Coinami: A Cryptocurrency with DNA Sequence Alignment as Proof-of-work

Coinami: A Cryptocurrency with DNA Sequence Alignment as Proof-of-work

Atalay M. Ileri[1,*,†] Halil I. Ozercan[1,*] Alper Gundogdu[1] Ahmet K. Senol[1]
M. Yusuf Ozkaya[1,♯]
Can Alkan[1,‡]

[1] Department of Computer Engineering, Bilkent University, Ankara, Turkey
[†] Current address: Department of EECS, Massachusetts Institute of Technology, Cambridge, MA, United States
[♯] Current address: College of Computing, Georgia Tech, Atlanta, GA, United States
[*] These authors contributed equally. [‡] Corresponding author: calkan@cs.bilkent.edu.tr.

Abstract

Rate of growth of the amount of data generated using the high throughput sequencing (HTS) platforms now exceeds the growth stipulated by Moore's Law. The HTS data is expected to surpass those of other "big data" domains such as astronomy, before the year 2025. In addition to sequencing genomes for research purposes, genome and exome sequencing in clinical settings will be a routine part of health care. The analysis of such large amounts of data, however, is not without computational challenges. This burden is even more increased due to the periodic updates to reference genomes, which typically require re-analysis of existing data.

Here we propose Coin-Application Mediator Interface (Coinami[1]) to distribute the workload for mapping reads to reference genomes using a volunteer grid computer approach similar to Berkeley Open Infrastructure for Network Computing (BOINC). However, since HTS read mapping requires substantial computational resources and fast analysis turnout is desired, Coinami uses the HTS read mapping as proof-of-work to generate valid blocks to main its own cryptocurrency system, which may help motivate volunteers to dedicate more resources. The Coinami protocol includes mechanisms to ensure that jobs performed by volunteers are correct, and provides genomic data privacy. The prototype implementation of Coinami is available at `http://coinami.github.io/`.

1 Introduction

High throughput sequencing (HTS) technologies evolved very quickly since 2007 [21], and now they are among the most powerful tools available for biological research. We are now able to read the entire genome of a human individual in a few days for a fraction of costs incurred by previous technologies [19, 21]. However, the volume of data generated by these platforms are enormous, leading to a picture where computational analyses represent the major bottleneck [7, 29, 25]. For example, the Illumina HiSeqX platform can sequence the genomes of approximately 18,000 humans a year, at an estimated cost of $1,500 per genome (`http://www.illumina.com/systems/hiseq-x-sequencing-system.ilmn`). This corresponds to about 2 petabytes of data per year, per sequencing center. Considering the fact that there are many genome centers that either already have, or will purchase this system, the amount of data generated each year will increase to hundreds of petabytes to exabytes.

The computational analyses of such data involves multiple steps, but the main bottleneck is to find the potential locations of short stretches of DNA sequences in a reference genome. This step, called *read mapping*, usually takes 30 CPU days per human genome [17, 16, 1, 14, 11, 31] (see [8] for a review of aligners). The computational burden of read mapping is monotonically increasing not only because of the growth of data to be analyzed, but also because of updates in the reference genome assemblies. For instance, the human reference genome is updated every 3 to 4 years that fixes assembly

[1] All students involved in this project were undergraduates when they contributed. No graduate students were harmed during the making of this project. Yet.

mistakes, and adds either new sequences that are found in the genomes of newly sequenced individuals, or "alternative haplotypes" that are frequent variations from the existing reference sequence. Such drastic changes in reference genomes usually necessitate remapping of the existing data to enable more accurate characterization of genomic variants. Thanks to the *embarrassingly parallel* nature of this problem clusters are typically used. However, building clusters that are large enough to handle hundreds of petabytes of data is not feasible. Therefore, volunteer grid computing technologies, on the other hand, offer a promising alternative to large clusters and data centers. Although there are certain distinctive characteristics of mapping reads generated by different platforms, given the popularity of Illumina as today's most popular sequencing platform, the remainder of the paper will assume that the genomes are sequenced using the Illumina technology.

Volunteer grid computing was made popular by the Berkeley Open Infrastructure for Network Computing (BOINC) platform, and specifically its Search for Extraterrestrial Intelligence at Home (SETI@home) project (`http://setiathome.ssl.berkeley.edu/`). BOINC volunteers choose a scientific problem to work on, download data from the server, solve the problem, and upload results back to the server. To reduce the burden on the volunteers, the BOINC clients can be set to run only when the computer is idle (i.e. "screen saver mode"). There are a number of bioinformatics applications ported to the BOINC platform such as Rosetta@home for protein structure prediction (`https://boinc.bakerlab.org/`), and FiND@home (`http://findah.ucd.ie/`) for docking simulations on malaria proteins. Although it is also possible to write HTS read mapping applications for BOINC, such applications must provide data security to prevent both "leak before publication" and genomic privacy, and protection against malicious users (i.e. volunteers that deliberately upload incorrect results). Furthermore, HTS read mapping "assignments" (i.e. the sequence data) continuously grow as the volume of data required to be processed grows. It is obvious that timely analysis requires substantial computational resources (CPU, RAM, disk space, and network bandwidth), which may make it difficult to motivate volunteers. One path to increasing volunteer motivation to dedicate resources runs, arguably, through incorporating cryptocurrencies within the distributed grid computing for HTS read mapping.

The cryptocurrencies were first introduced by Wei Dai in 1998[6], and received considerable attention from the public after the first decentralized cryptocurrency (Bitcoin) was released in 2009 by an unknown person with pseudonym Satoshi Nakamoto[23]. The motivation behind Bitcoin was to create a distributed currency, which is not dependent on any controlling authority. On the footsteps of Bitcoin's success, a plethora of cryptocurrencies were proposed including Litecoin [15], Peercoin [22] and Dogecoin [20]. More recently, another proof-of-work scheme, named Cuckoo, is introduced that finds small cycles in large random graphs [30]. These cryptocurrencies are also maintained by the "miners", but the power of their mining network is negligible compared to that of Bitcoin's.

Although there are some differences between cryptocurrencies, all of them are composed of two interconnected processes called Mining and Transaction. In a nutshell, mining refers to the creation of new "digital money" termed *coin*, that are recorded by a public ledger called *blockchain.* Mining involves a computationally difficult process called "proof-of-work", while transaction refers to an exchange of blockchains between users. In Section 2 we provide some insight into cryptocurrency protocols in detail in the context of Bitcoin - the most widely used cryptocurrency.

As of January 2016, Bitcoin mining network's total computation power has reached approximately 710 PetaHash/s[2]. In comparison, world's most powerful scientific computation grid -BOINC- boasts a computation power of 11.22 PetaFLOP/s[3]. Although two units are not directly comparable as hashing uses integer operations while BOINC's power is measured in floating point operations, even the arbitrary yet not too radical scaling of 1 to 10 (assuming floating point operations are 10X "harder" than integer operations) posits Bitcoin network almost as powerful as the BOINC network.

The computation power of the Bitcoin network is mainly used to maintain the currencys integrity by ensuring that the blockchain creation is always a difficult task. The difficulty stems from the proof-of-work, which entails finding a number called the *nonce.* Nonce computation becomes harder and harder as the *difficulty target* is increased after the creation of every 2016 blocks. This effectively limits the amount of blockchains generated by the miners, preventing devaluation of the money as more blockchains are mined. The proof-of-work schemes within Bitcoin and other cryptocurrencies are effective solutions for the security of the cryptocurrency systems, yet serve no other practical purpose.

[2] `http://blockchain.info/stats`

[3] `http://boinc.berkeley.edu/`

Here we propose Coin-Application Mediator Interface (Coinami) where such computation power will be used for scientific computation as well as integrity purposes. We chose to use DNA sequence alignment problem as the proof-of-work in our initial implementation. Briefly, Coinami is a three-level multi-centric system that distributes HTS alignment problems to volunteers (or, *miners*). The miners download problem sets from the middle level autorities, that are in turn certified by the root authority, map the HTS reads to a reference genome, and send the results back to the middle level authority for verification (Section 3). However, the proof-of-work is decoupled from the rest of the system, making Coinami easily adaptable to other scientific problems that require substantial computational resources.

2 Current cryptocurrency systems

In this section we provide a brief introduction to cryptocurrencies based on the Bitcoin protocol.

2.1 Transactions

Transactions represent money exchange between two parties. Simply, they are records of participants, input amount and output amount. Unlike physical money (i.e. traditional currencies such as Dollar, Euro, etc.), cryptocurrencies can easily be copied before spending them. Therefore, all transactions should be recorded to prevent double spending. Every time a transaction occurs, the original (input) coins are destroyed and new (output) coins are generated. This scheme ensures that every coin can be spent only once. Each transaction contains receiver's public key and is signed by sender's private key. This adds the information of the spender and the receiver to the transaction itself. Once a transaction created, it is broadcast to the network for inclusion in a block. Inclusion in a block makes a transaction valid.

There is a special transaction called *coinbase transaction*, which does not have any input coins. It is the first transaction of every block and it contains the block generation reward for the miner.

2.2 Mining

Generation of a new block is called *mining*. The main purpose of mining is validating and recording the transactions. A fresh-minted block contains the hash of the previous block in chain, a set of transactions and a *nonce*, which is a 32 bit integer used for altering the blocks hash value (Figure 1A). A block is considered to be *valid* if its hash contains a predetermined number of zeros at the beginning, which denotes the *difficulty* of the generated block. Upon the generation and publication of the block, miners are awarded with some new coins. These new coins are included in the block as the first transaction and this transaction is called *generation transaction* or *coinbase transaction*.

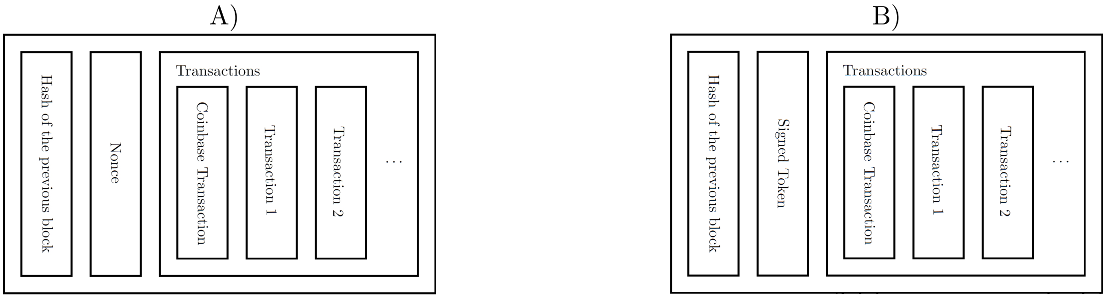


Figure 1: A) Contents of a Bitcoin block. B) Coinami block. We modified the Bitcoin blockchain by replacing the *nonce* with a *signed token*, which is generated by the authority. The remainder of the block is the same with that of Bitcoin, as it includes the hash of the previous block and the transaction history.

Computational power of a mining network can change dynamically. The difficulty index is introduced to prevent fluctuations in coin throughput due to these changes, and is updated after the generation of every 2016^{th} block. The Bitcoin mining protocol assumes that it takes approximately one week to generate 2016 blocks, and adjusts the difficulty level based on the actual time.

In practice, mining is performed by taking a potential blocks hash using the SHA-256 hash function to find a value with predetermined difficulty. Alteration of the hash value is achieved by changing the value of nonce. This is done by trying different nonce values in a brute-force fashion; the result is called *proof-of-work*.

3 Proposed system

Since virtually all computational power required in the process is used for the calculation of proof-of-work, we integrate our solution in that portion of the protocol. The two important properties of the proof-of-work scheme are 1) maintaining the difficulty of generation, and 2) validity of blocks. Therefore, the new proof-of-work scheme should keep these properties in place. Most current cryptocurrencies, including Bitcoin, are completely decentralized, as the assignments for the proof-of-work can be generated independently by the miners as long as they meet the system's difficulty level. However, for the proof-of-work in Coinami to be *useful*, the assignments for the proof-of-work need to have real and practical value. Therefore unlike major cryptocurrencies, Coinami is not completely decentralized due to the availability and generation of HTS data. Instead, Coinami has a three-level multi-center structure, where one *root authority* tracks and validates middle level *authority servers* that supply HTS data to the system and checks for validity of alignments, and the third level is composed of *miners* (Figure 2). The root authority should be trusted by the entire system to validate only "trustable" authority servers, which can be major sequencing centers. In the remainder of the paper, we refer to middle level authority servers as *authorities*.

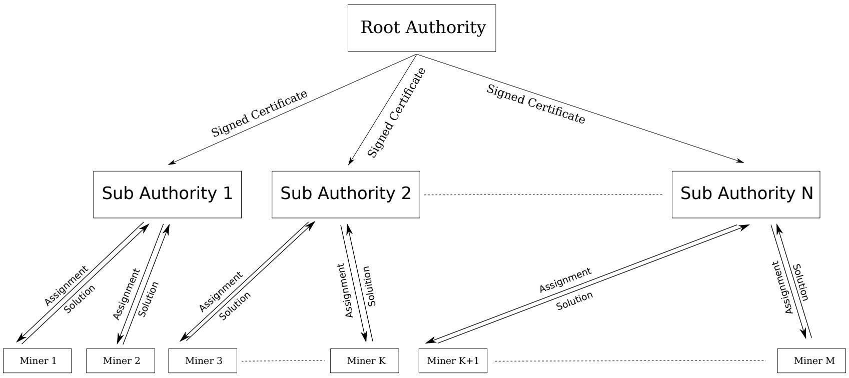


Figure 2: Three level structure for the Coinami network. The single root authority issues certificates to the sub authorities. The sub authorities (middle level) sends assignments to multiple miners, and validate the results they receive. If the alignments are valid, they sign the blockchains and return to the miners. A miner can work on assignments from multiple sub authorities.

Figure 3 summarizes the Coinami workflow. We adapted our scheme from Adam Back's Hashcash [2] protocol. Below, we provide the Coinami protocol in the context of the middle level authorities and the miners.

3.1 Authorities

The root authority assigns certificates to the middle level authorities, *validating* the authorities and allowing them operate within the Coinami network. The certificates and their corresponding private keys will be used in *token* signing.

Authorities fulfill two main roles in our system: 1) they *inject* new assignments (i.e. "alignment problems") into the system, and 2) check for the validity of the results to prevent counterfeit. If the results are valid, the authority signs the block by adding a signature token and returns it to the miner. For the validity test, the authority injects *decoy reads* into the assignment. The decoy reads will constitute

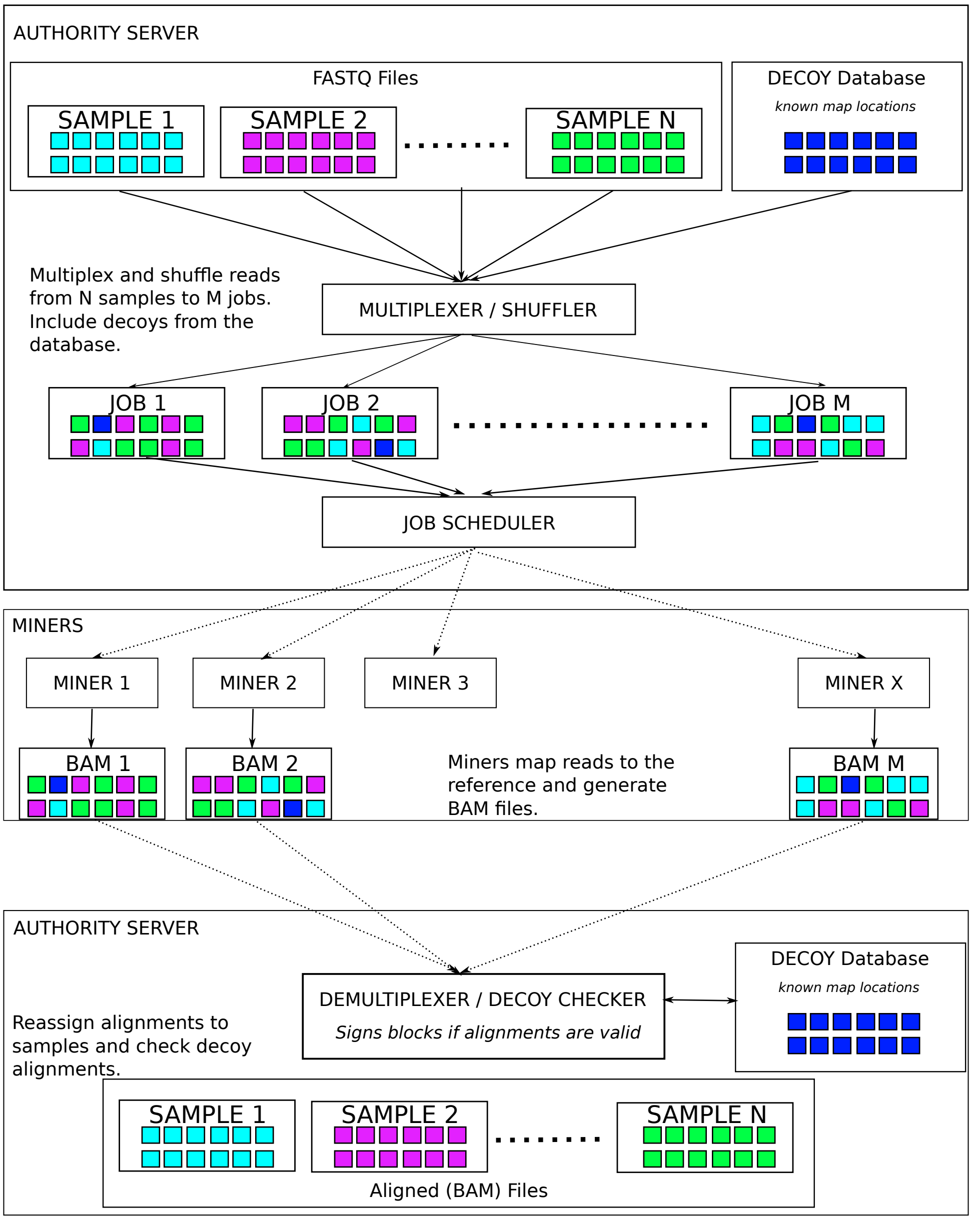


Figure 3: Coinami workflow. The authority hosts genomes of N samples in the form of FASTQ files, and a database of decoy reads with known mapping locations. The decoy database needs to be generated once for each reference genome and read mapper combination. To create the assignments, the authority multiplexes and shuffles reads from multiple samples and includes decoys that constitute 5% of the assignment. The miners requests jobs from the authority to to work on, which are tracked by a job scheduler that also imposes "deadlines" on the assignments to prevent deadlocks. The miners download the assignments, map the reads to the reference genome specified by the exchange protocol between the authority and the miners, and send the BAM files back to the authority. Finally, the authority reassigns the alignments to N samples and simultaneously checks decoys to verify the results. If the decoys are aligned to their predetermined location, the BAM file is considered to be valid, and the authority signs the block and returns it back to the corresponding miner.

5% of each assignment, and will allow quick comparison of the accuracy of alignment results, as theirs are pre-calculated. Note that in a set of 1,000 reads, there will be 50 decoys, and the probability of guessing the decoys correctly is $\frac{1}{\binom{1,000}{50}} \leq 1.06 \times 10^{-85}$. To prevent errors in downstream analyses of the alignment results for genomic variation discovery, the authority discards alignments for the decoy reads after verification of the results. Additionally, the decoy reads need to be indistinguishable from the rest of the assignment. In the case a read aligner that selects a random location for multi-mapping reads [17, 16] is used, either the decoys need to be selected among uniquely mappable reads (which may pose an attack threat), or all possible locations should be considered as correct (i.e. the `XA` field as reported by BWA-MEM). Alternatively, a deterministic mapper such as Bowtie2 [14] or mrFAST [1, 31] can be used. As an extra step to enhance data privacy, the authority mixes reads from multiple genomes together with the decoys into a single assignment (i.e. FASTQ file). Multiplexing data from the genomes of different individuals makes reconstruction of an individual's sequence impossible even if miners cooperate. Note that if the assignment contains $2n$ reads from two data sets, then there are 2^n different ways of grouping them into two size n subsets. The size of the assignments (i.e. number of reads) determines the difficulty of mining. The authority then encrypts the read names using AES encryption [5] to prevent the miners from deducing the decoys and individual genomes (Figure 4). Note that the authority is the only entity that can decrypt read names using its own key. Then, a job scheduler adds "deadlines" to the assignments to prevent deadlocks, and makes them available for downloading by the miners (Figure 3). The assignment also includes metadata for the reference genome and the read mapper to be used along with its mapping parameters. We note that the pre-calculated alignments for the decoys will need to be updated for different aligners and parameters, however, in practice only a few read mappers and their default parameters are widely used. After the miner sends the results (i.e. BAM file [18]), the authority separates the alignments from multiple genomes and verifies (and discards) decoys simultaneously by simply scanning the result file. Once the authority is satisfied, it signs and sends the block back to the miner.

3.2 Miners

A miner requests jobs from an authority and the authority sends the miner the next available assignment. The miner then maps the reads within the assignment to the reference genome using the designated mapper and its parameters, and generates a BAM file. The BAM file is sorted and indexed for easier processing at the authority side. However, the miner does not remove duplicates as done in most analyses, simply because the assignment contains reads from multiple genomes along with decoys. Therefore duplicate removal process will likely discard reads that are common across different genomes and decoys by chance, thus invalidating the alignment.

The miner sends the sorted and indexed BAM file back to the authority for validation. As described above, if the decoys are aligned to their correct positions, the authority signs and sends the block (Figure 1B) back to the miner. Upon receiving the validated block, the miner spreads it within the network. Other miners validate the new block by checking the signature of the authority, and other basic cryptocurrency block properties such as the hash of the previous block hash and the transaction history..

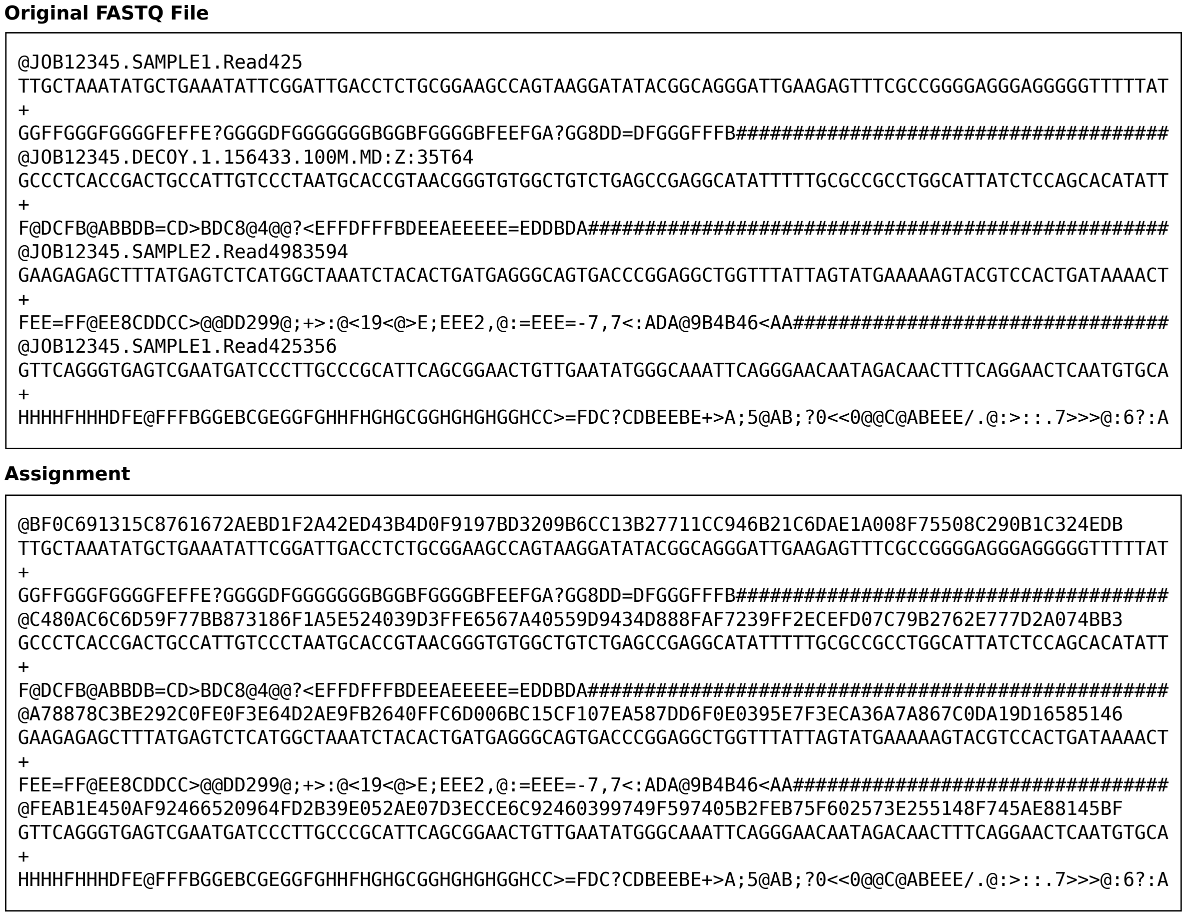
3.3 Implementation

We implemented a prototype for Coinami authority servers in C++, and the interface for miners including a graphical user interface for wallet in Python. The current version runs on Linux operating system. It uses BWA-MEM [16] for read mapping, SAMtools [18] for SAM to BAM conversion and BAM sorting, and the `libstatgen`[4] library for BAM processing (i.e. decoy identification and removal, and sample recovery). We used the Crypto++ library[5] to implement hashing, signing, and encrypting/decrypting read names. We modified the BasicCoin[6] project to implement the Coinami blockchain. Coinami components for authorities and miners are available at http://coinami.github.io/.

[4] `http://genome.sph.umich.edu/wiki/C%2B%2B_Library:_libStatGen`

[5] `https://www.cryptopp.com/`

[6] `https://github.com/zack-bitcoin/basiccoin`



Original FASTQ File

```
@JOB12345.SAMPLE1.Read425
TTGCTAAATATGCTGAAATATTCGGATTGACCTCTGCGGAAGCCAGTAAGGATATACGGCAGGGATTGAAGAGTTTCGCCGGGGAGGGAGGGGGTTTTTAT
+
GGFFGGGFGGGGFEFFE?GGGGDFGGGGGGGBGGBFGGGGBFEEFGA?GG8DD=DFGGGFFFB#####################################
@JOB12345.DECOY.1.156433.100M.MD:Z:35T64
GCCCTCACCGACTGCCATTGTCCCTAATGCACCGTAACGGGTGTGGCTGTCTGAGCCGAGGCATATTTTTGCGCCGCCTGGCATTATCTCCAGCACATATT
+
F@DCFB@ABBDB=CD>BDC8@4@@?<EFFDFFFBDEEAEEEEE=EDDBDA##################################################
@JOB12345.SAMPLE2.Read4983594
GAAGAGAGCTTTATGAGTCTCATGGCTAAATCTACACTGATGAGGGCAGTGACCCGGAGGCTGGTTTATTAGTATGAAAAAGTACGTCCACTGATAAAACT
+
FEE=FF@EE8CDDCC>@@DD299@;+>:@<19<@>E;EEE2,@:=EEE=-7,7<:ADA@9B4B46<AA################################
@JOB12345.SAMPLE1.Read425356
GTTCAGGGTGAGTCGAATGATCCCTTGCCCGCATTCAGCGGAACTGTTGAATATGGGCAAATTCAGGGAACAATAGACAACTTTCAGGAACTCAATGTGCA
+
HHHHFHHHDFE@FFFBGGEBCGEGGFGHHFHGHGCGGHGHGHGGHCC>=FDC?CDBEEBE+>A;5@AB;?0<<0@@C@ABEEE/.@:>::.7>>>@:6?:A
```

Assignment

```
@BF0C691315C8761672AEBD1F2A42ED43B4D0F9197BD3209B6CC13B27711CC946B21C6DAE1A008F75508C290B1C324EDB
TTGCTAAATATGCTGAAATATTCGGATTGACCTCTGCGGAAGCCAGTAAGGATATACGGCAGGGATTGAAGAGTTTCGCCGGGGAGGGAGGGGGTTTTTAT
+
GGFFGGGFGGGGFEFFE?GGGGDFGGGGGGGBGGBFGGGGBFEEFGA?GG8DD=DFGGGFFFB#####################################
@C480AC6C6D59F77BB873186F1A5E524039D3FFE6567A40559D9434D888FAF7239FF2ECEFD07C79B2762E777D2A074BB3
GCCCTCACCGACTGCCATTGTCCCTAATGCACCGTAACGGGTGTGGCTGTCTGAGCCGAGGCATATTTTTGCGCCGCCTGGCATTATCTCCAGCACATATT
+
F@DCFB@ABBDB=CD>BDC8@4@@?<EFFDFFFBDEEAEEEEE=EDDBDA##################################################
@A78878C3BE292C0FE0F3E64D2AE9FB2640FFC6D006BC15CF107EA587DD6F0E0395E7F3ECA36A7A867C0DA19D16585146
GAAGAGAGCTTTATGAGTCTCATGGCTAAATCTACACTGATGAGGGCAGTGACCCGGAGGCTGGTTTATTAGTATGAAAAAGTACGTCCACTGATAAAACT
+
FEE=FF@EE8CDDCC>@@DD299@;+>:@<19<@>E;EEE2,@:=EEE=-7,7<:ADA@9B4B46<AA################################
@FEAB1E450AF92466520964FD2B39E052AE07D3ECCE6C92460399749F597405B2FEB75F602573E255148F745AE88145BF
GTTCAGGGTGAGTCGAATGATCCCTTGCCCGCATTCAGCGGAACTGTTGAATATGGGCAAATTCAGGGAACAATAGACAACTTTCAGGAACTCAATGTGCA
+
HHHHFHHHDFE@FFFBGGEBCGEGGFGHHFHGHGCGGHGHGHGGHCC>=FDC?CDBEEBE+>A;5@AB;?0<<0@@C@ABEEE/.@:>::.7>>>@:6?:A
```

Figure 4: Sample assignment. For simplicity we only present one of the paired-end reads. The authority first generates a FASTQ file by mixing and shuffling reads from multiple samples and inserting decoy reads. For simplifying demultiplexing of the results, reads are renamed and labeled with sample IDs. The decoy read names also include the mapping information that can be directly compared with the miner-reported BAM file to avoid database search. In this example, the decoy read is pre-aligned to coordinate 156,433 in chromosome 1, with CIGAR and MD fields [18] 100M and MD:Z:35T64, respectively. We also add a job ID as a prefix to read names (JOB12345 in this example), which acts as a salt for the encrypted read name. This way, even if the same decoy is used in different job/assignment, it receives a different encrypted read name, preventing a potential attack. However, this information needs to be hidden from the miners to prevent decoy and sample identification. Therefore, the authority encrypts the read names with its private key, and applies base64 encoding to represent encrypted read names in ASCII format. Read name encryption also generates the same read name for the second end of the paired-end reads.

3.4 Processing power estimates

In this section we provide an estimate of the processing power of the Coinami network. First, we ran two sample assignments with 100 bp reads using three different CPUs (Table 1), and compared the run times with the estimated processing power in billion floating point operations (gigaflop; GFLOP) per second (GFLOP/s). We used the CPU performance table provided by BOINC[7] per core, and multiplied the run time in seconds with the estimated GFLOP/s values to estimate the number of floating point operations. We then calculated the number of reads processed per gigaflop. We note that, most of the assignment

[7] https://setiathome.berkeley.edu/cpu_list.php

work is in fact integer operations, however, this analysis gives us a rough estimate for proof-of-work run times across different CPUs. Even if we consider the lowest performance, on the average, 628 reads are processed per GFLOP. Therefore, if Coinami network processing power reaches 10% of that of BOINC's (i.e. 1 PetaFLOP/s), ∼658 million reads can be processed per second (if we omit the data transfer time), which corresponds to one genome sequenced at 22X coverage. Note that this estimate does not include the data transfer and authority processing times.

Table 1: Benchmarking results for number of reads processed per gigaflop.

CPU	Reads	Time (s)	GFLOP/s	GFLOP	Read/GFLOP
Intel^R Core^{TM} i7-3770 3.40GHz	1,000,000	195	4.4	858	1,165.50
Intel^R Core^{TM} i7-3770 3.40GHz	10,000,000	1,945	4.4	8,558	1,168.50
Intel^R Xeon^R E5-2643 3.30GHz	1,000,000	304	3.99	1,212.96	824.43
Intel^R Xeon^R E5-2643 3.30GHz	10,000,000	2,992	3.99	11,938.08	837.66
Intel^R Xeon^R E7- 4830 2.13GHz	1,000,000	698	2.28	1,591.44	628.36
Intel^R Xeon^R E7- 4830 2.13GHz	10,000,000	5,645	2.28	12,870.6	776.96

Analysis of computation power vs. proof-of-work. We sampled two sets of 100 bp reads from the 1000 Genomes Project [28] to generate assignments. As the proof-of-work, we aligned the reads to the reference human genome using BWA-MEM [16], then converted SAM file to BAM and sorted the BAM file using SAMtools [18]. The run times reported in this table corresponds to the total time required to run both BWA-MEM and SAMtools using a single core.

4 Future Work

There are many yet unsolved problems with the Coinami project. Below, we provide a list of open problems and future directions.

4.1 System-wide difficulty level

Currently, the Coinami network does not dynamically adjust the difficulty level in a way similar to Bitcoin. However, a reliable and self-regulating cryptocurrency system must limit the newly minted blockchains to prevent inflation. It is fairly straightforward in the Bitcoin protocol, where basically the difficulty level is determined by an integer, and changes to this number make it harder (or easier) to calculate the nonce value. Since the Coinami network requires an external data source, the system-wide difficulty level adjustment is not straightforward.

The first solution that comes to mind could be simply increasing the size of the assignment therefore increasing the time required to complete the task. However, this would mean that the assignment may need to be composed of billions of reads from multiple full genomes, which will incur a heavy load on storage and transfer, and eventually cause starvation due to the time-out mechanism that is designed to prevent deadlocks. A more sophisticated solution may be adding counters to the token and adjust the difficulty level (D). The authority may simply increment the counter and sign/finalize the blockchain only when the miner finishes D assignments. This would also require the miner to send a pre-existing but not yet signed token to the authority, or if no such token is sent, the authority will create a new token with a counter initialized to 1. Alternatively, the authority may keep a database of miners and their token counters. Another possibility is that the authority sends out D assignments to the miners, but sign only one of the randomly selected solutions. This would ensure D assignments are completed, but only one block is validated from solutions. However, this scheme needs to be carefully tuned to provide fairness among miners.

4.2 Authority-specific difficulty

We are also planning to add authority-specific difficulty levels to Coinami. The system will increase the difficulty for assignments from those authorities that signs substantially more (amount to be determined) tokens than other authorities. This is to prevent an authority to take over the system, and also to provide fairness among authorities. In addition, due to the multi-centric nature of Coinami, a "strong" (i.e. frequent signer) authority may essentially become a monopoly (compared to other authorities) and affect the value and integrity of the cryptocurrency[8].

4.3 Data transfer

The storage and therefore data transfer requirements for HTS data are very high. For example, the raw data (i.e. FASTQ files) of a genome sequenced at 40X coverage consumes approximately 150 GB of space even when compressed with `gzip`. The file size for the result (i.e. BAM file) will be similar. The assignments in Coinami will not contain full genomes, however, the amount of data to be transferred between miners and authorities will still be large, which will impose difficulties for both the authority and the miner. As a future work, we plan to incorporate data-specific compression to ameliorate this problem. For example, the assignments will be compressed using a FASTQ compressor such as SCALCE [12], and the resulting BAM file will be compressed using DeeZ [13] or CRAMtools [9]. Although they will add to the computational burden for the miner, they will also reduce the file sizes by $\geq$50%, which will also help with the data transmission problem. We are also planning to add native SCALCE and DeeZ capabilities to the read mappers used in Coinami and to the authority processing. To further improve data transfer speed, we plan to use a UDP-based data transfer protocol, such as UDT [10]. With a view to privacy, all communications between the miner and the authority will be encrypted.

4.4 Generalized application interface

As the Coinami protocol separates the proof-of-work from the rest of the system (i.e. transactions), it is possible to modify it to solve other scientific problems that require substantial computational capabilities. However, its current implementation only supports HTS read mapping[9].

We plan to make Coinami easier to port for solving different problems and to provide a plug-in mechanism for other researchers to use the Coinami network for their computational needs. For this purpose, we plan to develop interfaces that use Docker containers as the proof-of-work component. This will also make it possible to switch from a single-application multi-authority system to multi-application multi-authority system. We call these new type of authorities "employers", the miners "employees", and the root authority "the ministry of labor". The employees will be able to select the scientific problem to contribute to the solution of, and the tokens will be signed by the employer that provided the job. Note that the current BOINC platform works similarly, except for the cryptocurrency system. As a side note, this approach also may lead to interesting game theory problems[10].

5 Discussion

The amount of HTS data generated world-wide is ever-increasing, and it is expected to surpass other major domains of "big data": YouTube, Twitter, and Astronomy [27]. Hundreds of thousands of genomes and exomes are sequenced every year, which creates a huge computational burden. Additionally, even when we focus only on human data, the reference genome receives an update every few years, causing additional computation cost as the pre-existing data need to be remapped to the new version of the reference. As sequencing in the clinical environments gain traction [3], the volume of such data will only skyrocket, not to mention shifting priorities and urgency imposed by clinical cases, leading to "analysis starvation" for data with less priority. There are also efforts in place to move from a linear reference genome to graph-based structures to build "pan-genomes" that represents whole populations [24], and "variant graphs" to represent all previously known genomic variants (`https://github.com/ekg/vg`), which will likely prompt re-analysis of most existing data.

[8] Recent history showed that economies should avoid creating entities that are too big to fail.

[9] In fact, specifically Illumina data.

[10] One wonders if there will be any strikes and layoffs.

Currently HTS data analyses are performed using large scale clusters. Building and maintaining clusters are not easy tasks, and for most users certainly not feasible. Against this background the research community is witnessing a shift [26] to using dedicated academic clouds for genomics, such as the Embassy Cloud [4] at the European Bioinformatics Institute (EBI), and multi-purpose commercial cloud platforms such as Amazon AWS. However, no clusters, cloud platforms, or data centers offer "infinite" capabilities, and with HTS data growing faster than the fruits of Moores Law, the need for alternative approaches to distribute some of the workload to as many computers as we can is obvious. Volunteer grid computing may help analyze low-priority data (i.e. remapping to new reference), however, it is also less likely for volunteers to dedicate necessary resources as HTS mapping is compute intensive, requires large storage space and network bandwidth to download raw data and upload results. It might be possible to merge cryptocurrencies with volunteer grid computing as an approach to help increase volunteer motivation.

Coinami provides a protocol and prototype implementation of such a combination of Bitcoin and BOINC. We show that it is possible to distribute read mapping work load to untrusted parties (i.e. volunteers or miners), while ensuring data privacy and preventing malicious users that may try to submit "garbage" results. However, the miners have to trust the root authority, which in turn must trust the middle level authorities.

It is unknown whether Coinami will be used in practice in the future. There are three questions left unanswered yet. First: who shall act as the root authority? Probably, this is the easiest to answer, as a universally trusted research entity - such as the National Center for Biotechnology Information (NCBI) or EBI - may assume this role. The root authority only approves or rejects middle level authorities, therefore running the root authority server does not induce computation overhead. The second question is more complicated: will any sequencing center join the system as a middle level authority? The authorities have to invest on computational capacity to generate assignments and to validate the results returned by miners. They also have to actually be willing to share the needed data. The data sharing problem can be easily solved if only publicly available pre-existing data is injected to the system for realignment purposes[11]. However, the computational burden of running an authority server must be assessed to see if it is more feasible than performing the analyses in-house. Finally: will Coinami have the desired effect for volunteer motivation? This is probably the hardest one to answer, which will also depend on the "popularity" and value of Coinami-generated cryptocurrency. One possible approach to solve the "economics" problem may be fixed-rate conversion of Coinami blockchains with traditional currency, paid by the authority that issued the corresponding token. Another possibility is for the authorities to offer free service such as exome sequencing in exchange of Coinami blockchains. Or, Coinami blockchains may just remain as "bragging rights" for *real* volunteers and science and technology enthusiasts. Although less likely, people may deem Coinami blockchains to be valuable in-and-of themselves, and the Coinami project may even spearhead future developments to accelerate read mapping using field programmable gate arrays (FPGA) and application specific integrated circuits (ASIC).

6 Acknowledgments

We thank O. Mutlu, V. Boeva, F. de la Vega, R. Durbin, R. Chikhi, C. Orhan and N. Lack for discussions and future work ideas about this project, and C.T. Ozturk for proofreading the manuscript. We also thank M. Inan, E.B. Gulcan, and E. Akay for helping with the prototype implementation. This work was supported in part by an European Molecular Biology Organization Installation Grant (IG-2521) to CA.

7 Author contributions

AMI developed the initial concept. AMI and CA defined protocols. HIO, AG, AKS, and MYO. modified parts of the protocols and implemented the prototype. AMI, HIO, and CA wrote the manuscript.

[11] There is, however, a risk that editors of a certain journal may call miners as #researchparasites.

References

[1] C. Alkan, J. M. Kidd, T. Marques-Bonet, G. Aksay, F. Antonacci, F. Hormozdiari, J. O. Kitzman, C. Baker, M. Malig, O. Mutlu, S. C. Sahinalp, R. A. Gibbs, and E. E. Eichler. Personalized copy number and segmental duplication maps using next-generation sequencing. *Nat Genet*, 41(10):1061–1067, Oct 2009.

[2] A. Back. Hashcash - a denial of service counter-measure. http://www.hashcash.org/papers/hashcash.pdf, August 2002.

[3] L. G. Biesecker, J. C. Mullikin, F. M. Facio, C. Turner, P. F. Cherukuri, R. W. Blakesley, G. G. Bouffard, P. S. Chines, P. Cruz, N. F. Hansen, J. K. Teer, B. Maskeri, A. C. Young, N. I. S. C. C. S. P. , T. A. Manolio, A. F. Wilson, T. Finkel, P. Hwang, A. Arai, A. T. Remaley, V. Sachdev, R. Shamburek, R. O. Cannon, and E. D. Green. The ClinSeq project: piloting large-scale genome sequencing for research in genomic medicine. *Genome Res*, 19(9):1665–1674, Sep 2009.

[4] C. E. Cook, M. T. Bergman, R. D. Finn, G. Cochrane, E. Birney, and R. Apweiler. The European Bioinformatics Institute in 2016: Data growth and integration. *Nucleic Acids Res*, 44(D1):D20–D26, Jan 2016.

[5] J. Daemen and V. Rijmen. AES proposal: Rijndael. 1998.

[6] W. Dai. b-money: a scheme for a group of untraceable digital pseudonyms to pay each other with money and to enforce contracts amongst themselves without outside help. http://www.weidai.com/bmoney.txt, 1998.

[7] P. Flicek. The need for speed. *Genome Biol*, 10(3):212, Mar 2009.

[8] N. A. Fonseca, J. Rung, A. Brazma, and J. C. Marioni. Tools for mapping high-throughput sequencing data. *Bioinformatics*, 28(24):3169–3177, Dec 2012.

[9] M. H.-Y. Fritz, R. Leinonen, G. Cochrane, and E. Birney. Efficient storage of high throughput DNA sequencing data using reference-based compression. *Genome Res*, 21(5):734–740, May 2011.

[10] Y. Gu and R. L. Grossman. UDT: UDP-based data transfer for high-speed wide area networks. *Computer Networks*, 51(7):1777–1799, 2007.

[11] F. Hach, F. Hormozdiari, C. Alkan, F. Hormozdiari, I. Birol, E. E. Eichler, and S. C. Sahinalp. mrsFAST: a cache-oblivious algorithm for short-read mapping. *Nat Methods*, 7(8):576–577, Aug 2010.

[12] F. Hach, I. Numanagic, C. Alkan, and S. C. Sahinalp. SCALCE: boosting sequence compression algorithms using locally consistent encoding. *Bioinformatics*, 28(23):3051–3057, Dec 2012.

[13] F. Hach, I. Numanagić, and S. C. Sahinalp. DeeZ: reference-based compression by local assembly. *Nat Methods*, 11(11):1082–1084, Nov 2014.

[14] B. Langmead, C. Trapnell, M. Pop, and S. Salzberg. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biol*, 10(3):R25, Mar 2009.

[15] C. Lee. Litecoin. https://litecoin.org/, October 2011.

[16] H. Li. Aligning sequence reads, clone sequences and assembly contigs with bwa-mem. *arXiv preprint arXiv:1303.3997*, 2013.

[17] H. Li and R. Durbin. Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics*, 25(14):1754–1760, Jul 2009.

[18] H. Li, B. Handsaker, A. Wysoker, T. Fennell, J. Ruan, N. Homer, G. Marth, G. Abecasis, R. Durbin, and . G. P. D. P. Subgroup. The sequence alignment/map format and SAMtools. *Bioinformatics*, 25(16):2078–2079, Aug 2009.

[19] E. R. Mardis. The impact of next-generation sequencing technology on genetics. *Trends Genet*, 24(3):133–141, Mar 2008.

[20] B. Markus. Dogecoin. http://dogecoin.com/, December 2013.

[21] M. L. Metzker. Sequencing technologies - the next generation. *Nat Rev Genet*, 11(1):31–46, Jan 2010.

[22] S. Nadal and S. King. PPCoin: Peer-to-peer crypto-currency with proof-of-stake. https://www.peercoin.net/assets/paper/peercoin-paper.pdf, August 2012.

[23] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.

[24] N. Nguyen, G. Hickey, D. R. Zerbino, B. Raney, D. Earl, J. Armstrong, W. J. Kent, D. Haussler, and B. Paten. Building a pan-genome reference for a population. *J Comput Biol*, 22(5):387–401, May 2015.

[25] A. Sboner, X. J. Mu, D. Greenbaum, R. K. Auerbach, and M. B. Gerstein. The real cost of sequencing: higher than you think! *Genome Biol*, 12(8):125, 2011.

[26] L. D. Stein, B. M. Knoppers, P. Campbell, G. Getz, and J. O. Korbel. Data analysis: Create a cloud commons. *Nature*, 523(7559):149–151, Jul 2015.

[27] Z. D. Stephens, S. Y. Lee, F. Faghri, R. H. Campbell, C. Zhai, M. J. Efron, R. Iyer, M. C. Schatz, S. Sinha, and G. E. Robinson. Big data: Astronomical or genomical? *PLoS Biol*, 13(7):e1002195, Jul 2015.

[28] The 1000 Genomes Project Consortium. A global reference for human genetic variation. *Nature*, 526(7571):68–74, Sep 2015.

[29] T. J. Treangen and S. L. Salzberg. Repetitive DNA and next-generation sequencing: computational challenges and solutions.d. *Nat Rev Genet*, 13(1):36–46, Jan 2012.

[30] J. Tromp. Cuckoo Cycle: a memory bound graph-theoretic proof-of-work. https://github.com/tromp/cuckoo/blob/master/doc/cuckoo.pdf?raw=true, July 2015.

[31] H. Xin, D. Lee, F. Hormozdiari, S. Yedkar, O. Mutlu, and C. Alkan. Accelerating read mapping with FastHASH. *BMC Genomics*, 14 Suppl 1:S13, 2013.