform utilizes AWS cc2 8x-large instance where each node has 64-bit processor arch, with virtual storage of 3360 Gigabytes; published applications on BaseSpace can allocate up to 100 nodes distributing an analysis in parallel for mutli-node applications. BaseSpace allows for direct importing from SRA, making large sample importing more efficient, fostering a critical environment for re-analyzing derivations from published data. We reduce a second bottleneck because it is easier to move bioinformatic software files as opposed to transferring Gigabytes of sequenced reads. For example, we've reproduced raw reads from an experiment from SRA with a size of 141.3 Gigabytes, however the analysis files needed for downstream production were 1.63 Gigabytes which is far more portable. We selected Illumina's BaseSpace Platform because the development application platform preserves, tests, and provides public cloud software which encourages reproducible research.

2 Materials

2.1 Kallisto

Kallisto [1] quantifies transcript abundance from input RNA-Seq reads by using a process known as pseudoalignment which identifies the read-transcript compatibility matrix. The compatibility matrix is formed by counting the number of reads with the matching alignment; the equivalence class matrix has a much smaller dimension compared to matrices formed by transcripts and read coverage. Computational speed is gained by performing the Expectation Maximization [30] (EM) algorithm over a smaller matrix.

2.2 Artemis

Artemis[3] is a BioConductor package that automates the index caching, annotation, and quantification associated with running the Kallisto pseudoaligner integrated within the BioConductor environment. Artemis can process raw reads into transcript- and pathway-level results within BioConductor or in Illuminas BaseSpace cloud platform.

Artemis[3] reduces the computational steps required to quantify and annotate large numbers of samples against large catalogs of transcriptomes. Artemis calls Kallisto[1] for on-the-fly transcriptome indexing and quantification recursively for numerous sample directories. For RNA-Seq projects with many sequenced samples, Artemis encapsulates expensive transcript quantification preparatory routines, while uniformly preparing Kallisto [1] execution commands within a versionized environment encouraging reproducible protocols.

Artemis [3] integrates quality control analysis for experiments that include Ambion spike-in controls defined from the External RNA Control Consortium [29][28]. Artemis includes erccdashboard [14], a BioConductor package for analyzing data sequenced with ERCC (Ambion) spike-ins. We designed

a function titled 'erccAnalysis' which produces Receiver Operator Characteristic (ROC) Curves by preparing differential expression testing results for spike-in analysis.

Artemis imports the data structure from SummarizedExperiment[15] and creates a sub-class titled KallistoExperiment which preserves the S4 structure and is convenient for handling assays, phenotypic and genomic data. KallistoExperiment includes GenomicRanges[31], preserving the ability to handle genomic annotations and alignments, supporting efficient methods for analyzing high-throughput sequencing data. The KallistoExperiment sub-class serves as a general purpose container for storing feature genomic intervals and pseudoalignment quantification results against a reference genome called by Kallisto [1]. By default KallistoExperiment couples assay data such as the estimated counts, effective length, estimated median absolute deviation, and transcript per million count where each assay data is generated by a Kallisto [1] run; the stored feature data is a GRanges object from GenomicRanges[31], storing transcript length, GC content, and genomic intervals. Artemis[3] is a portable work-flow that includes a routine 'SEtoKE' which casts a SummarizedExperiment [15] object into a KallistoExperiment object handy for general pathway analysis, transcript- and/or gene-wise analysis.

Given a KallistoExperiment, downstream enrichment analysis of bundle-aggregated transcript abundance estimates are performed using QuSage [18] imported from BioConductor. For gene-set enrichment analysis, Qusage [18] calculates the variance inflation factor which corrects the inter-gene correlation that results in high Type 1 errors using pooled, or non-pooled variances between sample groups. We've customized the Qusage [18] algorithm accelerating its computational speed improving its performance on average by 1.34X (Figure 5)(Table 6) 5. We improved QuSage's [18] performance using RcppArmadillo[19] modifying only the calculations for the statistics defined for Welch's test for unequal variances between sample groups; the shrinkage of the pooled variances is performed using the CAMERA algorithm within limma[16].

Artemis'[3] accelerated enrichment analysis is useful for analyzing large gene-sets from Molecular Signatures DataBase MSigDB [20] Reactome [8], and/or other signature gene sets simultaneously.

Pathway enrichment analysis can be performed from the BaseSpace cloud system downstream from parallel differential expression analysis. BaseSpace Cloud Platform offers Advaita among one of the many published computational software publicly available. Advaita offers an extensive pathway analysis software available within the BaseSpace environment, so we've customized Artemis' routine 'geneWiseAnalysis.R' to output differential expression formatted for downstream importing into Advaita.

2.3 TxDbLite

The choice of catalog, the type of quantification performed, and the methods used to assess differences can profoundly influence the results of sequencing analysis. Ensembl reference genomes are provided to GENCODE as a merged database from Havana's manually curated annotations with Ensembl's automatic curated coordinates. AceView, UCSC, RefSeq, and GENCODE have approximately twenty thousand protein coding genes, however AceView and GENCODE have a greater number of protein coding transcripts in their databases. RefSeq and UCSC references have less than 60,000 protein coding transcripts, whereas GENCODE has 140,066 protein coding loci. AceView has 160,000 protein coding transcripts, but this database is not manually curated [5]. The database selected for protein coding transcripts can influence the amount of annotation information returned when querying gene/transcript level databases.

Although previously overlooked, non-coding RNAs have been shown to share features and alternate splice variants with mRNA revealing that non-coding RNAs play a central role in metastasis, cell growth and cell invasion [23]. Non-coding transcripts have been shown to be functional and are associated with cancer prognosis; proving the importance of studying ncRNA transcripts.

Each non-coding database is curated at different frequencies with varying amounts of non-coding RNA enteries that influences that mapping rate. GENCODE Non-coding loci annotations contain 9640 loci, UCSC with 6056 and 4888 in RefSeq [5]; GENCODE annotations have the greatest number of lncRNA, protein and non-coding transcripts, and highest average transcripts per gene, with 91043

transcripts unique to GENCODE absent for UCSC and RefSeq databases [5]. Ensembl and AceView annotate more genes in comparison to RefSeq and UCSC, and return higher gene and isoform expression labeling improving differential expression analyses [6]. Ensembl achieves conspicuously higher mapping rates than RefSeq, and has been shown to annotate larger portions of specific genes and transcripts that RefSeq leaves unannotated [6]. Ensembl/GENCODE annotations are manually curated and updated more frequently than AceView. Further Ensembl has been shown to detect differentially expressed genes that is approximately equivalent to AceView [6].

Although Repetitive elements comprise over two-thirds of the human genome, repeat elements were considered non-functional and were not studied closely. Alu (Arthrobacter luteus) elements are a subfamily of repetitive elements that are roughly 300 base pairs long making up 11 % of the human genome [24]. Alu elements are the most abundant retro-transposable element, shown to increase genomic diversity impacting the evolution of primates and contributing to human genetic diseases [24]. Alu elements have been shown to influence genomic diversity through genetic instability. Regarding breast and ovarian tumorigenesis, the infamous BRCA1 and BRCA2 genes associated with survivability and prognosis contain genomic regions with very high densities of repetitive DNA elements [25].

Our understanding of biology is deepened when investigating expression of transcripts that include a merged transcriptome of coding, non-coding, and repetitive elements from data bases frequently curated. A *complete* transcriptome achieves higher mapping rates over larger catalogs of transcript families. We present TxDbLite[4] a fast genomic ranges annotation package that is included within the Artemis workflow that annotates non-coding, and repetitive elements on-the-fly downstream of transcript quantification steps.

TxDbLite [4] is a minimalist annotation package generator, designed to extract annotations from FASTA files. The underlying assumption is that users want to quantify transcript-level RNA expression, may or may not have a GTF file for each transcriptome, and would like to extract as much information as possible about the transcripts from what they do have. This in turn allows the Artemis[3] package to automatically determine what transcriptomes a dataset came from, whether those are already known to the package, and how to generate certain types of annotation for specific transcriptomes from Ensembl, RepBase, and ERCC (spike-in controls from Ambion).

3 Methods

3.1 Artemis

Illumina sequencers generate demultiplexed FASTQ files; Artemis [3] programmatically orders file inputs as required by Kallisto into sets of demultiplexed reads. The routine 'runKallisto'[3] pairs the corresponding demultiplexed reads, builds and caches an index consisting of External RNA Control Consortium (ERCC), non-coding RNA, and coding RNA, & RepBase repeatome transcripts. Quantification is issued against the transcriptome per-sample abundance, and individual quantified samples are merged into a single KallistoExperiment from the 'mergeKallisto'[3] routine. The merged KallistoExperiment identifies genes from 'collapseBundles'[3] method which collapses the merged bundles of transcripts by the respective gene ID contained in the FASTA reference, and discards any transcript that had less than one count across all samples. The total count for a successfully collapsed transcript bundle is aggregated from individual transcripts within bundles that passed the filtering criteria. The collapsed bundles are annotated using routines 'annotateFeatures'[3].

Standard approaches to modeling transcript-level differences in abundance rely upon having substantial numbers of replicates per condition. One of the novel features of Kallisto [1], implemented by Harold Pimentel (reference), is fast bootstrap sampling at the transcript level within the expectation-maximization algorithm. Artemis implements hooks to quantify the impact of this uncertainty in repeat elements and spike-in controls, where compositional analysis tools in the R environment [9] are pressed into service. Computational analysis with additive log-contrast formulations has long been

standard in geology and other fields. We are exploring its use as a within-bundle method to quantify the most prominent isoform-centric impacts in an expression analysis.

After obtaining bundled transcript aggregated counts labeled by the any arbitrary FASTA reference, a Gene-wise analysis was performed. The gene level analysis is invoked from 'geneWiseAnalysis'[3] which imports limma[16]) and edgeR [17] to normalize library sizes fitting an arbitrary marginal contrast, then propagates the resulting signed log10(p) values through clustered and un-clustered pathway enrichment analyses.

As previously, if the user has provided a grouping factor or design matrix, marginal significance for individual pathways and overall perturbation is assessed. Artemis discards transcripts and/or bundles with few reads (default 1) to improve statistical power [10].

3.2 BaseSpace

Artemis is currently written as a virtualized operating system, which can run on the BaseSpace platform generating the Kallisto pseudoaligned files. Artemis-BaseSpace can import SRA files and quantify transcript abundance.

3.3 TxDbLite

Gene annotation is performed from user-selected bundled transcriptomes (ERCC, Ensembl, and/or RepBase) simultaneously merging annotated samples into one R object: KallistoExperiment. We currently support reference databases for Homo sapiens and Mouse (NCBI). Routines such as 'annotateBundles' annotate transcriptomes from databases for example External RNA Control Consortium (ERCC), non-coding RNA, coding RNA, & RepBase repeatomes.

The design structure of Artemis versionizes the Kallisto [1] reference index to enforce that the Kallisto software versions are identical amidst merged KallistoExperiment data containers prior downstream analysis. Enforcing reference versions and Kallisto [1] versions prevents errors when comparing experiments. When the KallistoExperiment is generated, the Kallisto [1] version is stored within a data slot and can be accessed using the command 'kallistoVersion(KallistoExperiment)'. Before kallisto quantified data can be merged, Artemis first checks the Kallisto [1] index name and version from the $run\_info.json$ file and enforces matching version.

3.4 Quality Control Using ERCC-SpikeIns (Ambion)

Artemis workflow integrates the BioConductor package 'erccdashboard' [14] which tests for quality, and false positive rates. If the library preparation includes ERCC-Spike Ins, Artemis will generate useful receiver operator characteristic plots, average ERCC Spike Amount volume, comparison plots of ERCC volume Amount and normalized ERCC counts. Artemis method 'erccAnalysis' also includes Normalized Spike In Amount against Percent Differences, and most significantly 'erccAnalysis' plots FPR vs. TPR (Figures 2, 3, 4) 2, 3 4

4 Results

4.1 Data Variance

Artemis enforces matching Kallisto versions in order run and merge Kallisto quantified data [1]. In order to show the importance of enforcing the same Kallisto version, we've repeatedly ran Kallisto on the same 6 samples, quantifying transcripts against two different Kallisto versions and measured the percent differences and standard deviation between these runs [1]. We ran Kallisto quantification once with Kallisto version 0.42.1, and 10 times with version 0.42.3 merging each run into a KallistoExperiment and storing the 11 runs into a list of Kallisto experiments [1].

Standard Deviation of percent differences between v.0.42.3			
Kallisto v.0.42.3	Estimated Counts	Effective Length	Estimated MAD
Run1	0	0	0
Run2	0	0	0
Run3	0	0	0
Run4	0	0	0
Run5	0	0	0
Run6	0	0	0
Run7	0	0	0
Run8	0	0	0
Run9	0	0	0
Run10	0	0	0

Table 1: Variation of v.0.42.3

Standard Deviation of percent differences between v.0.42.1 and v.0.42.3				
Kallisto v.0.42.3 and v.0.42.1	Estimated Counts	Effective Length	Estimated MAD	TPM
Run1	419.7398	23.49247	74.54197	423.5493

Table 2: Standard Deviation of Percent Error 0.42.1 and 0.42.3

We then analyzed the percent difference for each gene across all samples and calculated the standard deviation of version 0.42.3 of the 10 Kallisto runs generated by Artemis [1]. We randomly selected a KallistoExperiment v.0.42.3 from our KallistoExperiment list, and calculated the percent difference between each of the other 9 KallistoExperiments of the same version across all samples. The table 1 (Table 1) shows the standard deviation of the percent differences of the raw values such as estimated counts, effective length, and estimated median absolute deviation. Kallisto data quantified against the reference generated by the same kallisto version is 0 within the same version 0.42.3 for every transcript across all samples.

However, we compared the merged kallisto data of estimate raw abundance counts, effective length, estimated median absolute deviation, and transcript per million values between version 0.42.1 and a randomly selected KallistoExperiment data container generated by kallisto version 0.42.3. Table 2 (Table 2) shows that there exists large standard deviations of the percent differences calculated between each gene expression across all samples. This shows the importance of enforcing uniform versions.

We plotted the errors between Kallisto versions, and fit each of the calculated percent differences to a normally distributed data set generated by the mean and standard deviation of the percent differences of each assay data [1]. The QQ plots show that the errors are some-what normal (Figure 1); thus we confidently create default settings which prevent the public from analyzing or sharing data from different versions.

4.2 Annotation of Coding, non-Coding, and Repeat Elements

The annotations were performed with a run time of 2.336 seconds (Table 4) on a merged Kallisto [1] sample directory creating a KallistoExperiment class with feature data containing a GenomicRanges [31] object with 213782 ranges and 9 metadata columns. After the sample directories container raw fastq files were quantified using Artemis, the Artemis routine 'mergeKallisto' was performed against 6 quantified samples. The system runtime for creating a merged KallistoExperiment class for 6 samples was 23.551 seconds (Table 3).

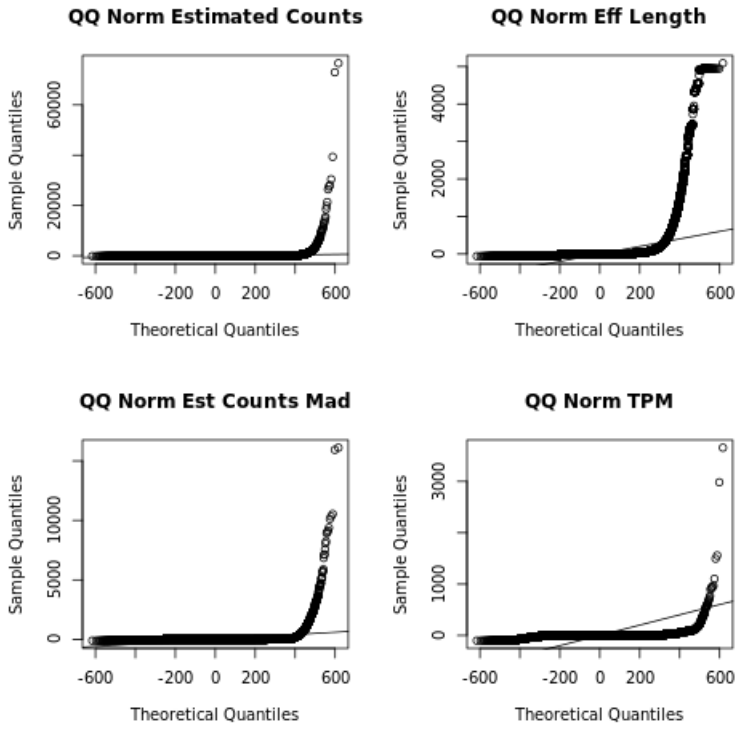


Figure 1: Quantile Plots of Percent Difference between v.0.42.3 and v.0.42.1

System Runtime for creating KallistoExperiment (secs)		
user	system	elapsed
23.551	1.032	19.107

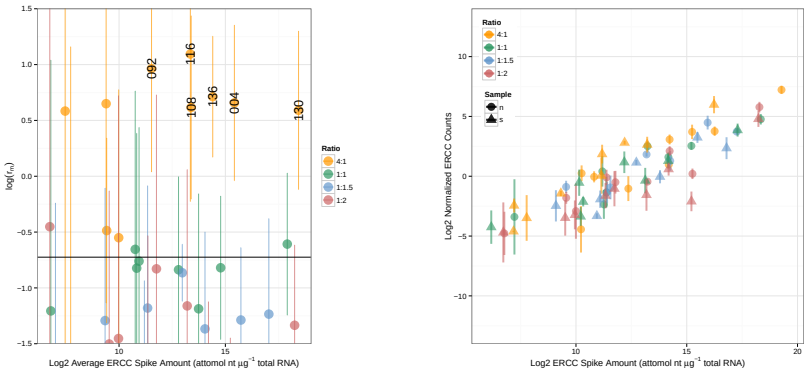
Table 3: Artemis Run Time KallistoExperiment

System Runtime for full Annotation of a merged KallistoExperiment (secs)		
user	system	elapsed
2.336	0.064	2.397

Table 4: Automated Annotation Run Time

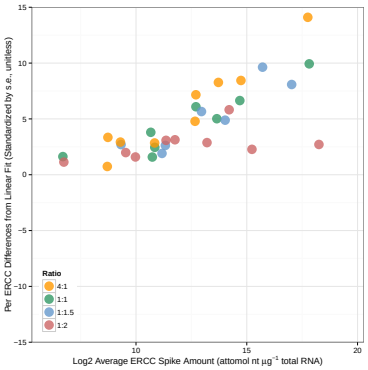
Number of annotated genes expressed in Annotated KallistoExperiment	
Anntotated Gene	Total Num
IG_C_gene	52
IG_D_gene	64
IG_J_gene	24
IG_LV_gene	0
IG_V_gene	250
TR_C_gene	32
TR_J_gene	93
TR_V_gene	162
TR_D_gene	5
IG_C_pseudogene	11
IG_J_pseudogene	6
IG_V_pseudogene	283
TR_V_pseudogene	43
TR_J_pseudogene	4
Mt_rRNAv	2
Mt_tRNA	22
miRNA	4554
misc_RNA	0
rRNA	565
snRNA	2036
snoRNA	1019
ribozyme	8
sRNA	20
scaRNA	51
Mt_tRNA_pseudogene	0
tRNA_pseudogene	0
snoRNA_pseudogene	0
snRNA_pseudogene	0
scRNA_pseudogene	0
rRNA_pseudogene	0
misc_RNA_pseudogene	0
miRNA_pseudogene	0
TEC	0
nonsense_mediate_decay	0
protein_coding	156188
processed_transcript	3361
non_coding	0
ambiguous_orf	0
sense_intronic	1001
sense_overlapping	337
antisense	10447
pseudogene	18502
retrotransposed	0
SINE	72
other_repeat	108
LINE	136
LTR_element	531
DNA_element	269

Table 5: Summary of Genes Annotated from an Experiment



(a) Ercc Counts (b) Amounts vs Counts

Figure 2: Ercc Counts



(a) Ercc Differences

Figure 3: Ercc Differences

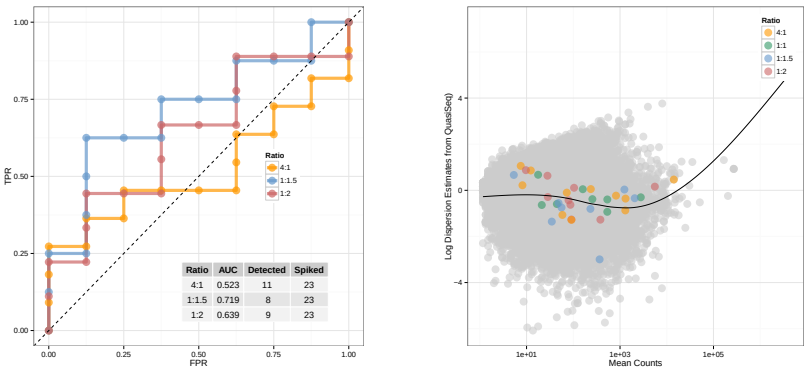
4.3 ERCC-Analysis

The results for ERCC analysis includes 5 plots generated by Artemis' method 'erccAnalysis', and the plots are saved calling 'saveArtemisPlots'. The Ercc plots figure 2, 3 4 titled ERCC Counts, Ercc Differences, and TPR vs FPR and Dispersion plots (Figures 2,3,4).

4.4 QuSage Acceleration Analysis

For testing a competitive revision of QuSage [18], we used the dataset from a study by Huang Y et al. (GEOID: GSE30550) which examines the expression of 17 patients before and 77 hours after exposure to the in uenza virus. This expression set was selected because it was described in the QuSage vignette [18]; so we selected the matching enrichment gene set and expression set used in QuSage BioConductor vignette [18], and developed accelerated computational performance while ensuring accuracy.

The calculations for Welch's Approximation, such as standard deviation and degrees of freedom, where performed by Armadillo C++ libraries seamlessly integrated into the R environment using RcppArmadillo [19]. The gained computational speed was achieved from altering the following QuSage functions 'calcVIF.R', 'makeComparisons.R', and 'calculateIndividualExpressions.R' [18]. We ensured that each of the C++ scripts had at most machine error precision between QuSage [18] defaults and the altered RcppArmadillo [19] libraries.



(a) TPR (b) Dispersion

Figure 4: FPR vs TPR, and Dispersion

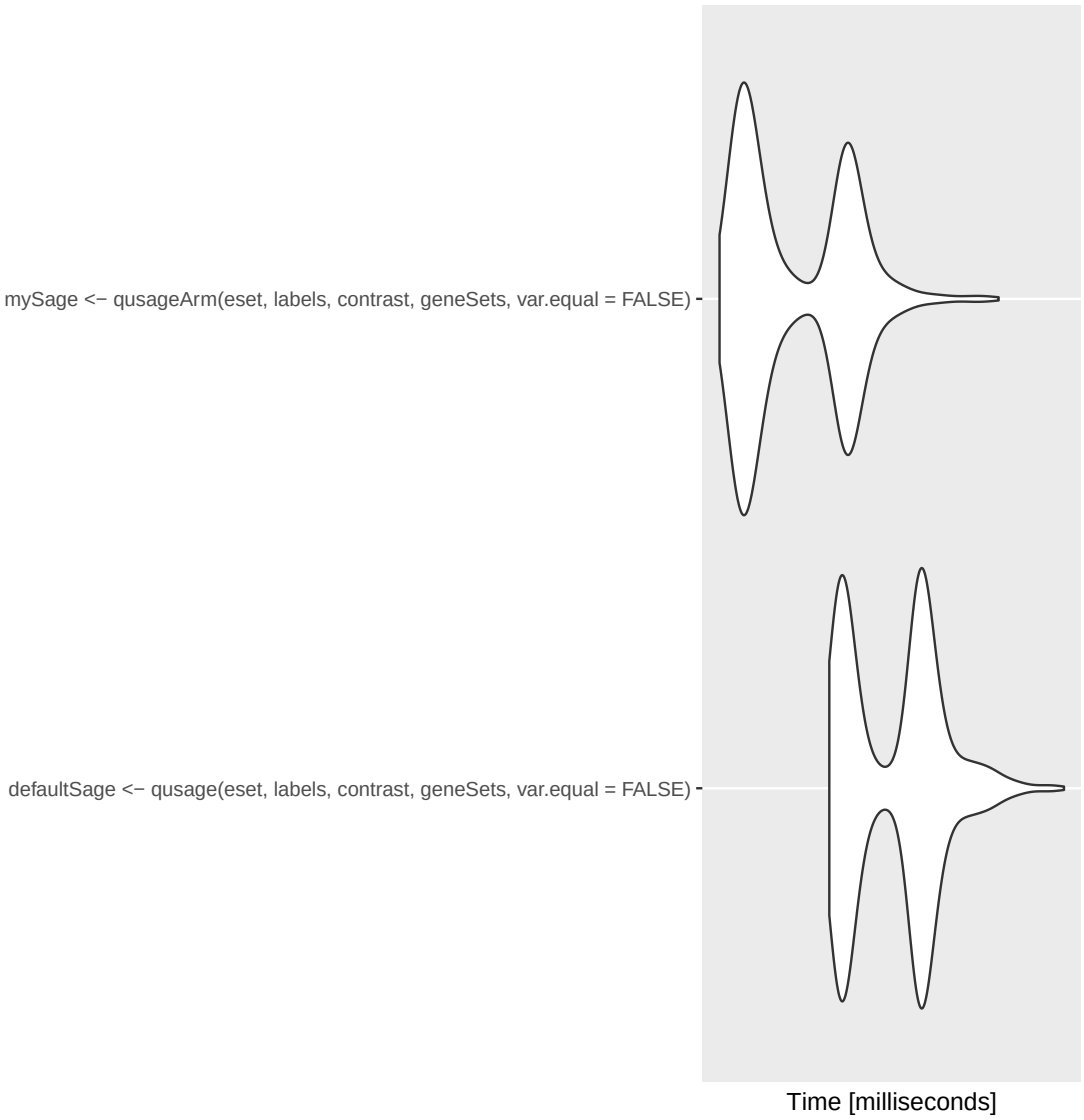
QuSage microbenchmark in milliseconds							
name	min	lq	mean	median	uq	max	neval
defaultQuSage	165.867	173.216	203.527	214.319	220.555	335.380	1000
armaQuSage	119.325	128.087	151.597	136.478	174.955	275.659	1000

Table 6: Qusage and Armadillo Benchmarks

5 Discussion

Many research publications include a written methodology section with varying degrees of detail and an included supplemental section; this is done as a symbol for attempting validation. Reproducible research is defined as a link between the global research community, defined as the set of all members and associated writings therein, to unique members of the global data environment, the set of all archived, published, clinical, sequencing and other experimental data. The aim for practicing transparent research methodologies is to clearly define associations with every research experiment minimizing opaqueness between the analytical findings, clinical studies, and utilized methods. As for clinical studies, re-generating an experimental environment has a very low success rate [32] where non-validated preclinical experiments spawned developments of best practices for critical experiments. Re-creating a clinical study has many challenges including an experimental design that has a broad focus applicability, the difficult nature of a disease, complexity of cell-line models between mouse and human that creates an inability to capture human tumor environment, and limited power through small enrollments during the patient selection process [32]. Confirming preclinical data is difficult, however the class of re-validated experiments each contain carefully selected reagents, diligently tested controls for various levels of an experiment, and, most significantly, a complete description of the entire data set. Original data sets are frequently not reported in the final paper and usually removed during the revision process. Experimental validation is dependent on the skillful performance of an experiment, and an earnest distribution of the analytic methodology which should contain most, if not all, raw and resultant data sets.

With recent developments for virtualized operating systems, developing best practices for bioinformatic confirmations of experimental methodologies is much more straightforward in contrast to duplicating clinical trials' experimental data for drug-development. Data validations can be improved if the local developmental and data sets are distributed. Recent advancements of technology, such as Docker allow for local software environments to be preserved using a virtual operating system. Docker allows users to build layers of read/write access files creating a portable operating system which controls



(a) qusage in Armadillo

Figure 5: computational microbenchmark

exhaustively software versions, data, and systematically preserves one's complete software environment. Conserving a researcher's developmental environment advances analytical reproducibility if the workflow is publicly distributed. We suggest a global distributive practice for scholarly publications that regularly includes the virtualized operating system containing all raw analytical data, derived results, and computational software. A global distributive practice links written methodologies and supplemental data to the utilized computational environment.

Cloud computational ecosystems preserve an entire developmental environment using the Docker infrastructure improving bioinformatic validation. Containerized cloud applications are instances of the global distributive effort and are favorable compared to local in-house computational pipelines because they offer rapid access to numerous public workflows, easy networking to archived read databases, and accelerate the upholding process of raw data. Distributed cloud pipelines are externally available which can easily manifest symbolic explanatory method sections. For researchers uninterested in designing an exhaustive cloud application, methodology writings can instead publicize locally containerized workflows with much less effort. Scholarly publications that choose only a written method section passively make validation gestures. We envision a future where published works will share conserved analytic environments and/or instantiated cloud software accessed by web-distributed methodologies, thus strengthening links between raw sequencing data, analytical results, and utilized software.

6 Conclusion

Artemis integrates the Kallisto [1] pseudoalingment algorithm into the BioConductor ecosystem that can implement large scale parallel ultra-fast transcript abundance quantification over the BaseSpace Platform. We reduce a computational bottleneck freeing inefficiencies from utilizing ultra-fast transcript abundance calculations while simultaneously connecting an accelerated quantification software to the Sequencing Read Archive. Thus we remove a second bottleneck occurred by reducing the necessity of database downloading; instead we encourage users to download aggregated analysis results. We also expand the range of common sequencing protocols to include an improved gene-set enrichment algorithm, Qusage [18] and allow for exporting into an exhaustive pathway analysis platform, Advaita, over the AWS field in parallel. We encapsulate building annotations libraries for arbitrary fasta files using custom software TxDbLite [4] which may annotate coding, non-coding RNA, with a self generated repeatome that includes genomic repetitive elements such as ALUs , SINEs and/or retro-transposons.

Acknowledgment

This project was funded by grants from Illumina, Leukemia Lymphoma Society-Quest for Cures 0863-15, and Tower Cancer Research Foundation.

References

[1] Bray, Nicolas, Harold Pimmentel, Pàll Melsted, Lior Pachter. Near-optimal probabilistic RNA-seq quantification. Nat Biotech. 2016, April 04. online. 1546-1696. <http://dx.doi.org/10.1038/nbt.3519>. 10.1038-nbt.3519. <http://www.nature.com/nbt/journal/vaop/ncurrent/abs/nbt.3519.html#supplementary-information>.

[2] Milacic et al. 2012 PMID:24213504 Croft et al. 2014 PMID: 24243840

[3] Anthony Colombo, Tim Triche Jr., Harold Pimmentel. Artemis: A package that complements Kallisto for quick, informative *seq analysis. September, 27,2015. <https://github.com/RamsinghLab/artemis>.

[4] Triche Jr., Timothy, Anthony Colombo. TxDbLite: Lightweight SQLite-based annotation classes/packages for use with artemis. September 27,2015. <https://github.com/RamsinghLab/TxDbLite>.

[5] GENCODE. "Comparing Different Publicly Available Genesets against GENCODE 7." Web log post. GencodeGenes. N.p., 08 Jan. 2013. Web. 27 Sept. 2015. <https://gencodegenes.wordpress.com/2013/01/08/comparing-different-publicly-available-genesets-against-gencode-7>.

[6] Chen, G., C. Wang, L. Shi, X. Qu, J. Chen, J. Yang, C. Shi, L. Chen, P. Zhou, B. Ning, W. Tong, and T. Shi. "Incorporating the Human Gene Annotations in Different Databases Significantly Improved Transcriptomic and Genetic Analyses." Rna (2013): 479-89. Print.

[7] Smit, AFA, Hubley, R & Green, P. RepeatMasker Open-4.0. 2013-2015. <http://www.repeatmasker.org>.

[8] Croft, D., G. O'kelly, G. Wu, R. Haw, M. Gillespie, L. Matthews, M. Caudy, P. Garapati, G. Gopinath, B. Jassal, S. Jupe, I. Kalatskaya, S. Mahajan, B. May, N. Ndegwa, E. Schmidt, V. Shamovsky, C. Yung, E. Birney, H. Hermjakob, P. D'eustachio, and L. Stein."Reactome: a database of reactions, pathways and biological processes". Nucleic Acids Research. 39. 2010-11-09. Print.

[9] Van den Boogaart, KG. "Analyzing compositional data with R". R Tolosana-Delgado. Springer. New York.2013. Print.

[10] Bourgon, R., R. Gentleman, and W. Huber. "Independent Filtering Increases Detection Power for High-throughput Experiments." Proceedings of the National Academy of Sciences (2010): 9546-551. Print.

[11] Geistlinger et al., "Gene Graph Enrichment Analysis, Bioinformatics, 27(13):i366-i373, 2011.

[12] Subramanian, Aravind. "Gene Set Enrichment Analysis: A Knowledge-based Approach for Interpreting Genome-wide Expression Profiles." www.pnas.org. The National Academy of Sciences, 14 May 2005. Web. 19 July 2015,

[13] Lawrence M, Huber W, Pags H, Aboyoun P, Carlson M, Gentleman R, Morgan M and Carey V (2013). Software for Computing and Annotating Genomic Ranges. PLoS Computational Biology, 9. http://doi.org/10.1371/journal.pcbi.1003118, http://www.ploscompbiol.org/article/info

[14] Munro SA et al. Assessing technical performance in differential gene expression experiments with external spike-in RNA control ratio mixtures. Nat. Commun. 5:5125 doi: 10.1038/ncomms6125 (2014).

[15] Martin Morgan, Valerie Obenchain, Jim Hester and Herv Pags (). SummarizedExperiment: SummarizedExperiment container. R package version 1.0.0.

[16] Ritchie, M.E., Phipson, B., Wu, D., Hu, Y., Law, C.W., Shi, W., and Smyth, G.K. (2015). limma powers differential expression analyses for RNA-sequencing and microarray studies. Nucleic Acids Research 43(7), e47.

[17] Robinson MD, McCarthy DJ and Smyth GK (2010). edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. Bioinformatics 26, 139-140

[18] Yaari G, Bolen CR, Thakar J, Kleinstein SH. Quantitative set analysis for gene expression: a method to quantify gene set differential expression including gene-gene correlations. Nucleic Acids Res. 2013 Aug 5.

[19] Dirk Eddelbuettel, Conrad Sanderson (2014). RcppArmadillo: Accelerating R with high-performance C++ linear algebra. Computational Statistics and Data Analysis, Volume 71, March 2014, pages 1054-1063. URL http://dx.doi.org/10.1016/j.csda.2013.02.005

[20] Subramanian, Aravind, Pablo Tamayo, Vamsi Mootha, Sayan Mukherjee, Benjamin Eberta, Michael Gillette, Amanda Paulovich, Scott Pomeroy, Todd Golub, Eric Lander, and Jill Mesirov. "Gene Set Enrichment Analysis: A Knowledge-based Approach for Interpreting Genome-wide Expression Profiles." Proceedings of the National Academy of Sciences of the United States of America 102.43 (2005): 15545-5550. Web. ¡http://www.pnas.org/content/102/43/15545.full¿.

[21] Kim, Daehwan, Pertea, Geo, Trapnell, Cole, Pimentel, Harold, Kelley, Ryan, Salzberg, Steven L. TopHat2: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions. Genome Biology.2013.14.4. pp 1-13.

[22] Trapnell, C. et al. Nat Biotech 28, 511515 (2010).

[23] Mitra, Sheetal A., Anirban P. Mitra, and Timothy J. Triche. A Central Role for Long Non-Coding RNA in Cancer. Frontiers in Genetics 3 (2012): 17. PMC. Web. 23 Mar. 2016.

[24] Zhang Wensheng, Edwards Andrea, Fan Wei, Deininger Prescott, Zhang Kun. Alu distribution and mutation types of cancer genes.BMC Genomics.2011.vol 12.issue 1.pp 1-18.

[25] Welcsh, P. L. "BRCA1 and BRCA2 and the Genetics of Breast and Ovarian Cancer." Human Molecular Genetics 10.7 (2001): 705-13. Web.

[26] Ocaa, Kary, and Daniel de Oliveira. Parallel Computing in Genomic Research: Advances and Applications. Advances and Applications in Bioinformatics and Chemistry: AABC 8 (2015): 2335. PMC. Web. 23 Mar. 2016.

[27] Reid JG, Carroll A, Veeraraghavan N, Dahdouli M, Sundquist A, English A, Bainbridge M, White S, Salerno W, Buhay C, Yu F, Muzny D, Daly R, Duyk G, Gibbs RA, Boerwinkle E.BMC Bioinformatics. 2014 Jan 29; 15():30.

[28] Proposed methods for testing and selecting the ERCC external RNA controls. External RNA Controls Consortium BMC Genomics. 2005 Nov 2; 6():150.

[29] The External RNA Controls Consortium: a progress report. Baker SC, Bauer SR, Beyer RP, Brenton JD, Bromley B, Burrill J, Causton H, Conley MP, Elespuru R, Fero M, Foy C, Fuscoe J, Gao X, Gerhold DL, Gilles P, Goodsaid F, Guo X, Hackett J, Hockett RD, Ikonomi P, Irizarry RA, Kawasaki ES, Kaysser-Kranich T, Kerr K, Kiser G, Koch WH, Lee KY, Liu C, Liu ZL, Lucas A, Manohar CF, Miyada G, Modrusan Z, Parkes H, Puri RK, Reid L, Ryder TB, Salit M, Samaha RR, Scherf U, Sendera TJ, Setterquist RA, Shi L, Shippy R, Soriano JV, Wagar EA, Warrington JA, Williams M, Wilmer F, Wilson M, Wolber PK, Wu X, Zadro R, External RNA Controls Consortium Nat Methods. 2005 Oct; 2(10):731-4.

[30] A. P. Dempster; N. M. Laird; D. B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. Journal of the Royal Statistical Society. Series B (Methodological),Vol. 39, No. 1. (1977), pp.1-38.

[31] Lawrence M, Huber W, Pagès H, Aboyoun P, Carlson M, et al. (2013) Software for Computing and Annotating Genomic Ranges. PLoS Comput Biol 9(8): e1003118. doi:10.1371/journal.pcbi.1003118

[32] Begley, C. Glenn, Ellis, Lee M. "Drug development: Raise standards for preclinical cancer research". Nature. 2012 Mar 29. print. Vol 483. Issue 7391. pp 531-533. Nature Publishing Group, a division of Macmillan Publishers Limited. All Rights Reserved.0028-0836. http://dx.doi.org/10.1038/483531a. 10.1038/483531a.10.1038/483531a

[33] Soneson, Charlotte, Matthes, Katarina L. , Nowicka, Malgorzata, Law, Charity W., Robinson, Mark D. 2016.Isoform prefiltering improves performance of count-based methods for analysis of differential transcript usage. Genome Biology. pp 1-15. Vol 17. Issue 1. 1474-760X. <http://dx.doi.org/10.1186/s13059-015-0862-3>.

[34] <http://guttmanlab.caltech.edu/publications.php>.