Eliminating uncertainty in the process of knowledge acquisition from the large-scale genomic data

大田　達郎

博士（理学）

平成３０（２０１８）年度

Eliminating uncertainty in the process of knowledge acquisition from the large-scale genomic data

Ohta, Tazro

December 2018

Table of Contents

ACKNOWLEDGMENTS

First of all, I would like to express my special gratitude to "the farmer" Dr. Hidemasa Bono: this thesis and my research career never exist if you did not rescue me from the laboratory of irreproducible research. It was a literally life-changing event that is being still my strong motivation to do my research to accomplish our mission to make science fair.

My dear colleagues in the three database centers in Japan - the Database Center for Life Science, the DNA Data Bank of Japan, and the National Bioscience Database Center: all the members are fantastic people who have been working for the future of the life science research. All that you share taught me how we make the world better.

With a special mention to Dr. Toshihisa Takagi, Dr. Yasukazu Nakamura, and Dr. Masanori Arita: For me without a mentor of research, I would like to express my special appreciation for being precious supervisors for my Ph.D. research and writing. I would also like to thank my dissertation committee members, Dr. Ken Kurokawa, Dr. Akatsuki Kimura, Dr. Shoko Kawamoto, and Dr. Koji Kadota.

I also want to thank the open source community: the Pitagora-Network, the Galaxy Project, the Open Bio Foundation, the Common Workflow Language, the NGS-field. I have learned many things through the experience of the open source development which made me a well-trained engineer. And to the BioHackathon family: The family helped me to grow as an international research scientist.

I thank all the friends on physical and online: I was always alone when I was writing this thesis, but I was not. It is super lovely to have so many friends who cheer me up when I am having a hard time.

Last but not the least, a special thanks to all my family members: my parents, my brother, my grandparents, my beloved daughter, and my dearest wife: my wonderful life without (truly) any concerns are because of you all. Thanks for all your encouragement!

December 2018

Eliminating uncertainty in the process of knowledge acquisition from the large-scale genomic data

Ohta, Tazro

Abstract

Genomic science has become a big data science since the advent of the high-throughput sequencing (HTSeq) technologies which produce a massive amount of nucleotide sequence data. Sequence Read Archive (SRA) is a public data repository for the HTSeq, where researchers submit the raw data from HTSeq experiments, now archives more than 4 million samples. To extract biological knowledge from these big data of genome sequences, researchers need to use computational software to perform various kinds of data analysis.

Performing genomic data analysis is often a complicated process because many factors affect the application of data analysis software. For example, researchers need to confirm the target molecules, the nature of the sequenced sample, the applied experimental instruments, and the used reagents to select appropriate software for data analysis. Researchers also have to understand the software operation often with many options and input parameters. Without sufficient background information and accurate operation of software, one cannot perform a proper data analysis, which results in producing the unreliable output. Thus, the precise description of the data analysis process is a key to evaluate the output of the research. However, it is not a manageable task for researchers to describe the precise process of knowledge extraction from the

data without a system to support. Therefore, in this research through the case of database development from the public sequencing data, I propose the methods to describe the data analysis process to remove uncertainty.

To describe the precise information of input data for the analysis, I developed the methods to integrate sample metadata with publication information and statistics of sequencing quality. The developed system integrated the sample metadata with the related publication, which also enabled researchers to find related data from the public database easily. The calculated quality statistics of each sequencing data can provide more comparable attributes of the data rather than free text. The additional information helped to supplement the lack of description in the metadata, which also helps researchers to interpret the output of data analysis.

The description of software used in the data analysis is also crucial to evaluate the output of the analysis. To describe the process of data analysis without any uncertain points, I developed a method to package the operations in an executable form with runtime information. The developed system called CWL-metrics works with Common Workflow Language (CWL), a community standard of workflow description, which one can describe the operations of tools and workflows. CWL enables researchers to write their workflows in a format that is executable by several workflow runner implementations. CWL-metrics provides additional information of runtime metrics to the CWL description, which makes workflows highly portable and reproducible.

The additional information to input data and the method to describe data analysis workflow in a reproducible form enable researchers to perform data analysis with a precise description of its processes. To demonstrate the data analysis with these methods, I developed a database and a web application using the public HTSeq data. The database, ChIP-Atlas, is to provide the results of data analysis of the public ChIP-Seq and DNase-Seq to show the comprehensive data of transcription factor binding sites and open chromatin regions. By using the proposed methods to make data analysis process transparent, the users of ChIP-Atlas can evaluate the result of the analysis with the precise description of sample metadata and the data analysis workflow.

CHAPTER 1

INTRODUCTION

Genomic science has made a significant achievement as an important research field in life science since its early history, which also has been providing useful techniques for the whole biological research domains. Researchers use nucleotide sequencing technologies not only as the method to survey the functions of genes and the relationship with phenotypes, but also for taxonomic classification, studies of evolution, or detection of microorganisms in an environment. Behind the wide range of applications of the method, there is a fundamental contribution of the nucleotide sequences accumulated by the past studies, and the databases which manage them.

In the late 1970s, following the rise of nucleotide sequencing method, there were emerging discussions for the needs of a database to share the nucleotide sequencing data [1]. In 1982, the EMBL-bank started in the European Molecular Biology Laboratory (EMBL) Heidelberg to collect and share the nucleotide sequencing data, followed by the GenBank database started in the Los Alamos National Laboratory [2]. In 1986, the DNA Data Bank of Japan (DDBJ) launched to accept submissions of nucleotide sequencing data [3]. The three databases started the International Nucleotide Sequence Database Collaboration (INSDC) where they exchange the submitted data and collaborate to maintain the regulations on data sharing [4].

At the same time of the foundation of the public nucleotide sequence databases, the researchers in National Institute of Health (NIH) developed the algorithm of similarity search against nucleotide and protein sequence database [5]. The genomic science made the first great leap by this method to compare sequences to find similar sequences which may have similar molecular functions. This method has changed the role of the nucleotide database from the box to store the evidence of the past studies to the platform to share the data as a material for further research.

In the middle of '00s, there were epic changes in the sequencing technology. The techniques which enable highly parallelized nucleotide sequencing are called the "next generation." The instruments which appeared first such as Roche 454, Illumina Genome Analyzer, or Applied Biosystems SOLiD enabled to sequence a massive amount of DNA sequences, though the length of one sequence was shorter and the accuracy of base call was lower in comparison with the previous Sanger sequencing method [6]. The methods to comprehensively sequence input DNA molecules became the new booster for biological research, and its advent has the INSDC start the new data repository called Short Read Archive, which the INSDC later changed its name to Sequence Read Archive (SRA) [7].

The "traditional" nucleotide sequence database accepts submissions of nucleotide information, then made them public in the form of a text file which includes nucleotide sequence represented by alphabetical characters. On the other hand, SRA accepts submissions of raw sequencing data in the form of a FASTQ file, which records nucleotide sequences and its base call accuracy encoded in ASCII codes [8]. The rapid

advance of the next generation sequencing, now called high-throughput sequencing (HTSeq) technologies results in a tremendous amount of data submissions to SRA, which made more than eight quadrillion bytes of sequences submitted in these ten years [9]. The fast growth of the number of submissions has increased the cost of data storage, which has The National Center for Biotechnology Information (NCBI) issued an announcement that they no longer accept submissions to SRA, which was later withdrawn [10]. Until today, the INSDC still manage to keep the data repository by using data compression formats such as SRA format or CRAM format [11].

The advance of the HTSeq technologies and the accumulated data in the database has changed the genomic science into a data-driven science. Unlike the conventional hypothesis-driven approach using the methods of molecular biology, the genomics as a data-driven, so-called big data science describes biological functions by analyzing data obtained from the high-throughput measurement methods which capture molecules in the sample material comprehensively. This approach based on capturing whole molecules is generically called Omics research. The genomic science from the late '00s was the beginning of the era of knowledge acquisition from the massive amount of nucleotide sequencing data.

The data produced by nucleotide sequencing instruments are a number of fragments of the nucleotide sequence (read) which is not understandable in the raw format. To acquire biological knowledge from the reads, one needs to perform data processing and data analysis on a computer. The data processing is a step to recover the genomic sequence from sequence read using existing biological knowledge. This

process includes trimming of the reads of low-quality base calling, read mapping to the reference sequence, or de novo assembly. The following data analysis step is to extract biological features from the genome or cDNA sequence obtained from the data processing. Those steps of data processing and analyzing are often called "data analysis" as one component of the scientific workflow. There are various factors that define the software used for the data analysis. For example, used sequencing instruments and reagents, sequenced samples, types of measured molecules, experimental treatment, or the biological features of interest may affect the selection of software.

In the data-driven science where the knowledge comes from the extensive amount of data, a factor that is responsible for the reliability of the knowledge is the quality of data analysis. The data analysis of good quality requires followings: selecting appropriate software and input parameters according to the character of the input data, executing the selected set of software properly, and providing provenance information of the output result.

However, researchers have no sufficient tools to ensure the quality of the data analysis in genomic science. Without the reliability to the data analysis, one cannot trust the results of studies. To make a research output more reliable, researchers need to describe more details of data analysis, which improves the transparency and removes the uncertainty. There are two major points concerning the description of the data analysis. First, researchers need to have methods to describe information of the input data to the analysis. The incorrect, insufficient, or unclear background information of input data leads to the usage of an inappropriate software or the misinterpretation of the

analysis result. Another is the description of tracking the steps of data analysis, which is required to perform replication. The information regarding to data analysis includes information of software, versions of software and libraries, input parameters, and the used computational environment, which are necessary to reproduce the data analysis.

Therefore, in this research, I propose methods to eliminate uncertainty from the data analysis of genomic research through the case study to construct the database from the public HTSeq database. First, I developed methods to integrate metadata of input data with external sources such as publications or statistics of sequencing quality (Chapter 2). The metadata recorded in the database often lack the information required for further data analysis. Thus, adding more description to the input data enabled to provide more information for the interpretation of analysis results. Following the extension of data description, I developed the method to describe the steps and the environment of data analysis on a computer (Chapter 3). Using a description framework for software tools and workflows, I described the process of data analysis in a reproducible manner. The description of data analysis also includes the information of runtime and environment. The use of description framework and environment information removes any uncertain points from the actions taken in the data analysis, which enable to replicate the data analysis by a different actor in a different computing platform. With these methods to make the process of data analysis transparent, I developed a database and a web application that users can access the process of data analysis, and the result data with referring the information of input data (Chapter 4). The database is already widely used for genomic researches, which shows that the

transparency of the knowledge acquisition process is the key to provide a useful tool for genomic research.

CHAPTER 2

INTEGRATION OF SAMPLE METADATA WITH PUBLICATION AND QUALITY STATISTICS TO FILL THE GAPS IN THE DATA DESCRIPTION

Published Article 1

Experimental Design-Based Functional Mining and Characterization of High-Throughput Sequencing Data in the Sequence Read Archive

Takeru Nakazato, Tazro Ohta, Hidemasa Bono.
PLoS One. 2013 Oct 22;8(10):e77910.

Published Article 2

Calculating the Quality of Public High-Throughput Sequencing Data to Obtain a Suitable Subset for Reanalysis From the Sequence Read Archive

Tazro Ohta, Takeru Nakazato, Hidemasa Bono.
GigaScience. 2017 Apr 25;6(6):gix029.

Background

The publication of primary data used as evidence is essential for ensuring transparency and reproducibility in scientific research, but also important for promoting the reuse of data in future research activities [12,13]. In the last decade, the rapid advance of HTSeq technologies has enabled omics research projects to produce massive amounts of data, which have huge potential for reuse from different perspectives [14]. An increasing number of sets of omics data are being produced not only by international consortiums but also from individual research projects [15]. However, only a portion of all archived data derived from large projects is frequently being reused, in contrast to data from individual studies. This is probably because users prefer to collect data from a single project that had a sufficient number of samples and that were produced by experiments under reliable conditions with precise sample metadata, thus ensuring the quality of the data.

With the simple keyword search model used in SRA, users often get too large or too small number of search results since there is a bias of the number of submission per data type. For example, a search performed on 26 September 2018 with the query "human liver RNA-seq" got 5298 matches on NCBI SRA data search on . On the other hand, the query "whale liver RNA-seq" only returned 16 matched records, while 4 of them are dolphins. Without knowing this submission number bias, users are not able to evaluate if the number of search results is reasonable. Furthermore, the search index is

built on the metadata described by the data submitters, which means that the data with insufficient metadata cannot be found on the data search.

To promote the reuse of combined sets of data from multiple projects, public repositories have to provide a filtering feature in data searches, so that users can control the number of experiments and quality of the data in their searches. Currently, data searches provided by repositories based on metadata described by the data submitter cannot be used for filtering by data quality. To enable such filtering, repositories have to provide information on the quality of sequence data.

Providing information on data quality can also provide insight into the data repository itself. Basic quality values, for example, mean and median levels of sequencing throughput, read length, or base call accuracy of the specific sequencing method, are important to obtain an overview of the archive. These values can be used to illustrate the overall distribution of data in the repository. The distribution can show the standard of data quality; thus, a user can use these values to filter out inappropriate datasets from among the thousands of search results.

Therefore, I developed a framework that integrates sample metadata with related information from different sources to enable precise provenance tracking of published data. Sample metadata often have only qualitative information such as biological sources or preservation condition. The framework added two different information sources. The first is the background information of the project used the sample, and the second is the quantitative information that can be calculated from the obtained data.

Those two types of additional information can provide more insights to the published data and the samples, and enable more practical data search.

Methods

Articles extraction related to each SRA entry

The publications that refer to SRA data are listed to improve the accessibility of archived data. First, a collection of the PubMed IDs (PMIDs) cited in the reference sections of the whole SRA metadata was performed, followed by the extraction of SRA IDs from journal articles in MEDLINE. The external database section of MEDLINE does not have the SRA IDs to refer the data used in the research, while identifiers of the other database such as GenBank or OMIM were recorded. Therefore, SRA IDs were extracted from the full-text versions of articles in PubMed Central (PMC) and the websites of the journals that were freely available for parsing using regular expression pattern matching. In particular, the articles which the MeSH term ‘‘High-throughput Nucleotide Sequencing’’ was assigned were used for the ID extraction. The SRA IDs extracted from journal articles are often not the same IDs used for submissions (i.e., start from SRA, ERA and DRA) or study (i.e. SRP, ERP and DRP), but are the IDs used for experiment (start from SRX, ERX and DRX) or run (SRR, ERR and DRR). Thus, the extracted IDs were converted to the corresponding SRA study IDs by the ID mapping table previously constructed.

Some transcriptome data captured by HTSeq are also submitted to the Gene Expression Omnibus (GEO). Since publications often cite GEO IDs as links to the data, GEO IDs and the related PMIDs are collected from the entire set of GEO data downloaded from the NCBI FTP site [16]. GEO has three types of identifiers, GDS for dataset, GSE for data series, and GSM for samples. All the GEO identifiers found in publications are converted to correspond SRA IDs using metadata submitted to GEO and SRA. Accordingly, I constructed a publication list referring to SRA data, showing publication title, journal name, PMID and referring SRA ID and data title.

Building search index and implementation of web application

Using the pairs of PMID and SRA ID, the metadata of publications are retrieved from NCBI Eutils service [17] and metadata of SRA from NCBI FTP site [18]. All the fields recorded in PubMed entries and PMC entries were gathered to be linked to SRA metadata. I built the search index with full text search engine framework groonga [19]. The source code of the web application is available on GitHub [20].

Data retrieval from the data repository for quality calculation

I downloaded data from the FTP server of the DDBJ [21] by using the *lftp* command. Most of the data were downloaded as a FASTQ format file. When data were only available in SRA format, I decompressed the data to FASTQ format by using the *fastq-dump* command of the SRA toolkit (ver. 2.5.1). The decompress command is performed with the *--split-3* option to split paired-end files into individual FASTQ files. Downloaded data were analyzed by md5 checksum to confirm that they were not corrupt.

Extraction of sequencing quality information

First, I performed FastQC [22] via the command line with options *--no-extraction* and *--threads 4*. The versions of FastQC software used in this study were 0.10.0, 0.10.1, and 0.11.3, depending on the date when each sequencing run was performed. I confirmed that there were no differences in the results of the modules that I used among the versions. I parsed the result files of FastQC (fastqc_data.txt) by the bioruby [23] module *bio-fastqc* [24], which I developed based on biogem [25]. The results from paired-end reads were concatenated by calculating the average values for each quality value, excluding values of the total number of sequences that were summed. If an experiment involved multiple sequencing runs, quality values were also concatenated to create comparable values for each experiment. By using relation of SRA Experiment ID and BioSample ID, calculated quality values, experimental metadata, and sample organism metadata were assembled. The code is available online [26].

Publishing quality data as linked open data

I published the individual results of FastQC for each sequencing run on the web server [27]. Each set of sequencing quality data was converted into RDF format and deposited in the NBDC RDF portal [28]. I developed an ontology to describe sequencing quality information, namely, sequence statistics ontology, and also published it in the NBDC RDF portal.

Visualization of the data distribution in the repository

Visualization of the distribution of data was performed using R language (ver. 3.2.3) [29] and library ggplot2 (ver. 2.1.0) [30]. The code is available online [26].

Results

Trends and growth of SRA entries

The HTSeq technologies are applied to the various research purposes including whole genome sequencing, transcriptome analysis, and metagenomics. To classify the projects available on SRA according to the sequencing applications, I extracted the study type fields from metadata and visualized by a line chart (Figure 2.1). As of September 2018, SRA has over 150,000 projects where the number was just 14,000 in March 2013. About one third of the total (52388) are projects for whole genome sequencing, became 8 times larger than the same category in 2013. The other major projects are of metagenomics (23360) and transcriptome analysis (21509), dramatically increased the numbers from 1240 and 1983 in March 2013, respectively. Further, I investigated experiments archived in SRA by using sequencing platform (Figure 2.2). Once the Illumina Genome Analyzer II was the dominant of ones used for data in SRA, but in 2018, it became just 2.4% of total experiments (115231/4805611). The current dominant is, surprisingly, still the Illumina HiSeq 2000 as it was in 2013. There were several newcomers in this sequencing instrument market, but none of them could reach to the top 10 instruments, at least by the number of experiments. Each instrument has its strength and suitable application, thus the number does not reflect the power of

instrument, but just popularity. This numbers are a heads-up for users that they need to care of this huge bias in numbers of submissions when they perform data search.

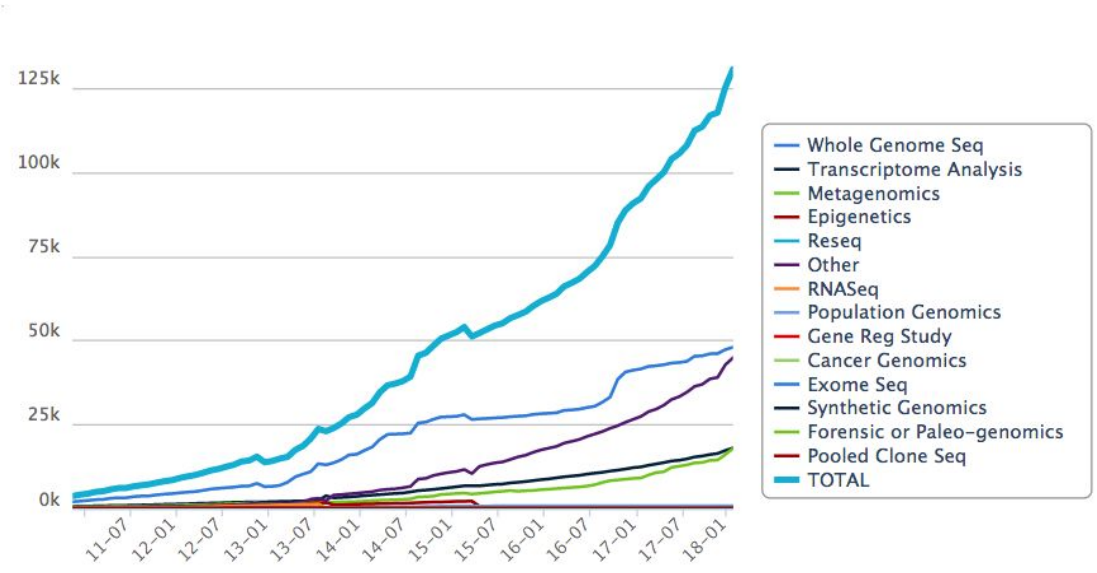


Figure 2.1: The growth of SRA data categorized by project types

The growth of the number of SRA studies categorized by project types. The number of experiments for Whole Genome Sequence has been the largest part of the SRA since the beginning of SRA.

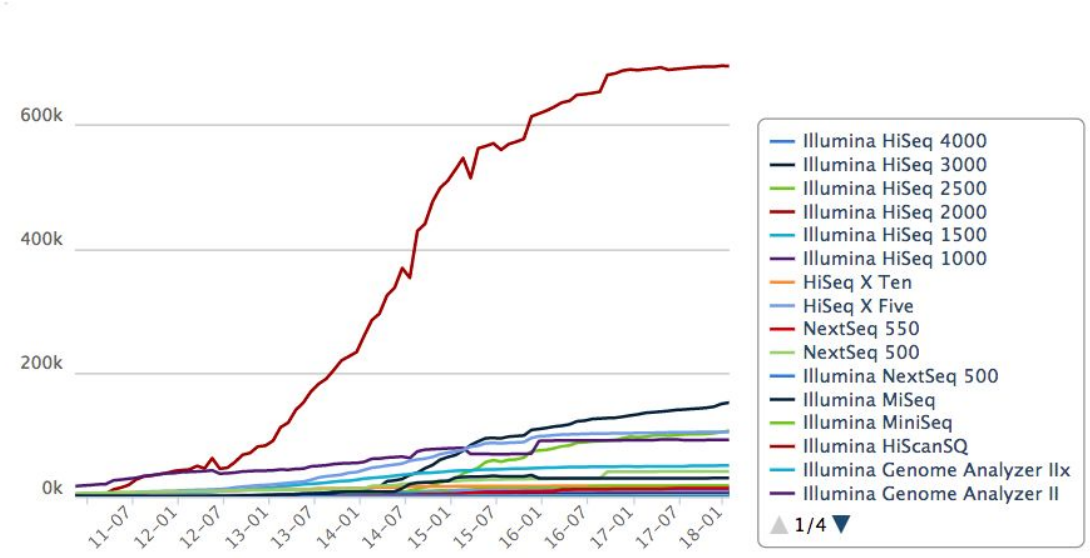


Figure 2.2: The growth of SRA data categorized by sequencing platforms

The growth of the number of SRA experiments categorized by sequencing platforms. HiSeq 2000 has been the dominant of the market since late 2012. Users must concern this huge bias of the instruments when they perform data search specifying a used sequencing instrument.

Building a list of publications that refer SRA entries

There are HTSeq data submitted to SRA before their papers get published. This means that one cannot access the details of all the sequencing projects that are found in SRA. To investigate the numbers of publications that are linked to SRA data, the publication list was generated from the identifiers found in SRA metadata and publication data in MEDLINE. The SRA IDs extracted from journal articles are often not the IDs used for submissions or study, thus the extracted IDs are converted to their corresponding study IDs using the ID mapping table provided by NCBI [31]. There are

24501 pairs of SRA ID and PubMed ID, which 30 times of the number in 2013. The number is relatively small in comparison with the number of projects archived in SRA. It may be because of the failure of ID retrieval from the publications that describe the identifiers in supplementary files such as PDF files available only on journal websites. However, for those entries having related publication information are able to be found by their metadata with publication information such as authors, affiliations, or versions of reagents or software described in materials and methods section in a publication. I used Groonga, a software for building full-text search engine, to build a search index by original SRA metadata with related publication information, including full-text data available on PMC.

Implementing web application interface for data search

To perform efficient data search, it is required to have an interface that can control the number of target entries to which users submit a keyword query. I implemented a web interface for data search, which has two steps for search. First, users are required to select facets to reduce the number of target entries. The facets are sample organism, sequencing application, and sequencing instrument (Figure 2.3). The interface shows the number of entries match to the facets and the combination of them, which remind users the number of data available in SRA for each metadata field. Users can select "show all" option to browse the list of selected entries or submit a keyword query to the target entries. This two-step search interface helps users to understand the number of data of their interest, which avoid repeating keyword search that only matches the small number of entries.

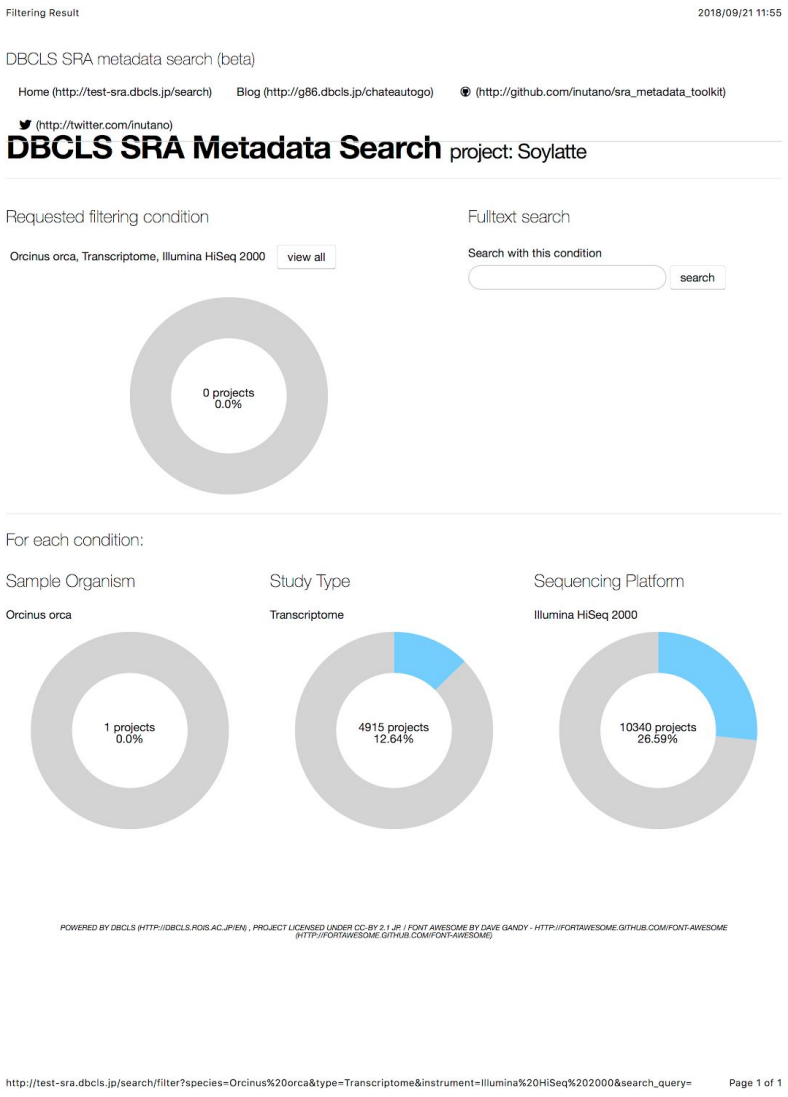


Figure 2.3: The visualization of filtering condition on data search

An example of a data search trial to find the RNA-seq data. The submitted filtering condition was a combination of *Orcinus orca* as an organism, Transcriptome analysis as sequencing application, and Illumina HiSeq 2000 as sequencing instrument. The search index has no result as shown at the top donut graph, but one can understand there is only one data entry sequenced *Orcinus orca*, while the sequencing application and platform were not the factors to reduce the number of matched entries.

Browsing search results by project report format

The search interfaces provided by the INSDC members show the search result separated for each object, such as project, experiment, sample, or run. This makes users to get lost on the websites unless they understand the complexity of the metadata object relationship. Therefore, in the newly implemented data search application, the search result shows information integrated multiple objects into one project (Figure 2.4). The result page shows the main three facets on top followed by related publications, materials, and methods of the articles, a list of publications that cite the article. In the bottom of the page, sample information is summarized in a table where users can perform filtering or sorting. The table data can be exported to TSV or JSON format file to download. This result page format helps users to find data of their interest effectively.

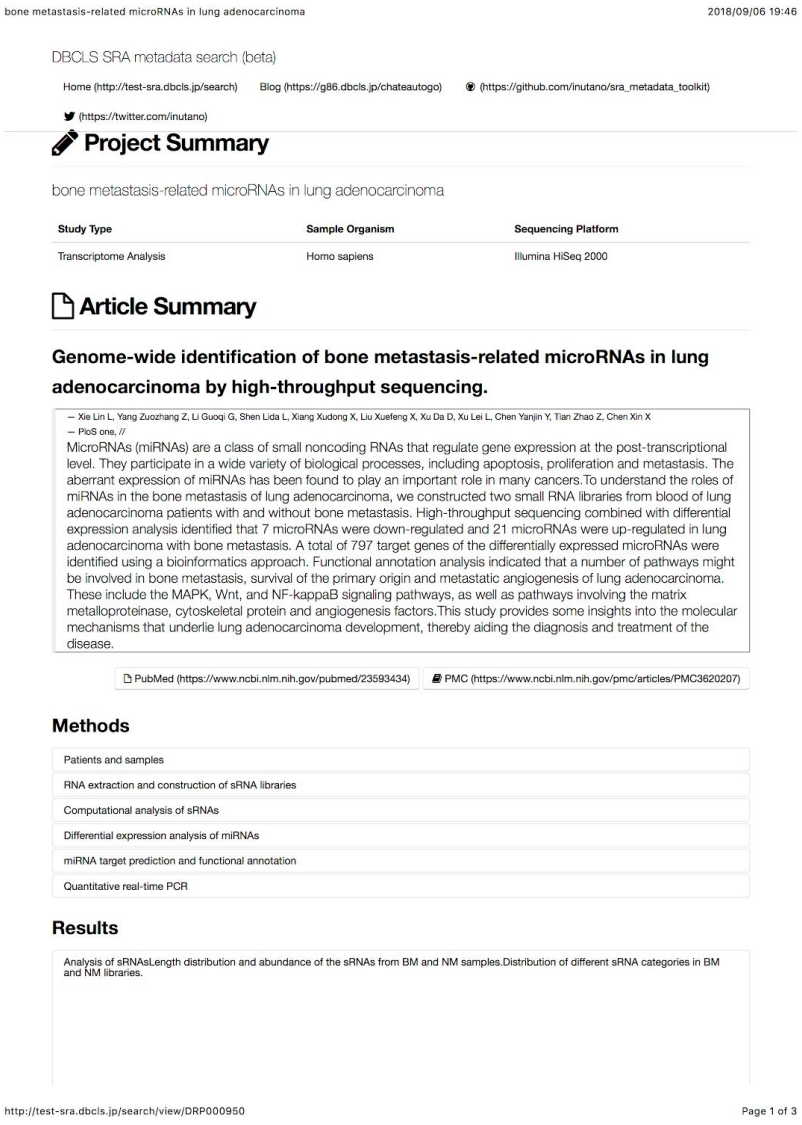


Figure 2.4: Report formatted SRA data search result

The search result page of DRP000950 on the faceting search web application. The page shows key metadata used for faceting search on the top, and related article summary with headings of methods and result of the manuscript, publications citing the paper, sequencing profile with read condition, information of sequenced samples, and hyperlinks to external resources. The actual looking of the result page differs since the

screenshot expands links of buttons, and the click-to-expand actions are embedded on headings of paper information.

API implementation for programmatic data search

The web application interface is not sufficient for the cases that users have to inspect metadata fields of many entries matched to the query. There are also cases that users want to collect and analyze metadata fields from a large number of entries. To help those use cases, I implemented an Application Program Interface (API) to perform large-scale data search. Users can access the server with the syntax to get results in JSON format (Table 2.1). API accepts standard HTTP GET/POST requests, which makes it easy to create scripts in any programming languages to access the system. This feature enables efficient data search trials which handle more than tens of thousands of search results.

Command	HTTP Method	URI format	Example	Response data type
Count number of entries	GET	/<object type>	/api/bioproject	Number of entries (integer)
Get an entry by ID	GET	/<object type>/<id>	/api/biosample/SAMD00062996	Full metadata object
Get entries by multiple IDs	POST	/<object type>	/api/biosample	A list of full metadata object
List unique values	GET	/<object type>/<key name>	/api/sra/experiment/instrument_model	A list of values with counts
Keyword search (specific field)	GET	/<object type>?<key>=<value>	/api/biosample?taxid=9606&title=breast%20cancer	A list of identifiers
Keyword search (all fields)	GET	/<object type>?term=<value>	/api/biosample?term=hot%20spring	A list of identifiers

Range search	GET	/<object type>?<date field>=<date1>TO<date2>	/api/bioproject?submission_date=2017-01-01TO2017-12-31	A list of identifiers

Table 2.1: The SRA data search API reference

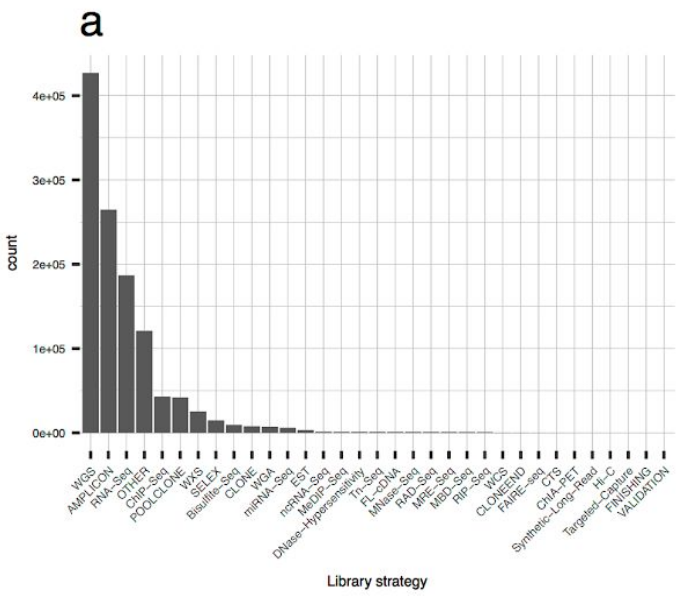
The commands to perform metadata search via RESTful API. Users can use a command line interface or web browser to access the URIs to get JSON formatted search result. Users can use a keyword search and range search functions to get identifiers of entries of interest, then get full metadata of the entries by using GET or POST method. The return values are all JSON format which allows users to use their favorite programming language or framework.

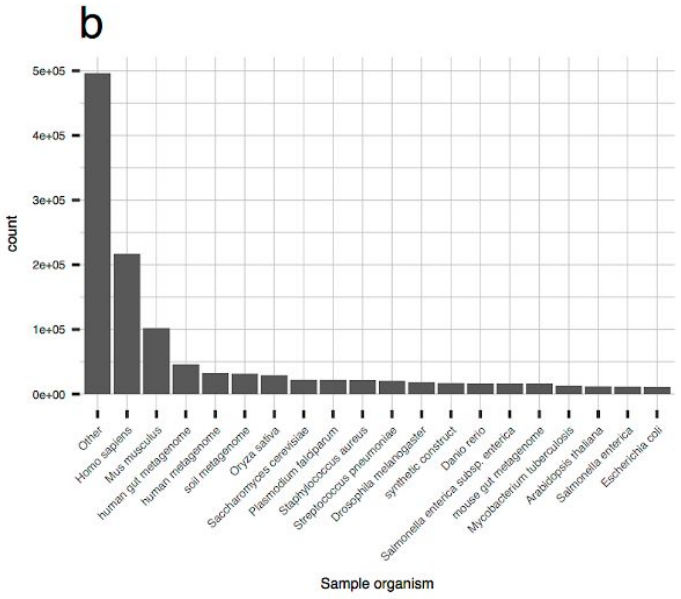
Downloading of sequencing data

To calculate quality values of sequencing data, I downloaded the data from the SRA, which is the largest public repository for HTSeq data [11]. Sequencing data containing personally identifiable information that should be shared in a controlled-access manner are not archived in SRA. In this study, I downloaded open-access SRA data stored in FASTQ format from the FTP server of the DDBJ [32].

I analyzed all of the publicly available HTSeq data submitted to SRA up until December 2015. The total number of sequenced samples was 1,171,313 and the number of sequenced bases was more than 2.7 trillion. The varieties of sequencing methods, sequencing instruments, and sequenced sample organisms are shown in Figure 2.5, which were extracted from the metadata described by the data submitter. The most common sequencing method is the whole-genome shotgun (WGS) approach, which was

employed for 426,841 samples, or 36.4% of the total. The number of different sequenced organisms is 33,961, based on the Taxonomy ID. The most commonly sequenced organism in SRA is human, with 216,896 samples, or 18.5% of the total, while the total number of samples whose scientific name contains “metagenome” is 244,457, or 20.9% of the total. The number of experiments counted by the sequencing instrument model used shows that Illumina HiSeq 2000 is the most commonly used instrument in SRA, with 542,332 experiments, or 46.3% of the total.


a
count
4e+05
3e+05
2e+05
1e+05
0e+00
WGS
AMPLICON
RNA-Seq
OTHER
ChIP-Seq
POOLCLONE
WXS
SELEX
Bisulfite-Seq
CLONE
WGA
miRNA-Seq
EST
ncRNA-Seq
MeDIP-Seq
DNase-Hypersensitivity
Tn-Seq
FL-cDNA
MNase-Seq
RAD-Seq
MRE-Seq
MBD-Seq
RIP-Seq
WCS
CLONEEND
FAIRE-seq
CTS
ChIA-PET
Synthetic-Long-Read
Hi-C
Targeted-Capture
FINISHING
VALIDATION
Library strategy


b
count
5e+05
4e+05
3e+05
2e+05
1e+05
0e+00
Other
Homo sapiens
Mus musculus
human gut metagenome
human metagenome
soil metagenome
Oryza sativa
Saccharomyces cerevisiae
Plasmodium falciparum
Staphylococcus aureus
Streptococcus pneumoniae
Drosophila melanogaster
synthetic construct
Danio rerio
Salmonella enterica subsp. enterica
mouse gut metagenome
Mycobacterium tuberculosis
Arabidopsis thaliana
Salmonella enterica
Escherichia coli
Sample organism

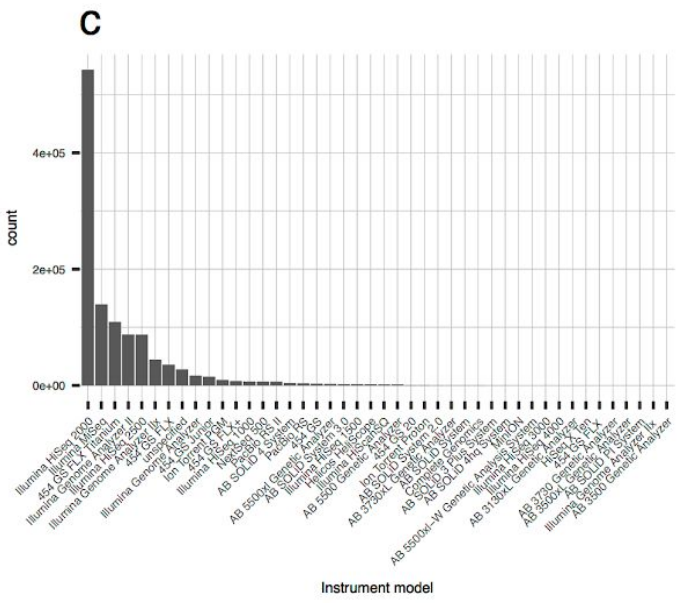

c
count
4e+05
2e+05
0e+00
Illumina HiSeq 2000
Illumina MiSeq
454 GS FLX Titanium
Illumina Genome Analyzer IIx
Illumina HiSeq 2500
Illumina Genome Analyzer
454 GS FLX
Unspecified
Illumina Genome Analyzer II
454 GS 20
Ion Torrent PGM
Illumina HiScanSQ
Ion Torrent Proton
NextSeq 500
AB SOLiD 4 System
AB 5500xl Genetic Analyzer
AB SOLiD System 3.0
Pacific Biosciences SMRT
454 GS
Illumina HiSeq 1000
Illumina HiSeq 1500
AB 5500 Genetic Analyzer
Helicos HeliScope
454 GS Junior
Ion Torrent Proton 2.0
AB 3730xl Genetic Analyzer
AB SOLiD System
Complete Genomics
AB SOLiD 4hq System
AB SOLiD 3hq System
Illumina Genome Analyzer IIe
AB 5500xl-W Genetic Analyzer
Illumina HiSeq 4000
AB 3130xl Genetic Analyzer
Illumina HiSeq 3000
AB 3730 Genetic Analyzer
AB 3500xL Genetic Analyzer
454 GS FLX+
Illumina Genome Analyzer
AB SOLiD PI System
AB 3500 Genetic Analyzer
Instrument model

Figure 2.5: Performed sequencing experiments and sequenced samples of public data for quality calculation

(a) Bar plot of the top 20 library strategies. Values are categorical, retrieved from metadata described by the data submitter. (b) Bar plot of the top 20 sequenced sample organisms. Taxonomy information is retrieved from the NCBI taxonomy database and declared by the data submitter. (c) Bar plot of sequencing instrument models.

Calculation of sequence read quality

To enable filtering of the search results in the repository by quality information, I extracted sequence read quality values from raw sequencing data using FastQC. FastQC is one of the most popular software programs for performing quality control of HTSeq data [22]. By using the results from FastQC, I calculated comparable values of sequence data, such as the total number of reads, mean and median sequence read length, %GC, read duplicate percentage, mean and median base call accuracy, and percentage of failed base calling (N content) (Table 2.2). The read quality values were calculated for each downloaded set of sequencing run data in FASTQ format, and then assembled using the SRA Experiment ID.

Calculated quality value	**Numbers of multiple runs in an experiment**	**Used FastQC modules**
Total Number of Reads	Added	Basic Statistics module
Mean/Median Read Length	Average	Sequence Length Distribution module

%GC	Average	Basic Statistics module
Total Duplicate Percentage	Average	Duplicate Sequences module
Mean/Median Base Call Accuracy	Average	Per Base Sequence Quality module
N Content	Average	Per Base N Content module

Table 2.2: Calculated sequence quality values

The list of calculated values with methods to merge multiple values in one experiment and used FastQC modules.

I integrated the categorical values described in metadata of the sample and experiment with calculated read quality data. Experimental metadata were extracted from an SRA metadata XML file downloaded from the FTP server of the NCBI Sample information was extracted from the XML file downloaded from BioSample, a database maintained by the INSDC to archive information on biological materials [33].

The state of the HTSeq repository visualized by the distribution of data quality

Providing sequence data quality enables users to control the number of search results from a data repository. The integration of information on data quality with metadata of samples and experiments can be used to develop a better search function. However, to offer a method of obtaining a suitable dataset from thousands of search results, it is necessary to know the standard of data quality and the data distribution in the repository. To illustrate the state of publicly available HTSeq data using quality values, histograms were created for sequencing throughput, base call accuracy, and N

content (Figure 2.6, Figure 2.7, Figure 2.8, Figure 2.9). As Figure 2.5 shows, there is a huge bias in numbers of sequencing methods, sequenced organisms, and used sequencing instruments. Thus, I focused on the factor that defines the range of the quality values, not the count of data, which is probably affected by the bias of the number of sequencing instruments. To understand the data attribute that is decisive to its distribution, histograms were color-coded (Figure 2.6b, 2.6d) or separated (Figure 2.7, Figure 2.8) in terms of the metadata of sequencing experiments and sequenced sample organisms. In the histograms of sequencing throughput, library source, particularly genomic, transcriptomic, or metagenomic source of sequencing, clearly explains the distribution of sequenced bases (Figure 2.6a, 2.6b, Figure 2.7). Overall, the mean value of throughput was 2.371e+09 and the median value was 3.349e+08. In the histogram of base call accuracy, as expected, the values are strongly affected by the choice of sequencing chemistry (Figure 2.6c, 2.6d, Figure 2.8). The mean value of base call accuracy was 29.45, while the median value was 35.52. The histogram drawn by N content showed that 1,103,515 items, namely, 94.2% of the data, had N at less than 1% of the total sequences (Figure 2.9). For the data with a higher proportion of N content, there may have been an error in the sample DNA preparation or sequencing operation.

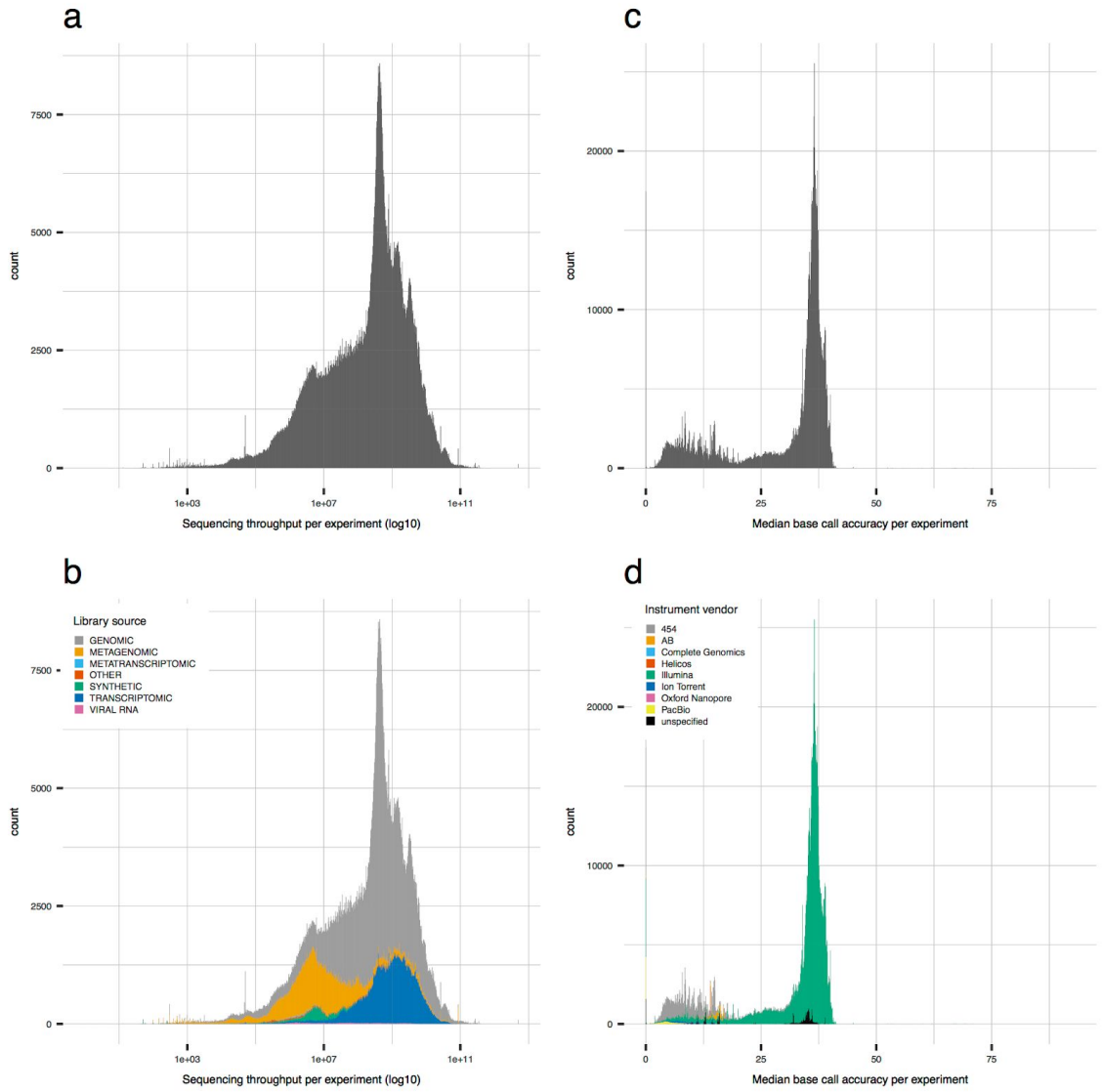


Figure 2.6: Data distribution in a public data repository by sequencing quality

(a, b) Histogram of sequencing throughput (a), and one color-coded by library source (b). (c, d) Histogram of base call accuracy (c), and one color-coded by instrument manufacturer (d).

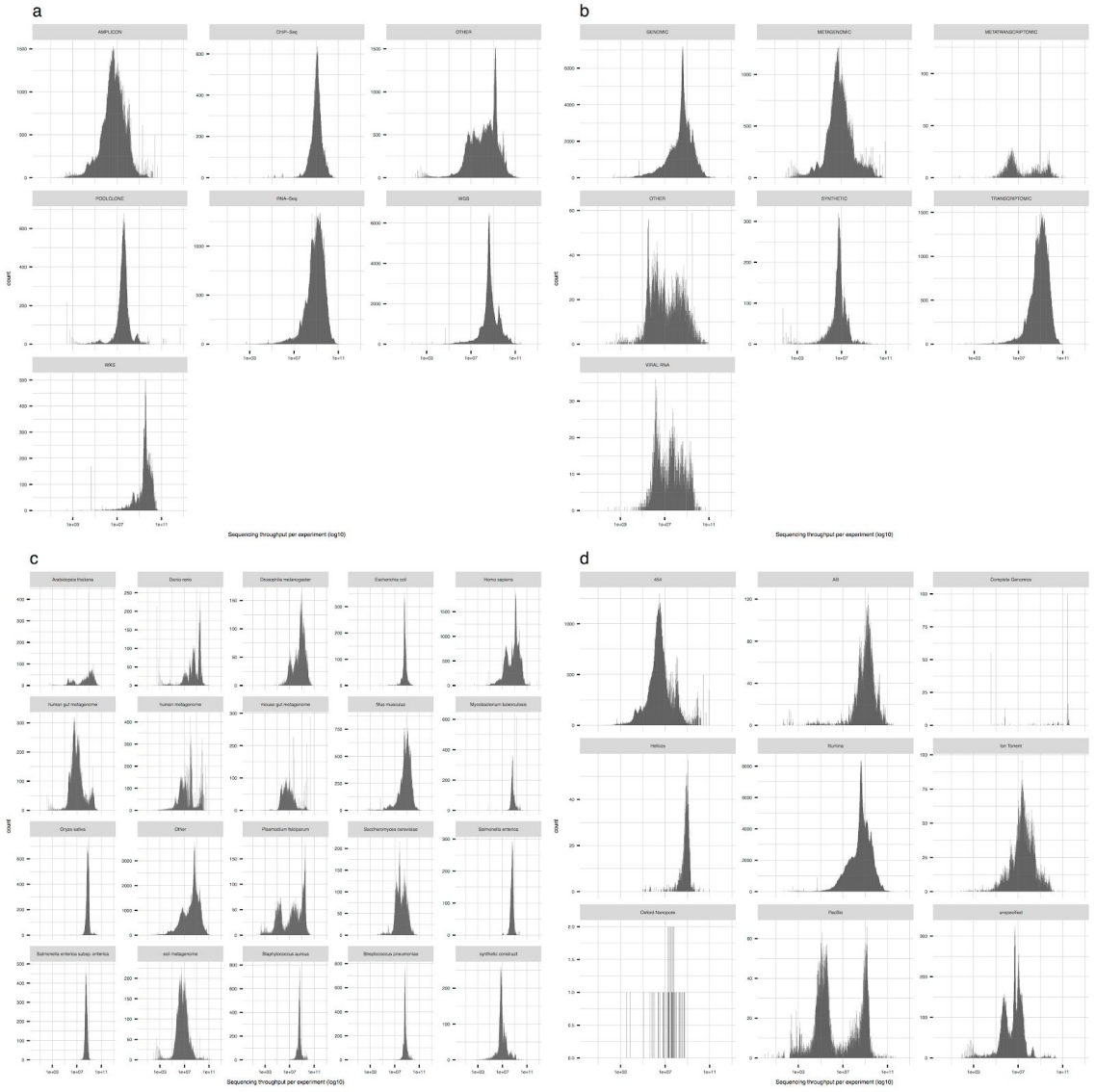


Figure 2.7: Data distribution of sequencing throughput for each set of metadata

(a–e) Histograms of sequencing throughput (a), separated by library strategy (b), library source (c), top 20 taxonomic scientific names (d), and instrument manufacturer (e).

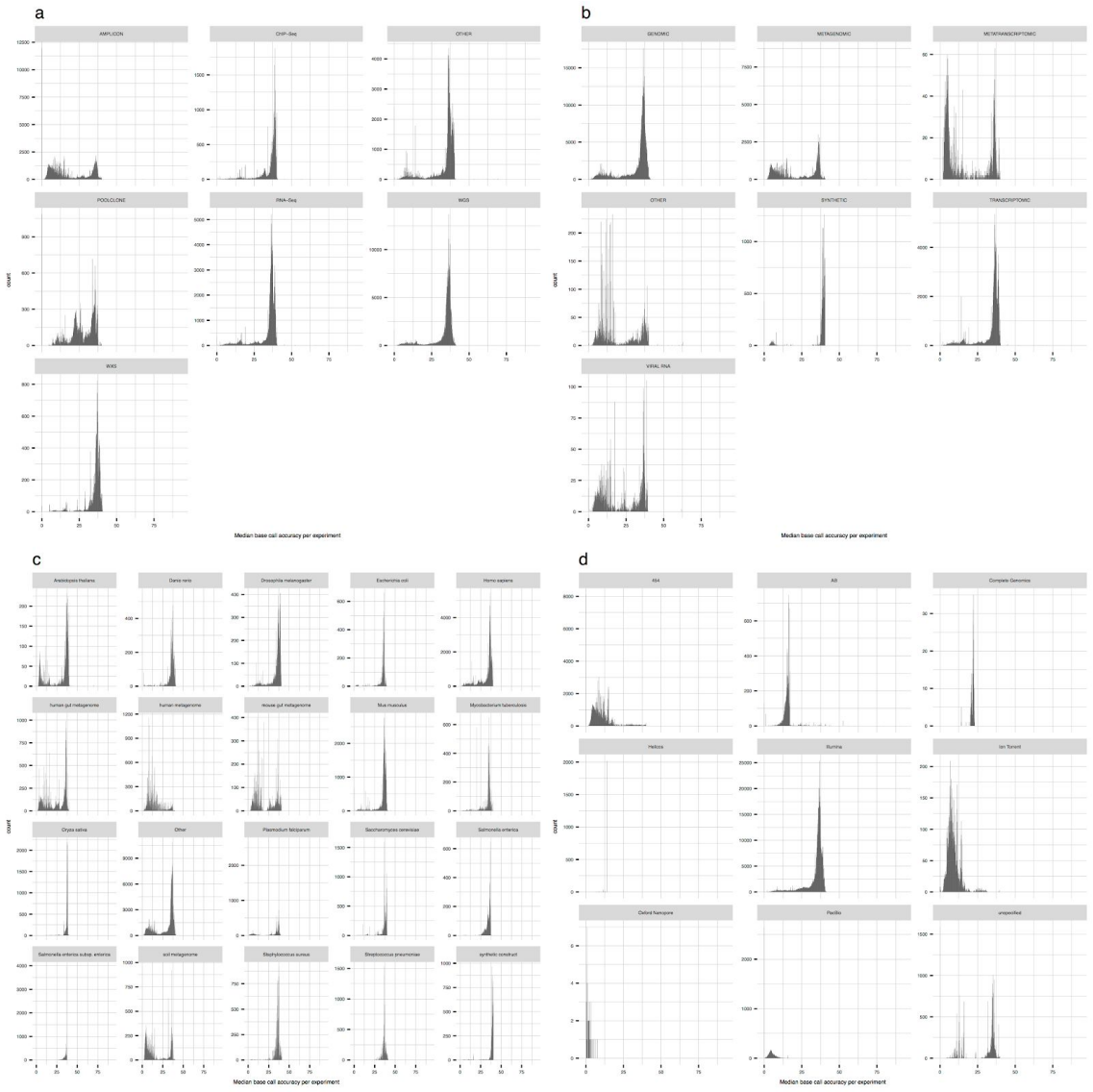


Figure 2.8: Data distribution of base call accuracy for each set of metadata

(a–e) Histograms of base call accuracy (a), separated by library strategy (b), library source (c), top 20 taxonomic scientific names (d), and instrument manufacturer (e).

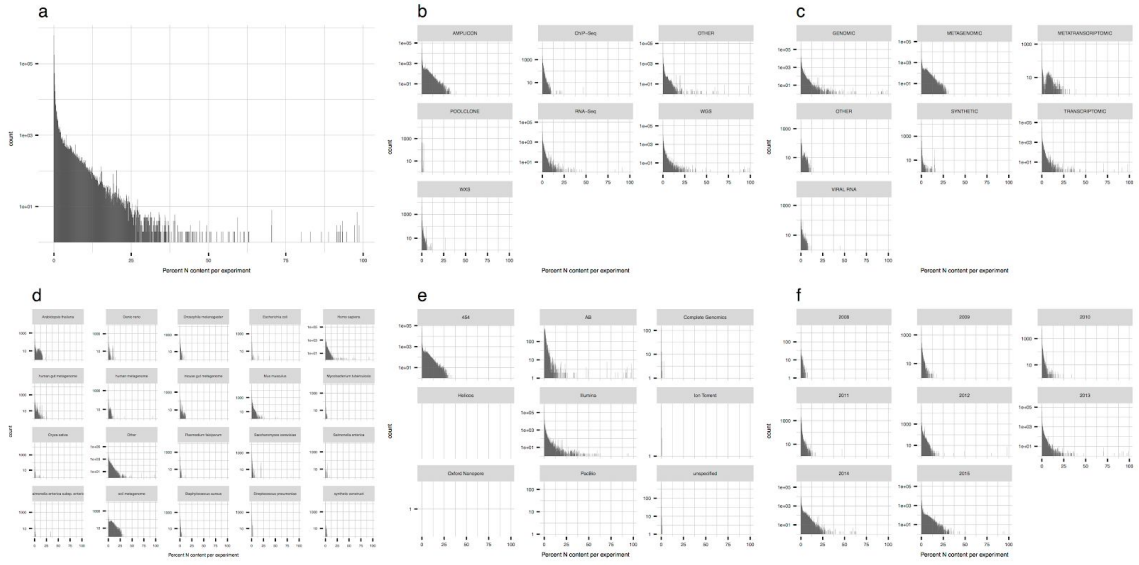


Figure 2.9: Data distribution by N content

(a–f) Histograms of N content percentage per experiment. Histograms of base call failure of overall (a), separated by library strategy (b), library source (c), sample organism (d), instrument manufacturer (e), and year of data submission (f). The y-axis is log 10 scale.

Data distribution by read quality for each sequencing method

SRA accepts the submission of various kinds of sequencing data, such as those obtained by WGS, RNA-Seq, ChIP-Seq, and metagenomic approaches, as well as many other DNA library construction strategies. To accomplish higher measurement accuracy and greater dynamic range, each sequencing method has ideal conditions regarding sequencing quality. I analyzed the distribution of data in each dataset by a library strategy to investigate how many performed experiments achieved such ideal conditions. I employed 988,678 sets of data for this analysis, which were obtained by the sequencing of human samples via WGS, amplicon sequencing, RNA-Seq, ChIP-Seq, pooled clone sequencing, or whole-exome sequencing (WXS). I visualized the data distribution by creating a histogram for each library strategy (Figure 2.10). The histograms were also separated by the sequencing instrument manufacturer to show which type of sequencing chemistry had been selected (Figure 2.11). In one of the six library strategies, namely, amplicon sequencing, multiple types of sequencing chemistry were used, while the others were performed mostly by the Illumina sequencing chemistry. The histograms indicate that the five library strategies require a larger number of sequence reads and higher base call quality. In contrast, experiments by other library strategies were performed with a short read length of around 100 bases long, while some amplicon sequencing experiments were performed with longer sequence reads of hundreds of bases. A total of 66.3% of amplicon sequencing experiments were performed by non-Illumina sequencers, for which the average read length was 388.4. This is consistent with the standards of each sequencing strategy [34].

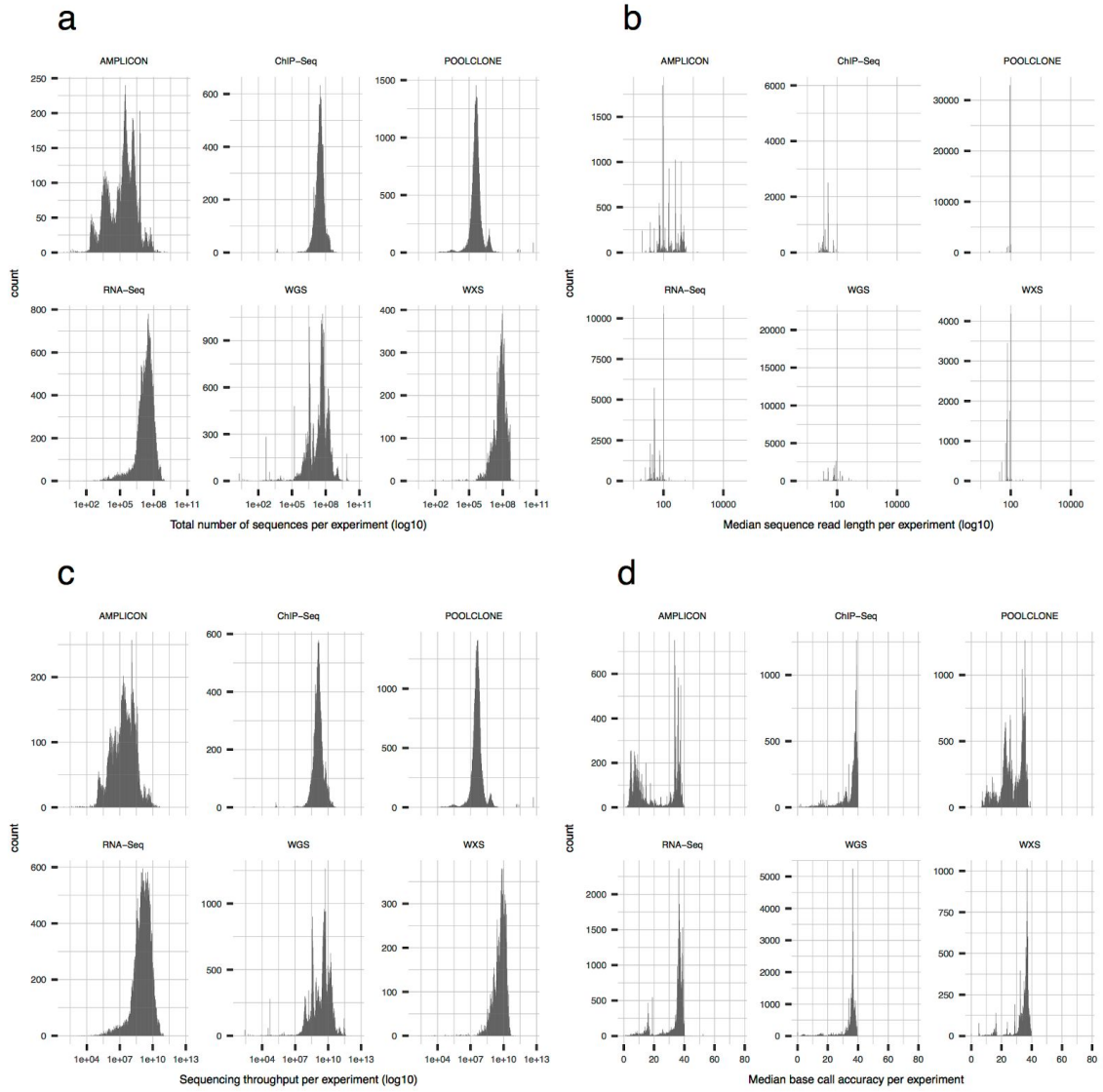


Figure 2.10: Human data distribution for each library strategy

(a–d) Histograms separated by the top 6 library strategies. Data distribution is by the total number of sequences (a), median read length (b), sequencing throughput (c), and median base call accuracy (d) per experiment.

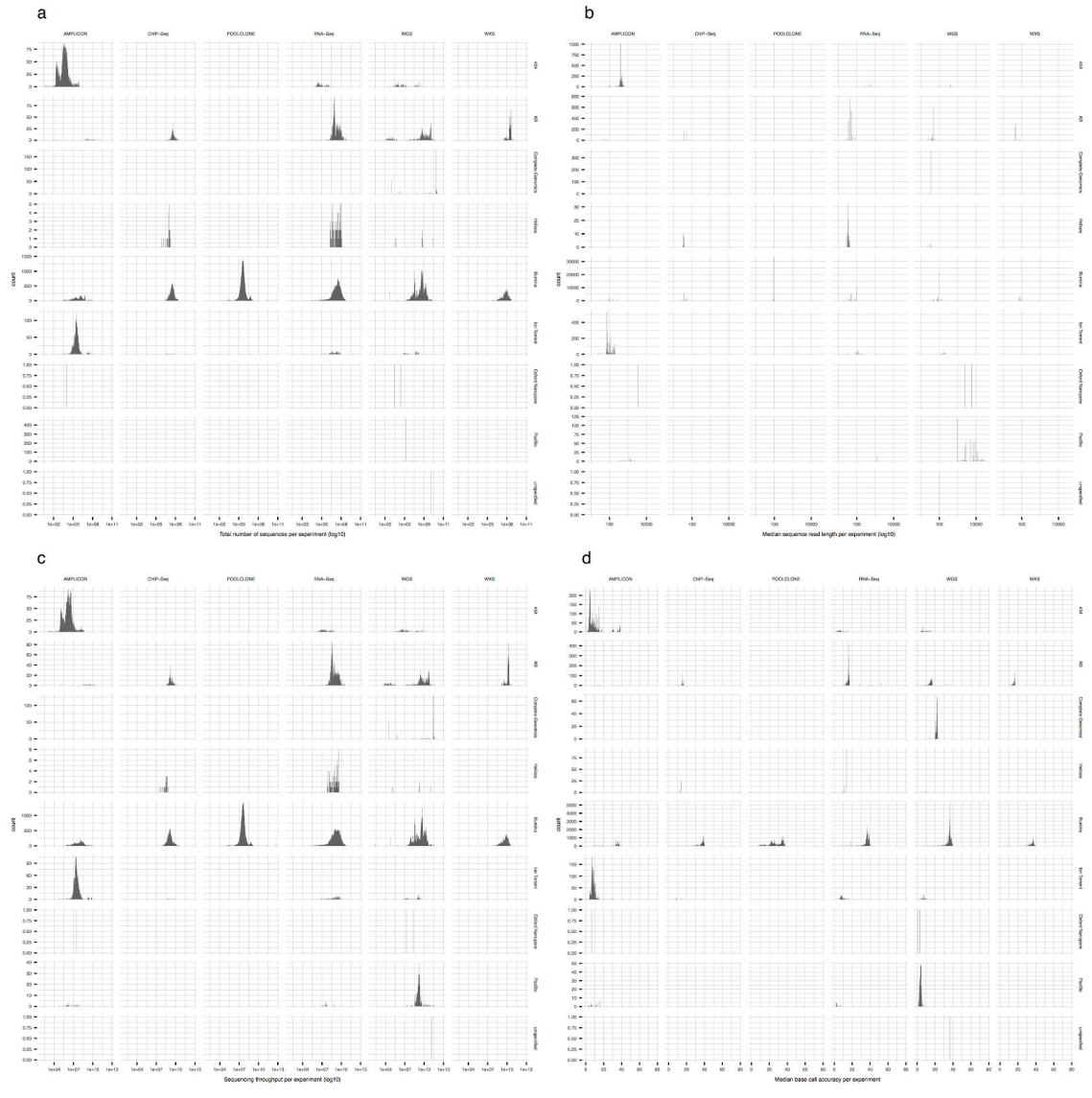


Figure 2.11: Human data distribution for each library strategy separated by instrument manufacturer

(a–d) Histograms separated by the top 6 library strategies and instrument. Data distribution is by the total number of sequences (a), median read length (b), sequencing throughput (c), and median base call accuracy (d) per experiment.

Changes of sequencing quality during SRA's history

Since 2007, when the first next-generation sequencing data were submitted to the SRA, there have been rapid advances in the sequencing technology regarding both the instruments and the chemistry, which have significantly improved the quality of sequencing data. The improved specs of sequencers have enabled various new sequencing methods to be developed, but have also helped improve the data quality output by existing methods. I visualized the changes of quality values for each sequencing method over time. A change in four sequencing qualities, total number of reads, read length, sequencing throughput, and base call quality, of six library strategies, WGS, amplicon, RNA-Seq, ChIP-Seq, pooled clone, and WXS, are visualized by box plots in quarterly time series (Figure 2.12, Figure 2.13). While the plots of pooled clone sequencing could not be evaluated due to a lack of continuous data submission, the plots of the other strategies show their trends over time. The plots of amplicon sequencing show no specific tendency, probably indicating that such sequencing quality values are determined by the characteristics of each sequencing project, surveying of which requires more detailed metadata. In ChIP-Seq and WXS, sequencing throughput increased slightly over time. In plots of base call accuracy, ChIP-Seq, RNA-Seq, WGS, and WXS showed increases of the value, possibly reflecting the improvement of sequencing technologies.

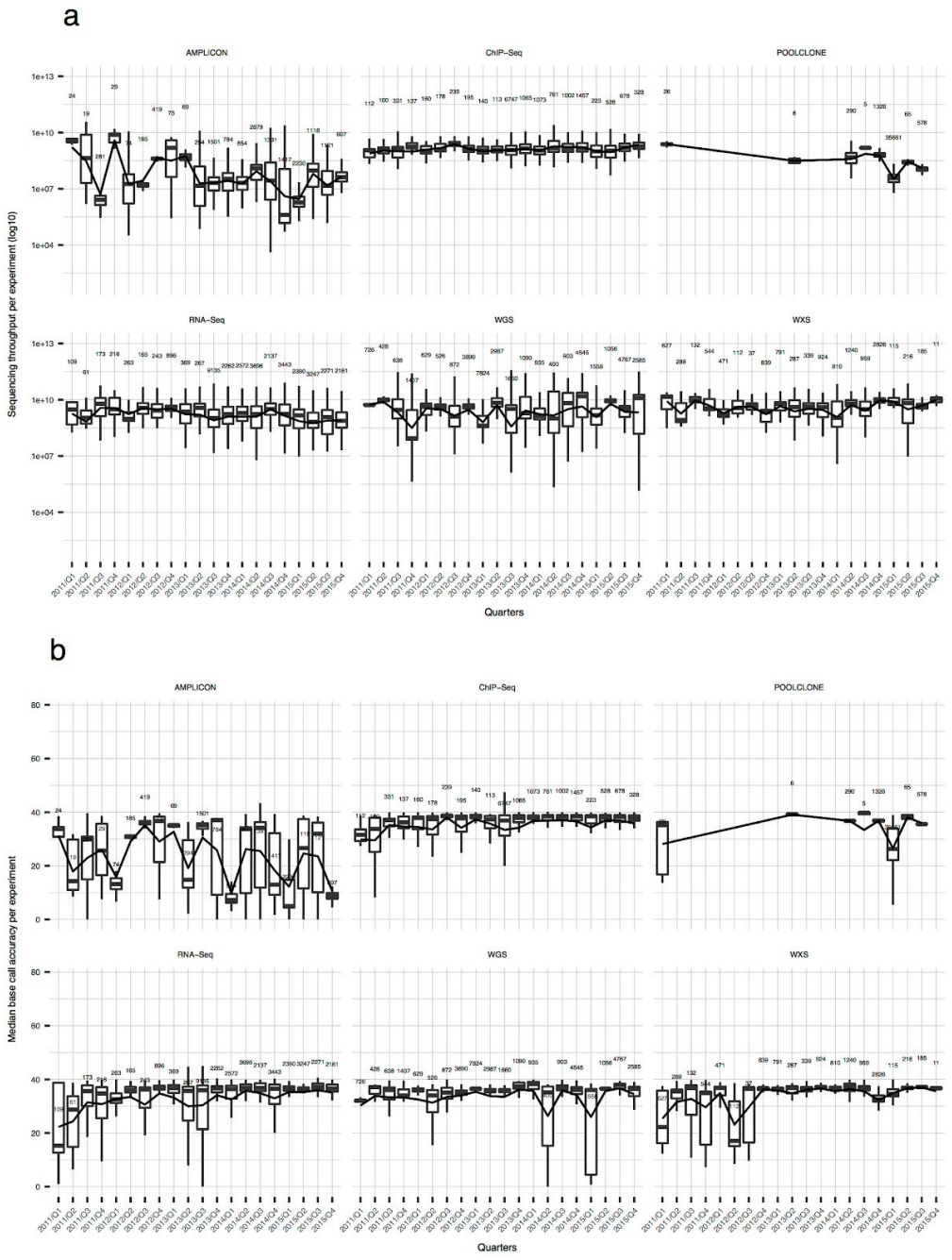


Figure 2.12: Change of data distribution by sequencing quality over time

(a, b) Box plots separated by the top 6 library strategies, showing quarterly change. Data distribution is by the sequencing throughput (a) and median base call accuracy (b) per experiment. The numbers in plots indicate the numbers of samples in a row. The lines connecting boxes indicate changes of mean value.

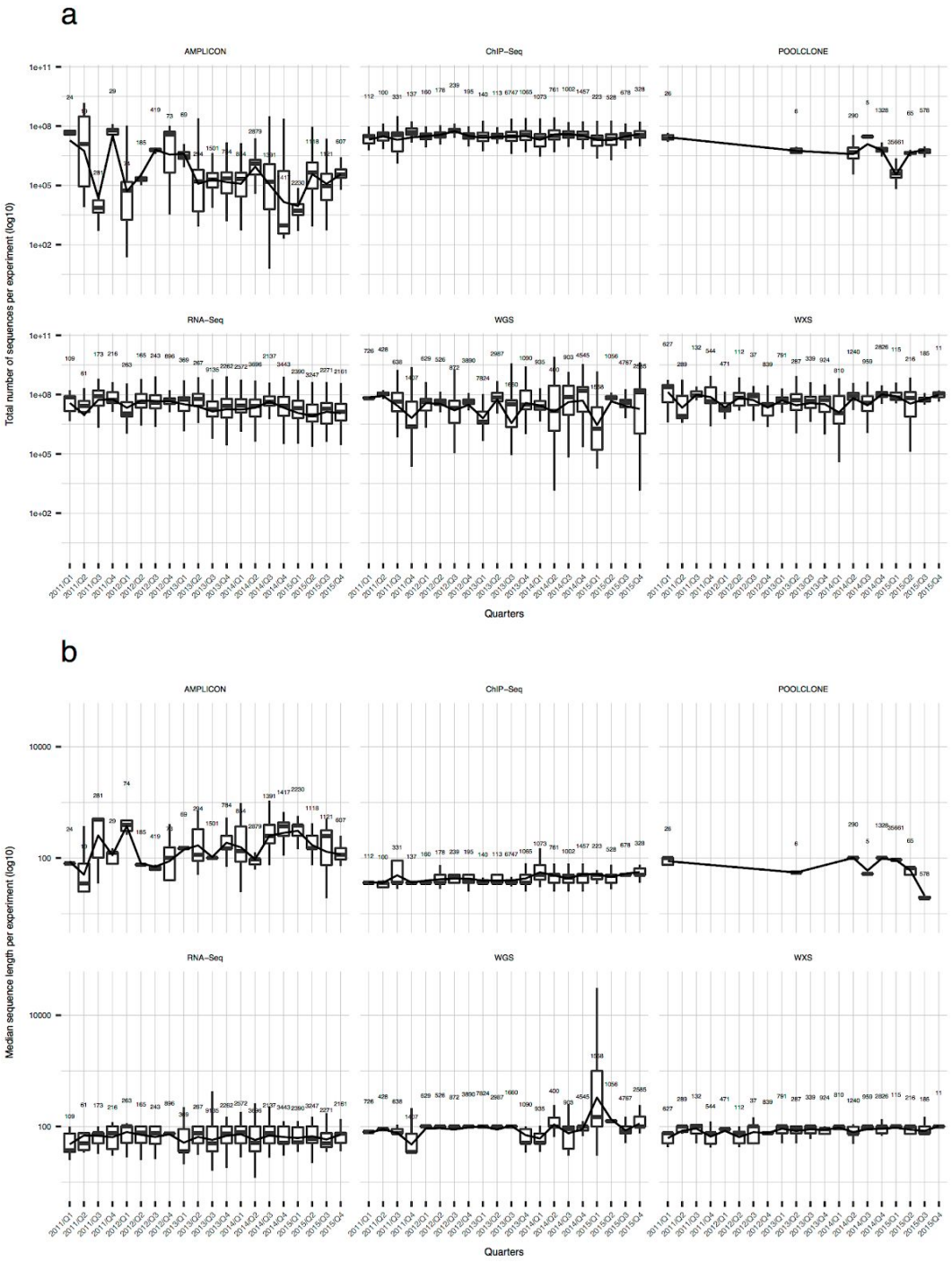

a
AMPLICON
ChIP−Seq
POOLCLONE
RNA−Seq
WGS
WXS
Total number of sequences per experiment (log10)
Quarters
b
Median sequence length per experiment (log10)
Quarters

Figure 2.13: Change of data distribution by number of reads and read length over time.

Box plot of sequence quality per experiment over time. (a) Data distribution by total number of sequence reads per experiment. (b) Data distribution by median sequence read length per experiment.

Discussion

I constructed a data search index to integrate related publication information. This extended metadata enabled users to search with queries of more variations. However, using the contents of the published article as metadata of archived data also have a problem. The system cannot remove irrelevant information from the contents, and the information can cause search noise.

To reduce the cost to evaluate search results, I developed the web application with an interface to show the integrated search result in a report-like format. The interface was designed to help users' decision to use the matched data by organizing related information and emphasizing main attributes. Also, for the search cases that have many matched entries, I implemented API to allow programmatic access which helps users to perform a large scale data exploration efficiently.

As the number of pairs of PMID and SRA ID was smaller than that of total projects, connecting publications to improve findability of data has a limitation. It is also important to help data submitters to describe metadata richer in quality and quantity. The INSDC started to ask submitters to use the packages per sequencing purpose, which define required and recommended fields of sample or experiment metadata with the data type and a controlled vocabulary set to be used [35]. It is obvious that cooperation of the database administrators and data submitters to improve the value of archived data, and the database itself.

By calculating the quantitative variables of sequencing data and integrating them with information on experiments and sample organisms, I enabled an appropriate size of the subset to be obtained from multiple projects archived in the repository. Without any quantitative information, users cannot choose a reliable dataset from among thousands of search hits. When users search data with a query of sample-related information, such as a treatment of biological materials, the number of search results tends to be very small or too large for users to be able to browse through, due to the lack of detailed metadata. It is also claimed that the metadata described by data submitter lack some important information, or may contain errors [36]. In contrast, my results can provide information in a way that enables users to look into a large dataset and control the amount of data output by their search by setting a threshold regarding the quality value.

The amount and the accuracy of sequencing data is drastically changing in these years. This means that database users have to care about the details of the experiment, for example, date of sequencing or used sequencing equipment for each database entry.

The quality information of public sequencing data provided by my work can be used to evaluate the reliability of entries in biological databases, such as genome variations or gene expressions.

CHAPTER 3

WORKFLOW DESCRIPTION WITH RUNTIME INFORMATION TO ENABLE REPRODUCIBLE DATA ANALYSIS

Published Article 3

Accumulating computational resource usage of genomic data analysis workflow to optimize cloud computing instance selection.

Tazro Ohta, Tomoya Tanjo, Osamu Ogasawara.
BioRxiv. 2018 Jan 1:456756.

Background

According to the improvement of DNA sequencing technology in accuracy and quantity, various sequencing methods are now available to measure different genomic features. Each method produces a massive amount of nucleotide sequence data that requires a different data processing approach [37]. Bioinformatics researchers develop data analysis tools for each sequencing technique, and they publish implementations as open source software [38]. To start data analysis, researchers need to select the tools by their experimental design and install them to their computing environment.

Installing open source tools in one's computational environment is, however, not always straightforward. Tools developed by different developers and different programming framework require different prerequisites, which forces one to follow the instruction provided by each tool's developer. Installing various software in one environment also can occur a conflict of software dependencies that are hard to resolve. Even if one could successfully install all the tools required for the analysis, maintaining the environment where all the tools keep working as expected is also a burden. There are also many events that can break the environment such as changes or updates of hardware, operating system, or software libraries. Therefore, the complexity of data analysis environment management gets higher when a project performs genomic data analysis that requires many tools. The high cost of setting up an environment results in the prevention of scaling out the computational resources as well. The difficulty also

brings researchers' dependency to the existing computing platform already set up, and the concentration of data processing jobs to the limited resource.

The container virtualization technology, represented by Docker, enables users to create a software runtime environment isolated from the host machine [39]. This technology that is getting popular also in the biomedical research domain is a promising method to solve the problem of installing software tools [40]. Along with the containers, using workflow description and execution frameworks such as those from the Galaxy project [41] or the Common Workflow Language (CWL) project [42] lowered the barrier to deploy the data analysis environment to a new computing environment. Moreover, the workflows described in a standardized format can help researchers to share the environment with collaborators with ease. The improvement of portability of data analysis environment, consequently, has made the on-demand cloud infrastructure an appealing option for researchers.

On-demand cloud is beneficial for most cases in genome science because users can increase or decrease the number of computing instances without maintaining hardware as the amount of data from laboratory experiments changes [43]. For example, some sequencing applications require data analysis software that uses a considerable amount of memory, but individual research projects often cannot afford such a large scale computing platform. Users can save their budget by using the on-demand cloud platform as most of the service providers charge per usage.

However, to use an on-demand cloud environment efficiently regarding time and economic cost, it is essential to select a suitable computing unit, so-called *instance type*, from many options offered by the cloud service providers. For example, Amazon Web Service (AWS), one of the popular cloud service providers, offers instance types of different scales for five categories (general purpose, compute optimized, memory optimized, accelerated computing, and storage optimized) [44]. Each data analysis tool has the different minimum requirement of computational resources such as memory or storage, and it can change by input parameters. Executing data analysis workflows on an instance without enough computational resource will result in a runtime failure or unexpected outputs. For example, tools to assemble short reads to construct genome by constructing De Bruijn graph usually take long processing time and a large amount of memory. If one failed to estimate the required amount of memory, the process might fail after a few days of execution, which results in losing one's time and budget. Thus, users need to know the minimum amount of computational resource required by the execution of their workflows to select a suitable instance type.

To optimize the instance type selection concerning processing time or running cost, users need to compare runtime metrics of workflow executions on environments of different computational specs. Here, I developed *CWL-metrics*, a system to accumulate runtime metrics of workflow executions with information of the workflow and the machine environment. CWL-metrics provides runtime metrics summary such as usage of CPU, memory, storage I/O with workflow's input files and parameters to help users to select the proper cloud instance for their workflows.

Methods

CWL-metrics software components

CWL-metrics runtime metrics capturing system is composed of five software components: Telegraf [45], Fluentd [46], Elasticsearch [47], Kibana [48], and a Perl daemon script. Telegraf is an agent to collect runtime metrics of running containers via Docker API using Telegraf Docker plugin. Fluentd works as a log data collector to send metrics data produced by Telegraf to Elasticsearch server. Elasticsearch is a data store to accumulate runtime metrics data and workflow metadata, accepting JSON format data via API endpoint. Kibana is a data browsing dashboard for Elasticsearch to view raw JSON data and to summarize and visualize data. Telegraf, Fluentd, Elasticsearch/Kibana launch as a set of containers during the initialization of CWL-metrics. CWL-metrics runs a Perl script which monitors processes on the host machine to capture cwltool processes. Once the script found a cwltool process, the script runs a function to collect workflow information via debug output of the cwltool process, "*docker info*" command output, Docker container log via "*docker ps*" command, and output of system commands to collect environment information. CWL-metrics provides a command *cwl-metrics*, which allows users to start and stop the metrics collection system, and fetch summarized runtime metrics data in a specified format, JSON or tab-separated format. The script to launch the whole system, CWL-metrics installation instructions, and the documentation are available on GitHub [49].

Packaging RNA-Seq tools and workflows

I used 7 different RNA-Seq quantification workflows to capture runtime metrics and analyze performance on cloud infrastructure. Each workflow starts with the tool to download sequence data from SRA, then convert SRA format file to FASTQ format. Consequently, each pipeline does sequence alignment to reference genome sequence (HISAT2, STAR, and TopHat2) or alignment-like approaches (Kallisto and Salmon) to the set of reference transcript sequence, then perform transcript quantification. Most of the tool containers used in the workflows are from the Biocontainers [50] registry. I containerized the tools those are not available on the registry and uploaded them to the container registry service Quay [51]. I described tool definitions such as input and output of tool execution and the workflow procedures in CWL tool files, which are available on GitHub [52]. Each workflow has two options for sequence read layout single-end and paired-end; thus I used fourteen workflows in total. The Table 3.1 shows the tool versions, the online location of the CWL tool files, and the original tool website locations.

Select RNA-Seq workflow input sequence data from the public data repository

To analyze the effect of sequence data quality to workflow runtime performance, I chose 9 samples of different read length and number of reads from the public raw sequencing data repository, SRA (Table 3.2). I used the Quanto database [53] to select the data by filtering length and number of sequence reads, with the condition of read length, 50, 75, or 100 and the approximate number of sequence, 1,000,000, 5,000,000, or 10,000,000. I filtered the data with the query "organism == Homo sapiens", "study type == RNA-Seq", "read layout == PAIRED", and "instrument model == Illumina

HiSeq", then manually picked suitable data. Both single-end and paired-end workflows used the same dataset while single-end workflows treated paired-end read files reads as two single-end read files. The version of the reference genome is GRCh38. I downloaded the reference genome file from the UCSC genome browser [54], and the transcriptome was from Gencode [55].

SRA Run ID	Read length	Number of reads per strand	BioSample ID	Sample description	Sequencing instrument
SRR4250750	50	1,000,425.00	SAMN05779985	cultured embryonic stem cells	Illumina HiSeq 2500
SRR5185518	50	5,008,398.00	SAMN06239034	cultured embryonic stem cells	Illumina HiSeq 2500
SRR2932901	50	10,017,495.00	SAMN04211783	fetal lung fibroblasts	Illumina HiSeq 2500
SRR4428678	75	1,043,870.00	SAMN05913930	embryonic stem cell derived macrophage	Illumina HiSeq 4000
SRR4241930	75	5,004,985.00	SAMN05770731	PGC-like cells (PGCLCs)	Illumina HiSeq 2000
ERR204893	75	10,234,883.00	SAMEA1573291	lymphoblastoid cell line	Illumina HiSeq 2000
SRR5168756	100	1,006,868.00	SAMN06218220	subcutaneous metastasis	Illumina HiSeq 2500
SRR5023408	100	5,004,554.00	SAMN06017954	primary breast cancer	Illumina HiSeq 2500
SRR2567462	100	10,007,044.00	SAMN04147557	prostate cancer cells LNCaP	Illumina HiSeq 2500

Table 3.2: The read characteristics of processed RNA-Seq data

We chose nine different RNA-Seq data from the SRA, a public HTSeq data. Each data are different in their read length and a total number of reads for performance

comparison. All data are from human sample sequenced by the Illumina HiSeq platform.

Run workflows on AWS EC2

To evaluate the performance on running different RNA-Seq workflows, I selected instance types of two different sizes 2xlarge and 4xlarge from three categories, general purpose, compute optimized, and memory optimized to run all workflows for all samples (Table 3.3). Each combination of instance type, workflow, and sample data was executed for five times while CWL-metrics is running on the same machine to capture the runtime metrics information. All workflow runs used Elastic Block Storage of General Purpose SSD volumes as file storage. I downloaded all the reference data used for workflows in advance. The scripts to get reference data and run workflows are available online [52].

Instance type	Category	vCPU	ECU	Memory (GiB)	Linux/UNIX Usage (per Hour)
m5.2xlarge	General Purpose	8	31	32	$0.384
m5.4xlarge	General Purpose	16	60	64	$0.768
c5.2xlarge	Compute Optimized	8	34	16	$0.34
c5.4xlarge	Compute Optimized	16	68	32	$0.68
r5.2xlarge	Memory Optimized	8	31	64	$0.504
r5.4xlarge	Memory Optimized	16	60	128	$1.008

Table 3.3: The machine specs of AWS EC2 instance types used in the metrics collection

To compare the performance of workflow runs on different computing platforms, we selected three categories from AWS EC2 categories, general purpose, compute optimized, and memory optimized. We further selected two different instance types from those three categories according to the number of virtual CPUs, 2xlarge and 4xlarge, with 8 and 16 CPU cores, respectively. Instance usage prices are as of 14 August 2018 for on-demand use in the US East (N. Virginia) region. Prices are not including charges for storage, network usage, and other AWS features.

Collect runtime metrics and summarize

After the workflow executions, I collected summarized metrics data from Elasticsearch by *cwl-metrics fetch* command. Exported JSON format data were parsed by a ruby script to create data summarized per workflow runs, loaded on Jupyter notebook [56] for further analysis. I calculated statistics of metrics by R language functions [29], and I created the box plots by the ggplot2 package [30]. The notebook file is available on GitHub [57].

Results

Implementation of CWL-metrics

CWL-metrics is designed to capture runtime metrics data of workflows described in CWL, a workflow description specification developed by an open source community. I designed the system as it does not require the users to perform any configurations to capture runtime metrics. Figure 3.1 shows the procedures of runtime metrics collection by CWL-metrics. To start collecting metrics, one only needs to install the system, and then run their workflows with *cwltool*, a reference implementation of CWL [58]. After the installation, the system starts monitoring the processes running on the host machine. Once the system found a cwltool process, it automatically starts collecting runtime metrics via Docker API and environmental information from the host machine. CWL-metrics also captures the log file generated by cwltool to extract workflow metadata such as input files and input parameters.

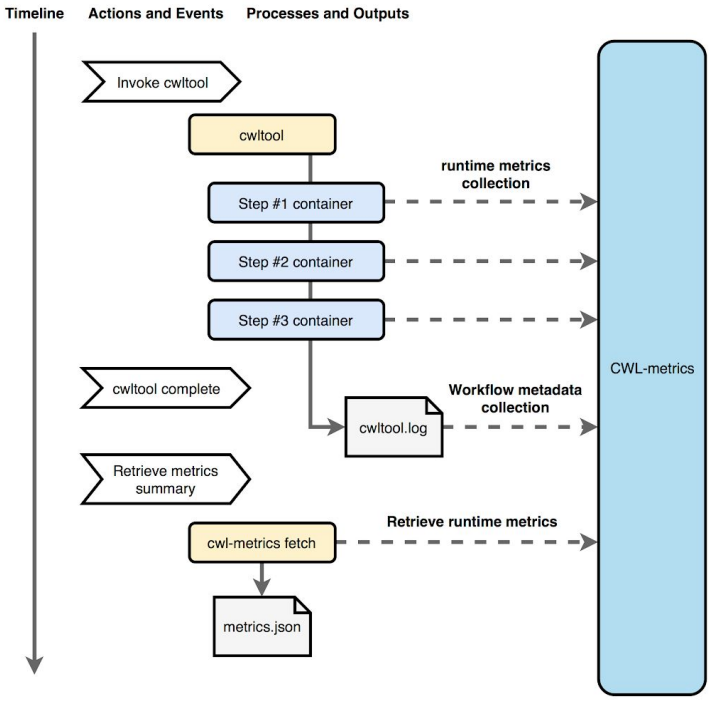


Figure 3.1: The container runtime metrics collection procedure with CWL-metrics

CWL-metrics was designed to capture runtime metrics of workflow steps automatically. After the initialization of the system, users only need to run a workflow by cwltool to start metrics capturing. The system collects runtime metrics of containers, and then the workflow metadata is captured after the workflow process finished. To retrieve runtime metrics, using the *cwl-metrics* command can output summary data in JSON or tab-delimited format.

To capture and store the information from multiple data source, CWL-metrics launches multiple components as Docker containers (Figure 3.2). These components keep running on the host machine after the initialization to cooperate the data collection. The Telegraf container collects runtime metrics data from the Docker API for every sixty seconds, and send the data to the Elasticsearch container. The Elasticsearch container provides data storage and the data access API. CWL-metrics automatically launches and stops these components on the single host machine. If users need to collect metrics of workflows running on multiple instances, they need to install CWL-metrics on each instance and assemble the summary data after the metrics data capture. Users can use their Elasticsearch server by setting environment variable ES_HOST and ES_PORT before initializing CWL-metrics.

To access and analyze the data collected by CWL-metrics, users can use the command *cwl-metrics* to get the data in JSON (Figure 3.3) or tab separated values (TSV) format. The JSON format contains workflow metadata such as the name of the

workflow, the time of start and end of the workflow execution. It also has the information of the environment including the total amount of memory and the size of storage available on the machine. The steps field of the JSON format contains information of the runtime metrics, the executed container, and the input files and parameters. Users can parse the data to analyze the performance of a tool execution or the whole workflow. The TSV format provides minimum information for each container execution so that one can easily compare the metrics data of steps.


```
{
  "CWL-metrics": [
    {
      "workflow_id": "3b66284a-969d-11e8-8d0f-0ae229374f7a",
      "workflow_name": "hisat2-cufflinks_wf_pe.cwl",
      "workflow_start_date": "2018-08-02T21:41:43+00:00",
      "workflow_end_date": "2018-08-02T21:44:25+00:00",
      "workflow_elapsed_sec": 162,
      "platform": {
        "instance_type": "c5.4xlarge",
        "region": "us-east-1a",
        "hostname": "4138af0fad86",
        "total_memory": "31897692",
        "disk_size": "508187044"
      },
      "steps": {
        "fcc52b5d2d3bf6dc1106c83117f5956c968047cbf0c5642144b86dbee32da619": {
          "stepname": "hisat2_mapping",
          "tool_status": "success",
          "input_files": {
            "SRR4428678_1.fastq.gz": 43828265,
            "SRR4428678_2.fastq.gz": 53452040,
            "out.sam": 778641728
          },
          "docker_image": "quay.io/biocontainers/hisat2:2.1.0--py36h2d50403_1",
          "docker_cmd": "hisat2 -S /var/spool/cwl/out.sam -x /var/lib/cwl/stg94f48183-8e7c-4fcb-
bc4b-58b2a7d33240/hisat2_GRCh38/genome --downstream-transcriptome-assembly --dta-cufflinks -1
/var/lib/cwl/stge9112392-7277-4049-8410-25324f93ec7c/SRR4428678_1.fastq.gz -2
/var/lib/cwl/stg31076a0b-02ab-4de9-9e8b-3b9af44152f8/SRR4428678_2.fastq.gz --threads 16 --time",
          "docker_start_date": "2018-08-02T21:41:52+00:00",
          "docker_end_date": "2018-08-02T21:42:09+00:00",
          "docker_elapsed_sec": 17.517223481,
          "docker_exit_code": 0,
          "metrics": {
            "cpu_total_percent": 1571.87279333333,
            "memory_max_usage": 5096611840,
            "memory_cache": 309497856,
            "blkio_total_bytes": null
          }
        }
      }
    }
  ]
}
```

Figure 3.3: An example of runtime metrics data summarized by CWL-metrics

CWL-metrics can output JSON formatted data which includes workflow metadata, tool container metadata, and tool container runtime metrics. The workflow metadata appears once for one workflow run with data of multiple steps in "steps" key while the example only has one step in the workflow to reduce the number of lines. Each step has a name, exit status, input files with file size, and details of the Docker container. Runtime metric values can be null for short-time steps since CWL-metrics collects these metrics with sixty seconds interval.

Use CWL-metrics to capture runtime metrics of RNA-Seq workflows

As an example use case to capture and analyze runtime metrics of workflows, I performed an analysis to optimize instance type selection for RNA-Seq quantification workflows. I run 7 RNA-Seq workflows (Table 3.1) for 9 public human RNA-Seq data with different read length and number of reads (Table 3.2) on 6 types of AWS Elastic Compute Cloud (EC2) service (Table 3.3) to capture the runtime metrics with CWL-metrics for each combination. Each workflow description has two different options for read layout: single-end and paired-end. For the selection of workflows, I chose two read mapping tools STAR and Hisat2, with two transcriptome assembly and read count programs Cufflinks and StringTie. I also used two popular tools using alignment-like algorithms, Kallisto and Salmon. TopHat2, the program which was once the most popular, but now obsolete, was added among them for comparing purpose. I performed metrics data collection five times for each combination of workflow, input data, and instance type.

Workflow name	Steps	CWL definition files
tophat2-cufflinks	download-sra, pfastq-dump, tophat2-mapping, cufflinks	https://github.com/pitagora-galaxy/cwl/tree/master/workflows/tophat2-cufflinks
hisat2-cufflinks	download-sra, pfastq-dump, hisat2-mapping, samtools_sam2bam, samtools_sort, cufflinks	https://github.com/pitagora-galaxy/cwl/tree/master/workflows/hisat2-cufflinks
hisat2-stringtie	download-sra, pfastq-dump, hisat2-mapping, samtools_sam2bam, samtools_sort, stringtie	https://github.com/pitagora-galaxy/cwl/tree/master/workflows/hisat2-stringtie
star-cufflinks	download-sra, pfastq-dump, star-mapping, samtools_sam2bam, samtools_sort, cufflinks	https://github.com/pitagora-galaxy/cwl/tree/master/workflows/star-cufflinks
star-stringtie	download-sra, pfastq-dump, star-mapping, samtools_sam2bam, samtools_sort, stringtie	https://github.com/pitagora-galaxy/cwl/tree/master/workflows/star-stringtie
kallisto	download-sra, pfastq-dump, kallisto-quant	https://github.com/pitagora-galaxy/cwl/tree/master/workflows/kallisto
salmon	download-sra, pfastq-dump, salmon-quant	https://github.com/pitagora-galaxy/cwl/tree/master/workflows/salmon

Table 3.1: The components of RNA-Seq quantification workflows

We described seven different RNA-Seq quantification workflows in CWL. Each workflow description has two different options for read layout, single-end and paired-end. We selected two major read mapping tools STAR and Hisat2, with two transcriptome assemble and read count programs Cufflinks and StringTie. We also used two popular tools using alignment-like algorithms, Kallisto and Salmon. We added TopHat2, one of the most popular but obsolete program for comparing purpose.

Table 3.4 shows the summary of runtime metrics, processing duration, and the calculated cost of instance usage per run for two workflows, HISAT2-Cufflinks and TopHat2-Cufflinks. The fastest processing time was one of the HISAT2-Cufflinks

workflow run on the c5.4xlarge instance, but the execution at the cheapest cost was the HISAT2-Cufflinks workflow on the c5.2xlarge instance. It indicates that workflows on cloud instances can have a trade-off of the processing time and the financial cost. The priority of the research project, the execution speed over the financial cost or vice versa, will be required for the final decision of instance selection optimization. The table also shows the possibility of loss of time or money when one failed to choose a proper instance type. For example, if one used the r5.4xlarge instance to run the HISAT2-cufflinks workflow, it is 7% slower than c5.4xlarge, and about 1.6 times expensive per sample. The impact of the instance type optimization failure will be more serious for the data processing jobs that take days or weeks.

Workflow name	Instance type	Workflow duration	Max CPU usage	Total amount of memory	Total amount of memory cache	Total amount of BlockIO	Cost per run
HISAT2-Cufflinks	c5.2xlarge	1014.5	796.8330796	10033995776	5183479808	4748816384	0.0958
HISAT2-Cufflinks	c5.4xlarge	778	1595.031529	9163902976	4314202112	1204879360	0.147
HISAT2-Cufflinks	m5.2xlarge	1013	799.0908131	11254398976	6396575744	1204858880	0.1081
HISAT2-Cufflinks	m5.4xlarge	846	1538.403444	11802640384	6938824704	331776	0.1805
HISAT2-Cufflinks	r5.2xlarge	1015	798.2115564	10912165888	6065545216	3608539136	0.1421
HISAT2-Cufflinks	r5.4xlarge	834	1588.403182	9973350400	5116166144	0	0.2335
TopHat2-Cufflinks	c5.2xlarge	5139	797.8534259	12310124544	8869050368	12343222272	0.4854
TopHat2-Cufflinks	c5.4xlarge	3695	1587.471528	15879102464	7833452544	1204891648	0.6979
TopHat2-Cufflinks	m5.2xlarge	5579	799.5529991	15149662208	9395200000	51970048	0.5951

TopHat2-Cufflinks	m5.4xlarge	3981	1595.226713	15875092480	7913992192	49848320	0.8493
TopHat2-Cufflinks	r5.2xlarge	5487	798.6095883	15152807936	9492783104	49848320	0.7682
TopHat2-Cufflinks	r5.4xlarge	4001	1291.353527	15877746688	7930822656	49848320	1.1203

Table 3.4: The runtime metrics comparison of TopHat2 and HISAT2

We summarized the runtime metrics values to compare two different workflows HISAT2-cufflinks and TopHat2-cufflinks. All runs are of input data SRR2567462. The read length was 100bp, the number of reads was 10,007,044.00, and the read layout was single-end. The shown values are workflow duration in seconds, the maximum CPU usage in percentage, the total amount of memory in bytes, the total amount of cache in bytes, the total amount of block IO in bytes, and the cost per run in USD. We calculated the median values for metrics values from the data of five times workflow iteration. Values can be zero for short-time steps since CWL-metrics collects these metrics with sixty seconds interval.

Figure 3.4 shows the results of processing duration of the HISAT2-StringTie workflow. There are clear differences of processing time between the samples, where the samples of the smaller number of reads have smaller differences between the instance types, but the runs on instance types with more CPU (4xlarge) marked shorter processing time with the samples of the larger number of reads. Each workflow runs used as many CPU cores as available on the environment; thus the difference can be considered as the difference of the number of threads. The read length and the

processing duration also have a strong linear relationship. This result will be useful to estimate the resource usage from the size of input data.

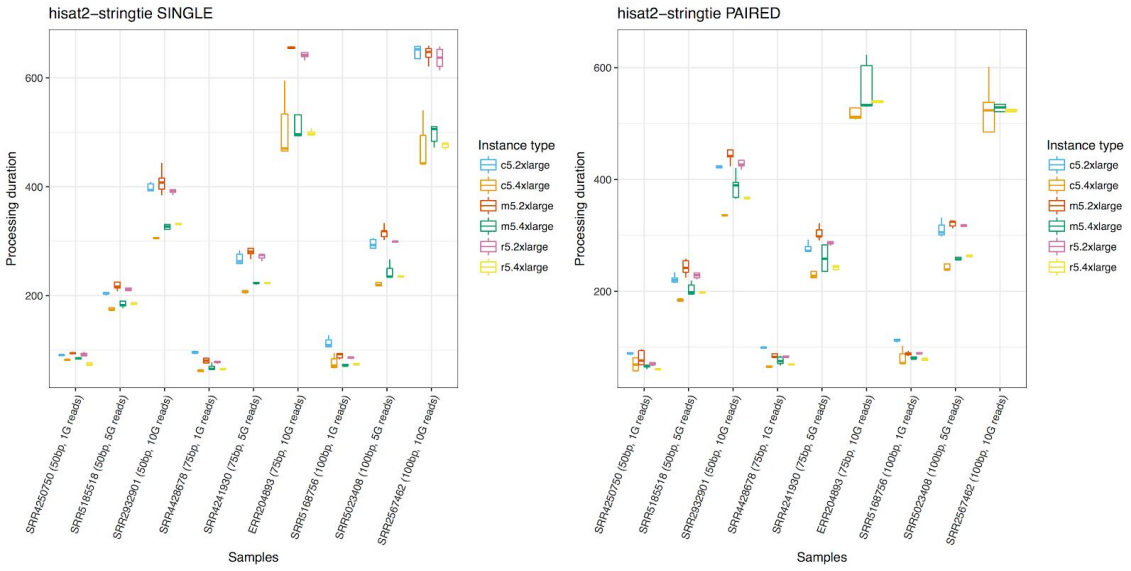


Figure 3.4: Box plot of per sample processing duration distribution of HISAT2-StringTie workflow

We plotted the values of processing duration of workflow runs excluding data download time. The x-axis shows SRA Run ID of samples used as input data with read length and number of reads. The y-axis shows the workflow processing duration in seconds. Values are separated and colored by the used instance type. Some runs on specific instance types are not in the plots because the failed executions are excluded. Each combination of sample and instance type were iterated five times to show the distribution of metrics. The plot shows that read length and the number of reads are both the factors that effect to the processing duration, and the differences between instance types are relatively small with the smaller number of reads (1G bases), while instances with more CPU cores (*.4xlarge) show shorter processing duration with 10GB reads.

On the other hand, the result of the comparison of the total amount of memory per input data in Figure 3.2 needs a different interpretation. Unlike HISAT2 and TopHat2, Kallisto and Salmon did not show clear differences in memory usage in different sizes of input data. The result indicates that the users need to know the behavior of the tool beforehand since the resource usage depends on the algorithms and the implementations.

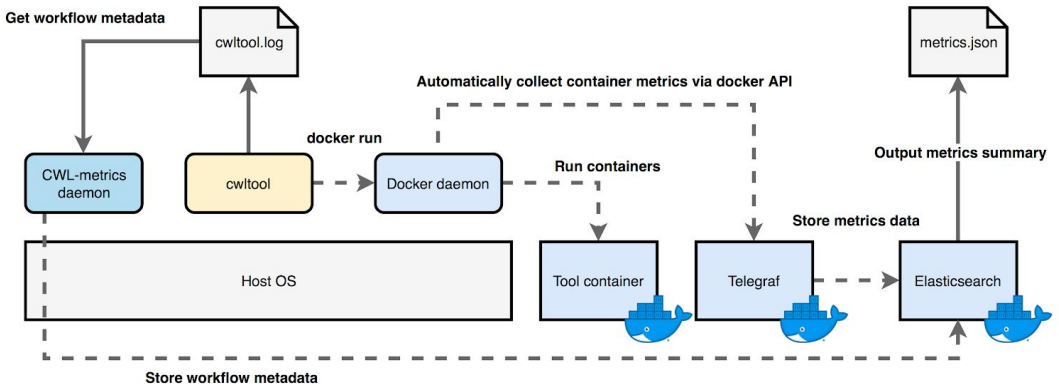


Figure 3.2: The CWL-metrics components and working process

CWL-metrics runs a daemon process and Docker containers on the host machine. The process and containers keep running until the system is terminated. Once a cwltool process starts running on the same machine, CWL-metrics system monitors the process

to get the list of workflow step containers and log files. Every sixty seconds, the Telegraf container try to access the Docker daemon to get runtime metrics of running containers. Fluentd container (not shown in the figure) sends runtime metrics data collected by Telegraf to the Elasticsearch container. CWL-metrics daemon process captures cwltool log file and sends workflow metadata to Elasticsearch.

The runtime metrics data provided by CWL-metrics also helps to perform a tool comparison. Figure 3.5 shows that the difference of processing time between the used workflows. Although users need to know the difference of the design concept and the strength of the tools to select the proper one for their research objectives, this result helps to understand the difference of the resource requirement of the workflows for similar purpose. For example, HISAT2 and STAR marked almost the same processing time, but STAR uses far more amount of memory. The plot of the processing time also shows that the obsolete tool TopHat2 is remarkably slower than the other tools.

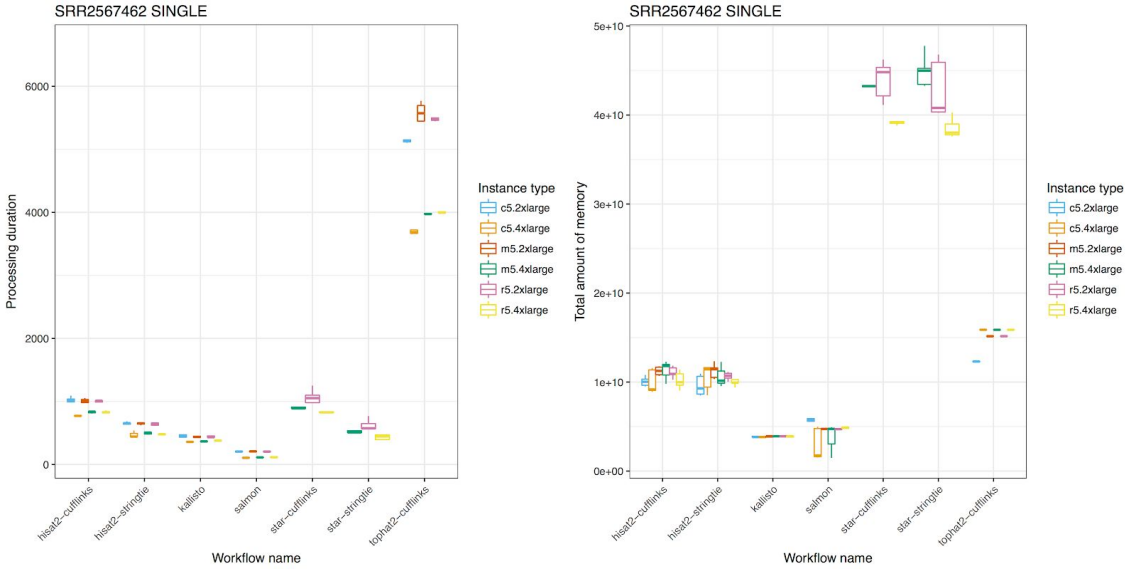


Figure 3.5: Box plot of processing duration and maximum memory usage of sample SRR2567462 per workflow

The values of processing duration were without data download time. Both plots used values of workflow executions as single end input of SRR2567462. The x-axis shows workflow names, and the y-axis shows the processing duration in seconds and total memory usage in bytes. We iterated each combination of workflow and instance type for five times. The plot of processing duration shows that there is a significant difference in execution time between the TopHat2 workflow and the others. While the difference of processing durations is relatively small, workflows with STAR aligner require four or five times much memory than HISAT2 workflows. These data suggest users know about runtime metrics of workflows before selecting cloud instance type.

Discussion

CWL-metrics enabled users to choose a proper cloud instance for workflow runs based on the runtime metrics data. The metrics data summarized by workflow inputs, such as the number of threads to use or total file size of input data, provides the most efficient cloud use for a research project. The data will also help the administrator of computational infrastructure to encourage researchers to use the cloud environment in case their local environment has too many running jobs to accept new job submissions. Each user might perform different analyses and visualizations concerning input parameters of their interest. Thus CWL-metrics outputs JSON and TSV data which are

easy to parse and used for visualization by any language of users' favorite, rather having a custom visualization tool other than Kibana.

CWL-metrics is applicable for most cases in bioinformatics data analysis. However, there are cases that the system does not work as effectively as expected. For example, the current implementation of CWL-metrics cannot capture the precise runtime metrics data of a tool that scatter its processes to multiple computation nodes. Also, it cannot estimate the performance of software that uses hardware acceleration systems such as GPU, since the information of those specific architectures is not available via Docker API. Nevertheless, in the example use case using RNA-Seq workflows, I showed CWL-metrics could provide beneficial information to help users to decide on the use of cloud infrastructure.

There are also the other workflow operation frameworks that have functions to capture runtime metrics, such as Galaxy [41], Toil [59], or Nextflow [60]. However, I chose CWL as the workflow description framework and its reference implementation cwltool as the workflow runner for the system because CWL is the project providing a way to share the workflow across the different workflow systems. Once users collected the runtime metrics of workflows by CWL-metrics, they can use the same workflow description with multiple workflow runner implementations. There are fifteen implementations listed as those supporting CWL [61]. Some implementations including Galaxy are still not covering full functions to import and export CWL description to share and run workflows, but the others including Arvados, Toil, and Apache Airflow are already available to users. If one wanted to use a workflow system that does not

support CWL yet, the summary of runtime metrics collected through Docker container is still valuable resource across the different frameworks.

CWL project has a subproject, CWLProv, to provide the provenance information of workflow executions to improve reproducibility of workflows by tracking intermediate files and logs [62]. The provenance information helps users to track inputs and outputs of workflow runs by using file checksum but does not record the detail of the resource usage. Adding runtime metrics data into the provenance information will cover the information regarding deployment, which helps users to reproduce the runs on a proper computing environment. Thus, the summary of runtime metrics collected by CWL-metrics should be bundled with the provenance information.

There will be more amount of sequencing data that one researcher needs to process by the technologies that produce a large amount of sequencing data such as high-throughput single-cell sequencing. In such a situation, it is essential to have a flexible computing environment that can quickly scale out according to the amount of data. The fast deployment of the data analysis environment to the proper cloud instance supported by Docker, CWL, and CWL-metrics is a way to achieve the computational scale out, which brings a huge benefit for bioinformatics researchers.

CHAPTER 4

DEVELOPMENT OF A DATABASE WITH TRANSPARENT DATA ANALYSIS PROCESS

Published Article 4

ChIP-Atlas: a data-mining suite powered by full integration of public ChIP-seq data.

Shinya Oki, Tazro Ohta, Go Shioi, Hideki Hatanaka, Osamu Ogasawara, Yoshihiro Okuda, Hideya Kawaji, Ryo Nakaki, Jun Sese, Chikara Meno.
EMBO reports. 2018 Dec 1;19(12):e46255.

.

Background

The methods to collect data from SRA and construct workflows to run on cloud infrastructure enabled large-scale reprocessing of published omics data at low cost. Users not having a large scale computational environment are also able to create a secondary database of their interest with reliable quality.

My collaborator and I developed a database named *ChIP-Atlas*, which provides results of re-analysis of ChIP-Seq and DNase-Seq data archived in SRA [63]. As of September 2018, the database has data of 78,000 experiments processed and made publicly available. The project performed reference alignment and peak calling to detect genomic locations determined as regulatory regions. The database shares the final output of its workflow such as bed format file from MACS2 peak calling software [64].

A large number of data of ChIP-Atlas makes it difficult to check the entire list to select data of interest. Each data has sample metadata manually curated, thus users use the information to summarize and select a dataset. However, it takes an unacceptably long time to download the dataset since the file size is huge. Therefore, a simple data sharing on the public server is not the best way to provide the results of re-analysis. Users need an interface to access the data fast and easily, and to visualize a part of data of their research interest.

To solve the problem to improve the accessibility of the large number of data in ChIP-Atlas, I developed a web application called chip-atlas.org [65]. It provides users a guide to select a subset from the available data, and connect to users' local genome browser to visualize genomic features. The web interface also has features to show details for each experiment with external data resources with calculated sequence quality.

Methods

Re-analyzed ChIP-Seq and DNase-Seq data available on ChIP-Atlas

ChIP-Atlas collects data from SRA with the condition that library strategy is "ChIP-Seq" or "DNase-Hypersensitivity", library source is "GENOMIC", and instrument model is from Illumina, Inc (San Diego, California, United States). As of September 2018, it has data from six organisms: 34390 samples of *Homo sapiens*, 31775 samples of *Mus musculus*, 729 samples of *Rattus norvegicus*, 3988 samples of *Drosophila melanogaster*, 2464 *Caenorhabditis elegans*, and 4809 samples of *Saccharomyces cerevisiae*. The data files are published on the DBArchive of the National Bioscience Database Center (NBDC) [66].

Implementing web application for dataset selection and browsing

As an interface to explore and visualize the data provided by ChIP-Atlas, I implemented the web application chip-atlas.org. It enables incremental search of sample

attributes by using curated metadata including antigen class, antigen, cell type class, and cell type. The web application was designed and implemented as to provide fast data selection feature without data processing or calculation, which enabled the lightweight application usage. The application has features such as dataset retrieval, visualization on local genome browser, and exploring details of samples and experiments with external data resources. I chose Integrative Genomics Viewer (IGV) as local genome browser, which users need to install on their local computers beforehand [67]. The program to launch the application is publicly available on GitHub [68]. The documentation of ChIP-Atlas data is also available on GitHub wiki [69].

Results

Obtaining subset of ChIP-Atlas database by using curated sample attributes

The metadata described by data submitters are manually curated using controlled vocabularies. There are many alternative names for antigens and cell types together with misspellings, efforts are taken to pick representatives (Table 4.1). The curated cell types are classified into cell type class by their origin tissues. The antigen used for immunoprecipitation are further classified to antigen class such as histone, RNA-polymerase, or transcription factors. The interface was designed as it uses the curated metadata as the main faceting for data selection (Figure 4.1). Users first need to select two attributes, antigen class and cell type class. The interface shows child elements of these classes, which users can choose one of them to filter the number of

the dataset. There is an option to select the threshold of quality value for each peak calculated by the MACS2 software. After the selection, users can choose to download the dataset as a single file or visualize the data on the local genome browser. Once users start the IGV genome browser on their local computer, they can browse data as fast as the data are on their computer by receiving data streaming from the ChIP-Atlas server, without downloading whole data files.

Original description in submitters' metadata	**#Exp**
H3K27me3	681
H3K27me3 (Millipore, 07-449)	88
H3K27me3 (Millipore 07-449) H3K27me3	81
anti-H3K27me3	75
H3K27me3(Diagnode, pAb-069-050)	44
K27me3	40
H3K27Me3	37
H3K27me3 (07-449, Millipore)	30
H3K27me3 (ActiveMotif,39155)	29
Millipore 07-449	28
Millipore, 07-449	26
H3K27me3 (Abcam ab6002)	23
H3K27me3 (Active Motif, 39155)	22
H3K27me3 (Active Motif 39155)	21

Table 4.1: The list of user described metadata terms curated to the term "H3K27me3"

An example of term curation of user-described terms with experiments categorized to H3K27me3 (n > 20). Lacking a standard of metadata description caused having many variations to specify the same antigen or cell line name in the same or related sample attribute fields. In this example, variations include different character cases, comma existence, catalog name or identifiers of reagent companies. The experiments with these terms are categorized to those of antigen "H3K27me3" based on the Brno nomenclature.

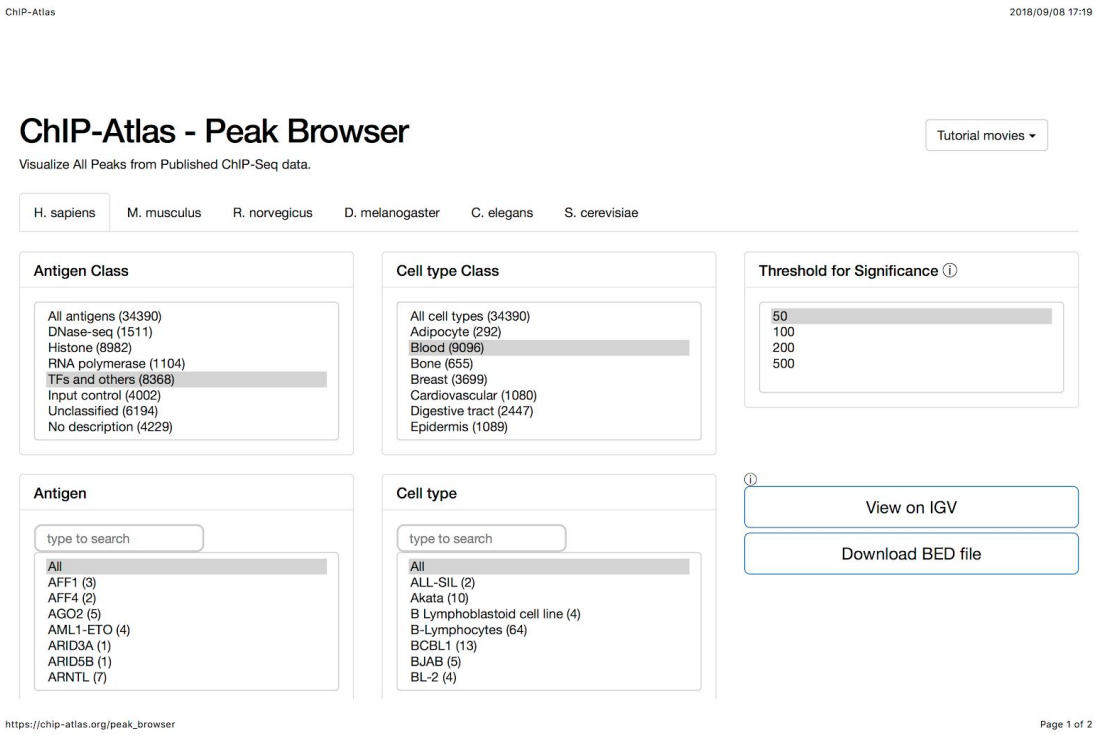


Figure 4.1: ChIP-Atlas data search interface

A screenshot of ChIP-Atlas data search interface (peak browser). Users first choose an organism of their interests, then select an antigen class and a cell type class from the select boxes. An antigen or a cell type is able to be chosen for the further data filtering. Options for peak call score threshold are available on the right-upper panel to reduce the number of peaks on the genome browser. Clicking "View on IGV" will fetch the data from the data server to the local machine to browse on the IGV genome browser. Users will query a gene or a genome location on the browser to see the region to find transcription factor binding sites or the other biological features.

Browse experimental details integrated with external data resources

Using the genome browser to check genomic locations of interest, users will find peaks having interesting aspects. Users can check the experimental information that produced the browsing peak by mouseover. The peaks showed on the genome browser have hyperlinks which get users back to the browser web application to show the experimental details (Figure 4.2). The page of individual experiment shows original and curated sample attributes, links to WikiGenes [70], PosMed [71], and PDBj [72] with query of the antigen, links to ATCC [73], MeSH [74], RIKEN BRC [75] with query of the cell type (Figure 4.2). There are also data processing information such as the number of reads, mapping ratio, duplicated reads removed, and number of peaks. The page also shows the base call accuracy of the data retrieved from the sequence quality database [53].

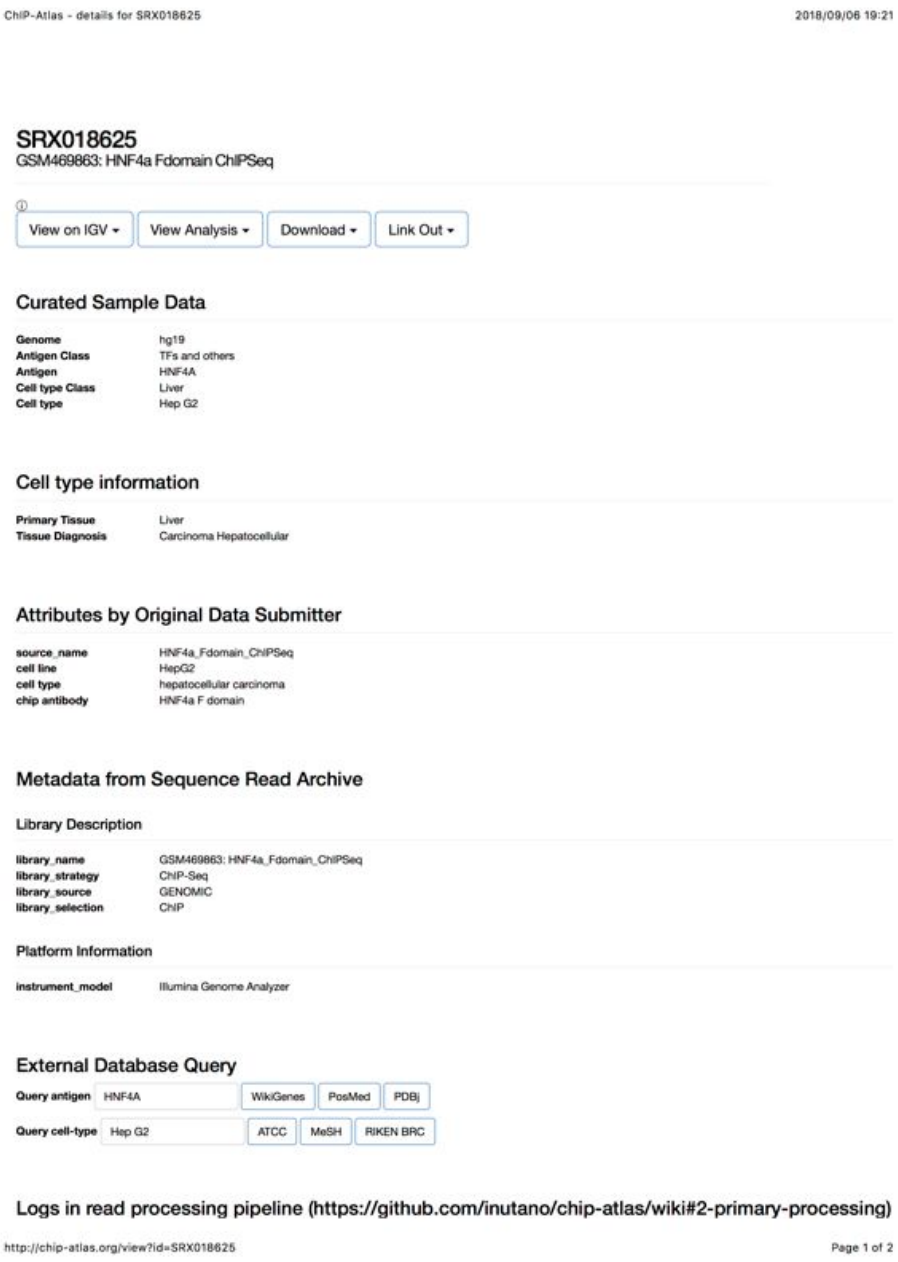


Figure 4.2: The information for each experiment with links to external databases

An example of detailed information of experiment ID SRX018625. The page contains curated sample data, cell type information, attributes by original data submitter, metadata submitted to SRA, read processing pipeline logs, and base call quality of

original sequence data. The page also offers hyperlink to external database, WikiGenes, PosMed, and PDBj for antigen, ATCC, MeSH, RIKEN BRC for cell type.

Discussion

The interface built based on the curated sample attributes helps users to explore data provided by the ChIP-Atlas database where more than 70,000 experiments from SRA were re-analyzed. While there are other omics databases available online, and some of them are providing similar dataset, most of the databases have their front pages with single keyword search box. For example, the UCSC genome browser, one of the most popular genome data browsers where the data from the ENCODE project published, has a genome browser oriented interface which users issue queries of genes or genomic locations to move to the location, then clicking list of data to show items on a new track [76]. This type of browser makes the application interface having too many list items to browse manually when the database has many types of data, like ChIP-Atlas. To keep the interface simple as the users who visit for the first time can understand what they need to do to find the data, chip-atlas.org was designed as it separates application to the web interface for data browsing and the genome browser running locally. By this decision, the web application becomes easy to use, and also is able to allow uploading local data to compare with the ChIP-Atlas data. The chip-atlas.org web application has been already used by many projects [77].

However, there is a problem on the sustainability of the application since it depends on the attributes manually curated. As of September 2018, the ChIP-Atlas updates its data monthly, but the most of the updating process is totally depending on the curation team, while the other procedures are already automated. There is an ongoing project which evaluates a machine-learning approach to automate the curation effort, but the approach has a problem that it cannot curate the new terms which the system does not know. Thus, for the time being, the curation effort is not fully automated, but the team is using an assistant software to support finding related terms.

As the automation of term curation, it is also important to have a system that promotes data submitters to follow the guidelines to describe proper metadata. DDBJ is developing the system called BioSample validator, which helps users to check data type or avoid misspelling [78]. The system can help a database to have the data with high quality of metadata description that reduce the cost of metadata curation by the third party.

CHAPTER 5

CONCLUSIONS

In this research, I developed the methods to describe precise information of data processing including sample data and software workflow. The metadata of samples and experiment submitted to SRA with sequence data lack description standards to control its quality. There are various factors such as an experimental reagent, sequencing instrument, or software version for base calling that changes sequencing quality. Researchers submit their sequencing data of different read length, number of reads, or base call accuracy. I developed a search system that integrates sample metadata with the external resources such as information extracted from related publications, and a database that calculates and collects sequencing quality statistics per entry. These systems and methods help users to select dataset that has precise information which are required for the evaluation of results of data analysis.

Describing data analysis workflow is also essential for evaluation of outputs from data analysis. The tools like Docker container or CWL can help to describe the information, however, runtime metrics information is also important for reproducibility of the data analysis. To add runtime information to the description of data analysis, I developed CWL-metrics, a system to analyze runtime metrics of workflows. Using this system, users can select the best possible option by analyzing the relationship between input parameters for workflow and resource usages such as CPU, memory, or disk IO.

The analysis result also helps users to estimate the amount of charge for the use of the cloud service.

The methods are developed according to the needs from problems that are unique to biological big data. The nucleotide sequence data after the appearance of the HTSeq technologies often are referred to as representative of big data in life science [79]. However, nucleotide sequence data has different characteristics from those of big data in commercial industries (Table 5.1). The representatives from industrial big data such as log data of web servers or messages on social networking services have much more entries of text, audio or images of the relatively small data size per entry, which are required to be analyzed in real time. The "three Vs", volume, velocity, and variety, which are called as the main characteristics of big data [80], do not suit biological big data. Thus, the methods to solve the problem on knowledge extraction from the big data in biology must be unique in comparison with those of the industries.

	Web server	SNS	Machine learning	Literature	Nucleotide sequencing
Data size per entry	+	+	+ ～ ++	++	+++
Processing time per entry	+	+	+ ～ +++	+	+++
Number of entries	+++	+++	+ ～ +++	++	+++
Variation of data production agents	+	+	+++	+	+++
Data quality distribution	-	-	+++	-	+++
Data production time interval	+	+	-	-	++
Real time analysis requirement	+	+	+	-	-
Variation of data analysis purposes	+	++	+++	++	+++

Human curation requirement	-	+	+	+	+
Data analysis repeat requirement	-	-	+	+	+++
Data sharing requirement	-	-	-	-	+

Table 5.1: The characteristics of big data among different domains

Comparison of so-called big data of 5 different domains, web server access log data (Web server), messages and user relationships in social network services (SNS), training data for machine learning techniques (Machine learning), text mining and natural language processing of literature data (Literature), and nucleotide sequencing data in biomedical sciences (Nucleotide sequencing). Though there are many methods, tools, or frameworks for "big data", the one in the life science makes itself unique with its characteristics which has many entries with large data size and several requirements in quality control and manual curation, while it does not require real-time processing that is said to be an important part of the definition of industrial big data.

I presented the problem that data analysis lacking information of input and process may cause the inappropriate interpretation of the result. In this research, I demonstrated the method to avoid the situation causes such misinterpretation of an output of data analysis with two different approaches, the extension of input metadata and the use of the frameworks to package software runtime information. Though I developed the methods and the frameworks for the nucleotide sequencing data, the approach to remove ambiguity and uncertainty from data analysis process is also typical for data analysis applications in different scientific domains. For example, scientific

research projects in different domains such as astrophysics also have problems in sharing the observation data in terms of reproducibility [81]. This indicates that sharing research data is to face the difference in the way of scientific studies between different research teams, which often needs many communications between the people having different practice on data management. The proposed approaches can help researchers to share their data without misinterpretation by providing further information that often dropped during the data exchange process. It is very important to find a common practice shared by different scientific domains for achieving a comprehensive exchange of data analysis process. I believe this study is the first step to solve this problem for future studies.

REFERENCES

1. Strasser BJ. GenBank--Natural History in the 21st Century?. Science. 2008 Oct 24;322(5901):537-8.
2. Benson D, Lipman DJ, Ostell J. GenBank. Nucleic acids research. 1993 Jul 1;21(13):2963-5.
3. Tateno Y, Gojobori T. DNA Data Bank of Japan in the age of information biology. Nucleic acids research. 1997 Jan 1;25(1):14-7.
4. Cochrane G, Karsch-Mizrachi I, Nakamura Y, International Nucleotide Sequence Database Collaboration. The international nucleotide sequence database collaboration. Nucleic acids research. 2010 Nov 23;39(suppl_1):D15-8.
5. Wilbur WJ, Lipman DJ. Rapid similarity searches of nucleic acid and protein data banks. Proceedings of the National Academy of Sciences. 1983 Feb 1;80(3):726-30.
6. Mardis ER. The impact of next-generation sequencing technology on genetics. Trends in genetics. 2008 Mar 1;24(3):133-41.
7. Leinonen R, Sugawara H, Shumway M, International Nucleotide Sequence Database Collaboration. The sequence read archive. Nucleic acids research. 2010 Nov 8;39(suppl_1):D19-21.
8. Cock PJ, Fields CJ, Goto N, Heuer ML, Rice PM. The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants. Nucleic acids research. 2009 Dec 16;38(6):1767-71.
9. SRA database growth. https://www.ncbi.nlm.nih.gov/sra/docs/sragrowth Accessed 26 September 2018

10. GB Editorial Team. Closure of the NCBI SRA and implications for the long-term future of genomics data storage. Genome biology. 2011
11. Kodama Y, Shumway M, Leinonen R. The Sequence Read Archive: explosive growth of sequencing data. Nucleic acids research. 2011 Oct 18;40(D1):D54-6.
12. Organisation for Economic Co-operation and Development. OECD Principles and Guidelines for Access to Research Data from Public Funding. OECD; Paris. 2007. http://www.oecd.org/science/sci-tech/38500813.pdf. Accessed 11 Nov 2016.
13. Sansone SA, Rocca-Serra P, Field D, Maguire E, Taylor C, Hofmann O, Fang H, Neumann S, Tong W, Amaral-Zettler L, Begley K. Toward interoperable bioscience data. Nature genetics. 2012 Feb 1;44(2):121-6
14. Ball CA, Sherlock G, Brazma A. Funding high-throughput data sharing. Nature biotechnology. 2004 Sep 1;22(9):1179-83.
15. Nakazato T, Ohta T, Bono H. Experimental design-based functional mining and characterization of high-throughput sequencing data in the sequence read archive. PLoS One. 2013 Oct 22;8(10):e77910.
16. NCBI GEO FTP ftp://ftp.ncbi.nlm.nih.gov/pub/geo/ Accessed 26 September 2018
17. Entrez Programming Utilities Help https://www.ncbi.nlm.nih.gov/books/NBK25500 Accessed 26 September 2018
18. NCBI SRA FTP ftp://ftp.ncbi.nlm.nih.gov/pub/sra Accessed 26 September 2018
19. Groonga: an open-source fulltext search engine and column store http://groonga.org Accessed 26 September 2018

20. Sequence Read Archive Metadata Search https://github.com/inutano/soylatte Accessed 26 September 2018
21. DDBJ DRA FTP ftp://ftp.ddbj.nig.ac.jp/ddbj_database/dra Accessed 26 September 2018
22. Andrews S. A quality control tool for high throughput sequence data. 2010. http://www.bioinformatics.babraham.ac.uk/projects/fastqc/. Accessed 26 September 2018
23. Goto N, Prins P, Nakao M, Bonnal R, Aerts J, Katayama T. BioRuby: bioinformatics software for the Ruby programming language. Bioinformatics. 2010 Oct 15;26(20):2617-9.
24. Ohta T. Ruby parser for FastQC, a quality control software for high-throughput sequencing data. https://rubygems.org/gems/bio-fastqc Accessed 26 September 2018
25. Bonnal RJ, Aerts J, Githinji G, Goto N, MacLean D, Miller CA, Mishima H, Pagani M, Ramirez-Gonzalez R, Smant G, Strozzi F. Biogem: an effective tool-based approach for scaling up open source software development in bioinformatics. Bioinformatics. 2012 Apr 1;28(7):1035-7.
26. Ohta T, Summary of quantitative sequence information of the Sequence Read Archive. https://github.com/inutano/sra-quanto. Accessed 26 September 2018
27. DBCLS SRA http://sra.dbcls.jp/. Accessed 26 September 2018
28. NBDC RDF Portal https://integbio.jp/rdf/. Accessed 26 September 2018
29. R Core Team. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. 2015. https://www.R-project.org/. Accessed 26 September 2018

30. Wickham H. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York; 2009.
31. SRA_Accessions.tab ftp://ftp.ncbi.nlm.nih.gov/sra/reports/Metadata/SRA_Accessions.tab Accessed 26 September 2018
32. Mashima J, Kodama Y, Kosuge T, Fujisawa T, Katayama T, Nagasaki H, Okuda Y, Kaminuma E, Ogasawara O, Okubo K, Nakamura Y. DNA data bank of Japan (DDBJ) progress report. Nucleic acids research. 2015 Nov 17;44(D1):D51-7.
33. Barrett T, Clark K, Gevorgyan R, Gorelenkov V, Gribov E, Karsch-Mizrachi I, Kimelman M, Pruitt KD, Resenchuk S, Tatusova T, Yaschenko E. BioProject and BioSample databases at NCBI: facilitating capture and organization of metadata. Nucleic acids research. 2012 Jan 1;40(D1):D57-63.
34. Goodwin S, McPherson JD, McCombie WR. Coming of age: ten years of next-generation sequencing technologies. Nature reviews genetics. 2016 Jun 1;17(6):333-51.
35. NCBI BioSample Packages https://www.ncbi.nlm.nih.gov/biosample/docs/packages Accessed 26 September 2018
36. Alnasir J, Shanahan HP. Investigation into the annotation of protocol sequencing steps in the sequence read archive. GigaScience. 2015 May 9;4(1):23.
37. Chang J. Core services: Reward bioinformaticians. Nature 2015;520:151–2.
38. Prins P, de Ligt J, Tarasov A et al. Toward effective software solutions for big biology. Nature biotechnology 2015;33:686–7.

39. Merkel D. Docker: lightweight linux containers for consistent development and deployment. Linux journal. 2014 Mar 1;2014(239):2.
40. Di Tommaso P, Palumbo E, Chatzou M, Prieto P, Heuer ML, Notredame C. The impact of Docker containers on the performance of genomic pipelines. PeerJ. 2015 Sep 24;3:e1273.
41. Afgan E, Baker D, Batut B, Van Den Beek M, Bouvier D, Čech M, Chilton J, Clements D, Coraor N, Grüning BA, Guerler A. The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2018 update. Nucleic acids research. 2018 May 22;46(W1):W537-44.
42. Amstutz P, Crusoe MR, Tijanić N, Chapman B, Chilton J, Heuer M, Kartashov A, Leehr D, Ménager H, Nedeljkovich M, Scales M. Common workflow language, v1. 0. DOI: 10.6084/m9.figshare.3115156.v2.
43. Stein LD. The case for cloud computing in genome informatics. Genome biology 2010;11:207.
44. Amazon EC2 Instance Types https://aws.amazon.com/ec2/instance-types/ Accessed 30 Oct 2018.
45. Telegraf https://www.influxdata.com/time-series-platform/telegraf/ Accessed 30 Oct 2018.
46. Fluentd https://www.fluentd.org/ Accessed 30 Oct 2018.
47. Elasticsearch https://www.elastic.co/products/elasticsearch Accessed 30 Oct 2018.
48. Kibana https://www.elastic.co/products/kibana Accessed 30 Oct 2018.
49. CWL-metrics https://inutano.github.io/cwl-metrics/ Accessed 30 Oct 2018.

50. da Veiga Leprevost F, Grüning BA, Alves Aflitos S, Röst HL, Uszkoreit J, Barsnes H, Vaudel M, Moreno P, Gatto L, Weber J, Bai M. BioContainers: an open-source and community-driven framework for software standardization. Bioinformatics. 2017 Mar 30;33(16):2580-2.
51. QUAY - inutano https://quay.io/user/inutano Accessed 30 Oct 2018.
52. pitagora-galaxy/cwl https://github.com/pitagora-galaxy/cwl Accessed 30 Oct 2018.
53. Ohta T, Nakazato T, Bono H. Calculating the quality of public high-throughput sequencing data to obtain a suitable subset for reanalysis from the Sequence Read Archive. GigaScience 2017;6, DOI: 10.1093/gigascience/gix029.
54. Casper J, Zweig AS, Villarreal C, Tyner C, Speir ML, Rosenbloom KR, Raney BJ, Lee CM, Lee BT, Karolchik D, Hinrichs AS. The UCSC genome browser database: 2018 update. Nucleic acids research. 2017;46(D1):D762-9.
55. Harrow J, Frankish A, Gonzalez JM, Tapanari E, Diekhans M, Kokocinski F, Aken BL, Barrell D, Zadissa A, Searle S, Barnes I. GENCODE: the reference human genome annotation for The ENCODE Project. Genome research. 2012 Sep 1;22(9):1760-74.
56. Project Jupyter http://jupyter.org Accessed 30 Oct 2018.
57. inutano/cwl-metrics-manuscript https://github.com/inutano/cwl-metrics-manuscript Accessed 30 Oct 2018.
58. common-workflow-language/cwltool https://github.com/common-workflow-language/cwltool Accessed 30 Oct 2018.

59. Toil: A scalable, efficient, cross-platform pipeline management system written entirely in Python and designed around the principles of functional programming. http://toil.ucsc-cgl.org/ Accessed 30 Oct 2018.
60. Di Tommaso P, Chatzou M, Floden EW, Barja PP, Palumbo E, Notredame C. Nextflow enables reproducible computational workflows. Nature biotechnology. 2017 Apr 11;35(4):316.
61. Common Workflow Language https://www.commonwl.org/ Accessed on 30 Oct 2018.
62. Khan FZ, Soiland-Reyes S, Sinnott RO, Lonie A, Crusoe MR. CWLProv–Interoperable retrospective provenance capture and its challenges. F1000Research. 2018 Jun 26;7.
63. Oki S, Ohta T, Shioi G, Hatanaka H, Ogasawara O, Okuda Y, Kawaji H, Nakaki R, Sese J, Meno C. ChIP‐Atlas: a data‐mining suite powered by full integration of public ChIP‐seq data. EMBO reports. 2018 Dec 1;19(12):e46255.
64. Zhang Y, Liu T, Meyer CA, Eeckhoute J, Johnson DS, Bernstein BE, Nusbaum C, Myers RM, Brown M, Li W, Liu XS. Model-based analysis of ChIP-Seq (MACS). Genome biology. 2008 Nov;9(9):R137.
65. ChIP-Atlas https://chip-atlas.org Accessed 26 September 2018
66. LSDB Archive - ChIP-Atlas http://doi.org/10.18908/lsdba.nbdc01558-000.V013
67. Thorvaldsdóttir H, Robinson JT, Mesirov JP. Integrative Genomics Viewer (IGV): high-performance genomics data visualization and exploration. Briefings in bioinformatics. 2013 Mar 1;14(2):178-92.
68. ChIP-Atlas https://github.com/inutano/chip-atlas Accessed 26 September 2018

69. ChIP-Atlas wiki https://github.com/inutano/chip-atlas/wiki Accessed 26 September 2018
70. Hoffmann R. A wiki for the life sciences where authorship matters. Nature genetics. 2008 Sep;40(9):1047.
71. Yoshida Y, Makita Y, Heida N, Asano S, Matsushima A, Ishii M, Mochizuki Y, Masuya H, Wakana S, Kobayashi N, Toyoda T. PosMed (Positional Medline): prioritizing genes with an artificial neural network comprising medical documents to accelerate positional cloning. Nucleic acids research. 2009 May 25;37(suppl_2):W147-52.
72. Kinjo AR, Suzuki H, Yamashita R, Ikegawa Y, Kudou T, Igarashi R, Kengaku Y, Cho H, Standley DM, Nakagawa A, Nakamura H. Protein Data Bank Japan (PDBj): maintaining a structural data archive and resource description framework format. Nucleic acids research. 2012 Jan 1;40(D1):D453-60.
73. ATCC https://www.atcc.org Accessed 26 September 2018
74. Lipscomb CE. Medical subject headings (MeSH). Bulletin of the medical library association. 2000 Jul;88(3):265.
75. RIKEN BRC CELL BANK http://cellbank.brc.riken.jp/en/ Accessed 26 September 2018
76. Karolchik D, Baertsch R, Diekhans M, Furey TS, Hinrichs A, Lu YT, Roskin KM, Schwartz M, Sugnet CW, Thomas DJ, Weber RJ. The UCSC genome browser database. Nucleic acids research. 2003 Jan 1;31(1):51-4.
77. ChIP-Atlas Publication http://chip-atlas.org/publications Accessed 26 September 2018

78. Validator implementation to the DDBJ BioSample submission system (16th, March, 2018) https://www.ddbj.nig.ac.jp/news/en/180314-e.html Accessed 26 September 2018

79. Marx V. Biology: The big challenges of big data. Nature. 2013 Jun 12;498:255-260

80. Zikopoulos P, Eaton C. Understanding big data: Analytics for enterprise class hadoop and streaming data. McGraw-Hill osborne media; 2011 Oct 19.

81. Pepe A, Goodman A, Muench A, Crosas M, Erdmann C. How Do Astronomers Share Data? Reliability and Persistence of Datasets Linked in AAS Publications and a Qualitative Study of Data Practices among US Astronomers. PLoS ONE 2014; 9: e104798.