SOFTWARE TOOL ARTICLE

Arkas: Rapid reproducible RNAseq analysis [version 1; referees: 1 approved, 1 approved with reservations]

Anthony R. Colombo, Timothy J. Triche Jr, Giridharan Ramsingh

Jane Anne Nohl Division of Division of Hematology and Center for the Study of Blood Diseases, Keck School of Medicine of University of Southern California, Los Angeles, CA, 90033, USA

Abstract
The recently introduced Kallisto pseudoaligner has radically simplified the quantification of transcripts in RNA-sequencing experiments. We offer cloud-scale RNAseq pipelines Arkas-Quantification, which deploys Kallisto for parallel cloud computations, and Arkas-Analysis, which annotates the Kallisto results by extracting structured information directly from source FASTA files with per-contig metadata and calculates the differential expression and gene-set enrichment analysis on both coding genes and transcripts. The biologically informative downstream gene-set analysis maintains special focus on Reactome annotations while supporting ENSEMBL transcriptomes. The Arkas cloud quantification pipeline includes support for custom user-uploaded FASTA files, selection for bias correction and pseudoBAM output. The option to retain pseudoBAM output for structural variant detection and pseudoBAM output 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. Illumina's BaseSpace cloud computing environment, where these two applications are hosted, offers a massively parallel distributive quantification step for users where investigators are better served by cloud-based computing platforms due to inherent efficiencies of scale.

This article is included in the Container Virtualization in Bioinformatics collection.
Introduction

High-performance computing based bioinformatic workflows have three main subfamilies: in-house computational packages, virtual-machines (VMs), and cloud-based computational environments. The in-house approaches are substantially less expensive when raw hardware is in constant use and dedicated support is available, but internal dependencies can limit reproducibility of computational experiments. Specifically, “superuser” access needed to deploy container-based, succinct code encapsulations (often referred to as “microservices” elsewhere) can run afoul of normal permissions, and the maintenance of broadly usable sets of libraries across nodes for users can lead to shared code dynamically linking to different libraries under various user environments. By contrast, modern cloud-based approaches and parallel computing are forced by necessity to offer a user-friendly platform with high availability to the broadest audience. Platform-as-a-service approaches take this one step further, offering controlled deployment and fault tolerance across potentially unreliable instances provided by third parties such as Amazon Web Service Elastic Compute Cloud (AWS EC2) and enforcing a standard for encapsulation of developers’ services such as Docker.

Within this framework, the user or developer cedes some control of the platform and interface, in exchange for the platform provider handling the details of workflow distribution and execution. This has provided the best compromise of usability and reproducibility when dealing with general audiences. In this regard, the lightweight-container approach exemplified by Docker lead to rapid development and deployment compared to VMs. Combined with versioning of deployments, it is feasible for users to reconstruct results from an earlier point in time, while simultaneously re-evaluating the generated data under state-of-the-art implementations.

Several recent high impact publications used cloud-computing workflows such as CloudBio-linux, CloudMap and Mercury AWS EC2. The CloudBio-linux software is centered around comparative genomics, phylogenomics, transcriptomics, proteomics, and evolutionary genomics studies using Perl scripts. Although offered with limited scalability, the CloudMap software allows scientists to detect genetic variations over a field of virtual machines operating in parallel. For comparative genomic analysis, the Mercury 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. The majority of RNAseq analysis pipelines consist of read preparation steps, followed by computationally expensive alignment against a reference. Software for calculating transcript abundance and assembly can surpass 30 hours of computational time. If known or putative transcripts of defined sequences are the primary interest, then pseudoalignment, which is defined as near-optimal RNAseq transcript quantification, is achievable in minutes on a standard laptop using Kallisto software.

After verifying these numbers on our own laptops, we became interested in a massively parallel yet easy-to-use approach that would allow us to perform the same task on arbitrary datasets, and reliably interpret the output. In collaboration with Illumina (San Diego, USA) we found that the available BaseSpace platform was already well-suited for this purpose, with automated ingestion of the Sequence Read Archive (SRA) datasets as well as newly produced data from core facilities using recent Illumina sequencers. The design of our framework emphasizes loose coupling of components and tight coupling of reference transcriptome annotations; nonetheless, the ease of use and massive parallelization provided by BaseSpace offers excellent default execution environment.

The BaseSpace Platform utilizes AWS cc2 8x-large instances by default, each with access to eight 64-bit CPU cores and virtual storage of over 3 terabytes. Published BaseSpace applications, which undergo rigorous review by Illumina staff scientists before deployment, can allocate up to 100 such nodes, distributing analyses simultaneously, in parallel. Direct imports of existing experiments from SRA, along with default availability of experimenters’ own reads, fosters a critical environment for independent replication and reanalysis of published data.

A second bottleneck in bioinformatic workflows, hinted at above, arises from the frequent transfer and copying of source data across local networks and/or the Internet. With a standardized deployment platform, it becomes easier to move executable code to the environment of the target data, rather than transferring massive datasets into the environment where the executable workflows were developed. For instance, an experiment from SRA with reads totaling 141.3GB is reduced to summary quantifications totaling 1.63GB (nearly two orders of magnitude) and a report of less than 10MB (a further two orders of magnitude), for a total reduction in size exceeding 4 orders of magnitude with little or no loss of user-visible information. Moreover, the untouched original data is never discarded unless the user explicitly demands it, something that can rarely be said of local computer environments. Moreover, the location of original sources is always traceable. The appropriate placement of Arkas cloud computational applications in close proximity to the origin of sequencing data removes cumbersome data relocation costs.

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 sequencing 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 and isoform-dependent switching of protein interactions. What sometimes gets lost within this excitement, however, is the reality that biological interpretation of 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. Recent work on the Kallisto pseudoaligner, amongst other k-mer based approaches, has resulted in such an approach.
In order to leverage these recent advances for large scale needs, we created a cloud computational pipeline, Arkas, which encapsulates Kallisto, automates the construction of composite transcriptomes from multiple sources, quantifies transcript abundances, and implements reproducible rapid differential expression analysis followed by a gene set enrichment analysis over Illumina’s BaseSpace Platform. The Arkas workflow is versionized into Docker containers and publicly deployed within Illumina’s BaseSpace cloud based computational environment.

Methods

Aeritas-Quantification Implementation

Aeritas is a two-step cloud pipeline. Aeritas-Count is the first step, which reduces the computational steps required to quantify and annotate large numbers of samples against large catalogs of transcriptomes. Aeritas-Count calls Kallisto for on-the-fly transcriptome indexing and quantification recursively for numerous sample directories. Kallisto quantifies transcript abundance from input RNAseq reads by using pseudoligment, 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 (EM) algorithm over a smaller matrix.

For RNAseq projects with many sequenced samples, Aeritas-Count encapsulates expensive transcript quantification preparatory routines, while uniformly preparing Kallisto execution commands within a versioned environment encouraging reproducible protocols. The quantification step automates the index caching, annotation, and quantification associated while running the Kallisto pseudoalnigner integrated within the BaseSpace environment. The first step in the pipeline can process raw reads into transcript and pathway collection results within Illumina’s BaseSpace cloud platform, quantifying against default transcriptomes such as ERCC spike-ins, ENSEMBL non-coding RNA, or cDNA build 88 for both Homo sapiens and Mus musculus; further, the first step supports user uploaded FASTA files for customized analyses. Aeritas-Count is packaged into a Docker container and is publicly available as a cloud application within BaseSpace.

Aeritas-Analysis Implementation

Previous work 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., we included this type of filtering for both gene- and transcript-level analyses within Aeritas-Analysis. The analysis pipeline automates annotations of quantification results, resulting in more accurate interpretation of coding and transcript sequences in both basic and clinical studies by just-in-time annotation and visualization.

Aeritas-Analysis integrates quality control analysis for experiments that include Ambion spike-in controls, multiple normalization selections for both coding gene and transcript differential expression analysis, and differential gene-set analysis. If ERCC spike-ins, defined by the External RNA Control Consortium, are detected then Aeritas-Analysis will calculate Receiver Operator Characteristic (ROC) plots using ‘erccdashboard’. The ERCC analysis reports average ERCC Spike amount volume, comparison plots of ERCC volume amount, and normalized ERCC counts (Figure 1).

Subsequent analyses import the data structure from SummarizedExperiment (Morgan, 2016) and create a sub-class titled KallistioExperiment that preserves the S4 structure and is convenient for handling assays, phenotypic and genomic data. KallistioExperiment includes GenomicRanges, preserving the ability to handle genomic annotations and alignments, supporting efficient methods for analyzing high-throughput sequencing data. The KallistioExperiment sub-class serves as a general-purpose container for storing feature genomic intervals and pseudalignment quantification results against a reference genome called by Kallisto. By default KallistioExperiment 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 run; the stored feature data is a GenomicRanges object from, storing transcript length, GC content, and genomic intervals.

Given a KallistioExperiment containing the Kallisto sample abundances, principal component analysis (PCA) is performed on trimmed mean of M-value (TMM) normalized counts (Figure 2A). Differential expression (DE) is calculated on the library normalized transcript expression values, and the aggregated transcript bundles of corresponding coding genes using limma/voom linear model (Figure 3A). Further, an additional PCA and DE analysis of both transcripts and coding genes is performed using in-silico normalization using factor analysis (Figure 2B, Figure 3B, Figure 3C). In each DE analysis FDR filtering method is defaulted to ‘Benjammini-Hochberg’, if there are no resultant DE genes/transcripts the FDR methods is switched to ‘none’. Aeritas-Analysis consumes the Kallisto data output from Aeritas-Count, and automates DE analysis using TMM normalization and in-silico normalization on both transcript and coding gene expression in a defaulted two group experimental design, which allows end-users to select the normalization type best suited for their needs.

Gene set differential expression, which includes gene-gene correlation inflation corrections, is calculated using Qusage. Qusage 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 experimental groups. The gene set enrichment is conducted using Reactome pathways constructed using ENSEMBL transcript/gene identifiers (Figure 4 and Table 1); REACTOME gene sets are not as large as other databases, so Aeritas-Analysis outputs DE analysis in formats compatible with more exhaustive databases such as Advaita. The DE files are compatible as a custom upload into Advaita iPathway guide, which offers an extensive Gene Ontology (GO) pathway analysis. Pathway enrichment analysis can be performed from the BaseSpace cloud system downstream from parallel differential expression analysis and can integrate with other pathway analysis software tools.
Figure 1. Arkas-Analysis ERCC spike-in Controls Report. A) The Receiver Operator Characteristic plot. The X-axis shows the False Positive Rate, the Y-axis shows True Positive Rate. B) and D) show the spike-in total RNA amounts with a linear model fit, and quantified ERCC transcript counts. C) shows a dispersion of mean transcript abundance counts and the estimated dispersion.
Figure 2. *Arkas-Analysis* Normalization Report: Normalization Analysis Using TMM and RUV. A) TMM normalization is performed on sample data and depicts the sample quartiles on normalized sample expression, PCA plot, and histogram of the adjusted p-values calculated from the DE analysis. Orange is the comparison group and green is the control group. B) A similar analysis is performed with RUV *in-silico* normalization.
Figure 3. *Arkas-Analysis* Differential Expression Report: DE using TMM and RUV. **A**) DE analysis using TMM normalization. The X-axis is the sample names (test data), the Y-axis are Gene symbols (HUGO). Expression values are plotted in $\log_{10}(1+\text{TPM})$. **B**) Similar analysis using RUV normalization. **C**) The design matrix with the RUV adjusted weights. The sample names are test data used in demonstrating the general analysis report output.
Figure 4. *Arkas-Analysis* Gene-Set Enrichment Plot. Gene-Set enrichment output report, each point represents the differential mean activity of each gene-set with 95% confidence intervals. The X-axis are individual gene-sets. The Y-axis is the log₂ fold change.

Table 1. *Arkas-Analysis* Gene-Set Enrichment Statistics. The columns represent the Reactome pathway name corresponding to the depicted pathways in Figure 4, the log₂ fold change, p-value, adjusted FDR, and an active link to the Reactome website with visual depictions of the gene/transcript pathway. *Arkas-Analysis* will output a similar report testing transcript-level sets.

<table>
<thead>
<tr>
<th>Pathway name</th>
<th>Log fold change</th>
<th>P.value</th>
<th>FDR</th>
<th>Gene URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>R-HAS-1989781</td>
<td>-0.87</td>
<td>0.0008</td>
<td>0.06</td>
<td><a href="http://www.reactome.org/PathwayBrowser/#/R-HSA-1989781">http://www.reactome.org/PathwayBrowser/#/R-HSA-1989781</a></td>
</tr>
<tr>
<td>R-HAS-2173796</td>
<td>-0.51</td>
<td>0.007</td>
<td>0.217</td>
<td><a href="http://www.reactome.org/PathwayBrowser/#/R-HSA-2173796">http://www.reactome.org/PathwayBrowser/#/R-HSA-2173796</a></td>
</tr>
<tr>
<td>R-HAS-6804759</td>
<td>-1.62</td>
<td>0.009</td>
<td>0.217</td>
<td><a href="http://www.reactome.org/PathwayBrowser/#/R-HSA-6804759">http://www.reactome.org/PathwayBrowser/#/R-HSA-6804759</a></td>
</tr>
<tr>
<td>R-HAS-381038</td>
<td>-0.43</td>
<td>0.013</td>
<td>0.226</td>
<td><a href="http://www.reactome.org/PathwayBrowser/#/R-HSA-381038">http://www.reactome.org/PathwayBrowser/#/R-HSA-381038</a></td>
</tr>
<tr>
<td>R-HAS-2559585</td>
<td>-0.4</td>
<td>0.032</td>
<td>0.341</td>
<td><a href="http://www.reactome.org/PathwayBrowser/#/R-HSA-2559585">http://www.reactome.org/PathwayBrowser/#/R-HSA-2559585</a></td>
</tr>
<tr>
<td>R-HAS-4086398</td>
<td>-0.95</td>
<td>0.033</td>
<td>0.341</td>
<td><a href="http://www.reactome.org/PathwayBrowser/#/R-HSA-4086398">http://www.reactome.org/PathwayBrowser/#/R-HSA-4086398</a></td>
</tr>
<tr>
<td>R-HAS-6804754</td>
<td>-0.57</td>
<td>0.102</td>
<td>0.389</td>
<td><a href="http://www.reactome.org/PathwayBrowser/#/R-HSA-6804754">http://www.reactome.org/PathwayBrowser/#/R-HSA-6804754</a></td>
</tr>
</tbody>
</table>

**Data variance between software versions**

We wished to show the importance of enforcing matching versions of Kallisto when quantifying transcripts because there is deviation of data between versions. Due to updated versions and improvements of Kallisto software, there obviously exists variation of data between algorithm versions (Figure 5, Supplementary Table 1, Supplementary Table 2). We calculated the standardized mean differences, and the variation of the differences between data output from Kallisto versions 0.43 and 0.43.1 (Supplementary Table 2), and found large variation of differences between raw values generated by differing Kallisto versions, signifying the importance of version analysis of Kallisto results.

The Dockerization of Arkas BaseSpace applications versionizes the Kallisto reference index to enforce that the Kallisto software versions are identical, and further documents the Kallisto version used in every cloud analysis. The enforcement of reference versions and Kallisto software versions prevents errors when comparing experiments.

**Operation**

*Arkas-Quantification* instructions are provided within BaseSpace (details for new users can be found here). The input are RNA sequencing samples, which may include SRA imported reads, and the outputs include the Kallisto data, .tar.gz files of the Kallisto sample data, and a report summary. Users may select for species type
(Homo sapiens or Mus musculus), optionally correct for read length bias, and optionally select for the generation of pseudoBAMs. More significantly, users have the option to use the default transcriptome (ENSEMBL build 88) or to upload a custom FASTA of their choosing. For users that wish for local analysis, they can download the sample .tar.gz Kallisto files and analyze the data locally.

The Arkas-Analysis instructions are provided within the BaseSpace environment. The input for the analysis app is the Arkas-Quantity sample data, and the output files are separated into corresponding folders. The analysis also depicts figures for each respective analysis (Figure 1–Figure 4) and the images can be downloaded as a HTML format.

Results
One main advantage of Dockerized analysis software is that it preserves software environments. As an exercise to show the importance of enforcing matching Kallisto versions, we’ve repeatedly ran Kallisto on the same 5 samples, quantifying transcripts (setting bootstraps=42) against two different Kallisto versions and calculating the standardized mean differences and variation of differences between each run. We ran Kallisto quantification once with Kallisto version 0.43.1, and 4 times with version 0.43.0, merging each run into a KallistoExperiment and storing the runs into a list of Kallisto experiments.

We then analyzed the standardized mean differences for each gene across all samples and calculated the variation of errors for each run quantified using version 0.43.0. Supplementary Table 1 shows the variation of the errors of the raw values such as estimated counts, effective length, and estimated median absolute deviation using the same Kallisto version 0.43.0. As expected, Kallisto data generated by the same Kallisto version had very low variation of errors within the same version 0.43.0 for every transcript across all samples. However, upon comparing Kallisto version 0.43.1 to version 43.0 using the raw data such as estimate abundance counts, effective length, estimated median absolute deviation, and transcript per million values, we found, as expected, large variation of data. Supplementary Table 2 shows that there is large variation of the differences of Kallisto data calculated between versions. Figure 5 depicts the standardized mean differences, i.e., errors, between Kallisto versions fitted to a theoretical normal distribution. The quantile-quantile plots show that the errors are marginally normal, with a consistent line centered near 0 but also large outliers (Figure 5). As expected, containerizing analysis pipelines will enforce versionized software, which benefits reproducible analyses.

Annotation of coding genes and transcripts
The extraction of genomic and functional annotations directly from FASTA contig comments, eliding sometimes-unreliable dependencies on services such as BioMart, are calculated rapidly. The annotations were performed with a run time of 2.336 seconds (Supplementary Table 3) which merged the previous Kallisto data from 5 samples, creating a KallistoExperiment class with feature data containing a GenomicRanges object with 213782 ranges and 9 metadata columns. The system runtime for creating a merged KallistoExperiment class for 5 samples was 23.551 seconds (Supplementary Table 4).

Discussion
Complete transcriptomes enrich annotation information, improving downstream analyses
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. GENCODE is annotated with special attention given to long non-coding RNAs (lncRNAs) and pseudogenes, improving annotations and coupling automated labeling with manual curating. The database selected for protein coding transcripts can influence the amount of annotation information returned when querying gene/transcript level databases.

Although previously overlooked, lncRNAs have been shown to share features and alternate splice variants with mRNA, revealing that lncRNAs play a central role in metastasis, cell growth and cell invasion. LncRNA transcripts have been shown to be functional and are associated with cancer prognosis; proving the importance of studying these transcripts, which are included as defaults within the Arkas pipeline.

Each transcript database is curated at different frequencies with varying amounts RNA entries that influences that mapping rate. GENCODE loci annotations contain 9640 loci, UCSC contain 6056 and RefSeq contain 4888. GENCODE annotations have the greatest number of IncRNA, protein and non-coding transcripts, and highest average transcripts per gene, with 91043 transcripts unique to GENCODE, absent for UCSC and RefSeq databases. ENSEMBL and AceView annotate more genes in comparison to RefSeq and UCSC, and return higher gene and isoform expression labeling improving differential expression analyses. 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. Although ENSEMBL has been shown to detect the same differentially expressed genes as AceView, ENSEMBL GENCODE annotations are manually curated and updated more frequently than AceView. The choice of transcriptome will definitely influence the power of an analysis, thus Arkas cloud analysis applications use ENSEMBL build 88 (ncRNA, and cDNA) by default for Homo sapiens and Mus musculus and also allow users to upload customized FASTA files.

Docker as a cornerstone of reproducible research
Reproducible research should consistently link the works developed by the research community to unique data environments such as clinical, sequencing and other experimental data, used in the construction of the published work. The aim for transparent research methodologies is to clearly define their association with every research experiment, minimizing opaqueness between findings.
and methods. For clinical studies, re-generating an experimental environment has a very low success rate\(^1\), which is why non-validated preclinical experiments have spawned the development of best practices for critical experiments. Re-creating a clinical study has many challenges, for example the difficult nature of a disease, the complexity of cell-line models in mouse and human that attempt to capture human tumor environment, and limited power through small enrollments in clinical trials\(^1\). Experimental validation is quite difficult and 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 than duplicating clinical trials’ experimental data. Recent technology advancements 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 exhaustively controls software versions and data, and systematically preserves the complete software environment. Conserving a researcher’s developmental environment advances analytical reproducibility if the workflow is publicly distributed. We suggest a global distributive practice.

---

**Figure 5.** Quantile-Quantile Plots of Data Variation Comparing Differences in Kallisto Data from Versions 0.43.1 and 0.43.0. The X-axis depicts the theoretical quantiles of the standardized mean differences. The Y-axis represents the observed quantiles of standardized mean differences.
for scholarly publications that regularly includes the virtualized operating system containing all raw analytical data, derived results, and computational software. Currently, Docker, compiled software through CMake, and virtual machines are being utilized, showing progress toward a global distributive practice linking written methodologies, and supplementary data, to the utilized computational environment[2].

Comparing Docker as a distributive practice to virtual machines seems roughly equivalent. Distributed virtual machines are easy to download, and the environment allows for re-generating resultant calculations. However, this is limited if the research community advances the basic requirements for written methodologies and begins to adopt a large scale virtualized distribution, converging to an archive of method environments which would make hosting complete virtual machines impractical or impossible. If an archive were constructed where each research article would link to a distributed methods environment, then an archive of virtual machines for the entire research community is impossible. However, an archive of Dockerfiles is more realistic because a Dockerfile consists of only a few bytes in size.

Novel bioinformatic software is often distributed as a cross-platform flexible build process independent of compiler, which reaches Apple, Windows and Linux users. The scope of novel analytical code is not to manage nor preserve computational environments, but to have environment independent source code as transportable executables. Docker, however, does manage operating systems, and the scope for research best practices does include gathering sets of source executables into a single collection of minimum space and maximum flexibility. Docker can provide the ability for the research community to simultaneously advance publication requirements and develop the future computational frameworks in cloud.

Another advantage for using Docker as the machine manifesting the practice of reproducible research methods, is that there is a trend of well-branded organizations such as Illumina’s BaseSpace platform, Google Genomics, or SevenBridges (all of which offer bioinformatic computational software structures), to use Docker as the principal framework. Cloud computational environments offer many advantages over local high-performance in-house computer clusters, which systematically structure reproducible methodologies and democratize medical science. Cloud computational ecosystems preserve an entire developmental environment using the Docker infrastructure, improving bioinformatic validation. Containerized cloud applications form part of the global distributive effort and are favorable over local in-house computational pipelines because they offer rapid access to numerous public workflows, easy interfacing to archived read databases, and they accelerate the upholding process of raw data. The Google Genomics Cloud has begun to make first steps with integrating cloud infrastructure with the Broad Institute, whereas Illumina’s BaseSpace platform has been hosting novel computational applications since its launch.

Scholarly publications that choose only a written method section passively make validation gestures, which is arguably inadequate in comparison to the rising trend or well-branded organizations. We envision a future where published work will share conserved analytic environments, with cloud software accessed by web-distributed methodologies, and/or large databases organizing multitudes of Dockerfiles with accession numbers, strengthening links between raw sequencing data and reproducible analytical results.

Cloud computational software does not only wish to crystallize research methods into a pristine pool of transparent methodologies, but also matches the rate of production of high quality analytical results to the rate of production of public data, which reaches hundreds of petabytes annually. In a talk given by Dr. Atul Butte in December 2015, he discussed that with endless public data, the traditional method for practicing science has inverted; no longer does a scientist formulate a question and then experimentally measure observations that test the hypothesis. In the modern area, empirical observations are being made at an unbounded rate, the challenge now is formulating the proper question (more details on his talk can be found here). Given a near-infinite amount of observations, what is the phenomena that is being revealed? Cloud computational software can accelerate the production of hypotheses by increasing the flexibility and efficiency of scientific exploration.

Many bioinformaticians have noted a rising trend in biotechnology, predicting that open data and open cloud centers will help democratize research efforts and create a more inclusive practice. With the presence of cloud interfacing applications such as Illumina’s BaseSpace Command Line Interface, DNA-Nexus, SevenBridges, and Google Genomics becoming more popular, cloud environments pioneer the effort for achieving standardized bioinformatic protocols.

Democratization of big-data efforts has some possible negative consequences. Accessing, networking, and integrating software applications for distributing data as a public effort requires massive amounts of specialized technicians to maintain and develop cloud centers that many research institutions are migrating toward. Currently, it is fairly common for research centers to employ high-performance computer clusters which store laboratory software and data locally; cloud computing clusters are beginning to offer clear advantages compared to local closed computer clusters. Collaborations are becoming more common practice for large research efforts, and sequencing databases have been distributing data globally, making cloud storage more efficient. This implies that services from cloud centers will most likely be offered by very few elite organizations because the large scale of cloud services will prevent incentives for smaller companies.

It is very likely that only a few elite organizations will provide services to cloud computing environments, acting as a gateway which directs the global research community toward a narrow set of well established, standardized, computational applications. With regard to recent changes relating to media consumption and e-commerce, democratization allows independent alternative selections far greater exposure, equalizing profits for lower ranked selections “at the tail”, however it may be possible that the abundant amount of data distributed over storage archives, which stimulates an economically abundant environment, could shift into a fiercely controlled economic environment of scarcity. For example, if a
gold-standard is reached for computational applications, the range of alternative selections could remain non-existent, which may diminish the future roles of bioinformaticians. This possible scenario suggests bioinformaticians could be re-directed to small garages instead of the technocratic places such as Silicon Valley, motivated not from a spirit of entrepreneurialism, but from a lack of funding.

Automotive downstream analyses is not without its drawbacks; most computational software is highly specialized for niche groups with a mathematical framework constructed by specialized assumptions, this may require a diverse array of computational developments, and thus a large community of developers. The automation of analytical results seems almost unavoidable, and the benefits seem to outweigh the negative consequences.

Conclusion
Arkas integrates the Kallisto pseudoalignment algorithm into the BaseSpace cloud computation ecosystem that can implement large-scale parallel ultra-fast transcript abundance quantification. We reduce a computational bottleneck by freeing inefficiencies from utilizing rapid transcript abundance calculations and connecting accelerated quantification software to the Sequencing Read Archive. We remove the second bottleneck because we reduce 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, and allow for exporting into an exhaustive pathway analysis platform, Advaita, over the AWS EC2 field in parallel.

Data availability

Data Used in Testing Variation between Versions

Controls: SRR1544480 Immortal-1
SRR1544481 Immortal-2
SRR1544482 Immortal-3

Comparison: SRR1544501 Qui-1
SRR1544502 Qui-2

Software availability

Latest source code: https://github.com/RamsinghLab/Arkas-RNASeq

Archived source code as at the time of publication:
DOI: 10.5281/zenodo.54565
License: MIT license

Reference FASTA Annotation Files
For Homo-sapiens and Mus-musculus ENSEMBL FASTA files were downloaded here for release 88.

ERCC Sequences
The ERCC sequences are provided in a SQL database format located here

Author contributions
AC wrote the manuscript, and developed the web-application and related software. TJ developed software, and helped the project design. GR wrote the manuscript and contributed to the development of software.

Competing interests
No competing interests were disclosed.

Grant information
This project was funded by grants from Leukemia Lymphoma Society-Quest for Cures (0863-15), Illumina (San Diego), STOP Cancer and Tower Cancer Research Foundation.

The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.
**Supplementary Table 3: Annotation runtime.** System runtime for full annotation of a merged KallistoExperiment (seconds). The columns represent system runtime, the Elapsed Time is the total runtime.

Click here to access the data.

**Supplementary Table 4: KallistoExperiment Formation runtime.** System runtime for the creation of a merged KallistoExperiment (seconds). The columns are similar to Supplementary Table 3.

Click here to access the data.

---

**References**

Open Peer Review

Current Referee Status:  ❓  ✔️

Version 1

Referee Report 24 May 2017
doi:10.5256/f1000research.12258.r22616

Ted Abel, Marie Gaine
Iowa Neuroscience Institute, University of Iowa, Iowa, USA

This paper introduces a RNA-Seq analysis pipeline, Arkas, which combines currently available tools typically used in RNA-Seq studies. The novelty of this pipeline is the encapsulation of tools needed to prepare the data, run quality control checks, analyze the data and perform secondary analyses. This is especially beneficial for investigators new to RNA-Seq analysis with little experience navigating through computational tools. The authors take care to outline the rationale behind creating an easy-to-use interface and how this will increase reproducibility and consistency across RNA-Seq studies. They emphasize the importance of consistency with versions by showing differing results between two Kallisto versions.

However, there are some minor limitations also found in this study:
It would be beneficial to include quality control checks at the beginning of the pipeline to generate data regarding the inputted sequencing files.

It would be interesting to see more processing time information to show the benefit of using this pipeline compared to similar methods.

As is discussed, the inclusion of lncRNAs increases the amount of potentially interesting results from this pipeline. However, the authors have chosen to ignore microRNAs, an important regulator of cellular function. The inclusion of microRNAs as a default option in this pipeline would provide even more potentially interesting results.

The normalization steps and Figure 2 should be discussed in more detail. Specifically, expand on the reasons for choosing these two methods and the differences between the methods and their outputs. In addition, a note about how a user should select a normalization type would help new users.

Whilst the authors suggest that the integration of Docker will help produce reproducible research methods, the in-depth look into Docker is unnecessary, as no data has been provided to show its benefit above other options.

Is the rationale for developing the new software tool clearly explained?
Yes

Is the description of the software tool technically sound?
Yes
Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?
Yes

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?
Yes

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?
Yes

**Competing Interests:** No competing interests were disclosed.

**Referee Expertise:** Molecular neuroscience

We have read this submission. We believe that we have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Author Response 08 Jun 2017

**Giridharan Ramsingh**, University of Southern California, USA

Thank you very much Dr. Abel for your insightful review. The revised manuscript removed the in-depth discussion of Docker because it was too broad. The revised version included a discussion section that compares processing times between Google Genomics, and another BaseSpace application.

Your comments helped address the analysis of microRNAs. For example, Kallisto can process smaller FASTA sequences, however this invokes limitations to the construction of the Target DeBruijn Graph by increasing the path ambiguity of longer read sequences. The revised manuscript now addressed this limitation, and suggested that users analyze microRNAs separately. This analysis feature is not yet a default, but would be a great future addition. We further address details in regard to normalization motivation and selection.

As suggested by the first reviewer Dr. Pimentel, we have significantly reduced the broad discussion section, and explicitly described the motivation for the development of Arkas. We have additionally revised the 'Methods' section to provide a brief overview of the applications, and clearer descriptions of the interface style that included Supplementary Figures depicting both interfaces.

"This paper introduces a RNA-Seq analysis pipeline, Arkas, which combines currently available tools typically used in RNA-Seq studies. The novelty of this pipeline is the encapsulation of tools needed to prepare the data, run quality control checks, analyze the data and perform secondary analyses. This is especially beneficial for investigators new to RNA-Seq analysis with little experience navigating through computational tools. The authors take care to outline the rationale behind creating an easy-to-use interface and how this will increase reproducibility and consistency across RNA-Seq studies. They emphasize the importance of consistency with versions by showing differing results between two Kallisto versions."
However, there are some minor limitations also found in this study:

*It would be beneficial to include quality control checks at the beginning of the pipeline to generate data regarding the inputted sequencing files."

Thank you for this suggestion. Analyzing read quality will guide users into the important decision to filter low quality reads, however Arkas was not designed to address this. In the revised manuscript, we have now mentioned another independent BaseSpace application FastQC which can assess read quality. For users interested in manually uploading sequencing data to BaseSpace, each read must pass a quality filter. This quality filter will automatically reject poor quality reads, and for this we designed Arkas with the assumption that sequenced reads input were of good quality.

"*It would be interesting to see more processing time information to show the benefit of using this pipeline compared to similar methods."

Thank you very much for addressing processing times. The revised manuscript significantly reduced the discussion section to comparisons of processing times. Your remarks inspired the addition of processing times of Arkas. We’ve included further information comparing the processing time to another BaseSpace application RNAExpress. Further, we added processing time information of a different Kallisto analysis pipeline implemented over Google Genomics Platform. The discussion section now is far more concise with greater relevance toward the functionality of our developed software.

"As is discussed, the inclusion of IncRNAs increases the amount of potentially interesting results from this pipeline. However, the authors have chosen to ignore microRNAs, an important regulator of cellular function. The inclusion of microRNAs as a default option in this pipeline would provide even more potentially interesting results."

Including microRNAs is a very great idea. Arkas can quantify microRNAs, but we decided not include microRNAs as default yet. In the revised manuscript we address that the small sequence sizes are a potential limitation to quantification of cDNAs/ncRNAs because it may increase path ambiguities during the construction of the Target DeBruijn graphs. Hence, we suggest that users analyze microRNAs separately and locally. This would be a great additional feature for the next version of Arkas.

"The normalization steps and Figure 2 should be discussed in more detail. Specifically, expand on the reasons for choosing these two methods and the differences between the methods and their outputs. In addition, a note about how a user should select a normalization type would help new users."

Thank you for addressing this. The revised manuscript has now explicitly stated how end-users may decide a selection of the normalization type. We further provide a brief explanation to why unsupervised normalization was selected.

"Whilst the authors suggest that the integration of Docker will help produce reproducible research methods, the in-depth look into Docker is unnecessary, as no data has been provided to show its benefit above other options."
We agree that the discussion of Docker was too broad, and the revised discussion is focused on comparative performance from other cloud platforms.

**Competing Interests:** None.

**Referee Report 18 May 2017**

**Harold Pimentel**
Department of Genetics, Stanford University, Stanford, CA, USA

Note: I am a co-author of the kallisto tool, one of the tools that is used in this pipeline.

Colombo et al. describe Arkas, a tool that takes raw RNA-Seq data and produces several different types of downstream analyses. Arkas leverages existing analysis tools (e.g. kallisto and limma) and platforms (Illumina BaseSpace) to create an easy to use, fast, and reproducible pipeline. A very useful (unique?) feature is that it documents software versions and enforces consistent software versions allowing users to see the potential differences with different software versions. This is made explicit in the "Results" section.

Having all of these tools together greatly reduces the time to setup analyses and also reduces the complexity for RNA-Seq novices who might have no idea where to start. Arkas makes all of the typical figures one might make in a standard RNA-Seq analysis. It also provides gene-set analyses which are often excluded from other pipelines. In my experience, gluing together analyses from differential expression to gene-set analyses can often be an annoyance due to inconsistencies and annotations and versions of these annotations. Arkas nicely solves this problem.

While I think the idea is very good and the tool seems comprehensive, I feel the manuscript needs a bit of work. Here are a few points:

- There are a few areas where the scope seems too broad. In general, I feel that the manuscript can be shortened to be more clear as well as more precise. In particular, the Docker section in the discussion is too broad and the role of Arkas seems lost. I strongly recommend shortening this section and discussing the role of Docker in Arkas more clearly.
- While the abstract and introduction provide a description of Arkas in RNA-Seq analysis, they do not provide a motivation. It is sort of hinted in several sections in the paper, but it is not explicit. The motivation of building another pipeline should be explicit.
- How does this pipeline compare to other pipelines such as Galaxy, DNANexus, etc.? Should probably be noted in the introduction/discussion.
- Perhaps I missed it, but the interface of Arkas does not appear to be described. There is a short subsection "Operation" that doesn't describe the type of interface. It appears to be available on Illumina BaseSpace, but does this make it a command line tool or an online web form style tool? A short description of this interface and possibly supplementary figures (if it is a web form style) should be provided. This is unclear to folks who are not familiar with BaseSpace.
- It should be greater emphasized how this tool can be used to reanalyze existing SRA data with relative ease. In my opinion this is a very strong argument as to why one might want a tool like this.
Areas that can be shortened:

- "Data variance between software versions" can be shortened as some of this is repeated in "Results."
- "Complete transcriptomes enrich annotation information..." Specifics of annotations can probably be removed/condensed. It is probably sufficient to say that some are 3x times larger which can change results drastically.
- "Docker as a cornerstone of reproducible research" The role of Docker in general can probably be shortened and how Arkas leverages it should be made more clear.

More minor points:

- A short sentence at the beginning of "Methods" should give an overview of the two-step process.
- The Galaxy Project (https://usegalaxy.org/) should probably be cited even though the scope is a bit different.
- Figure 1a: "Receiver Operator Characteristic plot" of what? This is stated in the main text, but should also the stated in the figure caption.
- Swap Figure 1d and 1c.
- It seems like BaseSpace sessions can easily be shared? If so, this is an additional strong point of using BaseSpace in Arkas.

Overall, I'm very excited to see this comprehensive tool exist and be described in this paper.

Is the rationale for developing the new software tool clearly explained?
Partly

Is the description of the software tool technically sound?
Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?
Yes

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?
Yes

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?
Yes

**Competing Interests**: I am a co-author of the kallisto tool, one of the tools that is used in this pipeline.

**Referee Expertise**: RNA-Seq analysis methods and data analysis

I have read this submission. I believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.
Thank you very much Dr. Pimentel for your thorough review. We have significantly reduced the broad discussion section, and narrowed the manuscript to the most important features. The ‘Abstract’ and ‘Introduction’ section was reduced to explicitly state the motivations for the design of Arkas. In the revised manuscript, the ‘Methods’ section provides a brief overview of the applications, and the ‘Operation’ section describes the interface style and includes Supplementary Figures depicting both apps.

The second reviewer Dr. Abel also suggested that the in-depth discussion of Docker was too broad. The revised version includes a discussion section that is compares processing times between Google Genomics, and another BaseSpace application. We also have now included brief points in regard to Galaxy.

Your helpful comments helped the manuscript become much more concise. In addition to your remarks, we have addressed important features regarding microRNAs on behalf of the second reviewer. Kallisto can process smaller FASTA sequences, however we have now addressed that users can analyze microRNAs, but we suggest a separate analysis for this.

We thank you very much for your revisions and appreciate your thoughtful remarks. We believe that addressing your remarks the manuscript is greatly elevated. Below are point-by-point responses to your questions.

"Note: I am a co-author of the kallisto tool, one of the tools that is used in this pipeline.

Colombo et al. describe Arkas, a tool that takes raw RNA-Seq data and produces several different types of downstream analyses. Arkas leverages existing analysis tools (e.g. kallisto and limma) and platforms (Illumina BaseSpace) to create an easy to use, fast, and reproducible pipeline. A very useful (unique?) feature is that it documents software versions and enforces consistent software versions allowing users to see the potential differences with different software versions. This is made explicit in the "Results" section.

Having all of these tools together greatly reduces the time to setup analyses and also reduces the complexity for RNA-Seq novices who might have no idea where to start. Arkas makes all of the typical figures one might make in a standard RNA-Seq analysis. It also provides gene-set analyses which are often excluded from other pipelines. In my experience, gluing together analyses from differential expression to gene-set analyses can often be an annoyance due to inconsistencies and annotations and versions of these annotations. Arkas nicely solves this problem.

While I think the idea is very good and the tool seems comprehensive, I feel the manuscript needs a bit of work. Here are a few points:

- There are a few areas where the scope seems too broad. In general, I feel that the manuscript can be shortened to be more clear as well as more precise. In particular, the Docker section in the discussion is too broad and the role of Arkas seems lost. I strongly recommend shortening this section and discussing the role of Docker in Arkas more clearly.*

Thank you very much for your input. In the revised manuscript, we have narrowed the Docker
discussion section to the scope of BaseSpace platform, and have described Arkas' relationship to Docker as an applied infrastructure to this platform. The previous version of the manuscript detailed the role of Docker in the broad concept of reproducible research. We have omitted these details. The revised manuscript describes the interdependent relationship between Arkas and Docker in the context of BaseSpace. For example, Arkas containerized Node.js and R to parse the BaseSpace JSON input information relating to BaseSpace's input fields. The new manuscript explained that Docker and Arkas are not independent entities, and pertain specifically to BaseSpace.

"While the abstract and introduction provide a description of Arkas in RNA-Seq analysis, they do not provide a motivation. It is sort of hinted in several sections in the paper, but it is not explicit. The motivation of building another pipeline should be explicit."

Thank you for this suggestion. We have now explicitly provided the motivation for Arkas' development by mentioning bottlenecks in RNA-sequencing such as sequencing importing and pre-processing steps, and how Arkas rectifies those bottlenecks. In the revised version, we illustrate how Arkas was developed downstream from BaseSpace SRA Import to greatly reduce importing and conversion steps. Also, we now explicitly stated the motivation for Arkas-Quantification such that Kallisto was implemented in parallel, which now scales quantification speed to the Amazon AWS EC2 cluster node availability rate. In addition, the revised manuscript explicitly stated the motivation for Arkas-Analysis, which provides a comprehensive analysis.

"How does this pipeline compare to other pipelines such as Galaxy, DNANexus, etc.? Should probably be noted in the introduction/discussion."

Thank you for this suggestion. In the revised discussion section, we now compare features of other cloud platforms, and other BaseSpace RNA-Seq applications. The revised discussion now included processing times of a large scale RNA-seq analysis that implemented Kallisto using Google Genomics Platform. In addition to Google Genomics, the revised manuscript briefly compares features offered by Galaxy to BaseSpace. Further we compare Arkas to other BaseSpace RNA-Seq applications.

"Perhaps I missed it, but the interface of Arkas does not appear to be described. There is a short subsection "Operation" that doesn't describe the type of interface. It appears to be available on Illumina BaseSpace, but does this make it a commandline tool or an online web form style tool? A short description of this interface and possibly supplementary figures (if it is a web form style) should be provided. This is unclear to folks who are not familiar with BaseSpace."

Thank you again for this suggestion. We have included a description explicitly stating that Arkas is a web form style. In addition, we included two Supplementary Figures to address the web input forms. Supplementary Figure 1 shows the input form for both web style apps, and Supplementary Figure 2 shows the output folder directory of the Arkas-Quantification.

"It should be greater emphasized how this tool can be used to reanalyze existing SRA data with relative ease. In my opinion this is a very strong argument as to why one might want a tool like this."
Thank you for addressing reanalysis of SRA data. In the updated manuscript, we now mention that Arkas’ design was motivated by the BaseSpace application SRA Import. The revised introduction now explicitly stated that Arkas is SRA compatible and we have provided citations for readers interested in utilizing this SRA application.

"Areas that can be shortened:

- "Data variance between software versions" can be shortened as some of this is repeated in "Results."

We combined the “Data variance between software versions” and “Results” section into an appropriate concise section.

- "Complete transcriptomes enrich annotation information..." Specifics of annotations can probably be removed/condensed. It is probably sufficient to say that some are 3x times larger which can change results drastically.

We reduced this discussion to brief specifics of database sizes. While obvious, we believe that a brief overview provides motivation for the default transcriptomes chosen by Arkas. In the revised manuscript, we provide a very concise explanation behind the selection of default transcriptomes.

- "Docker as a cornerstone of reproducible research" The role of Docker in general can probably be shortened and how Arkas leverages it should be made more clear.

Thank you again for this comment. We agree that this broad discussion went off topic and may distract future readers. The manuscript is greatly improved with the removal of the discussion about democratization of research efforts, and biotechnology. We significantly revised the discussion to a comparison of differing cloud platforms and corresponding processing times of other cloud applications.

"More minor points:

- A short sentence at the beginning of "Methods" should give an overview of the two-step process.

We provided an overview of Arkas in the section described.

- The Galaxy Project (https://usegalaxy.org/) should probably be cited even though the scope is a bit different.

Galaxy is briefly mentioned in the discussion. The revised manuscript reviewed and compared processing times of Google Genomics Platform and another RNAseq application within BaseSpace.

- Figure 1a: "Receiver Operator Characteristic plot" of what? This is stated in the main text, but should also the stated in the figure caption.
- Swap Figure 1d and 1c.

Thank you for pointing this out. The revised Figure 1a now states that the Receiver Operator
Characteristic plot is for ratios of detected and actual spiked ERCC sequences. We have swapped Figure 1d and Figure 1c.

"It seems like BaseSpace sessions can easily be shared? If so, this is an additional strong point of using BaseSpace in Arkas."

We now mention this brief point in the discussion.

"Overall, I'm very excited to see this comprehensive tool exist and be described in this paper."

Thank you very much Dr. Pimentel.

**Competing Interests:** None