SCIENTIFIC REPORTS

OPEN

A cloud-based workflow to quantify transcript-expression levels in public cancer compendia

Received: 16 August 2016
Accepted: 22 November 2016
Published: 16 December 2016

PJ Tatlow[1] & Stephen R. Piccolo[1,2]

Public compendia of sequencing data are now measured in petabytes. Accordingly, it is infeasible for researchers to transfer these data to local computers. Recently, the National Cancer Institute began exploring opportunities to work with molecular data in cloud-computing environments. With this approach, it becomes possible for scientists to take their tools to the data and thereby avoid large data transfers. It also becomes feasible to scale computing resources to the needs of a given analysis. We quantified transcript-expression levels for 12,307 RNA-Sequencing samples from the Cancer Cell Line Encyclopedia and The Cancer Genome Atlas. We used two cloud-based configurations and examined the performance and cost profiles of each configuration. Using preemptible virtual machines, we processed the samples for as little as $0.09 (USD) per sample. As the samples were processed, we collected performance metrics, which helped us track the duration of each processing step and quantified computational resources used at different stages of sample processing. Although the computational demands of reference alignment and expression quantification have decreased considerably, there remains a critical need for researchers to optimize preprocessing steps. We have stored the software, scripts, and processed data in a publicly accessible repository (https://osf.io/gqrz9).

Over the past decade, public cancer compendia have played a crucial role in enabling scientists to identify genomic, transcriptomic, proteomic, and epigenomic factors that influence tumor initiation, progression, and treatment responses[1–19]. Due to efforts like The Cancer Genome Atlas (TCGA), International Cancer Genomics Consortium, Cancer Cell Line Encyclopedia (CCLE), and Connectivity Map, thousands of studies have been published. Typically, consortia who oversee these efforts release raw *and* preprocessed data for the public to use. Accordingly, researchers who wish to reprocess raw data using alternative methods may do so[20–22]. For example, we previously reprocessed 10,005 RNA-Sequencing samples from TCGA and demonstrated that an alternative pipeline provided analytical advantages over the preprocessed data provided by the TCGA consortium[20]. However, this effort required us to copy more than 50 terabytes of data, across three time zones, from the data repository to our local file servers—and to employ tens of thousands of hours of computational time on local computer clusters. Other efforts, such as the Genomic Data Commons[23], are also reprocessing cancer compendia using updated pipelines. Such efforts require considerable institutional investment in computational infrastructure[24]. In the case of raw sequencing data, computational infrastructure must also implement appropriate security measures to ensure patient privacy[25,26]. Many research institutions do not have the resources to support such infrastructure, and duplicate efforts may unwittingly occur, resulting in wasted resources.

As an alternative, the National Cancer Institute initiated the Cancer Genomics Cloud Pilots[27], which enable researchers to access cancer compendia via cloud-computing services, such as Google Cloud Platform[28] or Amazon Web Services[29]. Via these public/private partnerships, cancer data are stored (and secured, as necessary) in shared computing environments. Researchers can rent virtual machines (VMs) in these environments and apply computational tools to the data, without needing to transfer the data to or from an outside location. This model promises to speed the process of scientific discovery, reduce barriers to entry, and democratize access to the data[30,31]. In the current era of highly collaborative science, this model also makes it easier for researchers from multiple institutions to collaborate in the same computing environment.

The landscape of bioinformatics tools available to process RNA-Sequencing data is rapidly evolving. Seemingly small differences in software versions or annotations can lead to considerable analytical differences or make it

[1]Department of Biology, Brigham Young University, Provo, Utah, USA. [2]Department of Biomedical Informatics, University of Utah, Salt Lake City, Utah, USA. Correspondence and requests for materials should be addressed to S.R.P. (email: stephen_piccolo@byu.edu)

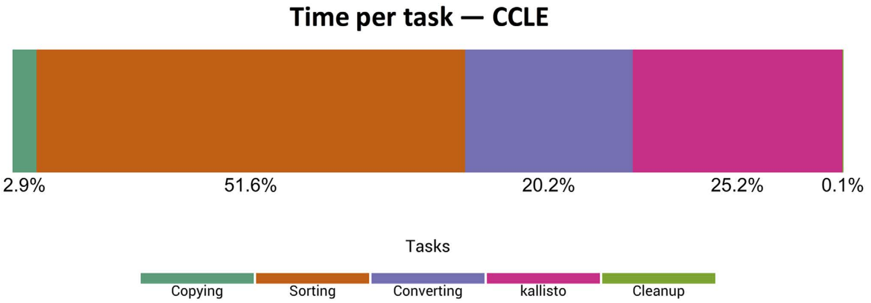

Figure 1. Relative time spent on computational tasks for CCLE samples using a cluster-based configuration. We logged the durations of individual processing tasks for all CCLE samples, averaged these values, and calculated the percentage of overall processing time for each task. Because the raw data were stored on Google Cloud Storage, copying the BAM and index files to the computing nodes took less than 3% of the total processing time. For preprocessing, the BAM files were sorted and converted to FASTQ format, which took 71.8% of the overall processing time. The *kallisto* alignment and quantification steps took only 25.2% of the overall processing time.

difficult to integrate datasets[32,33]. However, because the Cancer Genomics Cloud Pilots provide access to raw data, researchers can reprocess the data using whatever tools and annotations support their specific needs.

In coordination with the Institute for Systems Biology (ISB)[34], we used the Google Cloud Platform to process 12,307 RNA-Sequencing samples from the CCLE and TCGA projects. After preprocessing, we aligned the sequencing reads to the most current GENCODE reference transcriptome (see Methods) and calculated transcript-expression levels using *kallisto*, a pseudoalignment and read-quantification program that executes considerably faster than previous-generation tools, while attaining similar levels of accuracy[35]. We encapsulated all the software necessary to perform these steps into software containers[33,36]. Where possible, the containers execute tasks in parallel and dynamically determine the number of tasks that can be executed simultaneously. In addition, the containers collect detailed computer-performance metrics while the containers execute. We have made these containers freely available to the research community—along with scripts for executing and monitoring them (see Methods).

We processed the CCLE samples ($n = 934$) using either 1) a cluster-based configuration, orchestrated by the *Kubernetes* system[37], or 2) preemptible VMs. In this paper, we describe our experiences with these deployment approaches. The cluster-based configuration more closely resembles computing environments typically available at research institutions and thus may be more intuitive for researchers to use. However, using preemptible VMs, we were able to process the data at a considerably lower cost and with less monitoring overhead. Therefore, we used preemptible VMs to process all available TCGA RNA-Sequencing samples ($n=11,373$) for a total cost of $1,065.49.

Below we describe lessons learned as we processed these data, and we discuss logistical and financial issues that should be considered when using cloud-computing environments. We hope these observations will enable researchers to better evaluate options for processing large biological datasets in the cloud. We also explore opportunities for bioinformaticians to optimize data processing.

Results

Cluster-based configuration (CCLE data).

We created a software container to quantify transcript-expression levels for 934 RNA-Sequencing samples from CCLE on the Google Cloud Platform. Initially, we processed these samples on a cluster of 295 computing nodes. Each computing node had access to 4 virtual central processing units (vCPUs), 26 gigabytes of random access memory (RAM), and 400 gigabytes of disk-storage space. We used the *Kubernetes* system to distribute execution across these nodes in a queue-like manner: as one sample completed processing, a new sample began processing, and so on.

All RNA-Sequencing data were generated using *Illumina* sequencing instruments. After initial, low-level processing, data from these instruments are stored in the FASTQ format[38]. Each entry in a FASTQ file indicates the nucleotide sequence and quality scores for each molecule that was sequenced. Scientists use these files as input to sequence-alignment tools, such as *kallisto*, which aim to identify the transcriptomic region associated with each molecule. The output of sequence-alignment tools is typically in the BAM format[39]. These binary files indicate the nucleotide sequence, quality scores, and alignment location of each molecule.

The CCLE samples were available in BAM format only. Because *kallisto* only accepts input data in FASTQ format, the first step in our pipeline was to sort the BAM data by name and convert the data back to FASTQ format. This step consists of extracting the nucleotide sequences and quality scores and reformatting the data (see Methods). To enable faster execution, we performed these steps in parallel (see Methods). On average, these preprocessing steps took 84.7 minutes per sample (71.7% of the processing time), whereas *kallisto* took only 29.8 minutes (25.2% of the processing time; see Fig. 1). The remaining 3.6 minutes (3.0%) were used to copy files to and from the computing nodes. These observations indicate that RNA-Sequencing preprocessing steps require a considerable amount of computational time and must be taken into account when planning resource allocation and estimating costs.

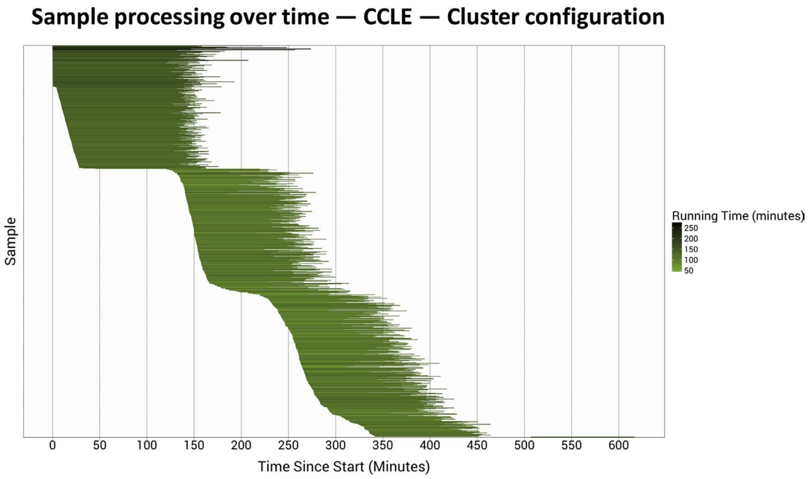

Figure 2. Processing time per CCLE sample using the cluster-based configuration. The 934 CCLE samples were processed on a cluster of 295 virtual machines. The horizontal lines represent the relative start and stop times at which each sample was processed. Darker lines identify samples that took longer to process. The longest-running sample took over 4.5 hours to process; the shortest-running sample completed in less than 1 hour. All but two samples finished processing within 7.5 hours. These two samples failed due to a disk-mounting error, so we reprocessed the samples on a smaller cluster (see Methods).

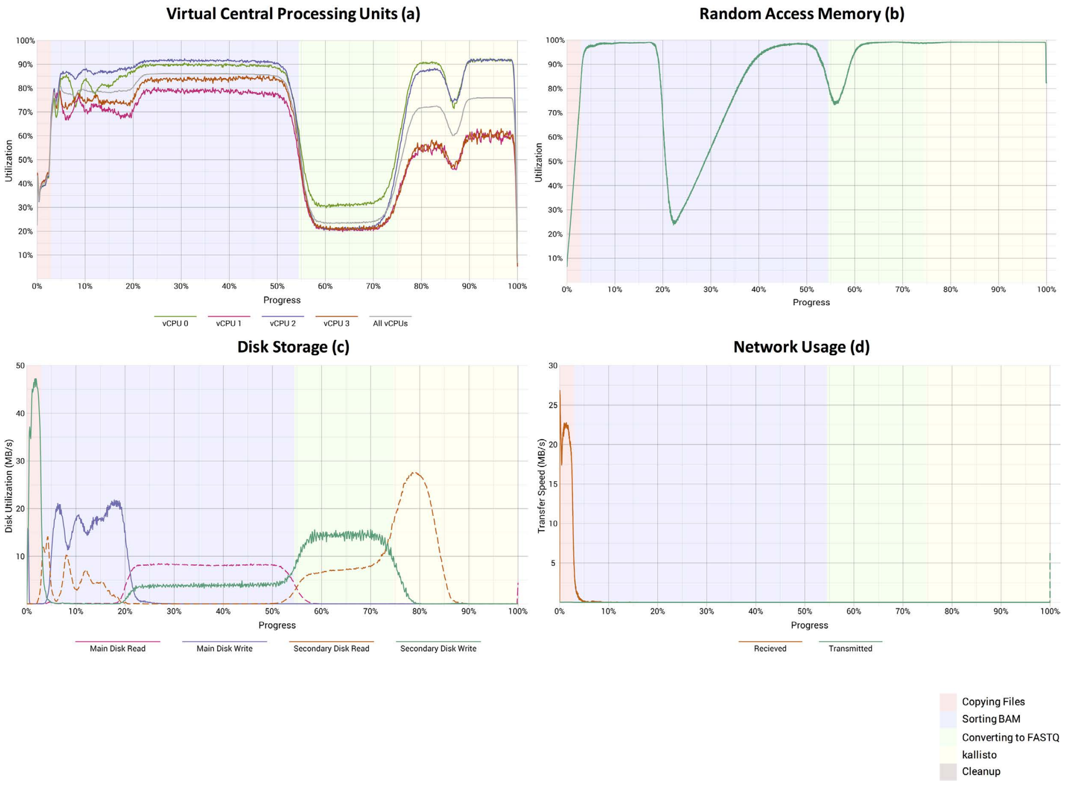

Figure 3. Computational resource utilization while CCLE samples were processed using a cluster-based configuration. These graphs show the (**a**) percentage of user and system vCPU utilization, (**b**) percentage of memory usage, (**c**) disk activity, and (**d**) network activity. The "main" disk for each virtual machine had 100 gigabytes of storage space. The "secondary" disks, which stored temporary files during the sorting and FASTQ-to-BAM conversion steps, had 300 gigabytes of space. The background colors represent the five computational tasks shown in Fig. 2. Each graph summarizes data from all 934 CCLE samples.

One limitation of a cluster-based configuration is that all computing nodes must remain active while even one sample is still processing. In general, larger RNA-Sequencing samples take longer to process than smaller

Category	CCLE (cluster)	CCLE (preemptible)	TCGA (breast, lung)	TCGA (other cancer types)
# samples	934	934	1,811	9,562
Data input format	BAM	BAM	FASTQ	FASTQ
Total data size[a]	12.59	12.59	10.67	57.8
# vCPUs per node[b]	4	16	2	2
RAM per node[c]	26	104	7.5	7.5
Primary disk storage per node[d]	100	100	10	10
Secondary disk storage per node[e]	300	300	350	350

Table 1. Summary of data sets and computational resources. [a]Total amount of raw sequencing data (measured in terabytes). [b]vCPU = virtual central processing unit. [c]RAM = random access memory (measured in gigabytes). [d]The primary disks were used to store operating system components and temporary files. [e]The secondary disks were used to store input and output data. Storage capacity was measured in gigabytes.

Expense category[a]	CCLE (cluster)	CCLE (preemptible)	TCGA (breast, lung)	TCGA (other cancer types)
Data transfer	$0.02	$0.08	$0.10	$0.70
Data storage	$111.27	$25.48	$66.10	373.08
Computing nodes	$605.16	$338.09	$95.75	$520.80
Miscellaneous	$0.07	$0.57	$1.47	$8.12
Total cost	*$716.52*	*$364.22*	*$163.42*	*$902.07*
Cost per sample	*$0.767*	*$0.390*	*$0.090*	*$0.094*

Table 2. Summary of cost information. [a]We obtained cost information for the Google Cloud Platform and assigned expenses to high-level categories. All values are in US Dollars (rounded to the nearest cent). All TCGA samples were processed using preemptible VMs. There was no charge for the time required to spinup VMs, pull images, localize files, or delocalize files.

samples. Therefore, in an attempt to reduce costs, we submitted samples to the queue in order of size (largest to smallest). Figure 2 illustrates the start and end times at which each sample was processed. The longest-running sample finished processing in 274 minutes, while one sample took only 46 minutes. Few computing nodes were idle during the first 350 minutes of overall processing time, and all but two samples had completed processing after 450 minutes. Disk-storage drives failed to mount properly while the remaining two samples were processing, so these samples had to be resubmitted to the queue.

We collected computer-performance metrics at one-second intervals while the software container was executing. Resource utilization varied dramatically across the processing tasks (Fig. 3). The sorting and *kallisto* phases were relatively vCPU intensive, whereas the BAM-to-FASTQ conversion used the disk-storage devices relatively heavily. RAM usage was consistently high, although it decreased temporarily during the sorting and conversion stages. As expected, network usage was high while data files were being transferred to and from computing nodes—but zero at other times. vCPU usage was highly variable across the individual vCPUs on a given computing node, and vCPUs were never used at full capacity for a considerable length of time.

The cost for each node in the cluster was $0.252 (USD) per hour. There was no cost for the master node that orchestrated the cluster. These computing costs and data-storage costs constituted the great majority of expenses that we incurred (see Tables 1 and 2). In total, the cost was $716.52 ($0.77 per sample). Because cloud-computing offerings, service prices, and bioinformatics tools change regularly, it is difficult to compare costs across projects. However, a 2011 study used the MAQ software[40] to align a single biological sample in the cloud for $320.10[29].

Preemptible-node configuration (CCLE data). Next we reprocessed the CCLE RNA-Sequencing samples using an alternative approach. First, to assess whether quality trimming would have a significant effect on the *kallisto* quantification values, we excluded low-quality bases and any adapter sequences that were present in the data. Second, we used preemptible VMs and compared the performance and cost profiles of this configuration against what we observed for the cluster-based option.

Typically, cloud-service providers have more computing nodes available than customers will use at any given time. While these extra nodes are idle, users can request to use the nodes with "preemptible" status—at a considerably reduced price. When demand rises, the service provider halts (preempts) execution of the nodes and makes them available to full-paying customers. In situations where nodes are preempted infrequently—and preemptible nodes are available—this approach can reduce costs, although additional care must be taken to ensure that preempted samples are re-executed when new nodes become available.

To reduce the time that the CCLE samples would take to process—and thus to reduce the likelihood of preemption—we used more powerful computing nodes: 16 vCPUs and 104 gigabytes of RAM per node. As the samples were processing, we executed a script, every 60 seconds, to check whether any samples had been preempted and then resubmitted those samples for processing (see Methods). Unlike the cluster-based configuration, which employed a static number of computing nodes, the number of preemptible nodes changed dynamically, according to availability. During the first 20 minutes, 932 nodes were allocated, and most samples completed

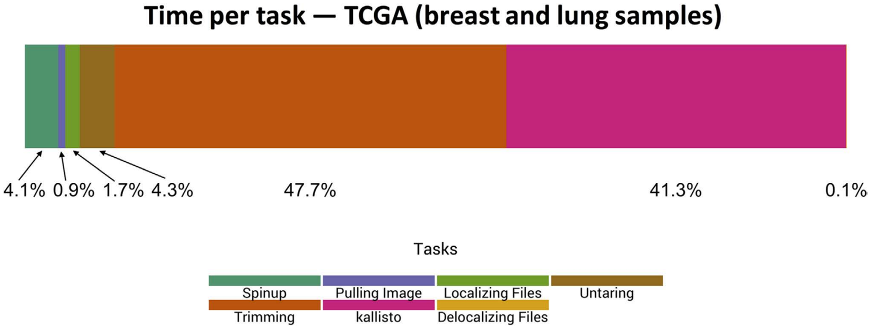

Figure 4. Relative time spent on computational tasks for TCGA breast and lung samples using a preemptible-node configuration. We logged the durations of individual processing tasks for the TCGA breast and lung samples, averaged these values, and calculated the percentage of overall processing time for each task. The "spinup," image pulling, and file localization steps enabled the virtual machines to begin executing. For sample preprocessing, the FASTQ files were unpacked, decompressed, and quality trimmed; together these steps took 52.0% of the processing time (on average). The *kallisto* alignment and quantification steps took 41.3% of the overall processing time.

processing before 150 total minutes had elapsed (Figure S1). Of the 934 CCLE samples, 78 were preempted at least once (Figure S1). Eight samples were preempted twice, and one sample was preempted three times. On average, preempted nodes executed for 76.64 minutes (min = 1.87; max = 126.57) before preemption. The BAM files for the preempted samples averaged 14.07 gigabytes in size, whereas the remaining samples averaged 13.79 gigabytes. Accordingly, preemption was a relatively random process, not necessarily influenced by the size of the data files.

Trimming next-generation sequencing reads is a common, though somewhat controversial, practice[41–43]. Trimming can remove low-quality bases and reads and can remove artefactual adapter sequences. We compared transcript-expression levels from our initial (untrimmed) CCLE data against the data that had been subjected to trimming. The average, sample-wise Spearman correlation coefficient was 0.988; however, the correlation coefficients for some samples were as low as 0.92 (Figure S2), suggesting that these samples had a considerable number of low-quality reads.

The performance profiles for the preemptible configuration were similar to those from the cluster-based configuration (Figures S3 and S4). The read-trimming step required an average of 7.97 minutes (9.0%) of processing time per sample (see Figure S3). In addition, there was considerable overhead to "spin up" and configure the VMs and to localize the necessary files (19.4% of processing time). We suspect this overhead was relatively high because the cloud environment needed to allocate a relatively large amount of resources (vCPUs and RAM) per computing node. Overall, the *kallisto* alignment and quantification process took only 9.8% of the processing time, reaffirming that preprocessing steps may contribute more to the expense of processing RNA-Sequencing data than alignment and quantification, which have been optimized substantially in recent years.

The cost of the preemptible nodes (16 vCPU) was $0.280 per hour. In addition, we rented a master node for $0.01 per hour. The total cost to process the samples was $364.22 ($0.39 per sample). Even though preemption occurred 88 times—contributing an estimated $31.48 to the total cost—the overall cost was 49.2% lower than the cluster-based configuration. This was true, even though we added the read-trimming step. Accordingly, we conclude that preemptible nodes show promise as a way to reduce expenses markedly, so long as preemption rates remain low.

Preemptible-node configuration (TCGA data).

Based on our experiences with the CCLE samples, we used preemptible VMs to process 11,373 RNA-Sequencing samples from TCGA that spanned 33 cancer types. The TCGA samples were available in FASTQ format; therefore, we did not need to sort or convert the data. However, the FASTQ file were packed inside "tar" files, so we included a step to unpack these files. Based on preliminary testing, we estimated that using less-powerful computing nodes—2 vCPUs and 7.5 gigabytes of RAM—would provide an optimal balance between cost and performance.

We started with 1,811 breast carcinoma and lung squamous cell carcinoma samples. These samples completed processing in 101 minutes on average, compared to 88.5 minutes for the CCLE samples. Although individual TCGA samples took 14% longer to process than the CCLE samples, the computing nodes used 87.5% fewer computational resources per sample. The VM "spinup" and configuration tasks were considerably shorter (Fig. 4), ostensibly because the computing nodes required fewer resources. The *kallisto* step took 41.3% of the processing time, due to the shorter preprocessing times. Only 210 preemptions occured (Fig. 5).

Because most of the TCGA samples were assigned to a single read group, we were not able to parallelize the quality-trimming step in most cases. Therefore, one of the vCPUs was consistently underutilized during the trimming phase (Fig. 6). However, *kallisto* maximized use of both vCPUs during most of its processing; it also consistently used all available RAM. Disk usage was heaviest while the FASTQ files were being unpacked, at the end of quality trimming, and at the beginning of *kallisto* execution. Because the data files were transferred to the VMs before our software container began executing, we detected no network activity (as expected).

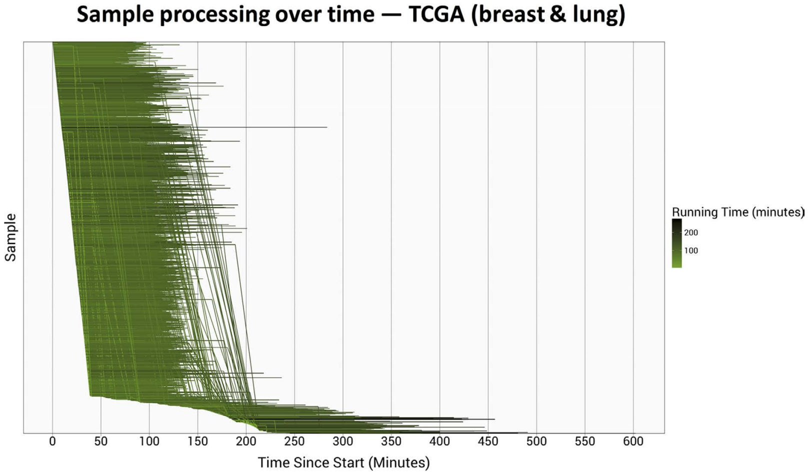

Figure 5. Processing time per TCGA breast and lung sample using a preemptible-node configuration. The 1,811 TCGA breast and lung samples were processed using a variable number of preemptible virtual machines. The horizontal lines represent the relative start and stop times at which each sample was processed. Darker lines identify samples that took longer to process. Vertical lines indicate times at which samples were preempted and then resubmitted for processing. In total, 210 preemptions occurred.

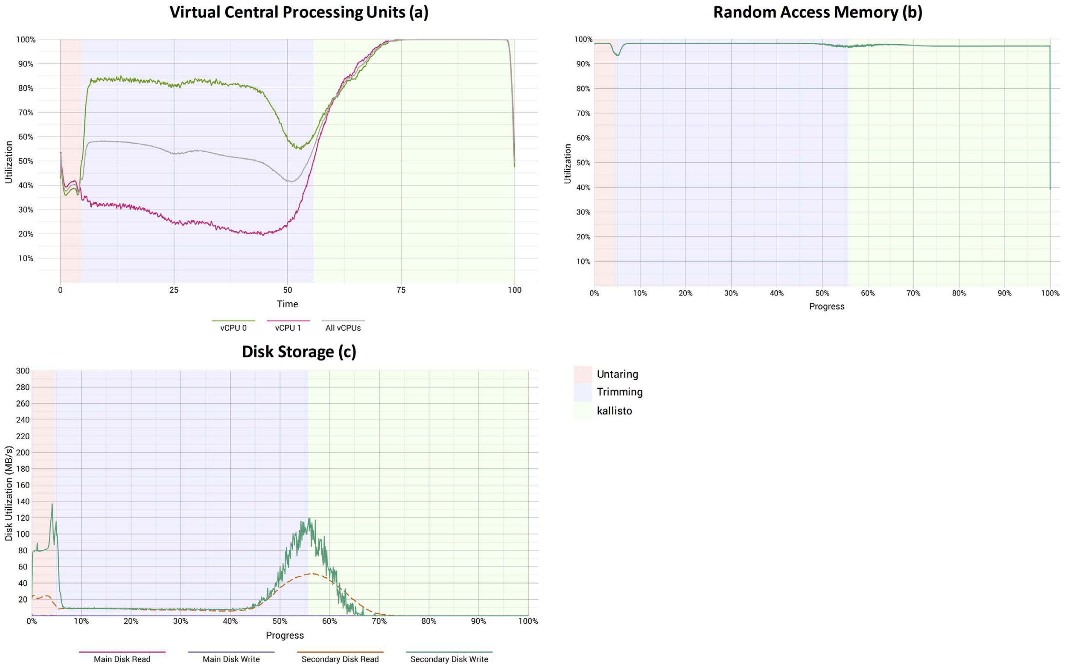

Figure 6. Computational resource utilization while TCGA breast and lung samples were processed using a preemptible-node configuration. These graphs show the (**a**) percentage of user and system vCPU utilization, (**b**) percentage of memory usage, and (**c**) disk activity. The "main" disks had 10 gigabytes of storage space and stored operating-system files. The "secondary" disks, which stored all data files, had 350 gigabytes of space. The background colors represent the computational tasks shown in Fig. 4. We were unable to collect performance metrics for preliminary tasks, such as file localization, because these tasks were not performed within the software container. Each graph summarizes data observed across all 1,811 TCGA lung and breast samples. Because there was typically only one pair of FASTQ files per sample, quality trimming could not be parallelized; therefore, we used only 2 vCPUs per sample.

Lastly, we applied our software container to the remaining 9,562 TCGA samples, which ranged from 0.27 to 27.69 gigabytes in size. In total, raw sequencing data for the TCGA samples were 64.47 terabytes in size and consumed a total of 30,795.2 vCPU hours. Overall, the cost to process these samples was $1,065.49 ($0.09 per sample; see Tables 1 and 2). These costs are far lower than any prior cloud-based project of which we are aware.

We have deposited the processed data—as read counts and transcripts per million—in tabular format and in Google's BigQuery database (see https://osf.io/gqrz9).

Discussion

The size of biomolecular data will continue to increase at a dramatic pace over the coming years[31]. To devise solutions for managing and processing these data, the research community will likely benefit from working with industry partners[44], including cloud-computing providers[45–48]. Here we describe lessons learned as we quantified transcription levels for 12,307 samples from CCLE and TCGA. We hope these insights will be useful both to the academic community and to industry partners.

Our analyses were inexhaustive. Indeed, further optimization may yield better performance and lower cost than we observed. However, we have demonstrated that raw RNA-Sequencing data can be preprocessed, aligned, and quantified for pennies per sample. Furthermore, to enable others to reproduce our findings, we have provided all source code that we used to process the samples, collect performance metrics, and create the figures in this paper (see Methods). Our workflow can be used to process additional RNA-Sequencing samples available via the NCI Cloud Pilots or to process samples that have been sequenced by individual research groups. With minor modifications to our pipeline, different reference genomes or algorithms could be applied to these data. We hope these resources will serve as examples to other researchers who wish to start using cloud environments.

Due to the way our workflow is structured, it was necessary to completely reprocess samples that had been preempted. An alternative solution would be to use persistent disk storage to store intermediate data files that result from individual processing steps within the workflow. When a sample is preempted, the workflow could check for these intermediate files and resume processing at the latest checkpoint. This approach might reduce overall processing times, but it would require users to rent additional storage space and thus might offset savings gained from reduced processing times. Yet another solution would be to split the workflow into smaller segments and execute the workflow in stages. Again, this would require additional persistent storage, and the VMs would need to be deployed repeatedly, a step that can take several minutes at a time. Although such approaches may provide advantages in some cases, we favor simplicity, especially in situations where preemption occurs infrequently.

Our performance-profiling data demonstrate that computational resources are often underutilized by bioinformatics tools. For example, in our pipeline, vCPUs were used unevenly and often not at full capacity. Although it is unreasonable to expect that all computational resources can be used at full capacity at all stages of data processing, opportunities exist for tool developers to examine ways to optimize the use of these resources.

Cloud environments enable scientists to flexibly determine which resources are employed for a given research task. In contrast, academic computing environments are relatively fixed resources that may be oversubscribed, forcing individual researchers to wait long periods of time to process their data. At other times, these environments may be underutilized, thus reducing cost efficiency. In addition, because individual researchers typically do not have administrative privileges in academic computing environments, they may be limited in how they can customize their workflows. A central feature of our approach is the use of software containers, which enabled us to package, deploy, and monitor our software more readily. We anticipate that as containerization technologies continue to mature, they will be an integral part of cloud-computing workflows (and perhaps academic environments).

Because it is impossible to control every possible variable that might influence computational performance, it is impossible to guarantee that the performance metrics we collected would be exactly reproducible if the analyses were repeated multiple times, even if the same hardware were used. Such is the nature of computations, especially in third-party environments. However, we are confident that the performance metrics we report are representative of the levels of performance attainable with the tools and configurations we used. We collected these observations across thousands of RNA-Sequencing samples. Thus even though some variability would occur for individual samples or steps, we expect that the overall performance differences would be consistent. In addition, Docker containers use the *cgroups* feature in Linux to ensure that containers can access only a finite amount of computational resources while they are executing[49]. Furthermore, each container used a distinct disk-storage device, which helped to ensure that the disk usage of one RNA-Sequencing sample was isolated from other samples that were processing concurrently.

Commercial cloud-computing environments have been available for more than a decade. Sequencing and bioinformatics service providers commonly use cloud environments to apply bioinformatics tools to genomic data[50]. Recently, publicly funded, data-generating consortia, including projects funded by the National Cancer Institute, have begun to explore the benefits and challenges of providing centralized access to large compendia of genomic data. Our experiences confirm that the cloud has potential to make it easier to apply custom workflows to sequencing data at a modest price. Accordingly, we believe cloud computing will play an increasingly important role in cancer research.

We opine that the cloud will be most useful in research settings when the size of raw data exceeds the storage capacity of in-house computer clusters, when there is a demand to reprocess raw data using alternative tools or parameter configurations, when raw data must be accessible to researchers from multiple institutions, or when it is essential to process data in a short time period. Such requirements are common in the current era of genome research—especially for public data compendia. In contrast, when individual research groups need to process moderately sized datasets using a standard computational pipeline, in-house resources may be more suitable. Alternatively, in some cases, it may be better to avoid the cloud as an added security precaution.

Initially, we processed the CCLE samples using a cluster-based container engine, which is loosely comparable to cluster-computing environments commonly used at academic institutions. A disadvantage of this configuration is that the entire container engine needs to remain available while any sample remains to be processed. It may be possible to mitigate this limitation by processing the larger samples first (as we did); however, if any error occurs and sample(s) must be re-added to the queue, cost efficiency will decrease. When we used preemptible

VMs to process the same samples, we did not need to pay for excess computing power and paid 50–70% less per computing node. Thus even though some samples were preempted and had to be reprocessed, the overall cost was much lower. We recommend this option as a way to reduce expenses.

Due to the rapidly changing landscape of next-generation sequencing analysis, it is extremely beneficial for cloud providers to provide access to raw data. Until recently, raw sequencing data from CCLE and TCGA were available via the CGHub service[51]. Users who desired to compute on these data were required to bring "data to the tools," whereas in cloud-computing environments, users can take their "tools to the data"[52]. In the case of CCLE, raw data were available in BAM format, which required us to convert the files to FASTQ format. In addition, we trimmed the reads for quality control. Together these steps took 70–75% of the processing time per sample (Figs 1 & S3). The TCGA raw data were available in FASTQ format; however, we needed to decompress and unpack *tar* files before trimming the reads (52% of the processing time). Accordingly, we recommend that alignment and quantification tools support both FASTQ and BAM input files—ideally, they would also support quality trimming. We also recommend that cloud providers store two versions of raw sequencing data: (1) compressed (but not packed) files that have been manipulated by no bioinformatics tool, and (2) files that have been sorted, trimmed, and compressed using standard tools. The former could be used by those who prefer to preprocess the data using alternative tools, whereas the latter could be used by all other users. Although storing these two versions of the raw data would increase disk-storage costs, it would likely decrease overall costs. The computational costs of file-format conversions may also be significant in other research domains.

We encourage funding agencies and research institutions to consider implications of cloud computing on research budgets. Whereas academic-computing infrastructure has traditionally been funded by indirect costs, individual researchers may be required to categorize cloud computing as direct costs. Efforts such as The NIH Commons have begun to address this issue[53]. Finally, we encourage university training programs to place an emphasis on teaching students how to use cloud-computing resources.

Methods

Cluster-based configuration. The Google Cluster Engine allowed us to control hundreds of dedicated computing nodes with one central node. This central node used *Kubernetes*[37] to manage resources and a queue. Using a YAML[54] configuration file, we submitted each RNA-Sequencing sample to *Kubernetes*, which then scheduled the samples to execute on the worker nodes. For each sample, *Kubernetes* deployed the relevant software container to the node, copied data files to the node, executed the software container, and copied output files to persistent storage. In cases where execution failed, *Kubernetes* automatically restarted sample processing. By default, the Google Cluster Engine allocated a 100 GB disk to each computing node; however, due to the size of the raw data and disk-space requirements for sorting and converting to FASTQ, we mounted a secondary disk drive (300 GB) to each node. While processing the samples, two nodes experienced errors while mounting disk drives. We used a two-node cluster to finish processing the remaining samples.

Preemptible-node configuration. We used the Google Genomics Pipeline service[55] to control execution of tasks on preemptible computing nodes. As a preemptible node became available, the service (1) created a VM on the node, (2) deployed the relevant software container to the VM, (3) copied data files to the secondary disk drive, (4) executed the software container, (5) copied output files to persistent storage, and (6) destroyed the VM after the sample finished processing (or was preempted). To help facilitate this process, we used a software framework provided by the Institute for Systems Biology[56]. Via a command-line interface, this framework facilitated the process of submitting samples to the Google Genomics Pipeline for processing. In addition, the framework monitored each sample's status and resubmitted samples that had been preempted (at 60-second intervals).

Software containers. We created three different Docker containers to house the software required to process each combination of data source and cloud configuration. The first container was used by the Google Cluster Engine to process BAM files from CCLE. The cluster-based configuration required us to use the container to copy input files to and from the computing nodes. After copying the files, the container used *Sambamba* (version 0.6.0)[57] to sort the BAM files by name and then *Picard Tools* (version 2.1.1, SamToFastq module)[58] to convert the BAM files to FASTQ format. In accordance with *kallisto*'s documentation, we used the "OUTPUT_PER_RG" flag in *Picard Tools* to ensure that paired-end reads were placed in separate output files. The FASTQ files were then used as input to *kallisto* (version 0.43.0)[35], which pseudoaligned the reads to the GENCODE reference transcriptome (version 24)[59] and quantified transcript-expression levels. Based on the *kallisto* authors' recommendation, we used 30 bootstrap samples; we also used the "–bias" flag to account for variance and bias. We used the parallelization features in *Sambamba* and *kallisto* to enable faster processing.

The second software container is similar to the first but was modified for use with the Google Genomics Pipeline service. Because this service handles copying the data files between Google Cloud Storage and the computing nodes, these tasks were not performed by the container. We also added a read-trimming step using *Trim Galore!* (version 0.4.1)[60], a wrapper around *Cutadapt* (version 1.10)[61]. This tool trims adapter sequences and low quality bases/reads. To process multiple FASTQ files (or pairs of FASTQ files for paired-end reads) in parallel, we used *GNU Parallel* (version 20141022)[62].

The third software container was designed specifically for the TCGA data. It extracts FASTQ files from a tar archive (whether compressed or not), performs quality trimming, and executes *kallisto*. Where applicable, it uses the *pigz* tool (version 2.3.1)[63] to decompress the input files in parallel.

All three containers use the *sar* module of the *sysstat* program (version 11.2.0)[64] to log each machine's vCPU, memory, disk, and network activity throughout the course of data processing. The containers copied these data to persistent storage, prior to the job's completion. We changed the time of each entry in the logs to a corresponding percentage of total job time, to allow the activity metrics to be summarized consistently across all jobs.

Availability of data and materials. An open-access repository (https://osf.io/gqrz9) contains all the scripts we used to construct the Docker containers and to process samples on the Google Cloud Platform. The repository also includes summarized performance data and scripts that we used to generate the figures in this manuscript.

References

1. Network, T. C. G. A. R., Cancer, T. & Atlas, G. Comprehensive genomic characterization defines human glioblastoma genes and core pathways. *Nature* **455,** 1061–1068 (2008).
2. Cancer Genome Atlas Research Network. Integrated genomic analyses of ovarian carcinoma. *Nature* **474,** 609–615 (2011).
3. The Cancer Genome Atlas Research Network. Comprehensive genomic characterization of squamous cell lung cancers. *Nature* **489,** 519–525 (2012).
4. Koboldt, D. C. *et al.* Comprehensive molecular portraits of human breast tumours. *Nature* **490,** 61–70 (2012).
5. Omberg, L. *et al.* Enabling transparent and collaborative computational analysis of 12 tumor types within The Cancer Genome Atlas. *Nat. Genet.* **45,** 1121–1126 (2013).
6. Verhaak, R. G. W. W. *et al.* Integrated genomic analysis identifies clinically relevant subtypes of glioblastoma characterized by abnormalities in PDGFRA, IDH1, EGFR, and NF1. *Cancer Cell* **17,** 98–110 (2010).
7. Piccolo, S. R. & Frey, L. J. Clinical and molecular models of glioblastoma multiforme survival. *Int. J. Data Min. Bioinform*. **7,** 245–265 (2013).
8. Kandoth, C. *et al.* Mutational landscape and significance across 12 major cancer types. *Nature* **502,** 333–339 (2013).
9. Zhang, J. *et al.* International Cancer Genome Consortium Data Portal–a one-stop shop for cancer genomics data. *Database* **2011,** bar026 (2011).
10. Truong, T. *et al.* Replication of lung cancer susceptibility loci at chromosomes 15q25, 5p15, and 6p21: a pooled analysis from the International Lung Cancer Consortium. *J. Natl. Cancer Inst*. **102,** 959–971 (2010).
11. Alexandrov, L. B. *et al.* Signatures of mutational processes in human cancer. *Nature* **500,** 415–421 (2013).
12. International Cancer Genome Consortium *et al.* International network of cancer genome projects. *Nature* **464,** 993–998 (2010).
13. Welch, J. S. *et al.* The origin and evolution of mutations in acute myeloid leukemia. *Cell* **150,** 264–278 (2012).
14. Barretina, J. *et al.* The Cancer Cell Line Encyclopedia enables predictive modelling of anticancer drug sensitivity. *Nature* **483,** 603–607 (2012).
15. Lamb, J. *et al.* The Connectivity Map: using gene-expression signatures to connect small molecules, genes, and disease. *Science* **313,** 1929–1935 (2006).
16. Yang, W. *et al.* Genomics of Drug Sensitivity in Cancer (GDSC): a resource for therapeutic biomarker discovery in cancer cells. *Nucleic Acids Res*. **41,** D955–61 (2013).
17. Duan, Q. *et al.* LINCS Canvas Browser: interactive web app to query, browse and interrogate LINCS L1000 gene expression signatures. *Nucleic Acids Res*. **42,** W449–60 (2014).
18. Bamford, S. *et al.* The COSMIC (Catalogue of Somatic Mutations in Cancer) database and website. *Br. J. Cancer* **91,** 355–358 (2004).
19. Futreal, A. A. *et al.* A census of human cancer genes. *Nat. Rev. Cancer* **4,** 177–183 (2004).
20. Rahman, M. *et al.* Alternative preprocessing of RNA-Sequencing data in The Cancer Genome Atlas leads to improved analysis results. *Bioinformatics* **31,** 3666–3672 (2015).
21. Culot, L. J., O'Hara, A., Keshavan, R., Che, Z. & Shams, S. Abstract 5175: Re-analysis of breast invasive carcinoma (BRCA) TCGA copy number data improves tumor profiles. *Cancer Res*. **74,** 5175–5175 (2014).
22. Gnad, F., Doll, S., Manning, G., Arnott, D. & Zhang, Z. Bioinformatics analysis of thousands of TCGA tumors to determine the involvement of epigenetic regulators in human cancer. *BMC Genomics* **16** Suppl 8, S5 (2015).
23. Home | NCI Genomic Data Commons. Available at: https://gdc.nci.nih.gov/. (Accessed: 9th June 2016).
24. Noor, A. M., Holmberg, L., Gillett, C. & Grigoriadis, A. Big Data: the challenge for small research groups in the era of cancer genomics. *Br. J. Cancer* **113,** 1405–1412 (2015).
25. Siu, L. L. *et al.* Facilitating a culture of responsible and effective sharing of cancer genome data. *Nat. Med*. **22,** 464–471 (2016).
26. Greenbaum, D., Sboner, A., Mu, X. J. & Gerstein, M. Genomics and privacy: implications of the new reality of closed data for the field. *PLoS Comput. Biol.* **7,** e1002278 (2011).
27. NCI Cancer Genomics Cloud Pilots—CBIIT: Welcome to the NCI Center for Biomedical Informatics and Information Technology. Available at: https://cbiit.nci.nih.gov/ncip/nci-cancer-genomics-cloud-pilots. (Accessed: 9th June 2016).
28. Google Cloud Computing, Hosting Services & APIs. *Google Developers*Available at: https://cloud.google.com. (Accessed: 9th June 2016).
29. Fusaro, V. a., Patil, P., Gafni, E., Wall, D. P. & Tonellato, P. J. Biomedical Cloud Computing With Amazon Web Services. *PLoS Comput. Biol.* **7,** e1002147 (2011).
30. Lowy, D. R. & Collins, F. S. Aiming High—Changing the Trajectory for Cancer. *N. Engl. J. Med.* **374,** 1901–1904 (2016).
31. Schatz, M. C. Biological data sciences in genome research. *Genome Res.* **25,** 1417–1422 (2015).
32. McCarthy, D. J. *et al.* Choice of transcripts and software has a large effect on variant annotation. *Genome Med*. **6,** 26 (2014).
33. Piccolo, S. & Frampton, M. Tools and techniques for computational reproducibility. *GigaScience* **5,** 30 (2016).
34. Cancer Genomics Cloud - Institute for Systems Biology. *Institute for Systems Biology*Available at: https://www.systemsbiology.org/research/cancer-genomics-cloud. (Accessed: 9th June 2016).
35. Bray, N. L., Pimentel, H., Melsted, P. & Pachter, L. Near-optimal probabilistic RNA-seq quantification. *Nat. Biotechnol.* **34,** 525–527 (2016).
36. Boettiger, C. An introduction to Docker for reproducible research. *Oper. Syst. Rev*. **49,** 71–79 (2015).
37. Kubernetes - Production-Grade Container Orchestration. Available at: http://kubernetes.io. (Accessed: 9th June 2016).
38. Cock, P. J. A., Fields, C. J., Goto, N., Heuer, M. L. & Rice, P. M. The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants. *Nucleic Acids Res*. **38,** 1767–1771 (2010).
39. Li, H. *et al.* The Sequence Alignment/Map format and SAMtools. *Bioinformatics* **25,** 2078–2079 (2009).
40. Li, H., Ruan, J. & Durbin, R. Mapping short DNA sequencing reads and calling variants using mapping quality scores. *Genome Res.* **18,** 1851–1858 (2008).
41. MacManes, M. D. On the optimal trimming of high-throughput mRNA sequence data. *Front. Genet*. **5** (2014).
42. Williams, C. R., Baccarella, A., Parrish, J. Z. & Kim, C. C. Trimming of sequence reads alters RNA-Seq gene expression estimates. *BMC Bioinformatics* **17,** 103 (2016).
43. Del Fabbro, C., Scalabrin, S., Morgante, M. & Giorgi, F. M. An extensive evaluation of read trimming effects on Illumina NGS data analysis. *PLoS One* **8,** e85024 (2013).
44. Bourne, P. E., Lorsch, J. R. & Green, E. D. Perspective: Sustaining the big-data ecosystem. *Nature* **527,** S16–7 (2015).
45. Stein, L. D. The case for cloud computing in genome informatics. *Genome Biol.* **11,** 207 (2010).
46. Fox, A. Cloud computing—what's in it for me as a scientist? *Science* **331,** 406–407 (2011).
47. Afgan, E. *et al.* Galaxy CloudMan: delivering cloud compute clusters. *BMC Bioinformatics* **11** Suppl 12, S4 (2010).
48. Schadt, E. E., Linderman, M. D., Sorenson, J., Lee, L. & Nolan, G. P. Computational solutions to large-scale data management and analysis. *Nat. Rev. Genet.* **11,** 647–657 (2010).

49. Singh, B. & Srinivasan, V. Containers: Challenges with the memory resource controller and its performance. In *Ottawa Linux Symposium (OLS)* 209 (Citeseer, 2007).
50. Marx, V. Biology: The big challenges of big data. *Nature* **498,** 255–260 (2013).
51. Wilks, C. *et al.* The Cancer Genomics Hub (CGHub): overcoming cancer through the power of torrential data. *Database* **2014** (2014).
52. Dudley, J. T. & Butte, A. J. In silico research in the era of cloud computing. *Nat. Biotechnol.* **28,** 1181–1185 (2010).
53. Commons Home Page | Data Science at NIH. Available at: https://datascience.nih.gov/commons. (Accessed: 11th July 2016).
54. Evans, C. C. & Others. The official YAML web site. *Retrieved April* **18,** 2010 (2004).
55. Pipelines API Command-line. *Google Developers*Available at: https://cloud.google.com/genomics/v1alpha2/pipelines-api-command-line. (Accessed: 11th July 2016).
56. isb-cgc. isb-cgc/ISB-CGC-pipelines. *GitHub*Available at: https://github.com/isb-cgc/ISB-CGC-pipelines. (Accessed: 8th July 2016).
57. Tarasov, A., Vilella, A. J., Cuppen, E., Nijman, I. J. & Prins, P. Sambamba: fast processing of NGS alignment formats. *Bioinformatics* **31,** 2032–2034 (2015).
58. McKenna, A. *et al.* The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Res.* **20,** 1297–1303 (2010).
59. Harrow, J. *et al.* GENCODE: the reference human genome annotation for The ENCODE Project. *Genome Res.* **22,** 1760–1774 (2012).
60. Babraham Bioinformatics - Trim Galore! Available at: http://www.bioinformatics.babraham.ac.uk/projects/trim_galore/. (Accessed: 8th July 2016).
61. Martin, M. Cutadapt removes adapter sequences from high-throughput sequencing reads. *EMBnet. journal* **17,** 10–12 (2011).
62. Tange, O. GNU Parallel - The Command-Line Power Tool. *login: The USENIX Magazine* **36,** 42–47 (2011).
63. Adler, M. pigz: A parallel implementation of gzip for modern multi-processor, multi-core machines. *Jet Propulsion Laboratory* (2015).
64. Godard, S. Sysstat. URL http://sebastien. godard. pagesperso-orange. fr/. (last visit: Jul. 2012) (2014).

Acknowledgements

S.R.P. thanks Brigham Young University for startup research funds. P.J.T. thanks the Simmons Center for Cancer Research at Brigham Young University for a summer fellowship that enabled him to perform this study. We thank the National Cancer Institute for providing cloud credits via the ISB Cancer Genomics Cloud pilot. We thank Sheila Reynolds and Abigail Hahn from the ISB for providing technical support and feedback on our manuscript. We thank researchers at the Broad Institute and The Cancer Genome Atlas Consortium who released data to the public.

Author Contributions

P.J.T. developed the software pipeline, processed all data, designed analyses, created all figures, and wrote the paper. S.R.P. developed the project idea, designed analyses, and wrote the paper. Both authors approved the final manuscript.

Additional Information

Supplementary information accompanies this paper at http://www.nature.com/srep

Competing financial interests: The authors declare no competing financial interests.

How to cite this article: Tatlow, P. J. and Piccolo, S. R. A cloud-based workflow to quantify transcript-expression levels in public cancer compendia. *Sci. Rep.* **6**, 39259; doi: 10.1038/srep39259 (2016).

Publisher's note: Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.