Managing the analysis of high-throughput sequencing data

Javier Quilez[1,2*], Enrique Vidal[1,2], François Le Dily[1,2], François Serra[1,2,3], Yasmina Cuartero[1,2,3], Ralph Stadhouders[1,2], Guillaume Filion[1,2], Thomas Graf[1,2], Marc A. Marti-Renom[1,2,3,4] and Miguel Beato[1,2]

*Correspondence: javier.quilez@crg.eu

[1]Gene Regulation, Stem Cells and Cancer Program, Centre for Genomic Regulation (CRG), The Barcelona Institute of Science and Technology (BIST), Dr. Aiguader 88, 08003 Barcelona, Spain

[2]Universitat Pompeu Fabra (UPF), Barcelona, Spain

Full list of author information is available at the end of the article

Abstract

In the last decade we have witnessed a tremendous rise in sequencing throughput as well as an increasing number of genomic assays based on high-throughput sequencing (HTS). As a result, the management and analysis of the growing amount of sequencing data present several challenges with consequences for the cost, the quality and the reproducibility of research. Most common issues include poor description and ambiguous identification of samples, lack of a systematic data organization, absence of automated analysis pipelines and lack of tools aiding the interpretation of the results. To address these problems we suggest to structure HTS data management by automating the quality control of the raw data, establishing metadata collection and sample identification systems and organizing the HTS data with an human-friendly hierarchy. We insist on reducing metadata field entries to multiple choices instead of free text, and on implementing a future-proof organization of the data on storage. These actions further enable the automation of the analysis and the deployment of web applications to facilitate data interpretation. Finally, a comprehensive documentation of the procedures applied to HTS data is fundamental for reproducibility. To illustrate how these recommendations can be

implemented we present a didactic dataset. This work seeks to clearly define a set of best-practices for managing the analysis of HTS data and provides a quick start guide for implementing them into any sequencing project.

Keywords: high-throughput sequencing; management and analysis best practices; bioinformatics

Background

DNA sequencing is the process of reading the nucleotides that comprise a DNA molecule. In the last decade we have witnessed a tremendous increase in sequencing throughput coupled with dropping costs per base [1], as well as the development of an increasing number of genomic assays based on HTS (see [2] for a comprehensive list). As a result, sequencing experiments have become cost-effective and faster to perform and they have countless applications, explaining why HTS is being rapidly adopted in the life sciences, from basic research to medical applications.

This causes the accelerated accumulation of sequencing datasets, numbering in the thousands. For instance, the number of sequences deposited in the Sequence Read Archive (SRA) [3], a major repository for HTS data, has skyrocketed from ~2 Terabases in 2009 to ~9,000 Terabases (the size of approximately 3 million human genomes) at the beginning of 2017 (**Additional file 1: Fig. S1)**. Moreover, this is surely an underestimation of the actual amount given that only sequencing experiments eventually included in a publication are deposited. Although data-intensive projects like TCGA [4], 1000 Genomes Project [5] and ENCODE [6] are the top HTS data generators [7], such a boost in the number of existing sequences reflects a pervasive use of HTS. For instance, while sequencing data for >90,000 studies have been submitted to the SRA, the top 10 and 100 contributors in terms of number of bases represent ~30% and ~60% of the archive, respectively (**Additional file 1: Fig. S2a**). Similarly, while ~80% of SRA data derive from *Homo sapiens* and *Mus musculus*, the central organisms in large sequencing projects, the remaining 20% come from a diverse number of organisms (~50,000) (**Additional file 1: Fig. S2b**).

Managing and analyzing the growing amount of sequencing data presents several challenges (**Table 1**). Some of these are not new or exclusive to HTS data (e.g. metadata availability), but their impact are

nonetheless exacerbated by the pervasiveness of HTS experiments. For instance, working groups that are relatively small and/or have a limited computational infrastructure are more prone to suffer from them. In contrast to large-scale data-intensive projects, which are more likely to allocate resources to anticipate, avoid and fix such issues; for instance, the 4DNucleome Project has established formal working groups employing tens of scientists responsible for the data standards and analysis protocols [8]. Similarly, in cross-sectional population studies, samples are normally collected at the same time and analyzed jointly, which may make more obvious the need to define sample naming schemes and to systematically collect the metadata that will be later required in the analysis. Conversely, in most research groups, sequencing experiments are performed independently by several people, accumulate over longer periods of time and are not initially meant to be analyzed together.

Here we first describe the challenges associated to the rapid accumulation of HTS data; we note that these issues are not trivial and may have a negative impact in terms of money, time and scientific quality/reproducibility. We then present recommendations for mitigating these issues and aiding in the management and analysis of HTS data. To illustrate how these recommendations can be readily implemented we accompany them with a didactic dataset including 7 HTS samples (2 RNA-seq, 1 ChIP-seq and 4 Hi-C samples). Specifically, for each sample we provide FASTQ files with 1,000 reads for the single-end sample (ChIP-seq) and 2x1,000 reads for paired-end samples (RNA-seq and Hi-C). The didactic dataset as well as an accessory suite of scripts are available at https://github.com/4DGenome/conseq.

Challenges and considerations

Mislabelled raw sequencing data

In HTS, the molecules of DNA in a sample are used as a template to generate millions of sequence reads of the original molecules. Because of the high yield of current sequencing technologies (in the order of tens to hundreds of gigabase pairs) [1], unless extremely high coverage is needed, a common practice is loading multiple biological samples into the sequencing instrument; the DNA from each sample is

labeled with a different sequence index so that the reads generated can be assigned to the biological sample from which they originate (**Additional file 1: Fig. S3**).

As a result of sequencing errors, a relatively small proportion of the reads of a sequencing run will contain none of the indexes used to label the different DNA samples included in the run. On the other hand, a relatively high number of such undetermined reads can be indicative of problems during the library preparation or hint a wrong relationship between the biological sample and the sequencing index, which may result in lower sequencing yields and/or interpretation errors in the downstream analysis. Detecting such anomalies when extracting the sequencing reads from each sample is necessary to recover reads that would be overlooked otherwise. Early detection of indexing problems can save a considerable amount of time and computing resources, as compared to finding out later after aligning the reads onto the genome sequence – a time consuming and computationally demanding step. Therefore, we recommend automatically inspecting the number of unassigned reads as well as comparing the expected and observed index sequences (**Table 1**).

Poor sample description

FASTQ files should not be the only data making a HTS experiment. Metadata describe and provide information about the sequenced DNA sample (**Additional file 1: Fig. S4**). Metadata are important in many ways. For one, some metadata values are required for the initial processing of the data. For instance, the species is needed in order to align the reads to the corresponding reference genome sequence, and the restriction enzyme applied in the Hi-C experimental protocol is used in the computational analysis. Other metadata are used for quality control (e.g. sequencing facility and/or date for detecting batch effects or rescuing swapped samples using the correct index) or in the downstream analysis (e.g. cell type, treatment). Finally, some information is necessary to reproduce the experiment and analysis, a critical point considering the relatively high personnel turnover in academia and the concerns about reproducibility in biomedical research [9, 10]. Not surprisingly, submitting HTS data and the associated metadata to public repositories (e.g. GEO, ENA, SRA) when submitting a manuscript or

before publication is required by a growing number of journals and should be done even when not required. The commitment to meet the standards of international data repositories is an efficient way to rise the quality of the metadata. In addition, data is typically more visible, better organized and better backed up on a data server than on the storage of a single laboratory. Beyond publication, metadata are also important internally when sharing data with researchers involved in a project. Despite their importance we have observed that very often metadata are scattered, inaccurate, insufficient or even missing, and that there is a decay in the availability of metadata for older sequencing samples. Factors contributing to this situation include (i) disconnection between the experiment and the analysis (in other words, the experimenter may not be aware of the information needed for the analysis), (ii) short-term view of the experiment (performed just for a specific ongoing project without considering its potential future use), (iii) the initial investment required for establishing a metadata collection system as well as the subsequent inertia of filling the form, and (iv) high turnover of people. Altogether, this results in a poor description of sequencing samples and can affect performance (**Table 1**).

Therefore, we propose collecting the metadata of a HTS experiment at some point between the preparation of the experiment and the beginning of the data processing. A good metadata collection system should be (i) relatively short and easy to complete by the experimenters themselves, (ii) instantly accessible by authorized members of the project and (iii) easy to parse for a human as for a computer. We implemented a system consisting of a Google Form with which metadata is filled in sample-wise and automatically transferred to a Google Spreadsheet as well as to a SQL database stored in a computing cluster (**Fig. 1a**, **Additional file 1: Fig. S5** and Didactic dataset). We are aware that the use of third-party software for collecting metadata may not be an option for projects sensitive with data privacy issues, especially those collecting data from human subjects. In **Additional file 2: Table S1** we propose several features required to achieve the properties mentioned above. As an example, the metadata collected for our Didactic dataset are described in **Additional file 2: Table S2**. Foreseeing the information that may be needed is key and we advise defining early in the project the metadata fields that will be collected;

otherwise, continuous addition of metadata fields will add substantial amount of work, as metadata values will need to be searched retrospectively, or result in patchy metadata databases.

Unsystematic sample naming

Very often sequencing samples and the associated files (e.g. FASTQ files) are identified with names that describe the HTS experiment or that are familiar and easy to remember for the person who performed it. However, this practice has several undesired consequences. Identical or similar identifiers referring to different HTS experiments as well as vague sample names, especially if there is no associated metadata, can preclude any analysis or lead to errors. Moreover, such unsystematic sample naming undermines the capability to process samples programmatically (e.g. search for data, parallelize scripts), which impairs automation and may also lead to errors.

We therefore strongly recommend establishing a scheme to uniquely identify sequencing samples (**Table 1**). A first approach we have explored is defining a unique sample identifier (ID) that encapsulates some recognizable information (e.g. user, HTS application) (**Fig. 2a**). This approach has at least two caveats: first, sample IDs are assigned manually, which is suboptimal unless the number of samples is small; second, from our experience, determining whether samples are biological or technical replicates may not be straightforward from the interpretation of the metadata. With these caveats in mind we also sought to generate sample IDs in an automated manner and based on a selection of fields from the metadata that uniquely points to a sequencing experiment (**Fig. 2b**). Despite the apparent non-informativeness of this sample ID approach, it easily allows identifying biological replicates and samples sequenced in the same batch since they will share, respectively, the first and second 9-mer. While the specific fields used to generate the sample ID can vary, it is important that they unambiguously define a sequencing sample (otherwise duplicated identifiers can emerge) and that they are always combined in the same order to ensure reproducibility. Indeed, another advantage of this naming scheme is that the integrity of the metadata can be checked, as altered metadata values will lead to a different sample ID. Whatever the choice of the sample ID scheme is, in addition to the critical feature of being unique, we

recommend trying to make it as computer-friendly as possible by using a fixed length and pattern (in this sense anticipating the number of samples that can be reached is key) as well as not mixing lower and upper case letters to avoid mistakes.

Untidy data organisation

Collecting metadata and labelling HTS experiments efficiently is useless if data cannot be located. Unfortunately, we have observed that the raw and processed sequencing data as well as the results derived from them tend to be stored in a rather untidy fashion (**Table 1**). Such situations may happen if files are stored without anticipating additional sequencing runs and analyses, processed and analysed on the fly or managed by several people with different or missing perceptions of organizing data (the latter is a frequent case when people leave and must be replaced). Difficulties to find data are aggravated by the existence of duplicated sample names and lack of recorded metadata.

Alternatively, we suggest a structured and hierarchical organisation that reflects the way in which sequencing data are generated and analyzed. In general, experiments are sequenced in different multi-sample runs separated in time, so it is convenient storing the raw data from the sequencing samples grouped by the run in which they were generated (**Fig. 1b**). Sequencing run directories can contain not only the FASTQ files with the sequencing reads but also information about their quality (e.g. FastQC [11] reports). Conversely, we discourage storing herein modified, subsetted or merged FASTQ files to ensure that analyses start off from the very same set of reads.

In our experience, HTS data goes through two types of sequential analyses. First, raw data (i.e. FASTQ files) are processed sample-wise with relatively standard but tunable analysis pipelines ("core analysis pipelines" in **Fig. 1b**) which generate a variety of files, some of them largely shared by most HTS applications (e.g. alignments in SAM format [12]) while others are more specific of the HTS assay (e.g. binding sites in ChIP-seq, expression quantifications in RNA-seq). Considering this, we recommend storing the processed data resulting from core analysis pipelines with (i) one directory for each sequencing sample; (ii) subdirectories for the files generated in the different steps of the analysis pipeline

(e.g. alignments, peaks) as well as for the logs of the programs used and the file integrity verifications; (iii) subdirectories that accommodate variations in the analysis pipelines (e.g. genome assembly version, aligner) so that these do not overwrite existing data.

In a second step, processed data from one or more samples are combined in order to perform downstream analyses (**Fig. 1b**), and finding effortlessly what, when and how they were performed is crucial. With this in mind we suggest: (i) name the analysis directory with a timestamp plus a descriptive tag from a controlled vocabulary (e.g. 'differential_expression_analysis')'; (ii) include well-defined subdirectories where the output of the analysis is saved; (iii) document thoroughly the analysis; and (iv) group analysis directories under a directory with the name of the user who requests the analyses. While saving the downstream analyses grouped by projects is also a sensible option, this may be straightforward only for analyses under the umbrella of a well-defined broad project. Moreover, very often the name initially given to a project when its directory is created may become imprecise as the project evolves, analyses are unrelated to any existing project or a given analysis is used in many projects. In **Additional file 2: Table S3** and in the Didactic dataset we provide further details about the proposed structured and hierarchical data organisation system.

Yet another analysis

Analysing HTS data is hardly ever a one-time task. At the core analysis level, samples are sequenced at different time points (**Fig. 1b**) so core analysis pipelines have to be executed for every new sequencing batch. Also, samples may need to be re-processed when analysis pipelines are modified substantially (for instance, by including new programs or changing key parameter values) to ensure that data from different samples are comparable in the downstream analysis. At the downstream level, repeating analyses with different datasets or variables, just to name a few variations, is a common task. We therefore identified four desired features for the code used in the analysis.

Firstly, it needs to be scalable, that is, effortlessly executed for a single sample or for hundreds. Processing hundreds of samples sequentially is impractical so, secondly, code has to be parallelizable

to exploit multi-core computing architectures in order to process multiple samples simultaneously and speed up the individual steps within the analysis. Third, automatic configuration of the variable values is necessary so that these need not to be set for each sample. Finally, we found very convenient breaking down analysis pipelines into modules that can be executed individually. In **Additional file 1: Fig. S6** we show how such features can be incorporated. Briefly, (i) scalability is achieved by having a submission script that generates as many pipeline scripts as samples in a configuration file; (ii) parallelisation is obtained by submitting each sample pipeline script as an independent job in the computing cluster, if there is one, and adapting the pipeline code to be suitable for running in multiple processors; (iii) each pipeline script is automatically configured by retrieving the pipeline variable values (e.g. species, read length) from the metadata SQL database; and (iv) the pipeline code is grouped into modules that can be executed all sequentially or individually by specifying it in the configuration file. More details about the implementation can be found in the Didactic dataset.

Undocumented procedures

From the moment HTS data are generated, they go through several procedures (e.g. compression, alignment, statistical analysis) that will eventually generate results, typically in the form of text, tables and figures. Very often the details of how these are generated are absent or vaguely documented, which may result in little understanding of the results, irreproducibility and hampers the identification of errors.

On the contrary, we recommend to document as much as possible all the parts involved in the analysis: (i) write in README files how and when software and accessory files (e.g. genome reference sequence, gene annotation) are obtained; (ii) allocate a directory (**Additional file 2: Table S3c**) for any task, even for those as simple as sharing files; (iii) code core analysis pipelines so that log files monitor the progress of the pipeline and the output of the programs used as well as check the integrity of important files like those containing the raw reads, the genome reference sequence and the alignments (**Additional file 2: Table S3**); (iv) use Markdown [13], Jupyter Notebook [14], RStudio [15] or alike to document procedures; and (v) specify the non-default variable values used.

Data overflow

HTS analysis workflows generate a great number of files; for instance, our analysis pipeline for Hi-C data generates ~70 files per sample. However, such files can be useless for some users, if for instance they do not have access to the computer where the data are stored, files are too big to be opened with text editors and/or users lack the skills to manipulate them with the right tools (e.g. Unix, BEDtools [16], SAMtools [12]). Even if this is not the case, as the number of samples increases, better ways to visualize the data than inspecting files individually are essential.

Therefore, we suggest to implement interactive web applications that display the processed data and allow performing specific analyses in a user-friendly manner. As an example, we take advantage of our structured and hierarchical data organisation as well as the available metadata to deploy a web application to visualise processed Hi-C data using Shiny [17] (**Additional file 1: Fig. S7**). In the same direction, platforms for exploring HTS datasets are being built [18–20]. We recommend defining the specific features of such web applications with its potential users, because implementing them requires effort and attempting to comprehend unnecessary functions may lead to a loss of time.

Discussion

Here we identified seven challenges associated to the increasing use of HTS data in the life sciences (**Table 1**). These span from the moment sequencing data are generated to their interpretation and have an important impact on the management and analysis of the data.

In our view these challenges partly reflect suboptimal habits (e.g. poor description of samples, unsystematic sample naming, untidy data organisation and undocumented procedures) that have just been aggravated by the rapid spread of HTS. In addition, the arrival of a technology that requires informatics skills into a historically wet lab-based field often generates situations in which those who perform the experiments are not aware of the computational challenges of the analysis. We think that the solutions proposed here can alleviate such problems and are therefore very timely.

We developed these considerations in the context of a large-scale sequencing project involving around 40 people, but we believe the challenges reported here do not affect exclusively this niche. Higher up, the problems we list are present to some extent in larger scale initiatives too. For instance, in the SRA repository there are ~30,000 experiments (32 Terabases) with an 'unspecified' instrument (**Additional file 2: Table S4**). Also in the SRA repository, only for the top 25 submitter institutions there are several Petabases of data assigned to multiple entries probably referring to the same submitter (**Additional file 2: Table S5**). Altogether, this represents a large amount of data that will be overlooked in many searches, which could have been avoided by enabling mandatory fields with predefined vocabulary. For another example, the ENCODE consortium published mislabelled or failed experiments [21], and approximately 20% of the uploaded ChIP-seq profiles correlate more with a negative control than with their replicate (unpublished observation). On the other hand, these issues also concern relatively smaller groups dealing with a more modest volume of samples. Therefore, we think that the considerations presented here will be applicable by a broad community.

Finally, we illustrate how to implement these considerations. While some of them are very straightforward, like checking the concordance of the sequencing indexes, others will need to be adapted to the specific needs (e.g. establishing a metadata collection system). More than providing golden solutions that work for all cases, we aim to foster the discussion in the community as to how we face the management and analysis challenges posed by the current explosion of projects using HTS. Beyond sequencing, we predict that many of the challenges and considerations presented here are common to other high-throughput technologies (e.g. spectrometry or microscopy).

Abbreviations

DNA: desoxyribonucleic acid; ENA: European Nucleotide Archive; ENCODE: Encyclopedia of DNA Elements; GEO: Gene Expression Omnibus; HTS: high-throughput sequencing; ID: identifier; INHIC: in situ Hi-C; TCGA: The Cancer Genome Atlas; SAM: Sequence Alignment/Map; SRA: Short Read Archive; SQL: Structured Query Language.

Competing interests

The authors declare that they have no competing interests.

Author's contributions

JQ wrote the manuscript. EV, FD, FS, YC, RS, GF, TG, MM and MB edited the manuscript. JQ, EV, FD, YC and RS developed the metadata collection and sample identification systems. JQ, EV and FS prepared the Didactic dataset. All authors read and approved the final manuscript.

Acknowledgements

We thank F. Javier Carmona and Corey T. Watson for advice on the manuscript. We received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013)/ERC Synergy grant agreement 609989 (4DGenome). The content of this manuscript reflects only the author's views and the Union is not liable for any use that may be made of the information contained therein. We acknowledge support of the Spanish Ministry of Economy and Competitiveness, 'Centro de Excelencia Severo Ochoa 2013-2017' and Plan Nacional (SAF2016-75006-P), as well as support of the CERCA Programme / Generalitat de Catalunya. RS was supported by an EMBO Long-term Fellowship (ALTF 1201-2014) and a Marie Curie Individual Fellowship (H2020-MSCA-IF-2014).

Author details

[1]Gene Regulation, Stem Cells and Cancer Program, Centre for Genomic Regulation (CRG), The Barcelona Institute of Science and Technology (BIST), Dr. Aiguader 88, 08003 Barcelona, Spain

[2]Universitat Pompeu Fabra (UPF), Barcelona, Spain

[3]Structural Genomics Group, CNAG-CRG, The Barcelona Institute of Science and Technology (BIST), Barcelona, Spain

[4]Institució Catalana de Recerca i Estudis Avançats (ICREA), Barcelona, Spain

References

1. Goodwin S, McPherson JD, McCombie WR. Coming of age: ten years of next-generation sequencing technologies. Nat Rev Genet. 2016;17:333–51. doi:10.1038/nrg.2016.49.

2. Pachter L. Bits of DNA. *Seq. https://liorpachter.wordpress.com/seq/. Accessed 28 Apr 2017.

3. Leinonen R, Sugawara H, Shumway M, International Nucleotide Sequence Database Collaboration. The sequence read archive. Nucleic Acids Res. 2011;39 Database issue:D19-21. doi:10.1093/nar/gkq1019.

4. Cancer Genome Atlas Research Network, Weinstein JN, Collisson EA, Mills GB, Shaw KRM, Ozenberger BA, et al. The Cancer Genome Atlas Pan-Cancer analysis project. Nat Genet. 2013;45:1113–20. doi:10.1038/ng.2764.

5. 1000 Genomes Project Consortium, Abecasis GR, Auton A, Brooks LD, DePristo MA, Durbin RM, et al. An integrated map of genetic variation from 1,092 human genomes. Nature. 2012;491:56–65. doi:10.1038/nature11632.

6. ENCODE Project Consortium. An integrated encyclopedia of DNA elements in the human genome. Nature. 2012;489:57–74. doi:10.1038/nature11247.

7. Muir P, Li S, Lou S, Wang D, Spakowicz DJ, Salichos L, et al. The real cost of sequencing: scaling computation to keep pace with data generation. Genome Biol. 2016;17:53. doi:10.1186/s13059-016-0917-0.

8. Dekker J, Belmont AS, Guttman M, Leshyk VO, Lis JT, Lomvardas S, et al. The 4D Nucleome Project. bioRxiv. 2017.

9. Baker M. 1,500 scientists lift the lid on reproducibility. Nature. 2016;533:452–4. doi:10.1038/533452a.

10. Munafò M, Nosek B, Bishop D, Button K, Chambers C, Percie du Sert N, et al. A manifesto for reproducible science. Nat Hum Behav. 2017;1:0021. doi:10.1038/s41562-016-0021.

11. FastQC. FastQC. https://www.bioinformatics.babraham.ac.uk/projects/fastqc/. Accessed 10 May 2017.

12. Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, et al. The Sequence Alignment/Map format and SAMtools. Bioinformatics. 2009;25:2078–9. doi:10.1093/bioinformatics/btp352.

13. Markdown. Markdown. https://daringfireball.net/projects/markdown/. Accessed 28 Apr 2017.

14. Jupyter. Jupyter. http://jupyter.org/. Accessed 28 Apr 2017.

15. RStudio. RStudio. https://www.rstudio.com/. Accessed 28 Apr 2017.

16. Quinlan AR, Hall IM. BEDTools: a flexible suite of utilities for comparing genomic features. Bioinformatics. 2010;26:841–2. doi:10.1093/bioinformatics/btq033.

17. RStudio, Inc. Easy web applications in R. Easy web applications in R. 2013. https://shiny.rstudio.com/. Accessed 28 Apr 2017.

18. Pimentel H, Sturmfels P, Bray N, Melsted P, Pachter L. The Lair: a resource for exploratory analysis of published RNA-Seq data. BMC Bioinformatics. 2016;17:490. doi:10.1186/s12859-016-1357-2.

19. Collado-Torres L, Nellore A, Kammers K, Ellis SE, Taub MA, Hansen KD, et al. Reproducible RNA-seq analysis using recount2. Nat Biotechnol. 2017;35:319–21. doi:10.1038/nbt.3838.

20. Khomtchouk BB, Hennessy JR, Wahlestedt C. MicroScope: ChIP-seq and RNA-seq software analysis suite for gene expression heatmaps. BMC Bioinformatics. 2016;17:390. doi:10.1186/s12859-016-1260-x.

21. Cuscó P, Filion GJ. Zerone: a ChIP-seq discretizer for multiple replicates with built-in quality control. Bioinformatics. 2016;32:2896–902. doi:10.1093/bioinformatics/btw336.

22. Schlötterer C, Tobler R, Kofler R, Nolte V. Sequencing pools of individuals - mining genome-wide polymorphism data without big funding. Nat Rev Genet. 2014;15:749–63. doi:10.1038/nrg3803.

23. Lasken RS, McLean JS. Recent advances in genomic DNA sequencing of microbial species from single cells. Nat Rev Genet. 2014;15:577–84. doi:10.1038/nrg3785.

24. Shapiro E, Biezuner T, Linnarsson S. Single-cell sequencing-based technologies will revolutionize whole-organism science. Nat Rev Genet. 2013;14:618–30. doi:10.1038/nrg3542.

25. Cock PJA, 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 Res. 2010;38:1767–71. doi:10.1093/nar/gkp1137.

Figures

Figure 1. Framework for the management and analysis of HTS data. (a) Metadata are collected via an online form and stored both online and in a local database. **(b)** The stages of HTS data.

Figure 2. Examples of sample identifier strategies. (a) The sample ID contains several bits of information separated by "_" and each defined manually based on the information present in the metadata. **(b)** After defining two sets of either biological or technical fields that unequivocally define a sequencing sample, for a given sample the values of the biological fields treated as text are concatenated and computationally digested into a 9-mer, and the same procedure is applied to the technical fields. The two 9-mers are combined to form the sample ID. In the Didactic dataset we provide a script that generates this type of sample ID from the metadata.

Tables

Table 1. Challenges associated to the accelerated accumulation of HTS data.

Challenge	Impact	Consideration
Mislabelled raw sequencing data	Underpowered analysis Erroneous results Loss of data, time and resources	Check unassigned reads and sequencing index concordance
Poor sample description	Prevents data processing and quality control Incorrect analysis and results Lack of reproducibility Delays publication	Metadata collection
Unsystematic sample naming	Duplicated or similar names Ambiguous identification Precludes computational treatment Data disclosure	Sample identifier scheme
Untidy data organisation	Data cannot be found Time consumption Inability to automate searches	Structured and hierarchical data organisation
Yet another analysis	Repeated manual execution of analyses Incapability to deconvolute analysis producing different results Compulsory linear execution	Scalability, parallelization, automatic configuration and modularity
Undocumented procedures	Poor understanding of results Irreproducibility Hampers catching errors	Documentation
Data overflow	No access to data Size and number of files make individual inspection inefficient	Interactive web applications

Additional Files

Additional file 1 - Supplemental Figures

Figure S1. Rapid accumulation of HTS data. (a) Accumulated volume of sequences in the SRA since 2007, expressed both in terms of bases and bytes (data accessed from https://trace.ncbi.nlm.nih.gov/Traces/sra/sra.cgi on 2017-04-06). Tera and Peta thresholds are shown as reference.

Figure S2. Diversity of the SRA datasets. (a) Cumulative number of bases (bp) in the SRA repository deposited by the top 100 contributors. **(b)** Number of bases per organism for the 25 organisms with most bases deposited in the SRA.

Figure S3. Overview of the sequencing process. A sequencing index is typically a 6-bp pre-defined nucleotide sequence that is added to the fragments generated from a single DNA sample. The source of the sequenced sample can be very diverse, e.g. cell populations from single (blue) or pooled individuals (green) [22], microbial communities (orange) [23] or, since more recently, single cells (red) [24]; in addition, current HTS applications go beyond genomic DNA sequencing (e.g. ChIP-sequencing, RNA-sequencing, Hi-C) (see Goodwin et al. for a recent review [1]). The fragments of indexed DNA are pooled and loaded into the sequencer so that each fragment occupies a specific spot. The sample sequencing index and the spot position are used to translate the images generated by the sequencer into sample-specific sequences stored in FASTQ files [25]. Image of the sequencer courtesy of Illumina, Inc.

Figure S4. Sequencing (meta)data. The description of a sequencing experiment is impaired unless sequencing reads are coupled with biological, technical and logistics information (metadata). Unambiguous sample identifiers provide a link between metadata and sequencing data.

Figure S5. Metadata SQL database schema. Depicted are the tables in the metadata SQL database, each with its name (top) and a subset of its fields; primary unique keys highlighted in orange. The metadata collected by the user is dumped into the "input_metadata" table, which can be associated through the "SAMPLE_ID" to other information generated from the analysis of the sequencing data. For instance, the "quality_control_raw_reads" stores information related to the quality of the reads reported by FastQC [11] . As many tables as analysis pipelines can be added to collect parameters used and metrics generated by these (e.g. "chipseq", "rnaseq"). Because the latter will store information for the last time a pipeline is executed for a given sample, including a table, like in 'jobs', to keep track of different executions may be useful (e.g. benchmarking of pipelines or parameters).

Figure S6. Scalability, parallelization, automatic configuration and modularity of analysis pipelines. Analysis pipelines can be simultaneously run on multiple samples with a single command (gray rectangle). The configuration file ('*.config') contains the list of samples to be processed as well as the hard-coded parameters shared by all samples (e.g. number of processors or genome assembly version). The submission script ('*submit.sh') wraps the pipeline code ('*seq.sh') and generates, for each sample, a pipeline script with sample-specific variable values obtained from the SQL metadata database, and this will be submitted to the queuing system of the computing cluster where it will be queued (orange) and eventually executed (green). Selected metadata generated by the pipeline (e.g. running time, number of aligned reads) will be recorded into the database. For more flexibility, the pipeline modules to be executed are specified in the configuration file.

Figure S7. Workflow of the Hi-C visualisation tool. Metadata and processed data are fed to the 'app.R' written for Shiny [17].

Additional file 2: Supplemental Tables

Table S1. Proposed features for a metadata collection system.

Table S2. Description of metadata fields.

Table S3. Example of structured and hierarchical data organisation. (a) Raw data, **(b)** processed data and **(c)** analysis results.

Table S4. Number of SRA deposited bases grouped by instrument name.

Table S5. Number of SRA deposited bases grouped by the submitter. For the top 25 contributors in terms of number of bases submitted, we searched for instances of multiple entries probably referring to the same submitter (e.g. 'ncbi' and 'NCBI').

Fig. 1

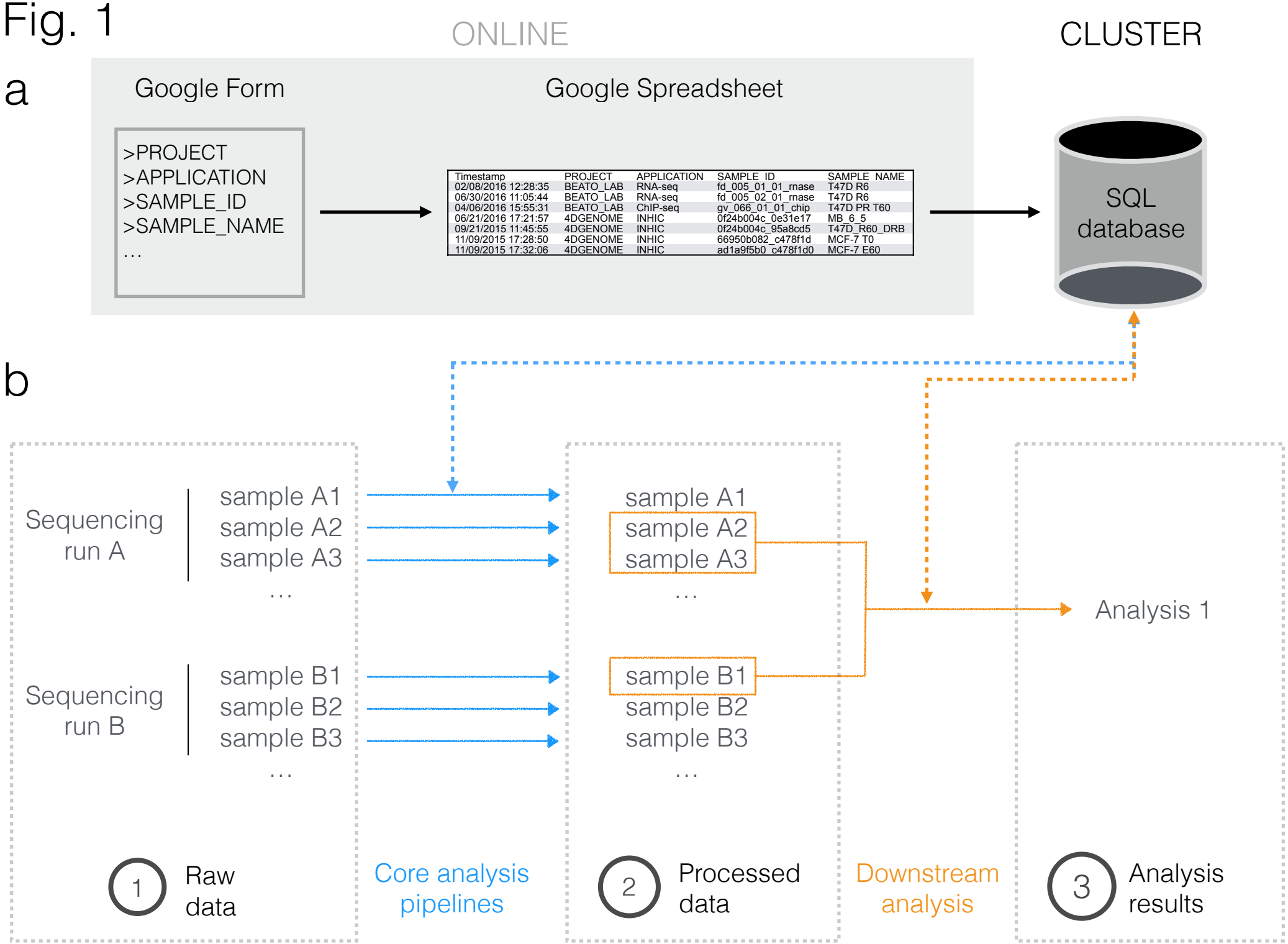
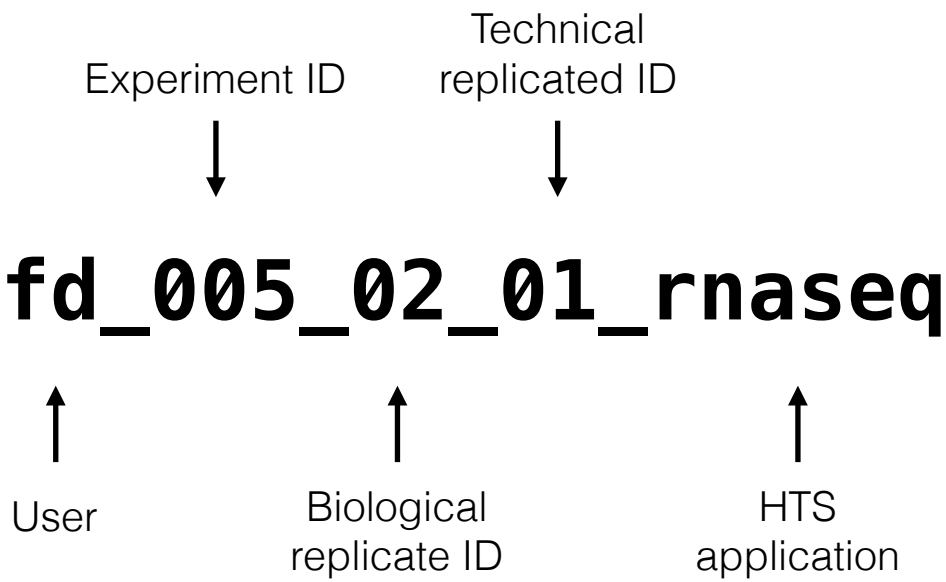

ONLINE
CLUSTER
a
Google Form
Google Spreadsheet
>PROJECT
>APPLICATION
>SAMPLE_ID
>SAMPLE_NAME
...
Timestamp PROJECT APPLICATION SAMPLE_ID SAMPLE_NAME
02/08/2016 12:28:35 BEATO_LAB RNA-seq fd_005_01_01_rnase T47D R6
06/30/2016 11:05:44 BEATO_LAB RNA-seq fd_005_02_01_rnase T47D R6
04/06/2016 15:55:31 BEATO_LAB ChIP-seq gv_066_01_01_chip T47D PR T60
06/21/2016 17:21:57 4DGENOME INHIC 0f24b004c_0e31e17 MB_6_5
09/21/2015 11:45:55 4DGENOME INHIC 0f24b004c_95a8cd5 T47D_R60_DRB
11/09/2015 17:28:50 4DGENOME INHIC 66950b082_c478f1d MCF-7 T0
11/09/2015 17:32:06 4DGENOME INHIC ad1a9f5b0_c478f1d0 MCF-7 E60
SQL database
b
Sequencing run A
sample A1
sample A2
sample A3
...
Sequencing run B
sample B1
sample B2
sample B3
...
sample A1
sample A2
sample A3
...
sample B1
sample B2
sample B3
...
Analysis 1
1 Raw data
Core analysis pipelines
2 Processed data
Downstream analysis
3 Analysis results

Fig. 2

a


Experiment ID
Technical replicated ID
fd_005_02_01_rnaseq
User
Biological replicate ID
HTS application

b

Biological
CELL_TYPE
PRE_TREATMENT
PRE_TREATMENT_TIME
TREATMENT
TREATMENT_TIME
CONTROL
*
0f24b004c_0e31e17c7
*
Technical
EXPERIMENT ID
HIC
SEQUENCING_PRE_LIBRARY
SEQUENCING_LIBRARY
SEQUENCING_CORE
SEQUENCING_RUN
*
9-mer text digestion of values