SRA Handbook

Last Updated: 2016 Jan 14

National Center for Biotechnology Information (US)
Bethesda (MD)

National Center for Biotechnology Information (US), Bethesda (MD)

NLM Citation: SRA Handbook [Internet]. Bethesda (MD): National Center for Biotechnology Information (US); 2010-.

This documentation provides an overview and help manual for the Sequence Read Archive (SRA) at the National Center for Biotechnology Information.

Table of Contents

Introduction to SRA

Concepts

Created: January 13, 2009; Updated: January 6, 2014.

Overview

The current Trace Archive for capillary-based sequencing platforms tracks sequencing data as individual traces. The arrival of new massively parallel sequencing technologies has complicated representation of such experimental data. In addition, investigators are conducting increasingly diverse experiments with greater throughput. More data producers (particularly those from smaller labs) are coming on line. Finally, we continue to see greater reliance of the community on public resources like those at NCBI to accession experimental data for archival, retrieval, and publication.

The Sequence Read Archive (SRA) is an entirely new resource at NCBI. It is being designed specifically meet the challenges presented by massively parallel sequencing technologies.

This document defines entities and relations that make up the Sequence Read Archive (SRA). .

Goals

- Provide a central repository for next generation sequencing data.
- Provide links to other resources referencing or using this data.
- Provide users with retrieval based on ancillary information and sequence comparison.
- Track studies and experiments (project metadata).
- Allow flexible submission and retrieval of ancillary data.
- Improve database efficiency through normalization of data structures.
- Separate submission from content.
- Establish basis for user-interactive submission and retrieval.

Related Documents

- NCBI Trace Archive Documentation

Concepts

A fundamental departure from the current Trace Archive design separates the experimental data from its metadata. The metadata are now organized as follows:

Study – A study is a set of experiments and has an overall goal.

Experiment – An experiment is a consistent set of laboratory operations on input material with an expected result.

Sample – An experiment targets one or more samples. Results are expressed in terms of individual samples or bundles of samples as defined by the experiment.

Run – Results are called runs. Runs comprise the data gathered for a sample or sample bundle and refer to a defining experiment.

Submission – A submission is a package of metadata and/or data objects and a directive for what to do with those objects.

Differences with the Trace Archive

The separation of metadata from data allows for separable submissions. Thus information about the study or experiment can be posted when it becomes decided (typically early in the project life cycle), and can await the experimental results (typically late in the project life cycle). This separation of concerns also allows for new features including hold-until-publish and user-controlled modification.

Metadata disassociation also permits gathering of richer information about studies. The old method of collecting study specific attributes (for example, the *salinity* attribute for seawater metagenomics studies) has been dropped in favor of a more flexible system of Center-controlled vocabulary. This policy change allows the SRA to serve as a repository of important ancillary data while avoiding the pitfall of ontology development.

All next generation sequencing technologies perform image processing in order to reduce sequencing data to base, quality, and intensity calls. These will be accepted in either the incipient Sequence Read Format (SRF) data file or manufacturer specific data files. Currently, the Trace Archive accepts only ZTR files.

The SRA will from its inception accept any secondary analyses typically performed on the next generation data. These might include alignments, small-scale assemblies, oligo profiles. As many of these analyses are currently being developed, the SRA will accept their reports and data in "blob" form with virtually no internal structure. This will establish a new "one-stop" submission paradigm appropriate for small projects, projects executed by automatic pipelines, and projects submitted from newer Centers with little experience interacting with NCBI. Future releases of the SRA will improve the routing of analysis data to downstream repositories. The Trace Archive accepts only traces.

The vast increase in the amount of data offered by the next generation technologies has required modification of the capillary-based sequencing notions of reads and read accessions. The SRA will be offering reads for retrieval based on type and container relationships specified in the retrieval query. Accessions may be assigned as specified by the user. However, there will be no tracking of individual read accessions even if these are designated in the original submission because of the excessive amount of storage space needed just to accommodate accessions.

Submission

A **submission** is a wrapper around study metadata and run data, plus directives for the SRA operators.

Properties include:

center_name

submission_name

submission_date

contact_info

study accession (and possibly one or more experiment accessions)

directive

Here are some directives (operations) concerning submissions:

- New – This is a new submission for this **study.**
- Modify – This submission entirely replaces a previously referenced submission. The goal is usually to repair a flawed submission, or to complete an interrupted submission.
- Release – Release to the public an embargoed submission.
- Suppress – The submission should be entirely suppressed.

Submissions are tracked using a publicly available web interface. Submissions can be referenced in two ways: by a public moniker assigned by the Center, and by a private key that is returned to the submitting Center once its submission has been logged at NCBI. A submitting Center can update or replace only its own submissions. Both keys must be used in all subsequent transactions concerning a submission.

A submission can be held until publication ("HUP", or embargo). Such submissions do not appear in public status information. The submitting Center must send a directive to NCBI releasing the submission to the public (there are no decision rules).

Run

Each experiment in a study may receive one or more runs of sequencing. Sequencing runs identify a sample accession. Thus the study and its experiments must be defined before the submission of run data. The advantage of this approach is flexibility: multiple samples can be sequenced by a single run, or multiple runs can encompass a single sample. Study metadata can still be included in the submission so long as they define all tags used in the run data.

Sample Organization

An experiment will support the following sampling formats:

Individual – The sequencing effort targets one sample only.

Multiplexed – The sequencing effort targets a set of samples and each read can be mapped to a sample. This mapping would be resolved on data retrieval.

Pooled – The samples that are being sequenced are known and can be listed, but the reads cannot be mapped to them.

Population – The samples cannot be distinguished but their overall number may be known.

Where possible, a sample should be identified by its taxon id rather than a name. In some cases a taxon id will not be relevant. Where bar codes (known oligo sequences incorporated into the sequencing material) are known, these should be listed by the submitter.

Library Organization

Some concepts in library construction are borrowed from capillary-based sequencing. Although it is not always necessary to construct a library for next generation technologies (this is indeed one of the advantages), some aspects of the source material may be important to track, including:

Library Name – Center assigned library name often used by collaborators

Library Source – Type of DNA or RNA used in the experiment

Library Selection – Whether the source material was selected for certain properties (for example methyl filtrated)

Library Layout – Number, order, orientation, and distance of associated reads (for example, 2 Kbp paired ends).

Library Protocol – Free form text that documents the "library construction" steps.

Spot Organization

A spot is a new kind of abstraction that captures all the data associated with one intensity function in time. Thus reads related by mate pairing or bar coding can be tracked implicitly by virtue of sharing a "reaction container." The concept is roughly analogous to that of a *growth template* in capillary-based sequencing, in which mate pair reads are related by their sharing of an insert in the cloning vector.

A read is classified as to whether it is a technical read (primer, linker, adapter, bar code, etc) or an application read (single read, paired ends, etc). Reads are indexed by one of three methods: base-based coordinates, cycle-based coordinates, or by alignment to an expected oligo sequence (such as a linker or bar code).

Specification of spot decoding is done at the level of the experiment design, so that this information is bound once per experiment, and not once per read, as is currently done.

Read Organization

NCBI proposes the following accession format for each read:

`SRA000000.ssss.rrr`

where SRA000000 denotes the sample accession, ssss denotes the “spot” number in the run, and rrr denotes the read index within the spot. A motivation for using integer encoding of read names in this way is to eliminate the heavy burden of accessing small units of information with large randomly accessible keys. Integer encoding allows for implicitly indexed access which takes far less computational time to manage. In addition, range specifications can replace lists when referring to contiguous read sets.

For example, application mate pairs might be retrieved for a certain sample by the following regular expression: *SRA458693.*.00[24]* .

Overview

Created: May 1, 2009; Updated: March 5, 2014.

An Introduction to the Sequence Read Archive

The advent of massively parallel sequencing technologies has opened an extensive new vista of research possibilities — elucidation of the human microbiome, discovery of polymorphisms and mutations in individual genomes, mapping of protein–DNA interactions, and positioning of nucleosomes — to name just a few. In order to achieve these research goals, researchers must be able to effectively store, access, and manipulate the enormous volume of short read data generated from massively parallel sequencing experiments.

In response to the research community's need for such a resource, NCBI, EBI, and DDBJ, under the auspices of the International Nucleotide Sequence Database Collaboration (INSDC), have developed the Sequence Read Archive (SRA) data storage and retrieval system. The SRA not only provides a place where researchers can archive their short read data, but also enables them to quickly access known data and their associated experimental descriptions (metadata) with pin-point accuracy.

The SRA currently contains more than 100 terabytes (100,000 gigabytes) of short read data and is growing rapidly. Due to the regular exchange of data between NCBI, EBI and DDBJ – all of whom are using the same design model and code libraries in their respective Sequence Read Archives -- researchers will be able to access the most up-to-date short read sequence data from around the world.

Terms of Usage

Permanence

Accessions issued by the SRA are always maintained and never reused. If a desired record has been withdrawn, then a message to this effect will be displayed to anyone who tries to access it. If a record has been superseded by a successor record, this fact will be presented to anyone trying to access it. Only in rare cases where the record needs to be expunged from the archive will a user not be able to access it.

Authentication

Submissions are managed through secure channels. These channels include PDA, NIH level login through CIT, and FTP accounts secured by passwords. We will correspond with submitters via email about submission and curation issues, but we do not exchange data by email. At this time NCBI PDA is used for authenticating to the SRA submission pages or accounts.

Please keep your PDA and file transfer accounts secure. Please do not reuse someone else's accounts. Center accounts are provided for the convenience of automated pipelines and

where multiple users need to manage submissions. The authentication information for such an account should be maintained securely by the Center. Accounts may be disabled or withdrawn after a long period of disuse in order to comply with NCBI security requirements.

Limitations

The Sequence Read Archive at NCBI is a public resource and the decision whether to submit data to this resource is the responsibility of the submitter. Prospective submitters should be aware of the following issues:

Never submit data without the permission of the **principal investigator**.

Most **human data** gathered from research subjects are under strict privacy controls and/or usage restrictions and must be handled with protections as determined by the research institution's Institutional Review Board (IRB), the funding agencies, and the laws of the United States or the submitter's home country. The dbGaP resource at NCBI may be a more appropriate broker for human sequencing data requiring controlled access due to these considerations. Data from whole genome, transcriptome, epigenome, and metagenome (which may include human contaminants) may fall into this category. Data gathered from human subjects, certain cell lines, and metagenomes may be covered as well.

Data submitted as part of a journal manuscript may have a **publication embargo** placed on it by the journal editors. The submitter can place a "hold until publish" restriction on the submission to the SRA as part of the submissions process.

Data that might relate to **patents** and **intellectual property** may be submitted to NCBI, but the submitter is responsible for ensuring that procedures and policies of his/her institution or company are observed.

Some **environmental data** gathered in the territory of certain countries, including territorial waters, may have sovereign legal restrictions on their use. NCBI cannot accept such data since NCBI is not able to enforce any usage restrictions.

Submitters must ensure that data obtained as part of a criminal investigation is free of any judicial restrictions on its use.

Submitters are responsible for obtaining all necessary permissions from the collecting institution for **forensic and paleontological data.**

The United States and many other countries have laws governing trade in **endangered species**. Please be aware that nucleotide material gathered from such samples may be subject to restrictions. Over-specific metadata accompanying submissions may also be inappropriate for when the samples are rare or endangered.

Modification

NCBI allows submitters to modify their records. The modifications can be made using the online submission tool that was used to create the records. If you wish to delete or move a record, please contact us at sra@ncbi.nlm.nih.gov. Only the center or individual that created the record can change it. If you have changed affiliations and wish to update old records, please login to the SRA page and update your organization in your profile under the Preference tab.

Curation

From time to time records deposited at the SRA must be updated with changes needed in order that the data continue to conform with the data model for the archive, to update data as it changes (for example finalizing publication information), to change data that are clearly wrong (for example correcting external references to other data or resources), and to add additional relevant metadata as they become available. NCBI will contact the submission owner on a best effort basis. The submission owner should maintain up to date contact information with NCBI to receive word of such changes.

Actual instrument data are not changed by NCBI. Only the submitter can make such modifications.

Availability

While NCBI tries to maintain maximum uptime of its servers on a 24x7 basis, no guarantee of availability is offered to users. Submissions that are interrupted by downtime may have to be restarted by the user.

Technical assistance is available on a limited basis during business hours USA Eastern Time. There is no guarantee for level of service regarding manual assistance.

SRA Features

A Central Data Repository with Submission Flexibility

The SRA preserves all content submitted from the major sequencing technologies, and currently accepts submissions that originate in any of these major short read formats:

- SFF (Roche 454)
- Illumina Native
- Illumina SRF
- AB SOLiD Native
- AB SOLiD SRF

Please note that additional formats will be supported as they become available.

SRA provides two venues for submissions: 1) An interactive web-based interface for occasional submissions, which requires only a brief registration prior to submission, and

2) An automated submission pipeline for centers making multiple submissions; this process uses XML to describe metadata and Sequence Read Format (SRF) as a common container file format.

SRA uses a high-speed file transfer protocol called fasp (Aspera, Inc., Emeryville, CA), which allows users to transfer files to and from the SRA at speeds up to 400 Mbps — many times faster than ftp. Details on obtaining the free client can be found in the Aspera Transfer Guide.

The SRA's common, compact design allows for the storage and rapid retrieval of all types of data from massively parallel experiments, including:

- Reads and associated quality scores
- Trimming and other technical information
- Experiment metadata
- Secondary analyses typically performed on short read data, including:
  - Alignments
  - Small-scale assemblies
  - Oligo profiles
- Intensity data

Submission requirements for the SRA are quite flexible; a user can include all of the data elements listed above, or just a subset of them. For example, the user could submit descriptive information about a study or experiment once details were decided, and submit the experimental results sometime in the future. The SRA submission process also allows users to modify their submissions and provides a number of Hold–Until–Published (HUP) options that include:

- Hold for a Number of Days: used for certain data-release policies
- Hold Until a Specific Date: used for the scheduled release of a publication
- Hold: used when a publishing journal has not yet been determined, or the publication date has not yet been set

Because the SRA's design allows for the submission of new (still under development) short read data analyses in "blob" form (virtually no internal structure), the SRA can be a "one-stop" submission resource for small projects, projects executed through automatic pipelines, and projects submitted by newer sequencing centers that have little experience interacting with NCBI.

The SRA also allows for the selective movement of older, less frequently used data element(s) to less expensive storage (tape or disc), or for the eventual discard of the data element(s) without having to reload any of the other data associated with these redistributed or discarded element(s).

For further information about submitting to SRA, please see the Submission Quick Start Guide.

An Integrated Search and Analysis System

The SRA's design allows users to quickly and precisely retrieve massively parallel experiment data of interest down to the level of individual reads. There currently are two ways to access data housed within SRA: the NCBI/SRA web interface (best for limited quantities of data) and the SRA System Development Kit (SDK) (best for larger quantities of data).

Accessing SRA Data Using the Web Interface

Since the SRA metadata is indexed in NCBI's Entrez search and retrieval system, users can access this content directly from the NCBI home page by selecting "SRA" from the drop-down list of available databases at the top of the page and entering a query (for example, "salmonella") in the search box; the SRA display page of results will include records for each matching study, with links to read/run data, project descriptions, etc. Users can also search SRA through the coordinated use of the "Browse" and "Search" tabs on the SRA home page. The SRA web interface allows the user to:

- Access any data type stored in the SRA independently of any other data type, (e.g., accessing read and quality data without the intensity data)
- Access reads and quality scores in parallel
- Access related data from other NCBI resources that are integrated with SRA
- Retrieve data based on ancillary information and/or sequence comparisons
- Retrieve alignments in "vertical slices" (showing underlying layered data) by reference sequence location
- Review the descriptions of studies and experiments (metadata) independently of experimental data

Accessing SRA Data using the System Development Kits

The SRA System Development Kits (SDK) provides Application Programming Interfaces (APIs) that facilitates the accession and manipulation of larger quantities of data.

The "Read" SDK allows the user to programmatically access data housed within SRA and convert it from the SRA format to any of these major short read formats:

- AB SOLiD Native
- FASTQ
- SFF (Roche 454) (under development)
- Illumina Native
- BAM

The "Read" SDK is designed to prevent accidental modification of the data, and is optimized to provide the user with the most efficient read possible.

The "Write" SDK, on the other hand, allows a user to both read SRA data as well as convert (write) it from the major short read formats listed below into the SRA flexible format:

- FASTQ
- AB SOLiD-SRF
- AB SOLiD-Native
- Illumina SRF
- Illumina Native (under development)
- SFF (under development)
- BAM

The "Write" SDK allows users to create a local archive using their own short read data, and in future will be used for direct submissions to SRA.

SRA Architecture

The Need for a New Paradigm in Massive Data Storage and Retrieval

NCBI began development of the SRA database in October, 2007 in response to the research community's need for efficient and flexible ways to store and retrieve the large amounts of massively parallel sequencing data beginning to appear. Now that the SRA has reached an initial state of completion and is publically available at NCBI, it is being deployed at EBI, and soon will also be deployed at DDBJ. NCBI and EBI have already begun exchanging data, and once the SRA is in place at DDBJ, there will be a regular data exchange between all three INSDC members.

The initial design of the SRA was conceived by looking at the advantages and disadvantages inherent in relational databases and in file-based storage systems, and using the best aspects of each to create something entirely new.

Relational Databases Alone are Not the Answer

Relational databases are good for recording and manipulating related data: they can index, make complex arbitrary joins, and process complex queries. Relational databases, however, are not a practical approach for the long-term storage of the terabyte and petabyte amounts of data generated from massively parallel sequencing experiments: they are bulky, require significant management, and are inflexible and costly in terms of storage space.

File-based Storage Systems Alone are Not the Answer

Simple file-based storage systems have many advantages to recommend them for storing large amounts of data: they are lightweight; they make use of the file and directory systems already available; they store data according to an object model (i.e. a run is in a single file, which is an object that can be accessed, shipped, modified, removed, etc.). However, file-based storage systems lack support for indexed queries and cannot create

necessary relationships. Archiving data in a file-based storage system often means using the tar (tape archive) format and applying a compression utility like gzip or bzip2 to produce a compressed tar file, making the data difficult (or impossible) to access in a repository setting.

A Hybrid Storage and Retrieval System

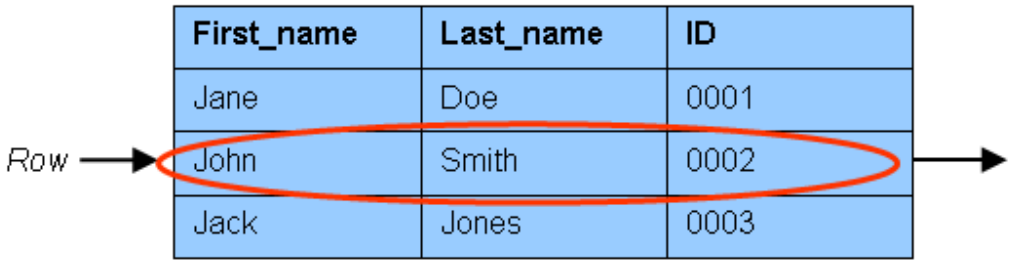
In order to store and retrieve the enormous amount of data generated by massively parallel sequencing technologies, NCBI, EBI and DDBJ needed to create a data repository that has much of the power of a relational database while being lightweight, transportable and flexible like flat-file storage. The solution was to create a hybrid relational database with a file-based and column-oriented design.

A Dynamic Mix of Database and Local Computing

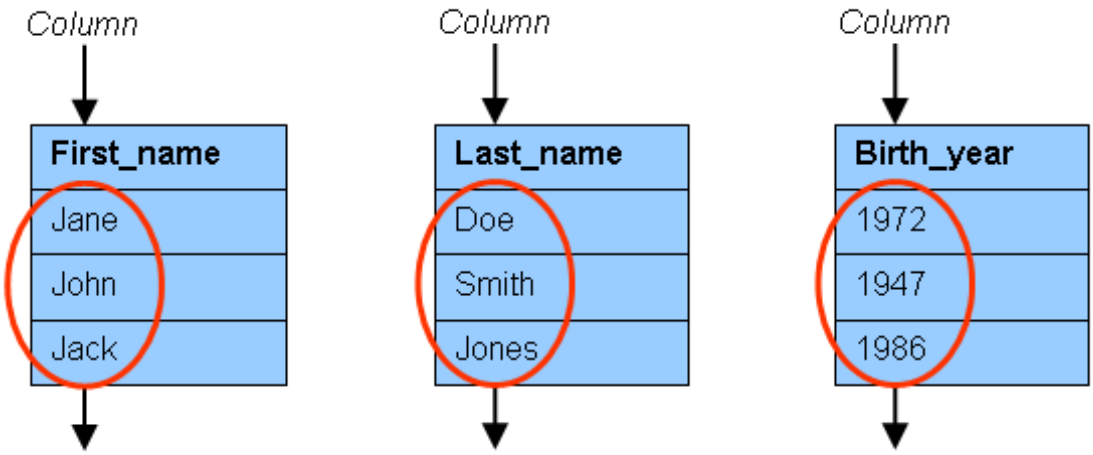
The SRA's design — a novel combination of a file-based, column-oriented design and a relational database — offers great versatility in storage and management of data. Search and retrieval services can be based not only at NCBI but also locally once users move their data into the SRA flexible format.

Row vs. Column-Oriented Database Design

Row-oriented databases store data as a series of row structures, where each structure contains one or more fields (unique data types arranged in columns) linked together in a table. In the row-oriented system, a user approaches the data from left to right in a single row.



Column-oriented databases turn this structure on its side, so to speak, and store the data as columns, where each data type is stored as a series independently in its own column (still associated with its unique ID). In the column-oriented design, a user approaches each data type as an independent series moving from the top of the column downward to the bottom of the column.



The advantage of row-oriented databases is that they link together all the fields (data types) in a single row so that the entire row can be retrieved with a single read. This becomes a disadvantage, however, when applied to the problem of massive data storage:

- Since multiple data types are involved, compression and/or packing of data in row-oriented tables is difficult and results in inefficient use of storage space.
- Because the fields are linked together in each row, the removal of one field necessitates the re-write of the entire data table, making the addition or removal of specific fields difficult.

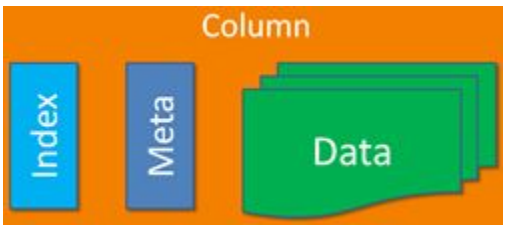
Column-oriented databases, in contrast, are able to achieve improved storage because there is only one data type per column, and also have greater retrieval efficiency. In addition, if a column-oriented database is properly designed, a column (data type) can be added or removed independently of the other columns in the table, so there is no need to re-write an entire data table for every addition or deletion of a data type.

The SRA Hybrid Design

SRA's file-based, column-oriented design makes use of the file system to keep the data columns physically separate. Each data column within the SRA design model is packaged in its own UNIX file rather than in a database. This makes it possible to store the most frequently accessed data series (e.g., "FASTQ" data) in fast, near storage, and less frequently accessed data (e.g., intensities and reads) in slower, bulk storage, such as tape or disk, located at the repository where the data were submitted (NCBI, EBI, or DDBJ). Users can obtain whichever data columns/files are desired via the compute cloud (internet). The design allows users to access and read any SRA short read data in random order, stream it quickly, or get reads and quality scores in parallel. The NCBI relational database portion of the SRA hybrid design model serves to track runs and components, while the SRA toolkit operates using directories and files, so it can easily bring the power of the SRA approach to a local computer.

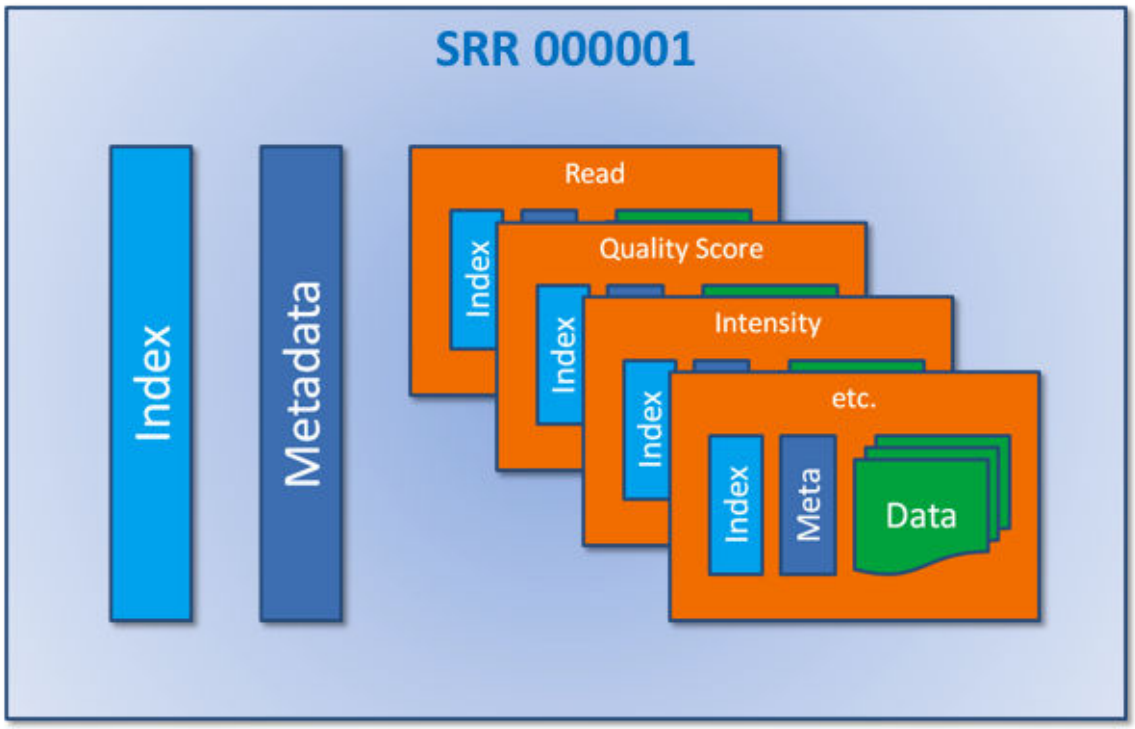
SRA Data Structure

As mentioned above, SRA uses file-based data management, where the base unit is a column packaged within its own UNIX file. Each column represents a single data type (read, quality score, or intensity data, etc.), and the UNIX file holding the column contains not only the particular data type, but also the index of identifiers for each member of that data type and a minimal description (metadata) of each:



If, for example, the column represented above holds read data for a single run within an experiment, then this "read column" (file) would contain the series of template reads (data) generated in the run, the identifiers for each read (which do not have explicit IDs but are given serial numbers based on the run ID to save space), and a small amount of metadata that describes the read (name [alias] and plate location).

The "columns" (files) are then organized together into a "run" — a "table" in database parlance. The "run" (table) groups columns that contain the data gathered for a sample or sample bundle in a particular experiment into a single structure (the number of columns in a run is arbitrary and depends on the number of data types available):



In the example above, the run accession number is SRR000001; this particular run contains a series of Read data, Quality Score data, Intensity data, as well as other data types that may exist ("etc."), where each data type is contained within its own column (file). In addition, the run "table" contains a substantial amount of metadata, including technical information about the instrument model, date of run, run center, plate statistics, brief experimental description, etc., as well as an overarching index for the data housed within the run.

A unique feature of this type of table is that it takes its runtime structural definition from the contents of the file system — that is, it determines its component columns dynamically when it is opened. This feature provides the user with great versatility when it comes to archiving data since old data types can easily be removed and replaced with new data types. For example, if new quality data becomes available, the old quality data column can be removed and replaced with a new quality column, and even if the new quality data is of an entirely different type than the old column, the table will resolve. Similarly, if a user no longer wished to archive intensity data, the intensity column could easily be removed and the table would still resolve upon opening.

The Necessity of Archival Data Storage

There is a great deal of discussion in the community involved with massively parallel sequencing regarding the necessity of archiving the various data types generated from short read sequencing experiments:

Intensity Data

Some within the community feel that intensities are no longer needed once base calls are made, while others believe they should be archived because bases may need to be re-called — for example, if new and improved base-calling algorithms are developed. SRA, therefore, was designed to enable archiving of intensity data. Many project leaders are choosing to archive intensities for early experiments completed prior to the establishment of optimal base calling, or for important projects where it may be difficult to re-sequence the samples. It may be sufficient, however, to deposit only the reads for projects such as ChIP-Seq experiments. For those projects depositing intensity data, the SRA provides the option to discard the intensity data at a later date if the community deems that it is no longer necessary.

Read Data

The SRA also was designed to archive read data because the data can be used in a variety of important ways:

- For alignments when improved alignment algorithms become available
- To regenerate an alignment when a reference assembly is updated
- To generate an alignment to another reference assembly (e.g. European, Asian, or African reference human genome assemblies)
- To pool data across different experiments or create experimental sub-sets from within an experiment

Alignment Data

Finally, SRA is developing the capability to archive alignment data, as this data may be used:

- For re-analysis for purposes different from those intended in the original experiment (e.g. alignment data from a gene expression experiment could be used for SNP verification)
- To verify simple summaries like histograms.
- To generate SNP calls, CNV calls, or different histograms

SRA Future Developments

SRA will continue to support new platforms and sequencing technologies as they become available.

Using the SRA

Download Guide

Created: September 9, 2009; Updated: January 14, 2016.

Overview

The purpose of this document is to explain to users how to download datasets of interest and associated metadata.

Important Notes on Download Facilities

- One basic format (.sra) is provided by the SRA for all publicly available data. The SRA Toolkit is provided to allow conversion to several popular formats.
- At a minimum, users are advised to use Aspera Connect (or the equivalent command line tool, ascp) for bulk downloads, rather than HTTP or FTP. Aspera provides faster bandwidth, a higher level of flow control, user level encryption, and the ability to download trees of components.
- We most strongly recommend the use of the SRA Toolkit to download data files directly. The individual utilities are able to resolve SRA accessions and initiate downloads automatically. The 'prefetch' utility is specifically provided for researchers that wish to download SRA data using a command line utility.

Related Documents

NCBI Large Data Download Best Practices

Notices

Reference herein to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government, and shall not be used for advertising or product endorsement purposes.

Software Version

This guide is current to SRA Toolkit version 2.5.x. Instructions for previous versions of the SRA Toolkit may be different from those provided in this guide. We recommend that users stay current with SRA Toolkit updates to benefit from feature additions and bug fixes.

Reference Compression

Compression by reference is a sequence alignment compression process for storing data. Compression by reference stores the difference in base pairs between sequence data and the segment(s) to which it is aligned. Throughout this document you will note that the

behavior and properties of reference compressed SRA data and conventional data differ significantly. Notably,

- The SRA Toolkit can output reference-compressed data as aligned sam and can perform pileup analysis.
- The SRA Toolkit requires internet connectivity in order to download reference sequences in order to process aligned SRA data.
- Only aligned data can be viewed in the NCBI Sequence Viewer.
- Aligned data cannot be filtered in the SRA Run Browser.
- Aligned data cannot currently be searched in SRA BLAST (this is actively being developed).

Download with Prefetch

The SRA Toolkit can be used to directly download SRA data files and reference sequences (see the "Reference Compression" section above). We strongly encourage users to use these methods to access SRA data as they are simple to use and they avoid many of the manual steps required by other methods (searching FTP directories, browsing and clicking, etc.).

The SRA Toolkit will have to be properly configured in order to access NCBI servers and download data. Recent versions of the Toolkit are packaged with a 'default' configuration that should work for most users. Please review the pros and cons for using the default configuration here. If the default configuration does not work for your installation, or you wish to customize aspects of file handling by the Toolkit (e.g., where downloaded files are stored locally), you will need to configure the Toolkit and then test it to confirm that it is operating as expected. Please email sra@ncbi.nlm.nih.gov if you have any problems configuring or using the Toolkit.

Prefetch

The 'prefetch' utility in the SRA Toolkit can be used to download SRA data and any required reference sequences in a single operation. Prefetch can use either HTTP (default) or ascp (if installed) to contact the SRA, resolve the accessions that you have specified, and then download the data. Prefetch can be used on single data files or to batch download several at a time. Below is an example prefetch command with the expected output. More information can be obtained on the prefetch documentation page and by executing 'prefetch --help'.

```
$ prefetch SRR390728
Maximum file size download limit is 20,971,520KB
2016-01-14T16:57:02 prefetch.2.5.7: 1) Downloading 'SRR390728'...
2016-01-14T16:57:02 prefetch.2.5.7: Downloading via fasp...
2016-01-14T16:57:08 prefetch.2.5.7: fasp download succeed
2016-01-14T16:57:08 prefetch.2.5.7: 1) 'SRR390728' was downloaded
successfully
2016-01-14T16:57:09 prefetch.2.5.7: 'SRR390728' has 25 unresolved
```

```
dependencies
2016-01-14T16:57:09 prefetch.2.5.7: 2) Downloading 'ncbi-
acc:GPC_000000394.1?vdb-ctx=refseq'...
2016-01-14T16:57:09 prefetch.2.5.7: Downloading via fasp...
2016-01-14T16:57:13 prefetch.2.5.7: fasp download succeed
2016-01-14T16:57:13 prefetch.2.5.7: 2) 'ncbi-acc:GPC_000000394.1?vdb-
ctx=refseq' was downloaded successfully
2016-01-14T16:57:13 prefetch.2.5.7: 3) Downloading 'ncbi-
acc:GPC_000000395.1?vdb-ctx=refseq'...
2016-01-14T16:57:13 prefetch.2.5.7: Downloading via fasp...
2016-01-14T16:57:15 prefetch.2.5.7: fasp download succeed
```

Note that the example file is reference-compressed and that prefetch automatically obtains the reference sequences required to extract data from the .sra file. If your Toolkit installation is not properly configured, or you elect to block the ability of the Toolkit to contact NCBI, you will then need to determine (1) if your downloaded dataset is reference-compressed, (2) if so, which references are required to access the data (see vdb-dump for an example of how to determine this), and (3) acquire the reference sequences manually here.

Other Toolkit utilities

All SRA Toolkit functions - most notably the 'dump' utilities that convert SRA data into other formats - are able to download data "on-the-fly" at runtime. This works like prefetch, as the tools will also automatically acquire all needed reference sequences. To invoke a Toolkit utility to download data as they are converted to your preferred format, simply execute the utility on an SRA accession rather than a local file. In other words, the command

```
$ fastq-dump --split-files SRR390728
```

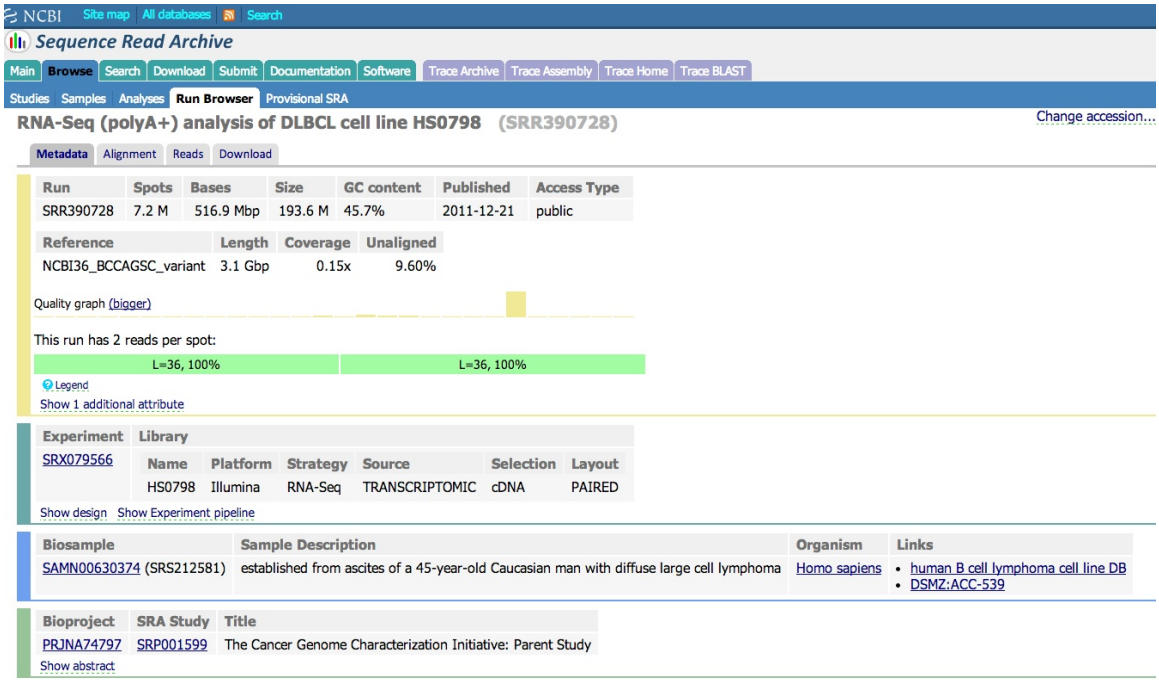
Is implicitly requesting that fastq-dump download SRR390728 and its references from the SRA and then output the data in fastq format. Conversely,

```
$ fastq-dump --split-files ~/Downloads/SRA/SRR390728.sra
```

Is instructing fastq-dump to operate on a local file that was previously downloaded from the SRA. In this case fastq-dump would still attempt to contact NCBI to obtain the references needed to convert the data to fastq (unless you have specifically configured the Toolkit to not contact NCBI).

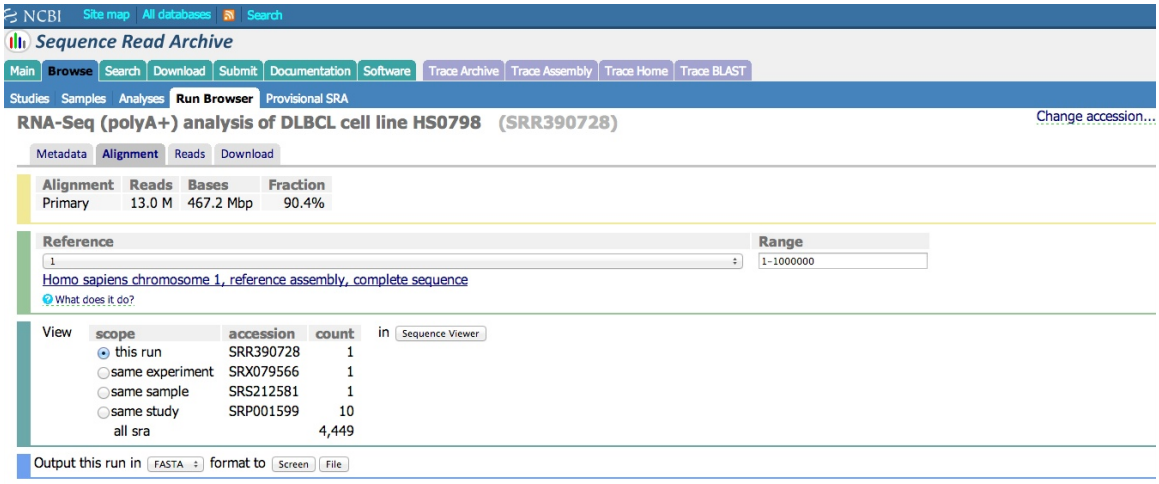
The Run Browser

The SRA Run Browser can display sequencing and instrumentation data on a given run. Typically the Run Browser is reached as a click through from an Entrez SRA Experiment report. Users may also navigate by entering a run accession (SRR, DRR, or ERR) directly in the Run Browser.

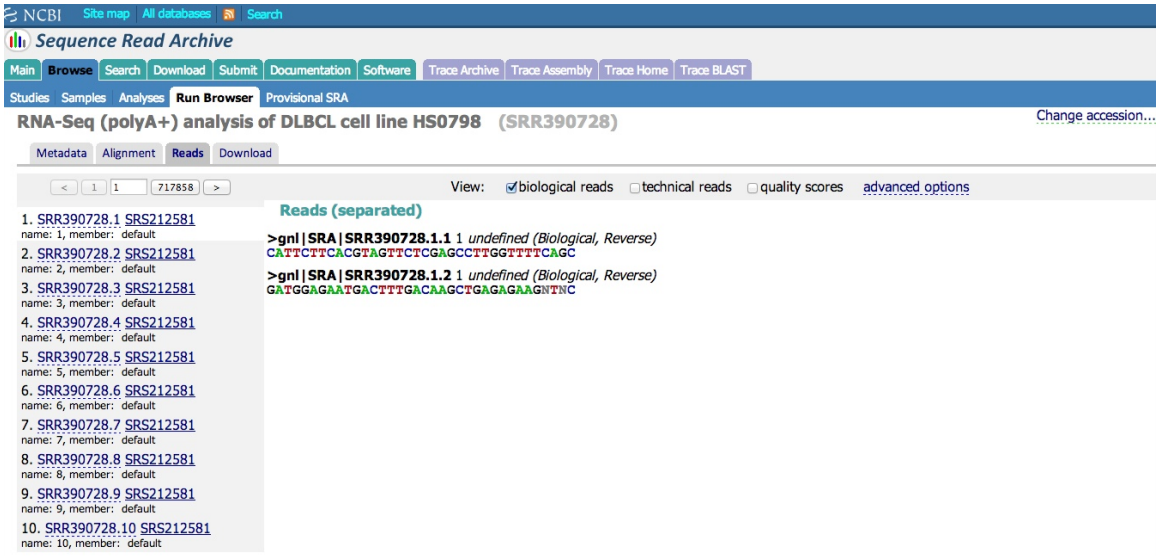


Viewing data in the Run Browser

Reference compressed (aligned) SRA data have an "Alignment" tab. Clicking on this tab will allow you to configure the NCBI Sequence Viewer to display the data aligned to a reference sequence.



To view the raw reads in a single Run, click on the "Reads" tab. Individual reads can be viewed and searched (see next section – note that only unaligned data can currently be searched). Various options can be applied using the "View" menu (e.g., display decimal quality scores, technical reads, etc.).



The Run Browser supports IUPAC single letter nucelotide codes (data submitted in color space are presented in base space; the SRA Toolkit can be used to download and output the data in color space, if required). Quality scores are presented in the Phred scale.

Filtering and Selection

The Reads tab in the Run Browser can be used to filter and search reads according to certain regular expression pattern matching:

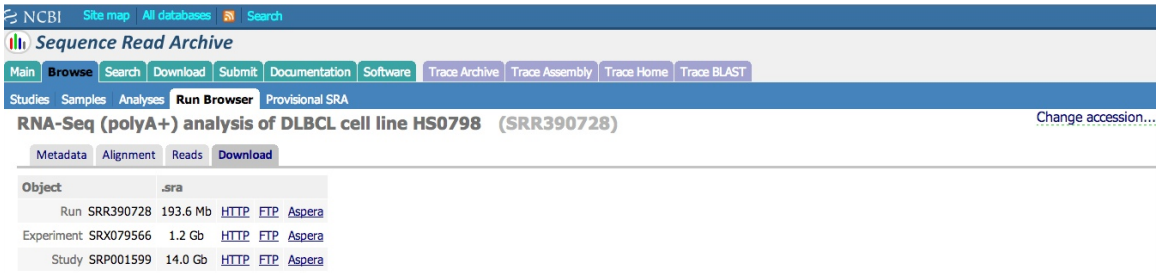
- Sequence substring: one of the biological reads for a spot should contain the substring. Examples: ATTGGA, ^ATTGGA, ATTGGA$, ATGDNNAT, and ATGGA&GCGC. The strings are case insensitive, and belong to either 2NA or 4NA alphabets. String length limited to 29 characters in 4NA alphabet (includes IUPAC substitution codes) or 61 characters in 2NA alphabet (ACGT only). Search is case insensitive and strings may be combined with boolean operators & | ! (AND, OR, NOT). See "SRA nucleotide search expressions" for more details.
- Name of a spot you are looking for. Example: EXWA4RL02G9Z6H
- Name of sample pool member, or "all" for all members. Example: M22_V2 will return all spots assigned to the sample pool member M22_V2 for run SRR031989.
- Spot Id. Example: 23

Please note that the filter searches across read boundaries within each spot. Thus, pattern matches within technical reads and across paired-end data boundaries will also be returned.

The filter provided in the Run Browser has limited functionality, but is quite fast if you are looking to quickly search a single Run for a defined sequence of interest. Please see the section below on SRA BLAST if you require more advanced searching or searches across multiple sequencing libraries.

Downloading Data from the Run Browser

Clicking on the "Download" tab in the Run Browser will present a selection of links that will allow you to download (1) an individual dataset (Run), (2) all datasets in a given sequencing library (Experiment), or (3) all datasets linked to a given project (Study). You are also provided with three download choices: Aspera (using the Aspera Connect plugin), HTTP (using your browser), FTP (using command line FTP or a client).



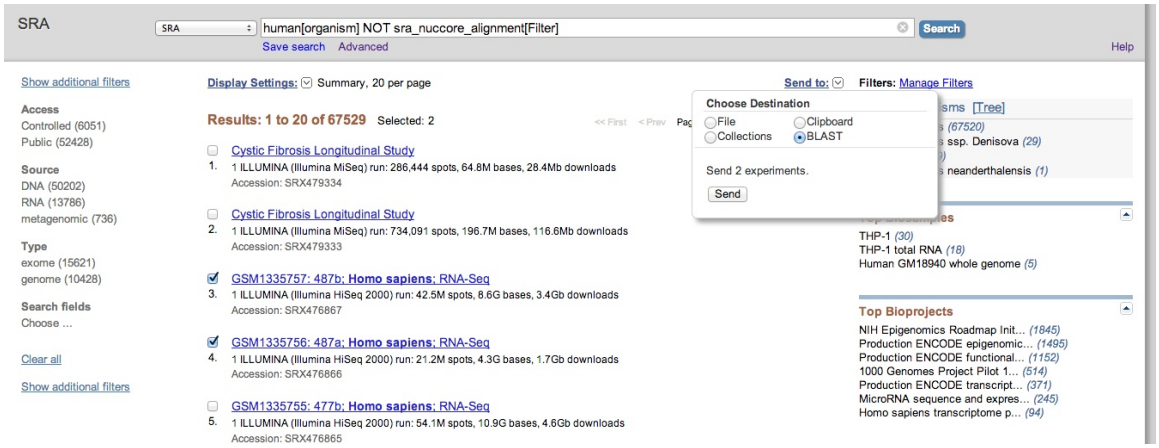
SRA BLAST

SRA BLAST can be used to for advanced searching of single or multiple sequencing libraries from the same or different projects. There are two ways to access SRA BLAST in order to build a "search space" from which you are attempting to pull matches to your sequence(s) of interest. Successful BLAST searches will lead you to a results / summary page that can be used to download reads of interest or be directed to the SRA Run Browser to further investigate or download the entire dataset.

Note that SRA BLAST currently has a limit of 2^{11} reads (approximately 2 billion) per search – attempts to add more than 2^{11} reads will result in an error and rejection of the search. Users that require more substantial search capability are advised to contact the SRA (sra@ncbi.nlm.nih.gov) to determine if other SRA BLAST tools might be of use.

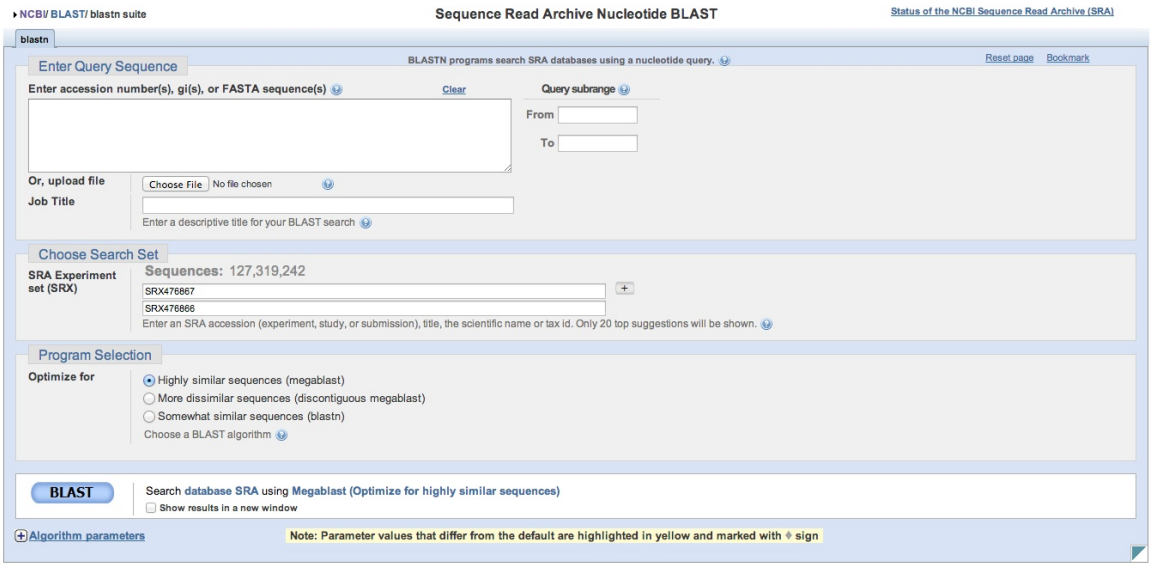
Sending Entrez results to SRA BLAST

After performing an Entrez query to restrict results to datasets of interest, you may use the "Send to" feature to select datasets of interest and send them to SRA BLAST.



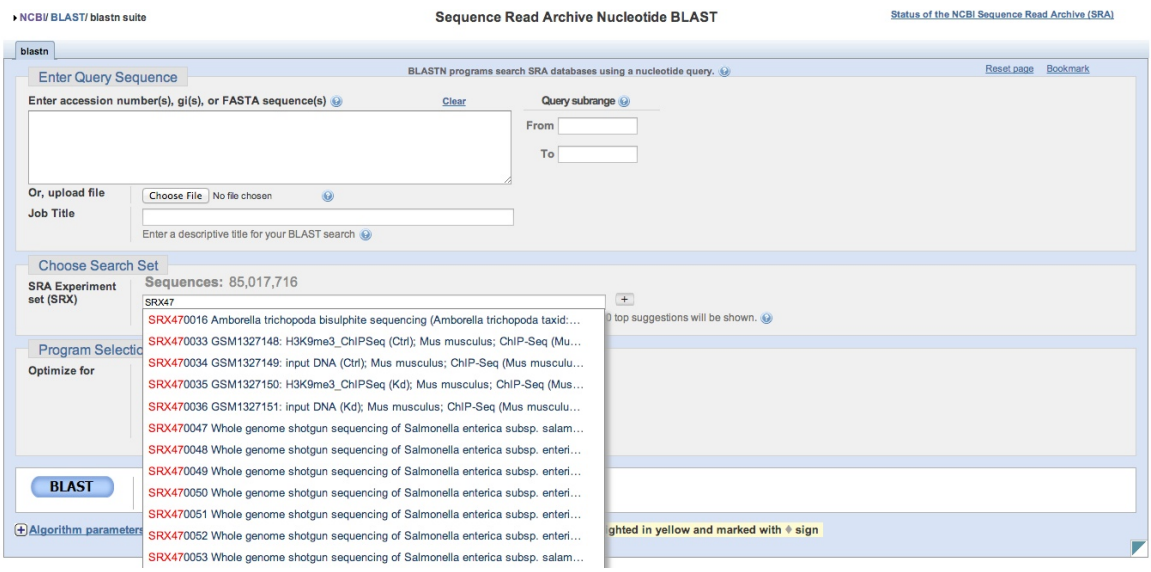
SRA-BLAST does not currently support reference compressed SRA datasets, so it is generally advised that you add the condition 'NOT sra_nuccore_alignment[Filter]' (as in the above example) to your queries to remove these datasets from the search results. Attempting to send incompatible datasets to BLAST will result in an error like the following:

If you believe that the data you are attempting to search against should be BLAST-able, but are not, please email sra@ncbi.nlm.nih.gov for assistance and advice. After successfully sending accessions to SRA BLAST, you are then able to input your sequence(s) of interest and perform the search.



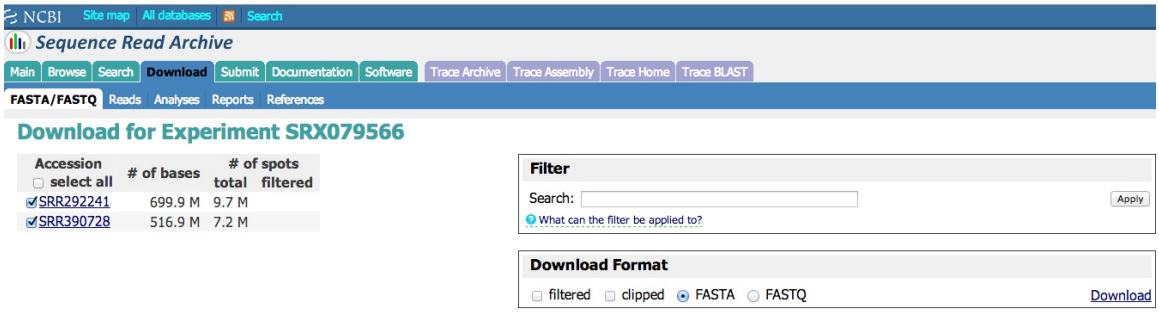
Building a search list in SRA BLAST directly

SRA BLAST can be accessed directly. You will then need to provide SRA Experiment (SRX, DRX, or ERX) accessions or use the autocomplete feature to help refine your search. You may enter 1 Experiment accession per line in the search list. The '+' button can then be used to add additional sequencing libraries to the search space. Note that a running tally of the number of sequences is presented above the list of accessions. Again, there is currently a limit of approximately 2 billion sequences per individual SRA BLAST query.



Direct downloading of fasta and fastq format

The SRA provides a tool that can be used to download data directly in fasta or fastq format. You must provide one or more SRA Experiment (SRX, DRX, or ERX) accessions in a comma-separated list. The same filtering inputs available in the Run Browser (described above) are available here to restrict the number of returned reads. Certain reads can also be clipped to remove low quality data from the download. If more than one Run accession in the list is checked, all data will be downloaded into a single fasta or fastq file, rather than per-accession files. Note that the output format of this tool is pre-defined and cannot be adjusted at the time of download. Users with specific formatting needs (e.g., for downstream analysis) are encouraged to use the SRA Toolkit to download and convert the data (described above).



Downloading metadata associated with SRA data files

SRA data files do not contain any information about the metadata (sample information, etc.) linked to the data themselves. The SRA provides a few tools to allow downloading of metadata in batch. Note that these tools differ from the Entrez Experiment, BioSample, and BioProject reports for a given dataset and may not contain all relevant metadata.

Viewing and downloading tabular metadata with the SRA Run Selector

The SRA Run Selector can be used to view metadata from one or more projects (SRA Study accessions – SRP, DRP, or ERP) entered into the field at the top of the page. The Run Selector provides a table view of library preparation and sample attribute metadata. The table can be filtered by sample attribute(s), accessions, etc. The "Get Metadata" button can be used to download a table (.txt, tab-delimited) of all or selected metadata.

NCBI **SRA RUN Selector** Change Study SRP001599 Change

Common attributes:

SRA Study	BioProject	analyte_type	gap_accession	is_tumor	study_name	Assay Type	Center Name	Platform	Conse
SRP001599	phs000235	RNA	phs000235	1	NCI Cancer Genome Characterization Initiative (CGCI)	RNA-Seq	BCCAGSC	ILLUMINA	public

PermaLink

	Runs	BioSamples	MBytes	MBases	
Total	20	10	13,363	22,364	Get Metadata
Filtered	20	10	13,363	22,364	+ - x
Selected	0	0	0	0	Show selected x Get Metadata

Run	BioSample	Sample Name	SRA Sample	DSMZ	body_site	cell_line	sex	study_subject_id	Library Name	MBases	MBytes
SRR292240	SAMN00630373	HS0685	SRS212580	ACC 47	pleural effusion fluid	DOHH-2	male	DOHH-2	HS0685	395	239
SRR292241	SAMN00630374	HS0798	SRS212581	ACC 539	peritoneal cavity fluid	DB	male	DB	HS0798	667	958
SRR292242	SAMN00630375	HS0841	SRS212582	ACC 32	pleural effusion fluid	Karpas 422	female	Karpas 422	HS0841	718	447
SRR292243	SAMN00630376	HS0842	SRS212583	ACC 583	lymph node	NU-DHL-1	male	NU-DHL-1	HS0842	681	526
SRR292244	SAMN00630377	HS0900	SRS212584	ACC 572	peritoneal cavity fluid	SU-DHL-6	male	SU-DHL-6	HS0900	766	691
SRR292245	SAMN00630378	HS0901	SRS212585	ACC 575	pleural effusion fluid	WSU-DLCL2	male	WSU-DLCL2	HS0901	896	1,053
SRR292246	SAMN00630379	HS1163	SRS212586	ACC 722	bone marrow	OCI-LY1	male	OCI-LY1	HS1163	1,775	1,386
SRR292247	SAMN00630380	HS1182	SRS212587	ACC 688	peripheral blood	OCI-LY7	male	OCI-LY7	HS1182	2,140	1,603
SRR292248	SAMN00630381	HS1183	SRS212588	ACC-528	bone marrow	OCI-Ly19-R1	female	OCI-Ly19-R1	HS1183	1,619	1,265
SRR292249	SAMN00630382	HS2047	SRS212589	ACC 579	cerebrospinal fluid	NU-DUL-1	male	NU-DUL-1	HS2047	3,462	1,454
SRR390726	SAMN00630382	HS2047	SRS212589	ACC 579	cerebrospinal fluid	NU-DUL-1	male	NU-DUL-1	HS2047	3,394	821
SRR390727	SAMN00630373	HS0685	SRS212580	ACC 47	pleural effusion fluid	DOHH-2	male	DOHH-2	HS0685	342	137
SRR390728	SAMN00630374	HS0798	SRS212581	ACC 539	peritoneal cavity fluid	DB	male	DB	HS0798	492	184
SRR390729	SAMN00630375	HS0841	SRS212582	ACC 32	pleural effusion fluid	Karpas 422	female	Karpas 422	HS0841	602	143
SRR390730	SAMN00630376	HS0842	SRS212583	ACC 583	lymph node	NU-DHL-1	male	NU-DHL-1	HS0842	442	180
SRR390731	SAMN00630377	HS0900	SRS212584	ACC 572	peritoneal cavity fluid	SU-DHL-6	male	SU-DHL-6	HS0900	501	185
SRR390732	SAMN00630378	HS0901	SRS212585	ACC 575	pleural effusion fluid	WSU-DLCL2	male	WSU-DLCL2	HS0901	557	238
SRR390733	SAMN00630379	HS1163	SRS212586	ACC 722	bone marrow	OCI-LY1	male	OCI-LY1	HS1163	1,021	634
SRR390734	SAMN00630380	HS1182	SRS212587	ACC 688	peripheral blood	OCI-LY7	male	OCI-LY7	HS1182	917	613
SRR390735	SAMN00630381	HS1183	SRS212588	ACC-528	bone marrow	OCI-Ly19-R1	female	OCI-Ly19-R1	HS1183	977	606

Command line access to metadata with the SRA Run Info CGI

Users can access the SRA Run Info CGI either through a browser or using a command line tool like wget.

```
wget -O <file_name.csv> 'http://trace.ncbi.nlm.nih.gov/Traces/sra/
sra.cgi?save=efetch&db=sra&rettype=runinfo&term=<query>'
```

As a parallel to the above example in the Run Selector,

```
wget -O ./SRP001599_info.csv 'http://trace.ncbi.nlm.nih.gov/Traces/sra/
sra.cgi?save=efetch&db=sra&rettype=runinfo&term= SRP001599'
```

Will return essentially the same information. Note that the CGI returns data in a comma-separated value (.csv) format, rather than the tab-delimited format of the Run Selector. The last component, <query>, can contain any set of Entrez parameters. Users may refine a search using Entrez and then copy over the search terms to a script for batch downloading. As an example, the search string

"Homo sapiens"[Organism] AND "cancer"[All Fields] AND "cluster_public"[prop] AND "strategy wgs"[Properties]

Will return these results in an Entrez search of the SRA. The equivalent Run Info CGI search would be

```
wget -O ./query_results.csv 'http://trace.ncbi.nlm.nih.gov/Traces/sra/
sra.cgi?save=efetch&db=sra&rettype=runinfo&term="Homo
sapiens"[Organism] AND "cancer"[All Fields] AND "cluster_public"[prop]
AND "strategy wgs"[Properties]'
```

Note that Entrez groups by Experiment accession, but that the CGI does not. It is, therefore, to be expected that the Run Info CGI will return a longer list of results than Entrez, but will still contain the same datasets.

Aspera Transfer Guide

Created: May 11, 2009; Updated: April 16, 2014.

Notice

```
Reference herein to any specific commercial products, process, or
service by trade name, trademark, manufacturer, or otherwise, does not
necessarily constitute or imply its endorsement, recommendation, or
favoring by the United States Government. The views and opinions of
authors expressed herein do not necessarily state or reflect those of
the United States Government, and shall not be used for advertising or
product endorsement purposes.
```

Overview

This document provides instructions on the use and installation of Aspera Connect for high throughput file transfer with NCBI. As the sizes of the datasets have increased, we have found that the traditional methods of *ftp* or *http* do not have the performance characteristics needed to support this load of data.

Requirements for large scale data transfer over the internet include high bandwidth, auto checksum, recursive copy, and security based on strong keys. NCBI has chosen to use a product from Aspera, Inc (Emeryville, CA) because of improved data transfer characteristics. FTP and HTTP access will continue to be available and are the default options for users without Aspera installed. Instructions are provided below for investigators to use this data transfer technology. NCBI also is open to using additional products with the appropriate performance characteristics.

Scope

This document is intended for users transferring large data files to and from NCBI. It applies to the Sequence Read Archive (SRA), dbGaP, and other archives where aspera download is enabled.

Aspera

Aspera Connect

Aspera Connect is software that allows download and upload via a web plugin for popular browsers on machines running Linux, Windows, and Macintosh. The software also includes a command line tool (ascp) that allows scripted data transfer. The software client is free for users exchanging data with NCBI.

Download and install Aspera Connect software from: http://downloads.asperasoft.com/connect2/

The website's download button will default to the detected operating system of the user's computer. To download for a different OS, click the link to 'See all installers'.

Please note the Requirements and consult with your network administrator to ensure transfers with aspera will not be blocked.

Aspera can be installed for individual users. However users of shared machine may want to have the software installed for all users by a system administrator.

The fasp Protocol

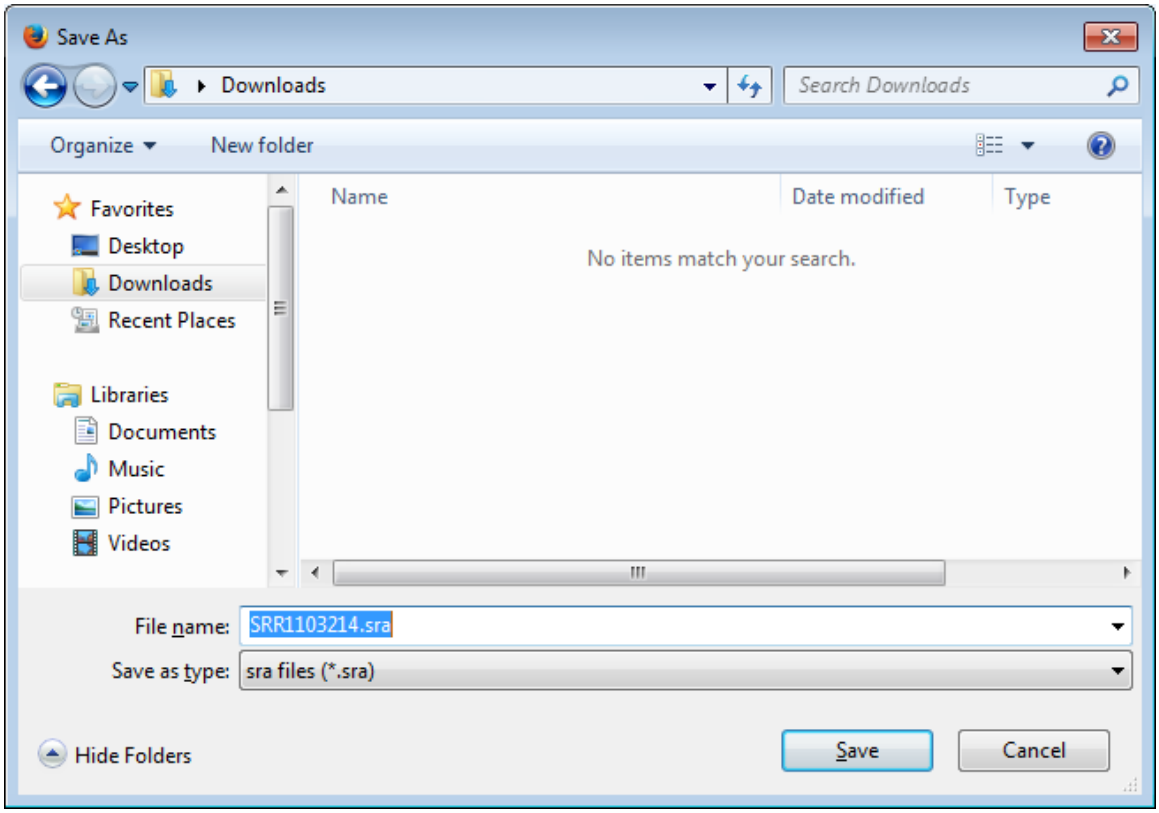
The FASP protocol from Aspera (www.asperasoft.com) uses UDP, eliminating the latency issues seen with TCP, and provides bandwidth up to 5 gigabit per second (Gbps) to transfer data. It has a restart capability if data transfer is interrupted midstream and is well behaved, so if there is other data traffic on your network connections, it will back off in order to avoid starving other protocols. We have seen effective throughput up to 800 megabits per second (Mbps) to a single site.

Downloading Data with Aspera Connect Browser Plugin

Once the plugin has been installed in your browser, you may download files or entire directories from NCBI using Aspera. Example: In your browser window, go to

http://www.ncbi.nlm.nih.gov/public/?/ftp/sra/sra-instant/reads/ByRun/sra/SRR/SRR292/SRR292241

Click 'SRR292241.sra' to begin saving the data. You will be prompted to select where the file is to be saved. For example:



You can download full directories or a single file at a time. The Aspera Connect plugin works with Chrome, Internet Explorer (IE), Safari, and FireFox web browsers. In some cases Aspera Connect may create a popup window to get a confirmation for file transfer and this popup window can be hidden behind your current web browser.

Using ascp to Download by Command Line

The command line program *ascp* is a utility delivered along with the Aspera Connect product.

```
ascp -i <asperaweb_id_dsa.openssh with path> -k1 -Tr –l100m
anonftp@ftp.ncbi.nlm.nih.gov:/<files to transfer> <local destination>
```

- -i <asperaweb_id_dsa.openssh with path> = fully qualified path & file name where

this public key file is located. This file is part of Aspera Connect distribution and is usually located in the 'etc' subdirectory.

- –T to disable encryption
- –k 1 enables resume of partial transfers
- –r recursive copy
- –l (maximum bandwidth of request, try 100M and go up from there)

Experiment with transfers starting at 100 Mbps and working up to 400 Mbps. Select the bandwidth setting that gives good performance with unattended operation.

- <files(s) to transfer> = names of files to transfer (including path)
- <local destination path> = location to store the downloaded data

Windows Executable Location

The *ascp* program for Microsoft Windows is located by default in "*C:\Program Files \Aspera\Aspera Connect\bin\ascp.exe*"

OS X Executable Location

The *ascp* Mac program location is */Applications/Aspera Connect.app/Contents/Resources/ ascp*

Linux Executable Location

The *ascp* Linux program location is /opt/aspera/bin/ascp

Additional information is available at the Aspera Web site: http://downloads.asperasoft.com/documentation/

Using ascp to Upload by Command Line

In order to use the Aspera upload service you will need to use a **private** SSH key, individual users can contact us at sra@ncbi.nlm.nih.gov to request an Aspera private key.

Upload Command

```
ascp -i <private key file> -T -l 100m <file(s) to transfer>
asp-****@upload.ncbi.nlm.nih.gov:<destination directory>
```

- -i < private key file > = fully qualified path & file name of the private SSH key
- –T to disable encryption
- –k 1 enables resume of partial transfers
- –l (maximum bandwidth of request, try 100M and go up from there)

Experiment with transfers starting at 100 Mbps and working up to 400 Mbps. Select the bandwidth setting that gives good performance with unattended operation.

- <files(s) to transfer> = names of files to transfer (including path)
- <destination directory> = deposit location of the uploaded data (typically either 'test' or 'incoming')

For password protected private keys, it is possible to run *ascp* in an autonomous, unattended manner that does not require repeated login. The environmental variable ASPERA_SCP_PASS can be used to store the private key path for a scripted series of bulk uploads.

Key Pairs

SSH keys are used for establishing secure connections to remote computers.

Submitters using a dedicated center account can find instructions for generating a key pair or converting PuTTY format private keys to OpenSSH format in this guide.

http://www.ncbi.nlm.nih.gov/books/NBK180157/

Requirements

Firewall Requirements

Your local firewall must permit UDP data transfer in both directions on ports 33001-33009 for the following IP ranges:

130.14.*.*

165.112.*.*

The firewall must also allow ssh traffic outbound to NCBI.

Troubleshooting

Here are some example commands demonstrating a test download.

Mac OS X:

```
ascp -T -l640M -i "/Applications/Aspera Connect.app/Contents/Resources/
asperaweb_id_dsa.openssh" anonftp@ftp.ncbi.nlm.nih.gov:1GB /tmp/
```

Linux:

```
ascp -T -l640M -i /opt/aspera/etc/asperaweb_id_dsa.openssh
anonftp@ftp.ncbi.nlm.nih.gov:1GB /tmp/
```

MS Windows:

```
C:\TEMP>"C:\Program Files (x86)\Aspera\Aspera Connect\bin\ascp.exe" -T -l640M -
i "C:\Program Files (x86)\Aspera\Aspera Connect\etc\asperaweb_id_dsa.openssh " anon
ftp@ftp.ncbi.nlm.nih.gov:1GB C:\Temp\
```

For additional assistance, please contact the NCBI Help desk at info@ncbi.nlm.nih.gov

When you are about to contact the NCBI Help desk please provide them some basic information like operating system, version of aspera connect, type of disk storage used for transferring files and the type of network connection your organization has to the internet.

If you have a Linux or MacOS X operating system you may run these commands and show us their output:

```
curl -o /dev/null ftp://ftp.ncbi.nlm.nih.gov/1GB
curl -o /dev/null http://www.ncbi.nlm.nih.gov/staff/beloslyu/large.tar
traceroute ftp.ncbi.nlm.nih.gov
```

First two commands download a 1GB file from NCBI using ftp and http protocols, the content is dumped to /dev/null. The third command will let us see the latency in your internet connection and possible congestions on the way to NCBI.

Another possibility is to make some test downloads from Aspera's demo server, for Linux the command line is:

```
env ASPERA_SCP_PASS=demoaspera ascp -L- -T -l100m
aspera@demo.asperasoft.com:aspera-test-dir-large/1GB /tmp/
```

Aspera Connect is a commercial product and program specific support is available from the manufacturer at http://asperasoft.com/support/

The currently up-to-date documentation for ascp can be found at http://downloads.asperasoft.com/en/documentation/8

Submitting to the SRA

Submission Quick Start Guide

Created: February 9, 2010; Updated: April 12, 2014.

Steps for SRA Submission

(Note: If you have already created BioSample(s) and BioProject(s) as a part of WGS, Genome or TSA submission, please use those for your SRA submission as well.)

1. Create a BioProject for this research
2. Create a BioSample submission for your biological sample(s)
3. Gather Sequence Data Files
4. Enter Metadata on SRA website
    a. Create SRA submission
    b. Create Experiment(s) and link to BioProject and BioSample
    c. Create Run(s)
5. Transfer Data files to SRA
6. Update Submission with PubMed links, Release Date, or Metadata Changes

(See Figure 1)

BioProject

The SRA Study has merged with the NCBI BioProject resource. If no BioProject yet exists for this research, one can be created by following the link to 'Submit New BioProject' on the Experiment entry page. If submitting data for an existing BioProject, the accession for the project can be entered in the provided text field. Please note that BioProjects bear an accession like PRJNA#. Incomplete projects bearing a temporary submission ID like SUB# will need to be completed before linking to the SRA Experiment. For more context, see Describing an Experiment. More information on submitting to BioProject is available here.

BioSample

Submitters will create new Samples through the BioSample Submission Portal. Samples created through BioSample submission will be linked by a reference in the Experiment portion of a SRA submission. Registered BioSamples have accessions like SAMN#. As with BioProject, incomplete submissions that have only SUB# IDs must be completed prior to creating an SRA Experiment. If the submission contains more than 25 BioSamples and less than 100 Gb of data, please use our batch submission interface here. The documentation for the batch submission can be found here. If you have more than 25 samples and over 100Gb of data please contact SRA staff at sra@ncbi.nlm.nih.gov.

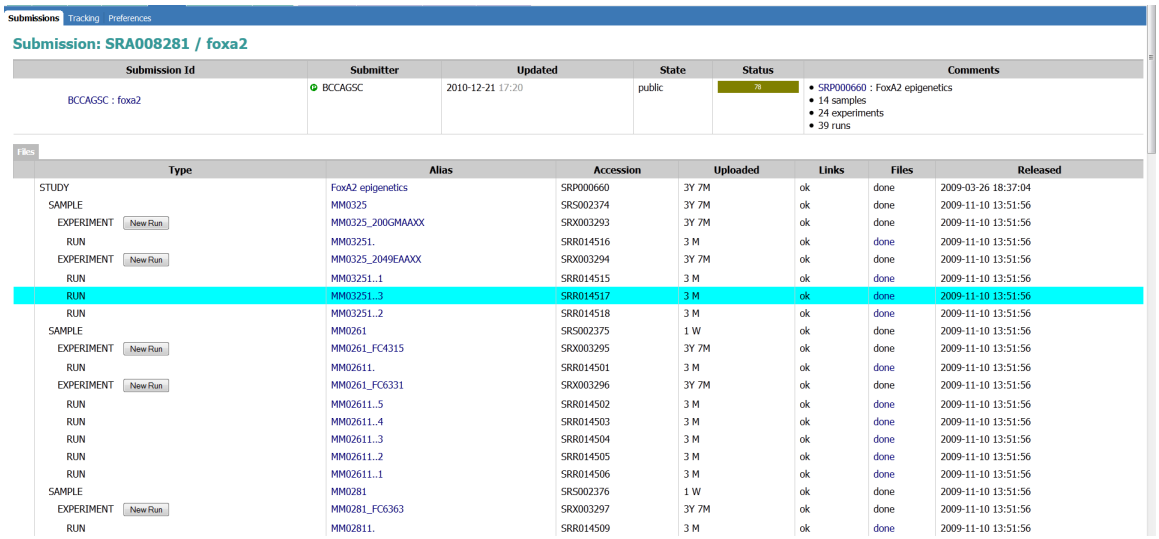


Figure 1 Example of a finished SRA Submission viewed from the Interactive Tool. Each Experiment references only one Sample, but many Experiments can reference the same Sample. An Experiment will have 1 or more Runs.

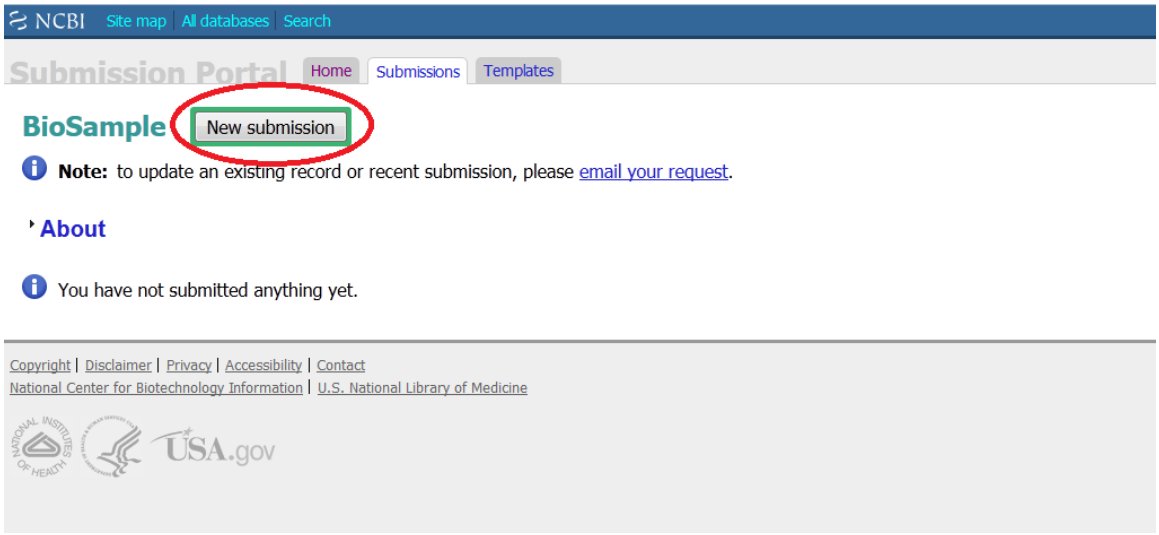


Figure 2 Once logged in to BioSample, click the 'New submission' button to begin creating a BioSample record.

Creating Samples

Each biological sample used in a study will be described by a BioSample record. A submission may contain many BioSamples. If samples were irreversibly pooled, a single BioSample record may describe the pooled components. Barcoded data files, on the other hand, should be demultiplexed prior to submission and a unique BioSample should be created for each barcoded sample; in other words, each BioSample must be linked to one

or more unique data files. If more than 24 hours have passed since completion of a BioSample submission and the sample has not received a SAMN# accession, check the BioSample submission for errors. If none are found please contact SRA at sra@ncbi.nlm.nih.gov.

(See Figure 2)

Login to the Sequence Read Archive

From the SRA Homepage:

Click the ***Submit*** tab.

Then login. (PDA and myNCBI have merged. You may log in with either by clicking on 'NCBI PDA'. If you do not have a PDA or myNCBI account already, one can be created. Alternatively, you may sign in with one of the 3rd party login options present after clicking on 'NCBI PDA'. If you have used a PDA account in the past but no longer see your previous SRA submissions, please contact SRA at sra@ncbi.nlm.nih.gov for assistance with your account view.)

(See Figure 3a)

(See Figure 3b)

Creating a New Submission

(See Figure 4)

Submission Alias and Comment

Alias – An ID used by submitters to track the submission of a set of Experiments and Runs. This field should be something that is used internally to refer to the project and makes sense to the submitter. Once saved, the Submission Alias cannot be changed. Like all Alias fields in SRA, this is not an indexed field and will not be visible to the public during normal usage of the database.

Example: *C. elegans resequencing project* (this field is NOT indexed in Entrez).

Submission Comment – area for submitter to enter a comment about the submission.

Example: *prepared with assistance by John Smith* (this field is NOT indexed in Entrez).

(See Figure 5)

Setting a Submission Release Date

A release date is required for all submissions. It is advisable to enter a release date before loading any data into a Submission. This will prevent accidental early release of data. Dates may be set for up to one year in the future in anticipation of a publication release

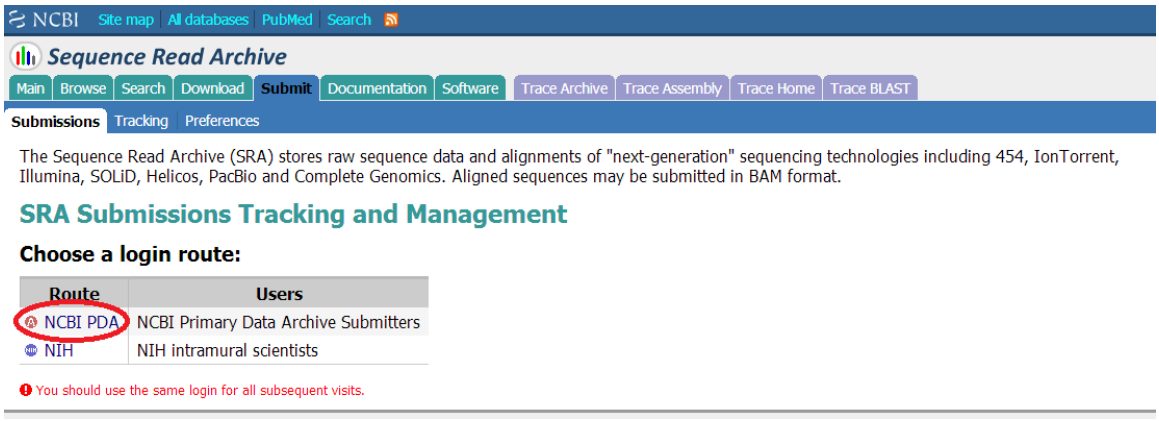


Figure 3a From the 'Submit' tab, click 'NCBI PDA' to login for Submission.

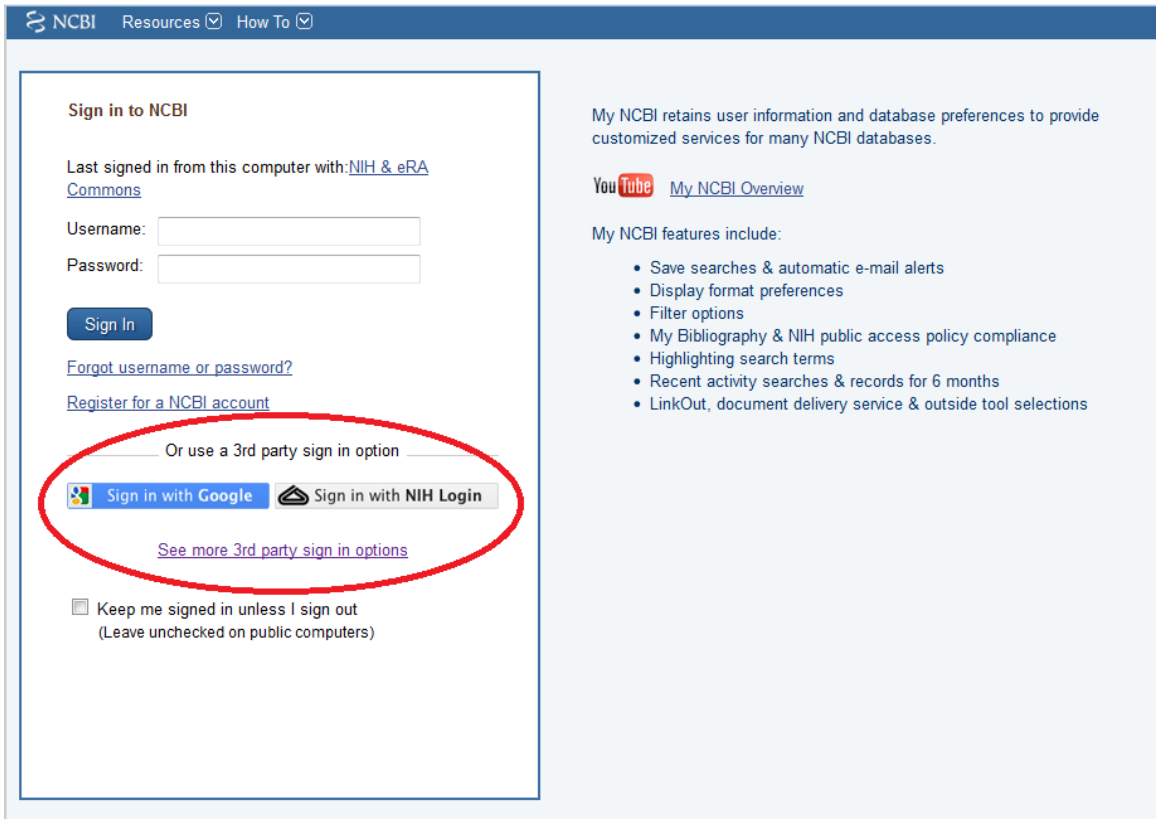


Figure 3b Alternative 3rd party log in options are available after clicking on 'NCBI PDA'.

date. They can be changed at any time by accessing your submission and changing the release date at the bottom of the page. This action does not require you to contact SRA.

(See Figure 6)

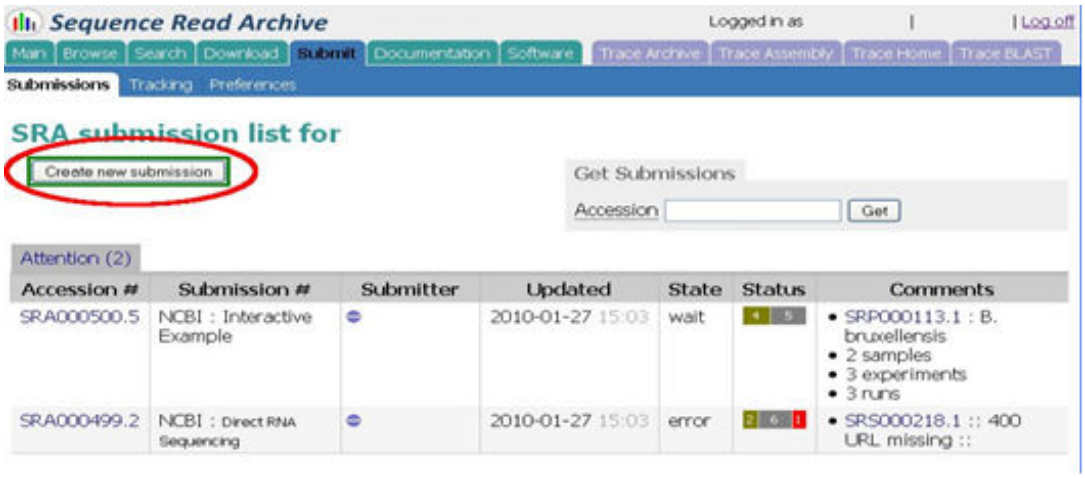


Figure 4 To start a Submission, click the 'Create new submission' button.

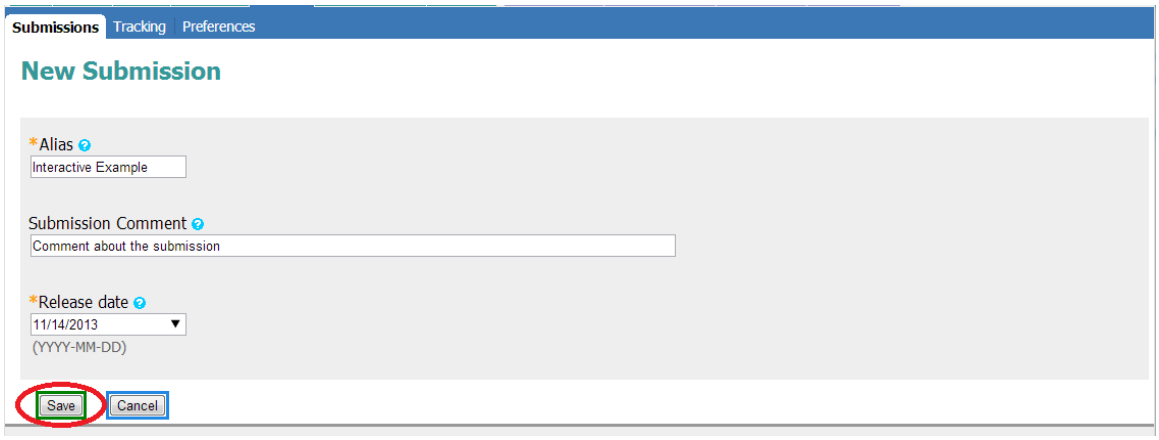


Figure 5 The Submission is not created until the 'Save' button is clicked.

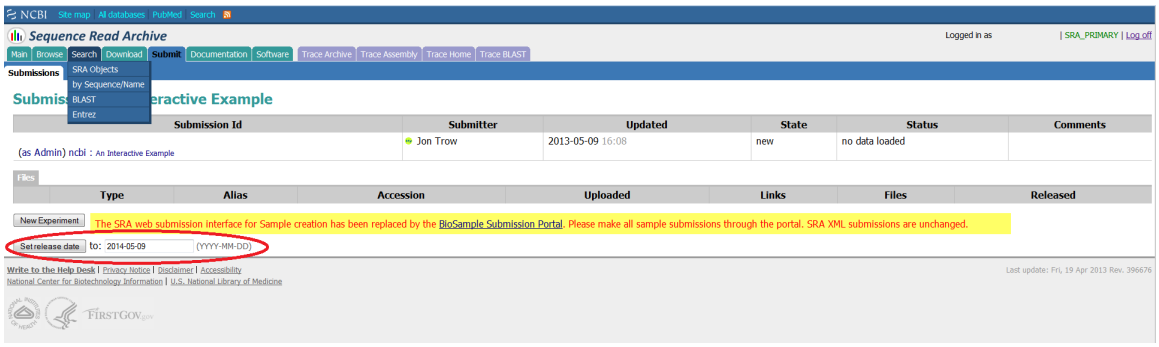


Figure 6 To save the date on which the submission is scheduled to be published/released to the public, enter a date in the box using a YYYY-MM-DD format, then click 'Set release date' The release date can be changed as long as the submission has not yet been made public

Figure 7 The Status Bar

Status

(See Figure 7)

The status bar provides a visual representation of the current state of the submission. **Done** (Dark Green) indicates the number of completed objects. **Wait** (Gray) further information or file uploads are needed. **Processing** (Light Blue- not shown) an object is currently being processed, if an object/file is processing for more than 48 hours, contact SRA at sra@ncbi.nlm.nih.gov. **Queue** (Dark Blue) the object will be processing when the pipeline is available. **Replaced** (Bright Green) an object/file was replaced by another. **Error** (Red) intervention is required, please contact SRA.

Experiment

Creating Experiments

An Experiment describes a sequencing library and instrument. An Experiment references 1 BioProject and 1 BioSample. (See Figure 8)

Describing an Experiment

Meta Information

Platform- This describes the sequencing platform used in the experiment.

Alias- An ID used as a reference for the user and archive. (Like all Alias fields in SRA, this is not an indexed field and will not be visible to the public during normal usage of the database.)

Title- A publicly viewable and formal title used to describe the Experiment.

BioProject Accession– Links this Experiment to a BioProject. If no BioProject yet exists for this research, one can be creating by following the link to 'Submit New BioProject' on the Experiment entry page. If the submitter is submitting data for an existing BioProject, the accession for the project can be entered in the provided text field. Please note that BioProjects bear an accession like PRJNA#. Projects with temporary accessions like SUB# will need to be completed before linking the SRA Experiment.

BioSample Accession- Links this Experiment to a BioSample. Like BioProject, links to SUB# are not accepted. Note that only 1 BioSample can be referenced in each Experiment. Thus, your submission will have at least 1 Experiment for each BioSample that you have

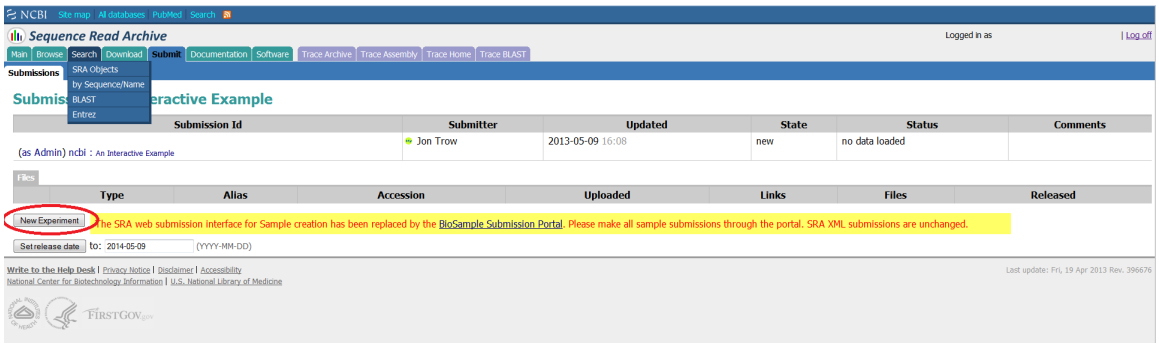


Figure 8 Click the 'New Experiment' button to begin creating an Experiment.

registered for the project. Please also note that it is possible to reference the same BioSample in mutiple Experiments. (See Figure 1)

Design Description- Describes the setup, experimental design, and goals of this Experiment.

Library

Additional descriptions of library terms can be found in Table 1 or the Glossary.

Name- Name of the Library that was sequenced

Strategy- Sequencing strategy used in the experiment

Source- Type of genetic source material sequenced

Selection- Method of selection or enrichment used in the Experiment

Layout- Configuration of the read layout. Paired, Fragment, etc.

Nominal Size (paired)- Size of the insert for Paired reads.

Nominal Standard Deviation (paired)- Standard deviation of insert size (typically ~10% of Nominal Size)

Table 1 List of available Experiment library descriptors.

Strategy	Sequencing strategy used in the experiment
WGA	Random sequencing of the whole genome following non-pcr amplification
WGS	Random sequencing of the whole genome
WXS	Random sequencing of exonic regions selected from the genome
RNA-Seq	Random sequencing of whole transcriptome

Table 1 continues on next page...

Table 1 continued from previous page.

Strategy	Sequencing strategy used in the experiment
miRNA-Seq	Random sequencing of small miRNAs
WCS	Random sequencing of a whole chromosome or other replicon isolated from a genome
CLONE	Genomic clone based (hierarchical) sequencing
POOLCLONE	Shotgun of pooled clones (usually BACs and Fosmids)
AMPLICON	Sequencing of overlapping or distinct PCR or RT-PCR products
CLONEEND	Clone end (5', 3', or both) sequencing
FINISHING	Sequencing intended to finish (close) gaps in existing coverage
ChIP-Seq	Direct sequencing of chromatin immunoprecipitates
MNase-Seq	Direct sequencing following MNase digestion
DNase-Hypersensitivity	Sequencing of hypersensitive sites, or segments of open chromatin that are more readily cleaved by DNaseI
Bisulfite-Seq	Sequencing following treatment of DNA with bisulfite to convert cytosine residues to uracil depending on methylation status
Tn-Seq	Sequencing from transposon insertion sites
EST	Single pass sequencing of cDNA templates
FL-cDNA	Full-length sequencing of cDNA templates
CTS	Concatenated Tag Sequencing
MRE-Seq	Methylation-Sensitive Restriction Enzyme Sequencing strategy
MeDIP-Seq	Methylated DNA Immunoprecipitation Sequencing strategy
MBD-Seq	Direct sequencing of methylated fractions sequencing strategy
OTHER	Library strategy not listed (please include additional info in the “design description”)
Source	**Type of genetic source material sequenced**
GENOMIC	Genomic DNA (includes PCR products from genomic DNA)
TRANSCRIPTOMIC	Transcription products or non genomic DNA (EST, cDNA, RT-PCR, screened libraries)
METAGENOMIC	Mixed material from metagenome
METATRANSCRIPTOMIC	Transcription products from community targets
SYNTHETIC	Synthetic DNA

Table 1 continues on next page...

Table 1 continued from previous page.

Strategy	Sequencing strategy used in the experiment
VIRAL RNA	Viral RNA
OTHER	Other, unspecified, or unknown library source material (please include additional info in the “design description”)
Selection	**Method of selection or enrichment used in the Experiment**
RANDOM	Random selection by shearing or other method
PCR	Source material was selected by designed primers
RANDOM PCR	Source material was selected by randomly generated primers
RT-PCR	Source material was selected by reverse transcription PCR
HMPR	Hypo-methylated partial restriction digest
MF	Methyl Filtrated
CF-S	Cot-filtered single/low-copy genomic DNA
CF-M	Cot-filtered moderately repetitive genomic DNA
CF-H	Cot-filtered highly repetitive genomic DNA
CF-T	Cot-filtered theoretical single-copy genomic DNA
MDA	Multiple displacement amplification
MSLL	Methylation Spanning Linking Library
cDNA	complementary DNA
ChIP	Chromatin immunoprecipitation
MNase	Micrococcal Nuclease (MNase) digestion
DNAse	Deoxyribonuclease (MNase) digestion
Hybrid Selection	Selection by hybridization in array or solution
Reduced Representation	Reproducible genomic subsets, often generated by restriction fragment size selection, containing a manageable number of loci to facilitate re-sampling
Restriction Digest	DNA fractionation using restriction enzymes
5-methylcytidine antibody	Selection of methylated DNA fragments using an antibody raised against 5-methylcytosine or 5-methylcytidine (m5C)
MBD2 protein methyl-CpG binding domain	Enrichment by methyl-CpG binding domain
CAGE	Cap-analysis gene expression
RACE	Rapid Amplification of cDNA Ends
size fractionation	Physical selection of size appropriate targets
Padlock probes capture method	Circularized oligonucleotide probes

Table 1 continues on next page...

Table 1 continued from previous page.

Strategy	Sequencing strategy used in the experiment
other	Other library enrichment, screening, or selection process (please include additional info in the “design description”)
unspecified	Library enrichment, screening, or selection is not specified (please include additional info in the “design description”)

Processing

This section varies with the sequencer selected. Please pay close attention to the answers provided in this section, as they may affect proper loading of data.

Pipeline

This section describes the processing pipeline used to generate the data. The Program and Version should be entered for each step in the processing pipeline. The sequencer platform software and version is expected for each experiment. Users can add additional lines to describe additional processing steps in the pipeline using the ‘Add’ button.

Saving the Experiment

In order to the save all the metadata, please click the “Save” button at the bottom of the page.

(See Figure 9)

For the public view of a completed Experiment, see Figure 10.

Links and Attributes

Used to add URLs, Entrez Links, or other Attributes in a key-value pair configuration. Table 2 is a list of the available NCBI database abbreviations and their descriptions. Here is a programmatic view of the database abbreviation.

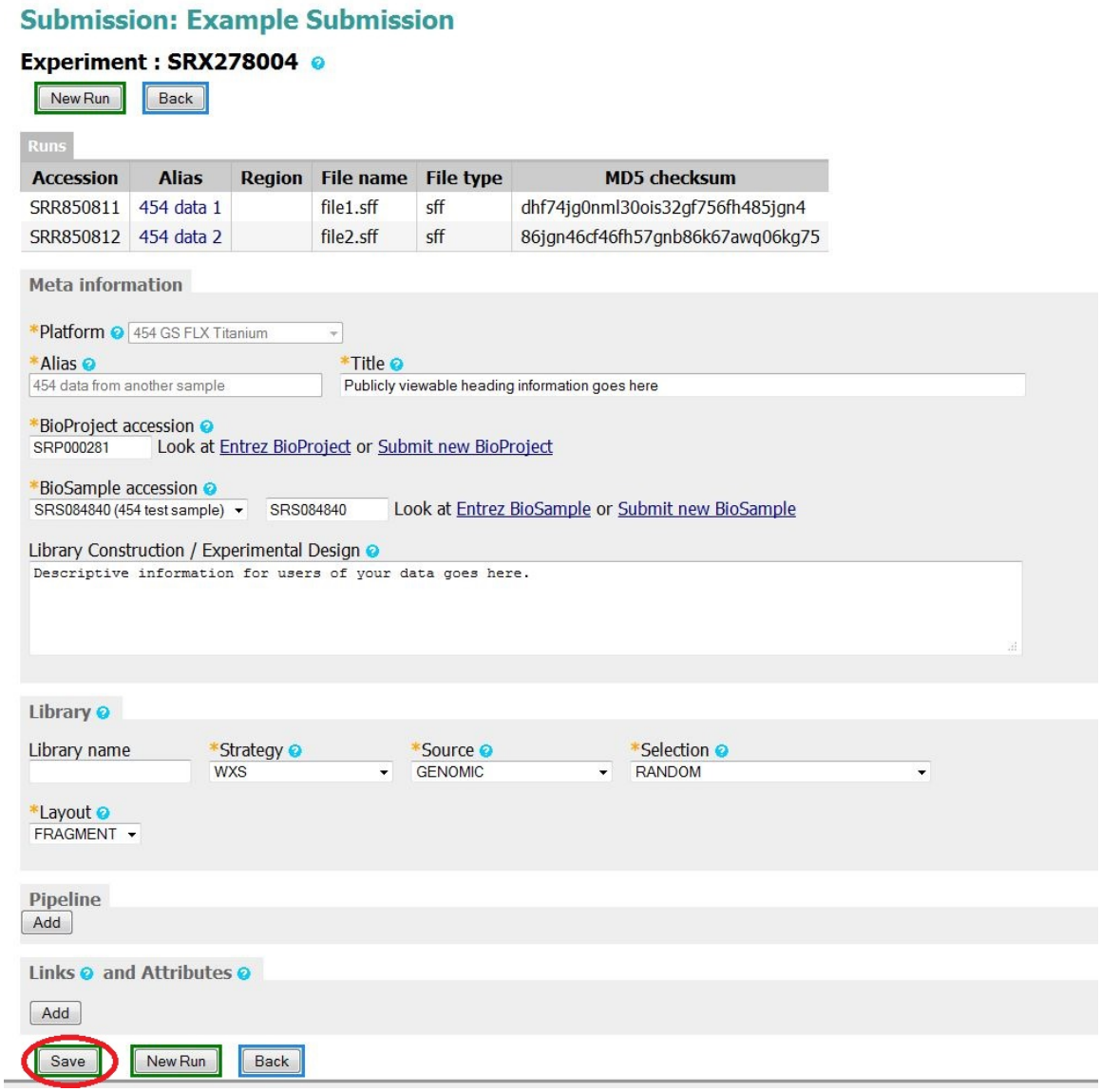


Figure 9 Click the 'Save' button to store the Experiment information. Saved Experiments can be updated as necessary.

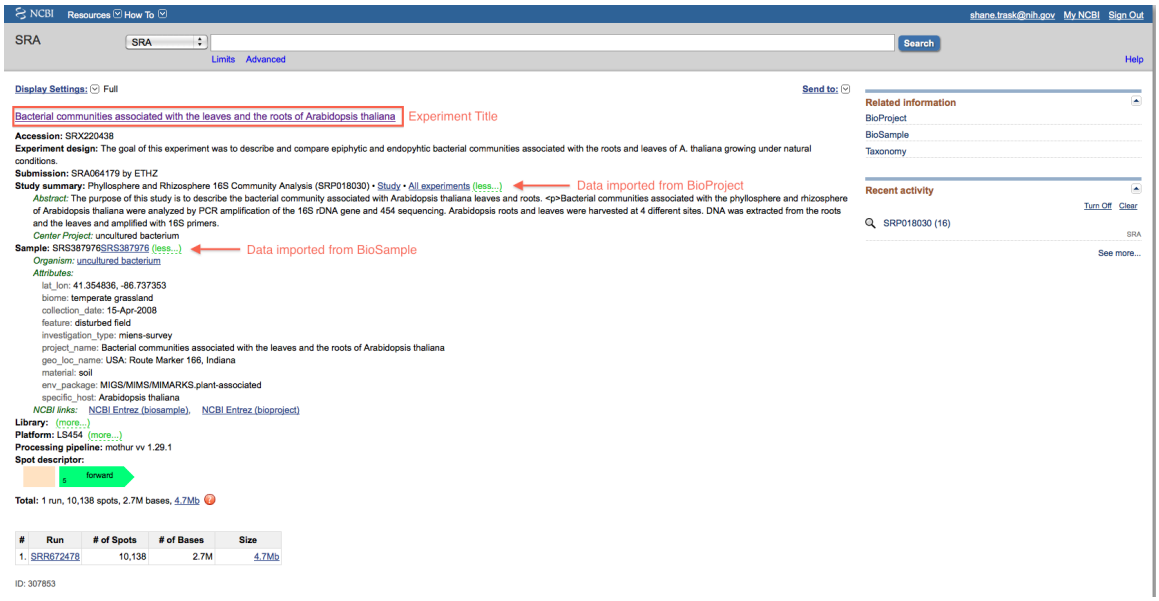


Figure 10 The public view of a completed Experiment-red boxes and text denote the source of displayed information.

Table 2 List of available NCBI database abbreviations

Abbreviation	Description	URL
pubmed	Pubmed database, please link using the pubmed ID	http://www.ncbi.nlm.nih.gov/pubmed
gtr	The Genetic Testing Registry (GTR) provides a central location for voluntary submission of genetic test information by providers.	https://www.ncbi.nlm.nih.gov/gtr/
nucleotide	The Nucleotide database is a collection of sequences from several sources, including GenBank, RefSeq, TPA and PDB.	https://www.ncbi.nlm.nih.gov/nucleotide
genome	This resource organizes information on genomes including sequences, maps, chromosomes, assemblies, and annotations.	http://www.ncbi.nlm.nih.gov/genome
assembly	Genome assembly organization and additional information.	http://www.ncbi.nlm.nih.gov/assembly
clinvar	ClinVar aggregates information about sequence variation and its relationship to human health.	http://www.ncbi.nlm.nih.gov/clinvar/
clone	Clone DB is a database that integrates information about clones and libraries, including sequence data, map positions and distributor information.	http://www.ncbi.nlm.nih.gov/clone/

Table 2 continues on next page...

Table 2 continued from previous page.

Abbreviation	Description	URL
gap	The database of Genotypes and Phenotypes (dbGaP) was developed to archive and distribute the results of studies that have investigated the interaction of genotype and phenotype	http://www.ncbi.nlm.nih.gov/gap/
dbvar	Database of genomic structural variation	http://www.ncbi.nlm.nih.gov/dbvar/
pmc	PMC is a free full-text archive of biomedical and life sciences journal literature at the U.S. National Institutes of Health's National Library of Medicine	http://www.ncbi.nlm.nih.gov/pmc/
epigenomics	Explore, view, and download genome-wide maps of DNA and histone modifications from our diverse collection of epigenomic data sets	http://www.ncbi.nlm.nih.gov/epigenomics/
probe	Probe Database is a public registry of nucleic acid reagents designed for use in a wide variety of biomedical research applications, together with information on reagent distributors, probe effectiveness, and computed sequence similarities.	http://www.ncbi.nlm.nih.gov/probe/
popset	A PopSet is a set of DNA sequences that have been collected to analyse the evolutionary relatedness of a population.	http://www.ncbi.nlm.nih.gov/popset/
geoprofiles	This database stores individual gene expression profiles from curated DataSets in the Gene Expression Omnibus (GEO) repository.	http://www.ncbi.nlm.nih.gov/geoprofiles/
snp	Database of single nucleotide polymorphisms (SNPs) and multiple small-scale variations that include insertions/deletions, microsatellites, and non-polymorphic variants.	http://www.ncbi.nlm.nih.gov/snp/

Run

Creating Runs

Runs describe the files that belong to the previously created Experiments. They specify the data files for a specific sample to be processed by SRA. Experiments may contain many Runs depending on how many sequencer runs were involved in data acquisition.

(See Figure 11)

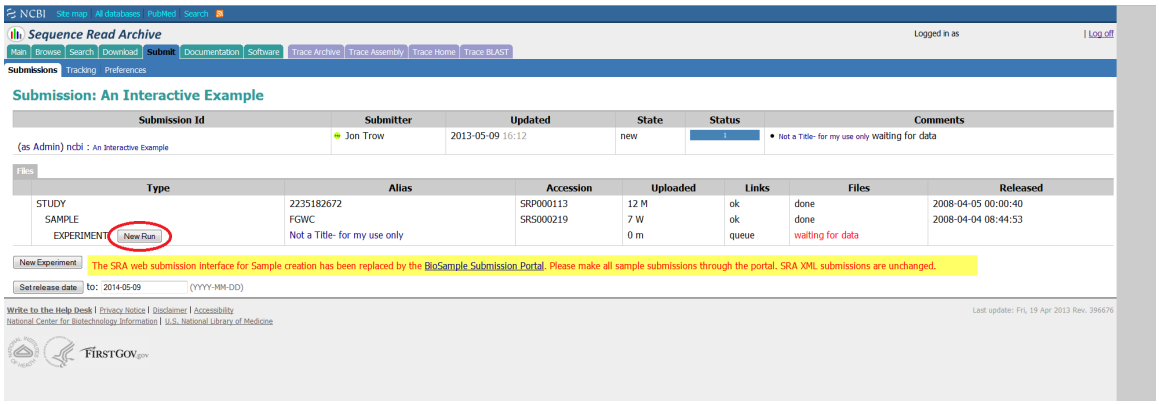


Figure 11 Click the 'New Run' button to the right of the Experiment for which a Run is needed. Each Experiment will have its own 'New Run' button.

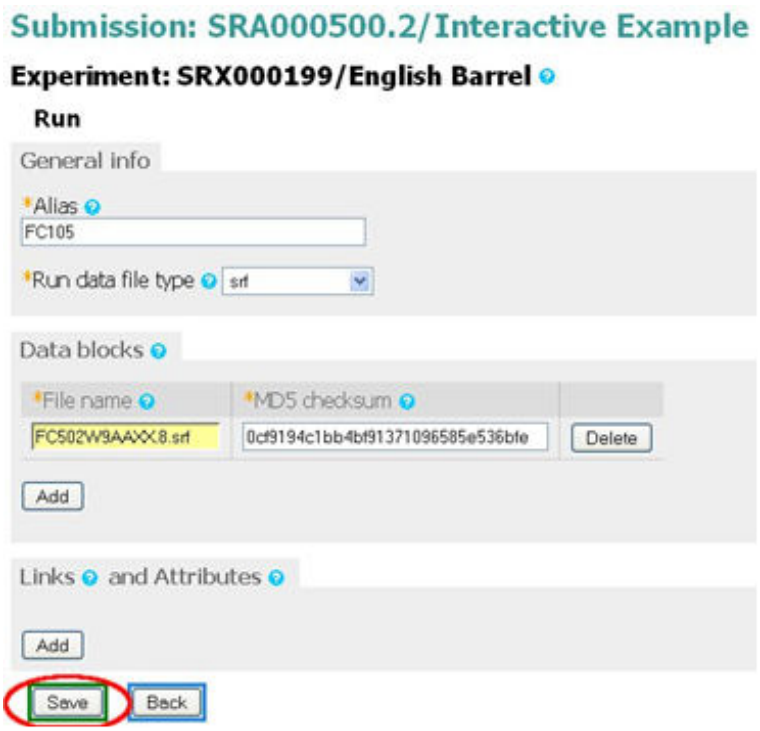


Figure 12 Click the 'Save' button to store the Run information. Runs can only be updated until data has been loaded for the Run. Once there is data in a Run, it will be locked from further updates. Contact SRA for changes to be made to locked Runs.

Describing a Run

Alias- An ID used as a reference for the user and archive. (Like all Alias fields in SRA, this is not an indexed field and will not be visible to the public during normal usage of the database.)

Run data file type- The storage format (srf, sff, fastq, etc.) of the sequence data being submitted. The SRA cannot accept FASTA format alone (FASTA/qual file pairs may be processed as FASTQ). More information about the file types currently accepted by the SRA can be found in the File Format Guide.

File Name- Name of the file transferred to the SRA including any file extensions. The SRA does not accept files compressed as .zip or .rar; it is NOT necessary to compress files transmitted to NCBI but files compatible with either gzip or bzip2 can be processed. Data files contained in a .tar archive need to be individually enumerated in a run. Note that original file names are not maintained after data is loaded to SRA. Each SRA Run produces a single .sra archive file that is amalgamated from all files listed in the Run.

MD5 checksum- A checksum or hash sum generated for the file listed in 'File Name' that is used to detect errors introduced through storage or transfer. SRA uses the file name and md5 checksum to track and link files to their proper Runs.

```
Unix- md5sum <file>
OS X- md5 <file>
```

Windows- Application required. Fsum Frontend (Please use Base16 for md5sum calculations) and WinMD5Sum are two possible options.

(See Figure 12)

Submission Checklist

- Does each biological sample have a BioSample record?
- Do you have at least 1 data file for each sample?
- Does each Experiment have at least 1 Run?
- Are file names entered exactly as they will be uploaded, including file extension?
- Is there enough information in titles and descriptions for other users to interpret the data? (Users cannot search based on "Alias" and will not see the "Alias" field during normal use)

Data Transfer

After the metadata is entered, data may be uploaded to the SRA.

Upload via FTP:

ftp://sra:password@ftp-private.ncbi.nlm.nih.gov/

(Windows Explorer may be used in Windows or an FTP client may be used in either Windows or OS X)

FileZilla is one of many free FTP clients that can be used by on PC or Mac.

Or from unix/linux/OS X command line

Address: ftp-private.ncbi.nlm.nih.gov

Login: sra

Password is provided in the browser once at least one Run is entered. If everything is correct files will be linked and loaded automatically.

Troubleshooting FTP

If you are having trouble with your FTP connection to NCBI, try

1. Setting passive mode rather than active mode
2. Ask your sysadmin to increase FTP buffer size to 32 MB
3. Try another host, or another platform (Windows instead of Unix)
4. Try another FTP client software:

Unix *ncftp (*http://www.NcFTP.com*)*

Windows *filezilla* (http://filezilla.sourceforge.net/)

If you still have trouble, please write us with the following details:

1. time of transfer (GMT or local time)
2. IP address of FTP client (the system you are transmitting from)
3. version of operating system software (*Unix* - uname -a, or cat /proc/version)
4. FTP account used
5. specific error messages (connection closed, etc)

Establishing a Center Account with SRA

A center account only needs to be established if you are going to be submitting data all year around on a regular basis and you are prepared to develop a programmatic method of generating XML. The pipeline will need to be kept up-to date with our schema updates so that your XML continues to stay valid.

To create a new Center, please provide the following information:

1. suggested center abbreviation (16 char max)
2. center name (full)
3. center URL
4. center mailing address (including country and postcode)
5. phone number (main phone for center or lab)
6. contact person (someone likely to remain at the location for an extended time)
7. contact email (ideally a service account monitored by several people)

Please click **here** to be taken to the Aspera Transfer guide, you will need to scroll down to the “Initiating an Account for Aspera Bulk Transfer for Centers Accounts” section.

Please write to sra@ncbi.nlm.nih.gov **for answers to submission questions.**

SRA Batch Submission Guide

Created: April 6, 2015.

Prerequisites for SRA Submission

(Note: If you have already created BioSample(s) and BioProject(s) as a part of WGS, Genome or TSA submission, please use those for your SRA submission as well.)

1. Create a BioProject for this research
2. Gather Sequence Data Files
3. Install Aspera Connect Instructions
    1 Download link for Aspera Connect

BioProject

The SRA Study has merged with the NCBI BioProject resource. If no BioProject yet exists for this research, one can be created by following the link to 'Submit New BioProject'. If submitting data for an existing BioProject, the accession for the project can be entered in the provided text field. Please note that BioProjects bear an accession like PRJNA#. Incomplete projects bearing a temporary submission ID like SUB# will need to be completed before linking to the SRA. More information on submitting to BioProject is available here.

Creating a New Submission

From the SRA Batch Submission portal:

Click the ***New Submission*** button.

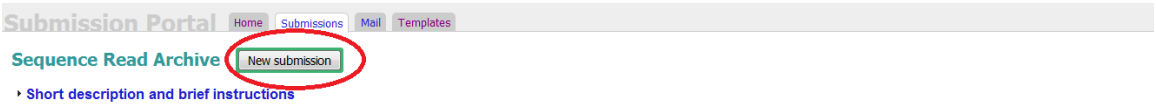


Figure 1 Creating a new submission

Steps in Submission Process

1. **"Submitter" tab**: Please fill in all requested fields.
    a You may click continue if all the fields are filled in.
2. **"General info" tab**: BioProjects are selected by typing in the BioProject name (either the PRJNA#### accession or the title)
    a. If you do not have a BioProject registered, create one here
    b. Set the release date

   c. You can select if you will register new BioSamples. Answering "Yes" will display 2 tabs for registering the new samples. Answering "No" will take you directly to the SRA metadata tab skipping the "BioSample type" and "BioSample attributes" tabs.
   d. Click Continue once you are satisfied with your selections
3. **"BioSample Type" tab** : Select the BioSample type that best describes your samples. This selection determines which BioSample attributes are required and which BioSample worksheet you will download at the "BioSample attributes" tab.
   a. You can look up the different BioSample types here.
   b. You will also be able to download the BioSample template from the above link as well, but you need to be in the SRA Batch submission interface in order to submit it.
4. **"BioSample attributes" tab** : Download the tab-separated BioSample worksheet and complete it for your samples. Upload the completed BioSample worksheet to this page.
   a. It is required that each sample in your worksheet has at least one unique attribute. Sample name, title, and description are not taken into account when searching for uniqueness.
   b. Additional attributes can be added by creating another column with a unique header
5. **"SRA metadata" tab** : Download and complete the tab-separated SRA metadata table. Upload the completed SRA metadata table to this page.
   a. If a sample was re-sequenced additional files can be added by populating additional columns with header "filename#", where # is the file count such as "filename1", "filename2".
      i Paired-end data with separate forward and reverse read files should have 2 filename columns so that both files are listed on the same row of the spreadsheet
   b. Additional attributes can be added by creating another column with a header
6. **"Files" tab** : is where you upload the files and this is done through the "Browse" button. The files can be transferred either through HTTP or Aspera plugin. It is advised to use Aspera plugin (http://downloads.asperasoft.com/connect2/) to perform the transfer.
   a. Please make sure plugins are not blocked on your browser otherwise Aspera plugin cannot run.
   b. If you are using Aspera plugin to transfer the files please pay attention to the top of the browser window as Aspera will prompt for permission.
7. **"Overview" tab** : Look over the submission and when you are satisfied "Click" submit button if everything is fine

Additional Details

BioSample

When filling out the BioSample spreadsheet please include as much metadata as you can. If you have filled out all required columns and your samples are still not unique, you can add your own columns which contain metadata to make each sample unique.

When entering dates please make sure to enter them in one of the following formats: “DD-Mmm-YYYY” (eg., 30-Oct-2010) or standard “YYYY-mm-dd” or “YYYY-mm” (eg 2010-10-30, 2010-10).

SRA MetaData

In order to specify multiple files derived from sequencing the same BioSample please header additional columns at the end of the SRA Metadata table as filename2, filename3 etc. and enter the names of your additional files in these columns. You can have a mixed number of files per BioSample.

In the SRA Metadata table you have the ability to add more metadata columns by entering a column header in a blank column and entering the data for that column. This can be any additional metadata you feel will add value for others such as Library Insert Size, Library Prep kit etc..

In the spreadsheet we require that specific terms be entered in the Library Strategy, Library Source and Library Selection columns. You can find the full list in Table 1.

Table 1 List of available Experiment library descriptors.

Strategy	Sequencing strategy used in the experiment
WGA	Random sequencing of the whole genome following non-pcr amplification
WGS	Random sequencing of the whole genome
WXS	Random sequencing of exonic regions selected from the genome
RNA-Seq	Random sequencing of whole transcriptome
miRNA-Seq	Random sequencing of small miRNAs
WCS	Random sequencing of a whole chromosome or other replicon isolated from a genome
CLONE	Genomic clone based (hierarchical) sequencing
POOLCLONE	Shotgun of pooled clones (usually BACs and Fosmids)
AMPLICON	Sequencing of overlapping or distinct PCR or RT-PCR products
CLONEEND	Clone end (5', 3', or both) sequencing
FINISHING	Sequencing intended to finish (close) gaps in existing coverage

Table 1 continues on next page...

Table 1 continued from previous page.

Strategy	Sequencing strategy used in the experiment
ChIP-Seq	Direct sequencing of chromatin immunoprecipitates
MNase-Seq	Direct sequencing following MNase digestion
DNase-Hypersensitivity	Sequencing of hypersensitive sites, or segments of open chromatin that are more readily cleaved by DNaseI
Bisulfite-Seq	Sequencing following treatment of DNA with bisulfite to convert cytosine residues to uracil depending on methylation status
Tn-Seq	Sequencing from transposon insertion sites
EST	Single pass sequencing of cDNA templates
FL-cDNA	Full-length sequencing of cDNA templates
CTS	Concatenated Tag Sequencing
MRE-Seq	Methylation-Sensitive Restriction Enzyme Sequencing strategy
MeDIP-Seq	Methylated DNA Immunoprecipitation Sequencing strategy
MBD-Seq	Direct sequencing of methylated fractions sequencing strategy
OTHER	Library strategy not listed (please include additional info in the "design description")
Source	**Type of genetic source material sequenced**
GENOMIC	Genomic DNA (includes PCR products from genomic DNA)
TRANSCRIPTOMIC	Transcription products or non genomic DNA (EST, cDNA, RT-PCR, screened libraries)
METAGENOMIC	Mixed material from metagenome
METATRANSCRIPTOMIC	Transcription products from community targets
SYNTHETIC	Synthetic DNA
VIRAL RNA	Viral RNA
OTHER	Other, unspecified, or unknown library source material (please include additional info in the "design description")
Selection	**Method of selection or enrichment used in the Experiment**
RANDOM	Random selection by shearing or other method
PCR	Source material was selected by designed primers
RANDOM PCR	Source material was selected by randomly generated primers
RT-PCR	Source material was selected by reverse transcription PCR
HMPR	Hypo-methylated partial restriction digest
MF	Methyl Filtrated
CF-S	Cot-filtered single/low-copy genomic DNA
CF-M	Cot-filtered moderately repetitive genomic DNA

Table 1 continues on next page...

Table 1 continued from previous page.

Strategy	Sequencing strategy used in the experiment
CF-H	Cot-filtered highly repetitive genomic DNA
CF-T	Cot-filtered theoretical single-copy genomic DNA
MDA	Multiple displacement amplification
MSLL	Methylation Spanning Linking Library
cDNA	complementary DNA
ChIP	Chromatin immunoprecipitation
MNase	Micrococcal Nuclease (MNase) digestion
DNAse	Deoxyribonuclease (MNase) digestion
Hybrid Selection	Selection by hybridization in array or solution
Reduced Representation	Reproducible genomic subsets, often generated by restriction fragment size selection, containing a manageable number of loci to facilitate re-sampling
Restriction Digest	DNA fractionation using restriction enzymes
5-methylcytidine antibody	Selection of methylated DNA fragments using an antibody raised against 5-methylcytosine or 5-methylcytidine (m5C)
MBD2 protein methyl-CpG binding domain	Enrichment by methyl-CpG binding domain
CAGE	Cap-analysis gene expression
RACE	Rapid Amplification of cDNA Ends
size fractionation	Physical selection of size appropriate targets
Padlock probes capture method	Circularized oligonucleotide probes
other	Other library enrichment, screening, or selection process (please include additional info in the “design description”)
unspecified	Library enrichment, screening, or selection is not specified (please include additional info in the “design description”)

Files

We encourage the use of the Aspera plugin for faster data transfer. If the files are packaged (tar) it will take longer to validate that all files are present in the submission as specified in the spreadsheet.

If Aspera Connect is not installed, then the upload will proceed through an HTTP connection. This is a much slower transfer method.

In the files tab please make sure you upload all the files before clicking “Continue” otherwise you will receive a warning.

Data Transfer

Data can only be uploaded through the browser, preferably by using the Aspera browser plugin.

Aspera can be downloaded from here:

http://downloads.asperasoft.com/connect2/

We recommend using Firefox and Windows Explorer as Chrome does not function well with Aspera.

If you need to upload your data using Aspera command line or FTP client, please write to sra@ncbi.nlm.nih.gov for answers to submission questions.

Submitting Sequence Data for a dbGaP project

Created: February 20, 2013; Updated: September 17, 2015.

Raw read data or aligned BAM file data for dbGaP projects is stored in a protected access area of the Sequence Read Archive (SRA) at NCBI. The sequence data will be linked to the study and samples registered with dbGaP.

Steps to submitting read data for a dbGaP study to SRA.

1. Register Study and Samples with dbGaP
2. Provide Submission information via XML or Spreadsheet to SRA
3. Upload Read Data Files
4. Confirm Data Counts

Submission Information

Contact Curator

You will be put in contact with a curator once your registration of study and samples has been completed.

Spreadsheet Submissions

Please let the curator assisting you know if you will be submitting by spreadsheet. To avoid errors in linking, study and sample information provided in the spreadsheet must match the sample and study information submitted to dbGaP. For this reason, the blank spreadsheet is not currently distributed. The SRA curator will provide a spreadsheet with the study and sample information entered for the submitter to complete. Please note that the library_ids must be unique, the title should describe the sequencing libraries individually to the users, and the design description should be a short materials and methods for the individual sequencing library. Additional instructions and descriptions are provided in the spreadsheet.

XML

The process is slightly different for studies that will use the Authorized Access system than those submitted for unrestricted access in SRA. For XML submission of dbGaP study data there will be a Submission, an Experiment per library, and a Run per BAM or production run. These XML files will be stored in a single tar archive and uploaded to an account at NCBI for the submitting center. The XML schemas are available here: http://trace.ncbi.nlm.nih.gov/Traces/sra/sra.cgi?view=xml_schemas

Examples of XML for submission with additional descriptions of key elements can be downloaded from the public FTP. Not all possible combinations of XML will be present.

ftp://ftp.ncbi.nlm.nih.gov/sra/examples/dbGaP_examples/

Linking to a Registered dbGaP Study

In the SRA <EXPERIMENT> XML:

```
<STUDY_REF accession="phs000000"/>
```

Linking to Registered dbGaP Samples

In the SRA <EXPERIMENT> XML:

```
<SAMPLE_DESCRIPTOR refcenter='phs000000' refname='submitted_sample_id'/>
```

The submitted_sample_id is synonymous with the SAMPID provided in dbGaP sample submissions.

Protected Data Transmission

Submitters must use the command-line program ascp from Aspera for transmission of data files. Aspera Connect is available for free from the download page here: http://www.asperasoft.com/en/downloads/8?list

Aspera Connect is free to use for submitters transmitting data to and from NCBI. Check with the local networking team to ensure UDP transfer is enabled for the following IP range: 130.14.*.* and 165.112.*.*

Aspera Key Pairs

Submitters will need to generate key pairs to use the Aspera upload account. The instructions for generating key pairs can be found in the Aspera Keys guide.
The **public** key only must be sent to the curator currently assisting you or the SRA Helpdesk (sra@ncbi.nlm.nih.gov) for access to the upload account.

ascp Usage

Example command line for dbGaP uploads:
ascp -i <key file> -Q -l 200m -k 1 <file(s) to transfer> asp-sra@gap-submit.ncbi.nlm.nih.gov:<directory>
-<directory> is either 'test' or 'protected'
-Do not set the -T option for protected transfers.

-<key file> private key full pathname must be used.

Additional information for using Aspera is available in the Aspera Transfer Guide.

Confirm Data Receipt

The curator can provide a report of files that were loaded. There is also a nightly report by samples provided on the dbGaP website. Change the accession “phs000710” in the address below to your study for the report.

http://www.ncbi.nlm.nih.gov/projects/gap/cgi-bin/GetSampleStatus.cgi?study_id=phs000710&rettype=html

SRA Submission Telemetry

Created: March 12, 2012; Updated: March 12, 2012.

Overview

This document shows the various methods by which submissions can be tracked interactively and programmatically.

The submitter is responsible for making sure that all components of a submission have been delivered to NCBI. While NCBI works to operate a smooth submission process and does correspond with submitters about problems, the submitter is responsible for repairing errors and resubmitting replacement components when necessary. Various channels of submission telemetry equip submitters with information and tools for them to complete their submissions in a correct and timely way.

Related Documents

SRA Quick Start Guide

Submission Model

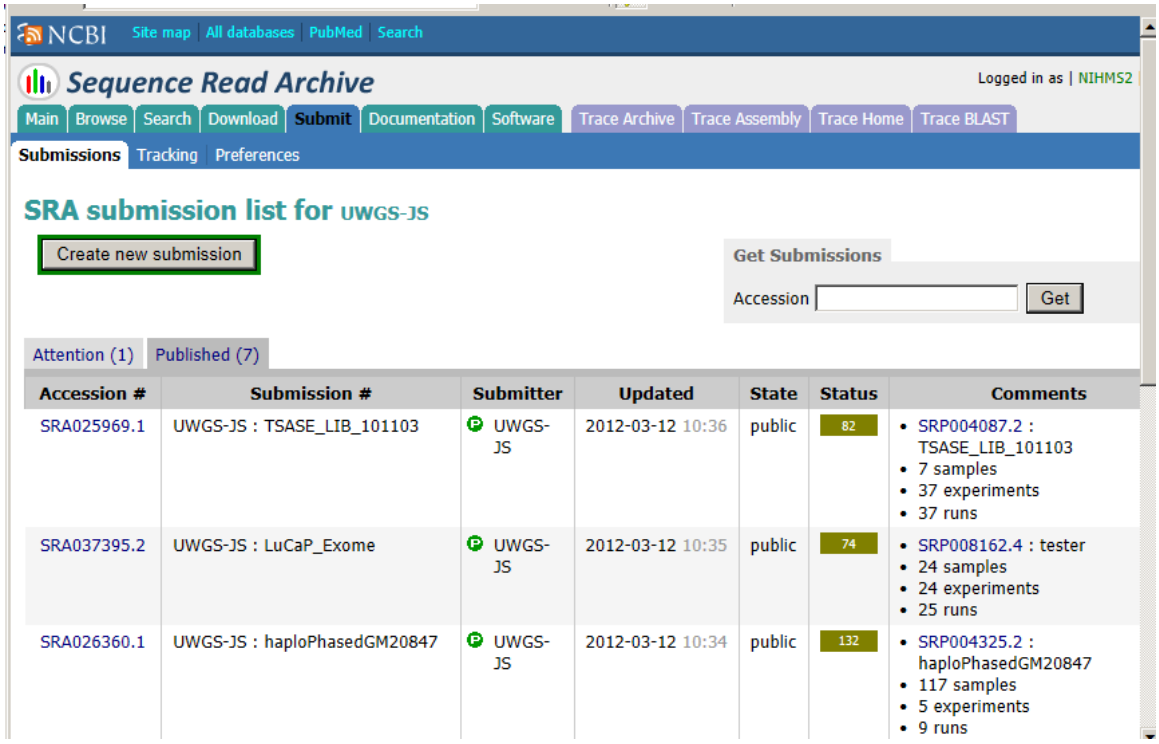
SRA submissions take place at two levels: metadata described interactively through web page forms or in bulk submission through xml, and content data encoded in recognizable file formats or content data in tar archive files of recognized files. Incoming data may be compressed or uncompressed. Metadata and content may arrive at the SRA at different times (asynchronous delivery). The purpose of submission telemetry is to empower the user to monitor the progress of their submissions at several points in the submission workflow.

NCBI operates a dual SRA submission interface: interactive and batch.

Interactive Telemetry

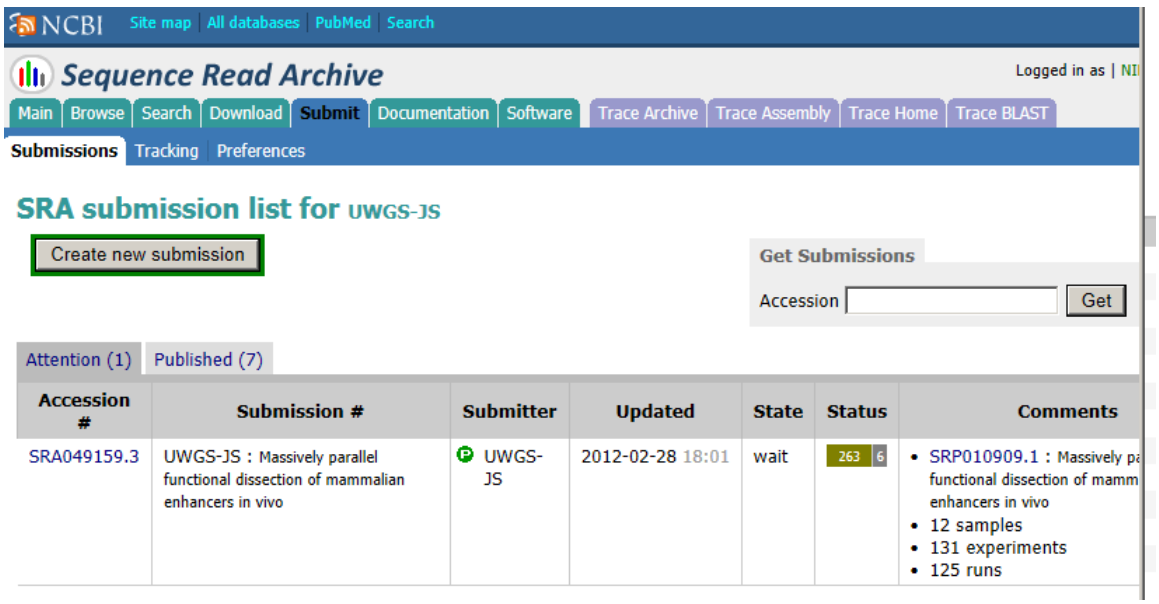
Interactive metadata tracking using the submission web tool

The interactive submission tool shows you the status of each component in your submission. For individual account submitters or those who use ftp exclusively, this is the only method for monitoring progress of your metadata submission. Completed submissions are listed under the “Submissions-Published” tab. The submission components will be displayed in green if they have been loaded.



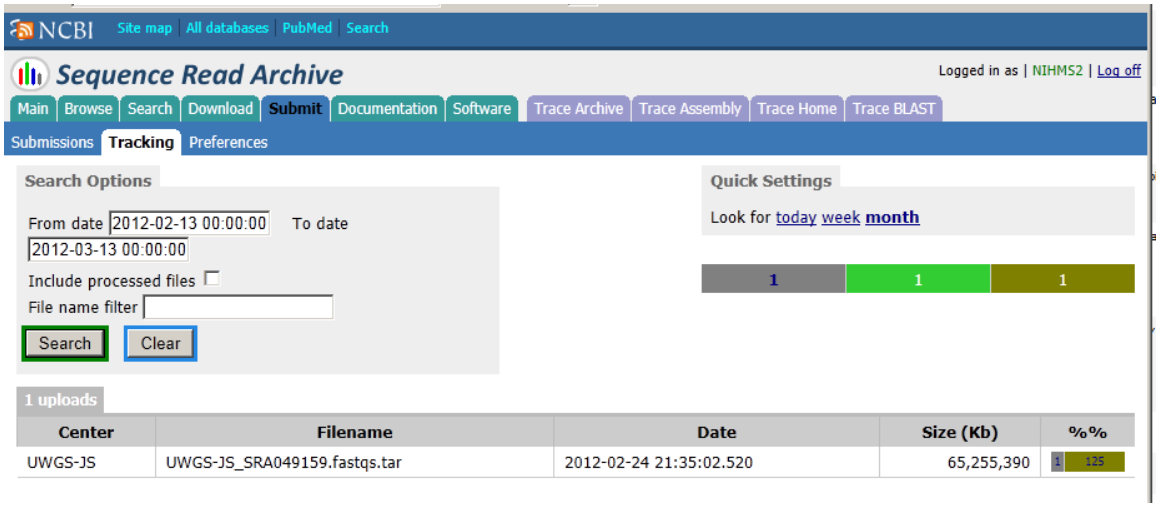
Submissions that need attention are listed under the "Submissions-Attention" tab. Problems are color coded as follows:

- Green – metadata component has been loaded
- Grey – metadata component has been received but is not linked to any data
- Red – error detected in metadata component, cannot load it. A portion of the error stream from the attempt to load the component will be displayed under the "Comments" column.



Interactive files tracking using the submission web tool

The status of SRA content files can be viewed through the "Tracking" tab of the interactive submission tool. This lists the files that are not yet loaded. The default reporting period is the most recent week, be sure to set the search date options for another time period.



The components will be displayed in green if they have been loaded. Problems are coded in the following color scheme:

- Green – content component has been linked and loaded into the SRA
- Grey – content component has not been linked to SRA metadata so is not loaded
- Red – error detected in content component so is not loaded

Limits to interactive telemetry

The submission telemetry displayed in the interactive submission tool may be limited in the following ways:

- A submission containing more than 1000 components cannot be completely displayed.
- A large number of submissions cannot be effectively tracked this way because of the large number of web pages that need to be examined.
- Submissions of high granularity (one component per submission) cannot be effectively tracked this way.

Batch Telemetry

The batch submissions telemetry stream consists of the following objects updated daily:

- Accessions tab file that shows current status of submitted SRA metadata components
- Metadata xml annotated with accessions assigned during the submission process
- Files tab file showing the current state of content data files sent to the SRA

- Submission area space usage report for both open and protected SRA submission channels

The telemetry stream can provide input to the "roundtrip" processing module for a submitters laboratory information management system (LIMS). Newly submitted documents can be downloaded to obtain the accessions assigned by NCBI. Documents can be downloaded in order to compare with internal state to make sure that the version at NCBI is current. Documents can be inspected to see that certain modification operations succeeded. A LIMS can track submissions to NCBI and generate reports that can be used to compare against the submission telemetry stream from NCBI.

Batch accessions status tracking with tab files

The Accessions Report is a list of SRA metadata objects and their status. This report is a tab delimited file called *SRA_Accessions* found within the metadata dump file:

upload@ncbi.nlm.nih.gov://asp- /outgoing/Files/NCBI_SRA_Metadata_Full_*_20110101.tar.gz

upload@ncbi.nlm.nih.gov://asp- /outgoing/Files/NCBI_SRA_Metadata_*_20120312.tar.gz

The fields are defined as follows:

Tag	Definition	Values	Units or meaning
accession	accession of the metadata object as assigned by NCBI	SRP, SRS, SRX, SRR, SRZ	
submission	submission containing the metadata object		
status	status of the metadata object in the archive	live	The object is indexed and available for retrieval
		suppressed	The object has been removed from indexing but can still be retrieved. This state usually reflects objects that have been superceded by successor objects.
		unpublished	The object has not been published, or, it was returned to an unpublished state after being published.
		withdrawn	The object has been redacted from the Archive. This state reflects rare situations where data was inappropriately released and copies in the Archive must be completely removed.

Table continues on next page...

Table continued from previous page.

Tag	Definition	Values	Units or meaning
updated	The date of last update of the object		ISO date
published	The date of the initial publication (release) of the object or its re-publication.		ISO date
received	The date at which the Archive received the data from the submitter.		ISO date
type	The object's document type	STUDY	
		SAMPLE	
		EXPERIMENT	
		RUN	
		ANALYSIS	
center	Short name for the submitting center.		
visibility	Whether the object has been archived at the open SRA no usage restrictions (public), or at the controlled access SRA (usage restrictions in place, the user must apply for access to the data). Note that visibility is orthogonal to the publication or embargo status of the data.		
alias	The submitter's name for the object		
md5sum	The MD5 checksum of the metadata object.		This value is computed in a canonical way, see below section.

Batch metadata tracking with xml files

Files of current metadata annotated with assigned accessions and any submission-time transformations are generated monthly with a daily incremental version. This is deposited into the open aspera account of the submitter.

upload@ncbi.nlm.nih.gov://asp- /outgoing/Files/NCBI_SRA_Metadata_Full_*_20110101.tar.gz

upload@ncbi.nlm.nih.gov://asp- /outgoing/Files/NCBI_SRA_Metadata_*_20120312.tar.gz

Batch files tracking with tab files

A file containing the current state of content data file transfers and loading is generated monthly with a daily incremental version. This is deposited into the open aspera account of the submitter. A public version of this file is not provided because of its prerelease focus.

upload@ncbi.nlm.nih.gov://asp- /outgoing/Files/NCBI_SRA_Files_*_20120312.gz

upload@ncbi.nlm.nih.gov://asp- /outgoing/Files/NCBI_SRA_Files_Full_*_20120229.gz

The tab file has the following format:

Tag	definition	Values	Units or meaning
realm	Whether the data have been submitted to the open SRA for eventual unrestricted use, or through to the protected SRA for authorized access.	Open	After release, data are publicly accessible and used without restriction.
		Protected	Data deposited into inner firewall, after release only accessible through authorized access credentials.
upload_id	NCBI upload tracking id.		This id is sequential in order of uploads.
upload_date	NCBI upload tracking date.		Sometimes this date is curated to the original NCBI receipt date rather than the date at which the file entered the system.
file_name	file name of extracted file		
file_size	file size of extracted file		bytes
file_md5sum	file checksum of extracted file using MD5 method		
upload_name	file name of upload package this file was found in (or = if same as file_name)		
upload_size	upload file size (or = if same as file_size)		bytes

Table continues on next page...

Table continued from previous page.

Tag	definition	Values	Units or meaning
upload_md5sum	upload file checksum using MD5 method (or = if same as file_md5sum)		
file_status	final status of extracted file	Done	Processing of the file is complete
		Error	An error was encountered
		Failed	The processing of the file failed
		Loaded	Content was loaded into the SRA
		Obsolete	File has been marked as not needed
		Received	File has been received only
		replaced_by_<upload_id>	File has been replaced by another
file_type	computed file type of extracted file	BAM	
		BZIP2	
		DATA	
		EMPTY	
		FASTQ	
		FLI	
		GZ	
		HDF5	
		HTML	
		MSOFFICE	
		RAR	
		SFF	
		SHELL	
		SHORTCUT	
		SRF	
		SYSTEM	
		TAR	

Table continues on next page...

Table continued from previous page.

Tag	definition	Values	Units or meaning
		TEXT	
		UNKNOWN	
		XSL	
		ZIP	
load_date	date of content load		ISO date
file_error	error message from file tracking system	bad_chunk_at_offset_<file_offset>	SRF file integrity problem
		bad_read_header_length	SFF file integrity problem
		bunzip2_error	bzip2 decompression error
		copy_error	internal error
		corrupt_at_offset_<file_offset>	SRF file integrity problem
		corrupt_file	file could not be processed
		Duplicate	duplicate file
		duplicate_to_upload_<upload_id>	duplicate file
		empty_file	zero length file
		failure_during_copy	internal error
		file_changed_during_copy	file was written to or removed by submitter
		gunzip_error	gzip decompression error
		missing_read_data	file does not have read data
		repeated_file_in_archive	internal error
		size_changed_during_copy	file was written to or removed by submitter
		tar_error	untar error
		tar_missing_terminating_blocks	tar file is truncated
		truncated_file	file is truncated
		unzip_error	decompression error
		upload_file_not_found	internal error

Table continues on next page...

Table continued from previous page.

Tag	definition	Values	Units or meaning
submissions	submission(s) containing this file (CSV)		
loaded_runs	loaded run(s) linked to this file (CSV)		
unloaded_runs	unloaded run(s) linked to this file (CSV)		
suppressed_runs	suppressed run(s) linked to this file (CSV)		
loaded_analyses	loaded analyses(s) linked to this file (CSV)		
unloaded_analyses	unloaded analyses(s) linked to this file (CSV)		
suppressed_analyses	suppressed analyses(s) linked to this file (CSV)		

Batch account space tracking with tab files

Each day a report is compiled showing the submitters aspera account quota usage and a list of files that are currently still located in the account. This file is produced in the open realm and in the protected realm so each instance should be retrieved to give a complete view of space utilization. If the quota is reached (or threatened) you may write to NCBI request an increase. You may also back off new submissions until the space "drains out".

To retrieve the files, use *ascp* against these addresses:

upload@ncbi.nlm.nih.gov://asp- /outgoing/Files/usage_report.txt

gap-upload@ncbi.nlm.nih.gov://asp- /outgoing/Files/usage_report.txt

Here is an example output portion for one submitter :

```
Usage report at Sun Mar 11 23:58:18 EDT 2012 created on cfengine1:
Filesystem            Size  Used Avail Use% Mounted on
panfs://pan1:global   9.1T  191G  9.0T   3% .
******************
List of all files:
.:
total 632
drwxrwsr-x  2 shumwaym trace 151552 Mar 11 23:09 incoming
drwxrwsr-x  4 shumwaym trace   4096 Mar  1 08:12 outgoing
```

```
drwxrwxr-x  4 shumwaym trace  90112 Mar  7 09:00 test

./incoming:
total 128110132
-rw-rw-r--  1 asp-bi trace 1789376627 Mar 11 22:55 D0E3KACXX111201.1.tagged_190.v2.bam
-rw-rw-r--  1 asp-bi trace 1693468095 Mar 11 22:43 D0E3KACXX111201.1.tagged_236.v2.bam
-rw-rw-r--  1 asp-bi trace  141840099 Mar 11 22:51 D0E3KACXX111201.1.tagged_289.v2.bam
-rw-rw-r--  1 asp-bi trace 1654940006 Mar 11 22:43 D0E3KACXX111201.1.tagged_332.v2.bam
-rw-rw-r--  1 asp-bi trace 1455232677 Mar 11 23:01 D0E3KACXX111201.1.tagged_34.v2.bam
```

Tools and Methods

Release check in Entrez

You can use Entrez SRA to check for the appearance of your submission. This only works if your submission has been released (published). To run this check, enter your submission accession or submission component accession as follows (example is using SRA025969):

http://www.ncbi.nlm.nih.gov/sra?term=SRA025969

This Entrez SRA search may be limited in the following ways:

- It can take 1-2 business days before released objects are fully indexed in Entrez.
- Components that have been released but do not have any data loaded (and also released) will not appear in Entrez.
- Submission to the protected SRA for distribution through dbGaP are released by dbGaP, and may not appear in Entrez until the data have reached the next periodic study release.

Public archive mirror reports

The annotated metadata xml and the Accessions status tab file are available in a public, released form at this address. http://trace.ncbi.nlm.nih.gov/Traces/sra/sra.cgi?cmd=show&f=faspftp_metadata&m=downloads&s=download_reports

Note that submissions that have not yet been released are redacted from this report. The information in this report is equivalent to that in Entrez, except that accessions that have been suppressed or those that returned to unpublished from a published state are still listed (though their xml are not dumped).

How to compare metadata versions

SRA metadata are not versioned in an explicit, public way. Rather, metadata are tagged when they change in a substantive way. A checksum is used to record the current content. One can tell whether the content has changed by comparing the checksum to a previously computed value.

```
Accession     Submission      Status        Md5sum
SRA000001     SRA000001     public        d703b0b98a686a84ff232b9967e3d55c
SRP000057     SRA000001     public        bdac682ff9dca87f158b3d327832ec66
SRR000289     SRA000001     public        2fc12909aff893cdc20c48c0aa875bdf
SRS000246     SRA000001     public        7bf577c49d282529bc5f35f63137e6c0
SRX000068     SRA000001     public        a474043e97911936fe49f06f7a301aa5
SRA000002     SRA000002     public        7810982f118198eaf207a351e1550aa9
SRP000058     SRA000002     public        4c5d2a1c8a7fca885a09d690e49a5d06
SRR000290     SRA000002     public        aed8942276489b55ab98e282725ee920
SRS000247     SRA000002     public        999486234ce2e7420e16f169c2a86578
```

The md5sum value is equivalent to putting an *xmllint 'noblanks'* version of the xml associated with an accession in a file by itself (without a line feed) and executing *md5sum -b* on that file. If there is no meta data difference, then no increment is generated for that center on that day.

On the 1st of every month, a complete meta data dump is created in addition to an incremental dump. The first meta data dump for a new center is both an incremental and complete dump.

Obtain a copy of the script used to get md5 values for each accession chunk in a meta data xml file:

```
wget ftp-trace.ncbi.nlm.nih.gov:/sra/utilities/getMetaMd5.pl
```

The usage is:

```
getMetaMd5.pl < meta xml file path (based on ending in .xml) >

       OR

getMetaMd5.pl < file containing list of meta xml file paths >

       OR

<list of meta xml file paths> | getMetaMd5.pl
```

So you can provide the path to a single .xml file, a file containing a list of paths to xml files, or pipe to it a list of paths to xml files.

The *getMetaMd5.pl* script runs on Linux and requires

- *perl* to be at */usr/local/bin/perl* (version 5.8.3 or higher),
- *xmllint* in your executable path (*libxml* 20630 or higher),
- *xsltproc* in your executable path (version 10102 or higher),
- *Digest::MD5*, a *perl* library for calculating md5 sums, and,
- *parseMeta.xsl*, to be in the same directory as the *getMetaMd5.pl* executable.

How to view files you have uploaded to your aspera account

To access NCBI servers with limited shell access a submitter must use their secret key (usually used with *ascp* for file transfer).

For access from unix/linux/macos the secret key must be in *openssh* format. In this case ssh command is used and command line is as (where zzz is your center name):

For open SRA account:

```
ssh -i secretkey.openssh asp-zzz@upload.ncbi.nlm.nih.gov
```

For protected SRA account:

```
ssh -i secretkey.openssh asp-zzz@gap-upload.ncbi.nlm.nih.gov
```

For similar access from windows the key must be in putty format. And the *putty.exe* command should be used. The command line is as (where zzz is your center name):

For open SRA account:

```
putty.exe -i secretkey.ppk asp-zzz@upload.ncbi.nlm.nih.gov
```

For protected SRA account:

```
putty.exe -i secretkey.ppk asp-zzz@gap-upload.ncbi.nlm.nih.gov
```

This limited shell has *aspsh>* as a prompt and allows only few commands like *ls,cp,mv,rm*. The *cd* command is not allowed so you must use *ls* with directory name as an argument.

Examples:

```
aspsh> ls -l
total 240
drwxrwsr-x  2 5608 trace 65536 May  7 10:08 incoming
drwxrwsr-x  3 5608 trace  4096 Apr 15  2009 outgoing
drwxrwsr-x  2 5608 trace  8192 Apr 27 20:04 test

aspsh> ls -l analysis
total 0

aspsh> ls -l incoming
total 15663023352
-rw-rw-r--  1 asp-zzz trace   16504539868 May  6 10:06
0083_20090930_2_SP_ANG_HSAP_NG_005sA_01003244491_4.srf
…
```

File Format Guide

Created: September 23, 2009; Updated: January 9, 2016.

Overview

This document reviews the file formats currently supported by the Sequence Read Archives (SRA) at NCBI, EBI, and DDBJ, and gives guidance to submitters about current and future file formats and policies regarding SRA submissions.

The SRA is one of the International Nucleotide Sequence Databases and this Collaboration (INSDC) sets policies and goals for the partner databases. This document is intended to be compatible with INSDC policies.

Goals

This document guides submitters of sequencing data in order to:

- Specify which data formats are currently supported by SRA.
- Enable submitters to validate and convert data prior submission to avoid unnecessary data transfers.
- Improve the speed of submission processing.
- Reduce the probability of failed submissions.
- Improve other services provided by SRA by freeing up time previously spend to correct and transform data.

This document guides depositors to the Archives so that they may:

- Understand how to prepare data for submission to one of the Archives.
- Know what formats are supported by Archives facilities or toolkits, and which ones may have to be developed by the user.
- Understand why technical issues limit the usage of certain file formats.

External Documents and Links

- SAMtools software and SAM-format specification documents: http://samtools.sourceforge.net/
- Standard Flowgram Format (SFF) documentation: http://www.ncbi.nlm.nih.gov/Traces/trace.cgi?cmd=show&f=formats&m=doc&s=formats#sff
- Tools for viewing and processing SFF files:
  - Mothur: http://www.mothur.org/
  - 454 Analysis Software: http://www.454.com/products/analysis-software/
  - QIIME: http://qiime.org/
- PacBio documentation on bax.h5 / bas.h5 format (PDF): http://files.pacb.com/software/instrument/2.0.0/bas.h5 Reference Guide.pdf
- HDF5 tools: http://www.hdfgroup.org/products/hdf5_tools

- Applied Biosystems documentation on 2 base encoding (PDF): https://www3.appliedbiosystems.com/cms/groups/mcb_marketing/documents/generaldocuments/cms_058265.pdf
- Complete Genomics documentation on formats: http://www.completegenomics.com/customer-support/documentation/100357139-2
- Sequence Read Format (SRF) homepage: http://srf.sourceforge.net

Revision History

- Reviewed by NCBI 2014-03-18
- Reviewed by NCBI 2012-07-11
- Reviewed by NCBI 2009-10-01
- Reviewed by EBI 2009-10-07
- Reviewed by DDBJ 2009-11-27

Overview of Input Formats

General Considerations

The SRA is a "raw data" archive, and requires per-base quality scores for all submitted data. Thus, unlike GenBank and some other NCBI repositories, FASTA and other sequence-only formats are not sufficient for submission. FASTA can, however, be submitted as a reference sequence(s) for BAM files or as part of a FASTA/QUAL pair (see below).

The SRA data model has transitioned from "dumps" of whole flowcell lanes or production runs into a semi-curated database of sample-specific sequencing libraries. This has implications for the types of data that we accept. Most specifically, barcoded/batch files should be split into per-sample data files ("demultiplexed"). Demultiplexing makes the sample - data linkage unambiguous in our database and should improve both the clarity and usability of submitted data. Please email sra@ncbi.nlm.nih.gov if you have specific questions about data requirements vis-à-vis samples.

Conversion to the SRA archive format (described below) is NOT required for submission. However, the SRA Toolkit can be used to "test load" your files locally if you would like to validate them prior to submission. BAM files can be evaluated with 'bam-load' and FASTQ files can be evaluated with 'latf-load' (first released in Toolkit version 2.3.5). These load utilities are effectively stand-alone and can be run by most submitters. Other SRA loading software, such as 'sff-load', 'abi-load', etc. are dependent on SRA XML documents and are only recommended for advanced users. If you elect to test load your data file(s) and encounter problems, please email sra@ncbi.nlm.nih.gov if you have questions.

Preferred Formats

The SRA generally prefers to obtain "container files". Container in this context means an unambiguous binary file. These are objects that contain both the data and a description or

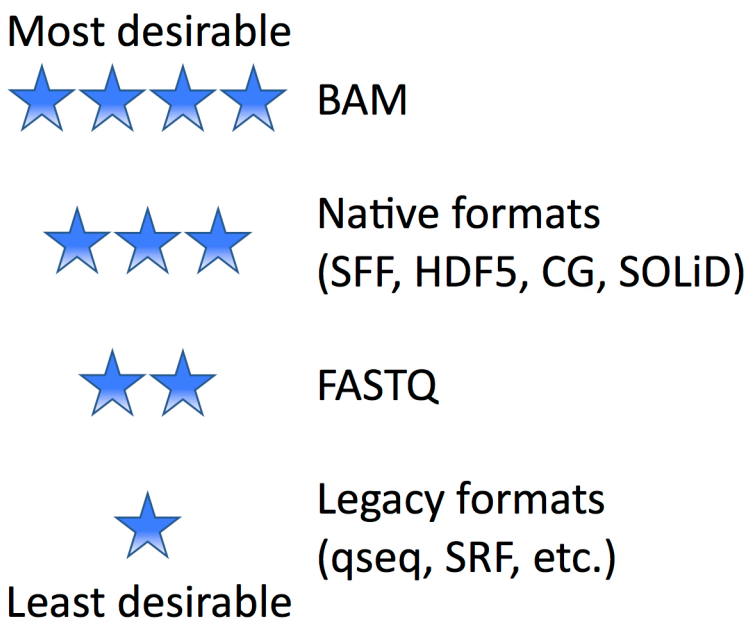


Figure 1 – Input file types supported by the SRA

specification of the data. Examples include BAM, SFF, and PacBio HDF5 formats. Containers have the following advantages:

- All data for a given library is contained in one file.
- Data are indexed for random access.
- Data are compressed so *gzip* and other compression utilities are discouraged.
- Data are streamable (can be read from one input handle).
- Data are self-identifying (file type can be interrogated with *file*).
- Data come with run-time configuration and execution parameters, including run date, instrument name, flowcell name, processing program and version, etc.

Text formats, such as FASTQ, are supported, but are not the preferred submission medium. Poorly defined specifications and high variability within these formats tend to lead to a higher frequency of failed or problematic submissions. Wherever possible, submitters are encouraged to submit data in a container format, as described above.

Figure 1 shows the hierarchy of input file types supported by the SRA. Table1 shows which properties of input data file formats are supported.

Table 1 – Input file types and their general properties

File model	Archive ready?	Stream-able on load?	Auxiliary data?	Run meta data?	Com-pressed?	Indexed?	Read names parseable ?	Read names indexable?
SRA	Y	Y	Y	Y	Y	Y	Y	Y
BAM	Y	Y	Y	Y	Y	Y	N	N
SFF	Y	Y	Y	Y	Y	Y	Y	Y
HDF5	Y	Y	Y	Y	Y	Y	Y	Y
SOLiD	Y	Y	Y	Y	N	N	Y	Y
FASTQ	Y	Y	N	N	N	N	N	N
SRF	N	Y	Y	Y	Y	Y	Y	Y
Illumina native	Y	Y	N	N	N	N	N	N

BAM (Binary Sequence Alignment/Map)

BAM is the preferred submission format for the SRA. BAM is the binary (compressed and indexed) version of SAM. BAM files can be read out as human-readable SAM through the use of BAM/SAM-specific utilities (like SAMtools), or with a conventional decompression utility like gzip/gunzip. SAM is a generic tab-delimited format that includes both the raw read data and information about the alignment of that read to a known reference sequence(s). There are two main sections in a SAM file, the header and the alignment (sequence read) sections, each of which are described below. Note that this documentation will focus on a description of the SAM format with respect to submission of BAM files to the SRA. A more comprehensive discussion of the format specifications can be found at the SAMtools website.

SAM Header Section

Each line in a SAM header begins with '@', followed by a two-character code that identifies the type of information encoded in the line. A typical SAM file can contain HD (header), SQ (reference sequence) line(s), RG (read group) line(s), and PG (program) descriptions in the header section. An example SAM header is shown below. Note that this is to highlight the format – not all sections and tags are required.

```
@HD    VN:1.4    SO:coordinate
@SQ    SN:CHROMOSOME_I    LN:15072423    UR:ftp://ftp.ncbi.nlm.nih.gov/
genbank/genomes/Eukaryotes/invertebrates/Caenorhabditis_elegans/
WBcel215/Primary_Assembly/assembled_chromosomes/FASTA/chrI.fa.gz
AS:ce10    SP:Caenorhabditis elegans
@SQ    SN:CHROMOSOME_II    LN:15279345    UR:ftp://ftp.ncbi.nlm.nih.gov/
genbank/genomes/Eukaryotes/invertebrates/Caenorhabditis_elegans/
WBcel215/Primary_Assembly/assembled_chromosomes/FASTA/chrII.fa.gz
AS:ce10    SP:Caenorhabditis elegans
@SQ    SN:CHROMOSOME_III    LN:13783700    UR:ftp://
```

```
ftp.ncbi.nlm.nih.gov/genbank/genomes/Eukaryotes/invertebrates/
Caenorhabditis_elegans/WBcel215/Primary_Assembly/assembled_chromosomes/
FASTA/chrIII.fa.gz    AS:ce10    SP:Caenorhabditis elegans
@SQ    SN:CHROMOSOME_IV    LN:17493793    UR:ftp://ftp.ncbi.nlm.nih.gov/
genbank/genomes/Eukaryotes/invertebrates/Caenorhabditis_elegans/
WBcel215/Primary_Assembly/assembled_chromosomes/FASTA/chrIV.fa.gz
AS:ce10    SP:Caenorhabditis elegans
@SQ    SN:CHROMOSOME_V    LN:20924149    UR:ftp://ftp.ncbi.nlm.nih.gov/
genbank/genomes/Eukaryotes/invertebrates/Caenorhabditis_elegans/
WBcel215/Primary_Assembly/assembled_chromosomes/FASTA/chrV.fa.gz
AS:ce10    SP:Caenorhabditis elegans
@SQ    SN:CHROMOSOME_X    LN:17718866    UR:ftp://ftp.ncbi.nlm.nih.gov/
genbank/genomes/Eukaryotes/invertebrates/Caenorhabditis_elegans/
WBcel215/Primary_Assembly/assembled_chromosomes/FASTA/chrX.fa.gz
AS:ce10    SP:Caenorhabditis elegans
@RG    ID:1    PL:ILLUMINA    LB:C_ele_05    DS:WGS of C elegans
PG:BamIndexDecoder
@PG    ID:bwa    PN:bwa    VN:0.5.10-tpx
```

Ideally, the "SN" value should be a versioned accession (e.g., NC_003279.7, rather than "CHROMOSOME_I"). This will allow the SRA to unambiguously identify the reference sequence(s) and process the BAM file with minimal intervention. Barring that, submitters are strongly encouraged to use the "UR" (URL/URI that can be used to obtain the reference sequence(s)) and "AS" tags to clearly define which assembly has been used (as above). If the data are instead aligned to a "local" or submitter-defined set of references (including any modifications to accessioned assemblies), then the submitter must include a "reference fasta" along with each submitted bam file. The FASTA header line(s) must match the "SN" names provided in the BAM file exactly. Deviation from these recommended practices will require manual intervention by SRA staff in order to process a BAM file and can delay completion of a submission.

SAM Alignment Section

The alignment section contains the sequence and quality information, ideally in a sorted order to reduce file size and improve indexing. Each read is contained on a single line, and all fields are tab-delimited and in an order defined by the SAM specification guide. Aside from the read ID (QNAME in SAM jargon), SEQ, and QUAL fields, most other fields are determined and reported by the software used to generate the SAM/BAM file and should not be manually edited. Below is an example alignment section that continues from the above example header.

```
3658435    145    CHROMOSOME_I    1    0    100M    CHROMOSOME_II
2716898    0
GCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAA
GCCTAAGCCTAAGCCTAAGCCTAAGCCT
@CCC?:CCCCC@CCCEC>AFDFDBEGHEAHCIGIHHGIGEGJGGIIIHFHIHGF@HGGIGJJJJJIJJJJJJ
JJJJJJJJJJJJJJJHHHHHFFFFFCCC    RG:Z:1    NH:i:1    NM:i:0
5482659    65    CHROMOSOME_I    1    0    100M    CHROMOSOME_II
11954696    0
GCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAA
```

```
GCCTAAGCCTAAGCCTAAGCCTAAGCCT
CCCFFFFFHHGHGJJGIJHIJIJJJJJIJJJJJIJJGIJJJJJIIJIIJFJJJJJFIJJJJIIIIGIIJHHH
HDEEFFFEEEEEDDDDCDCCCAAA?CC:    RG:Z:1    NH:i:1    NM:i:0
1022    163    CHROMOSOME_I    5    29    100M    =    499    594
AAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCT
AAGTCTAAGCGTAATGATAAGCATAAGC    :?@BADDBFHDHFDHE;C:?FHIIGGD>GGGGC>??
GADG>D@C?D9?B88BFAFE@FHD:;=C7C;@G<)7.?))).).;;',,5(((,;>(((,,(:5
RG:Z:1    NH:i:1    NM:i:6
22242785    16    CHROMOSOME_I    6    0    100M    *    0    0
AGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTAAGCCTA
AGCCTAAGCCTAAGCCTAAGCCTAAGCC
>ACCCC>CCDDDCCDEC@@FFFEFHEHHFGCIGGHEHGJIIHFIJJIIHHHDIHHJJIJJIIJIJIGIJIJJ
IIJJJJJJJJJJJJJHHHHHFFFFFCCC    RG:Z:1    NH:i:1    NM:i:0
```

Note that the header and alignment section are internally consistent: Each read has an RNAME (reference, 3rd value) that matches an SN tag value from the header (e.g., "CHROMOSOME_I"), and the read group tag ("RG:Z:") is consistent with the read group ID in the header ("1"). It is also important to ensure that the FLAG fields (2nd value in each line) are correctly set for the data; the SRA pipeline will attempt to resolve incorrect FLAG values, but sufficiently incorrect values can lead to processing errors.

External Documents and Links

SAMtools software and SAM-format specification documents: http://samtools.sourceforge.net/

Standard Flowgram Format (SFF)

454 Life Science (now part of Roche) and NCBI developed SFF to encode 454 flowgrams. In the absence of a BAM file, SFF is the preferred input format for 454 data. IonTorrent data can also be submitted as SFF. Extensive technical details about the format can be obtained here. In general, though, submitters of SFF data should ensure that the data are demultiplexed (if barcoded) – this is particularly common in pyrotag / 16S rRNA amplicon sequencing.

External Documents and Links

Tools for viewing and processing SFF files:

- Mothur: http://www.mothur.org/
- 454 Analysis Software: http://www.454.com/products/analysis-software/
- QIIME: http://qiime.org/

PacBio HDF5

Pacific BioSystems uses HDF5, a container file with a directory-like structure, to store raw data. The SRA accepts both bas.h5 and bax.h5 file submissions. Note that submission of data from the RS II instrument requires one (1) bas.h5 file and three (3) bax.h5 files.

External Documents and Links

PacBio documentation on bax.h5 / bas.h5 format (PDF): http://files.pacb.com/software/instrument/2.0.0/bas.h5 Reference Guide.pdf

HDF5 tools: http://www.hdfgroup.org/products/hdf5_tools

Analysis Files for PacBio

If you wish to submit additional analysis files for a PacBio submission, please contact SRA at sra@ncbi.nlm.nih.gov. Common analysis files that are submitted alongside HDF5 files are modification.csv and motif_summary.csv files. In your email please include the name of your submission and what types of files you wish to include as part of your submission. This will help us to locate your submission and determine if we can accept your data files.

FASTQ

FASTQ is not a specified file format, but a style similar to "FASTA". It consists of readname headers, nucleotide base calls and per-base quality scores in text form. There are many variations.

The following terms and formats are defined in general:

```
READNAME = Text string terminated by white space.
    BASES = [ACGTNactgn.]+
QUALITIES = [0-9]+ | <quality>\s[0-9]+ (Decimal-encoding, whitespace or tab-delimited)
            or
            [\!\"\#\$\%\&\'\(\)\*\+,\-\.\/0-9:;<=>\?\@A-I]+ (Phred-33 ASCII)
            or
            [\@A-Z\[\\\]\^_`a-h]+ (Phred-64 ASCII)
```

The permissible FASTQ format is simply:

```
@READNAME
BASES
+[READNAME]
QUALITIES
```

Where each instance of READNAME, BASES, and QUALITIES are newline-separated.

As indicated above, the QUALITIES string can be whitespace-separated numeric Phred scores or an ASCII string of the Phred scores with the ASCII character value = Phred score plus an offset constant used to place the ASCII characters in the printable character range. There are 2 predominant offsets: 33 (0 = !) and 64 (0=@).

Paired-end FASTQ

Paired-end data submitted in FASTQ format should be submitted in one of two formats: (1) As separate files for forward and reverse reads, in which the reads are in the same order. (2) As interleaved, or "8-line", FASTQ, in which forward and reverse reads alternate

in the file and are in order (i.e., read "1F", followed by read "1R", then read "2F", then "2R", etc.

Concatenated FASTQ (in which all forward reads are followed by all reverse reads) is not supported.

FASTA/QUAL pairs

FASTA files may be submitted if accompanied by corresponding QUAL files. These are recognized in the SRA data processing pipeline as equivalent to FASTQ and should be specified as "fastq" when submitting the data files. Borrowing from the FASTQ description above, the general format for FASTA/QUAL pairs is:

FASTA

```
>READNAME
BASES
```

QUAL

```
>READNAME
QUALITIES
```

Where READNAME must be identical between files for a given read, and QUALITIES are generally in whitespace or tab-separated decimal values. Note the following guidelines for FASTA/QUAL pairs of files:

- In a given pair of files, there must be the same number of reads in both.
- For a given read, there must be the same number of BASES and QUALITIES, i.e., if the BASES are trimmed to remove barcodes, then the same scores must be removed from the QUALITIES, etc.

Vendor-specific FASTQ variants

Illumina FASTQ

There are two general styles of FASTQ produced by Illumina machines. The older format is emitted from Gerald, the secondary analysis pipeline. This format contains 64-offset (ASCII '@' = 0) quality encoding. Paired end data are presented in the orientation in which they were sequenced (5'-3'-3'-5').

The index and read number labels are defined as:

- Index: string. Currently 0, should be the index of the multiplexed sample in barcoded experiments, for example @EAS51_105_FC20G7EAAXX_R1:1:1:471:409#ATCACG/2
- Index values are processed and stored as SRA spot groups
- Read Number: 1 for single reads; 1 or 2 for paired ends.

Formally,

```
@<READNAME>[#<index>]/<read_number>
BASES
+<READNAME>[#<index>]/<read_number>
QUALITIES
```

The newer Illumina FASTQ variant (as of CASAVA 1.8), use 33-offset quality encoding (ASCII '!' = 0) and have a different READNAME format:

```
@<instrument>:<run number>:<flowcell ID>:<lane>:<tile>:<x-pos>:<y-pos>
<read>:<is filtered>:<control number>:<index sequence>
```

Specific example:

```
@EAS139:136:FC706VJ:2:5:1000:12850 1:Y:18:ATCACG
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
+
BBBBCCCC?<A?BC?7@@???????DBBA@@@@A@@
```

454. FASTQ

The 454 READNAME is a 14 character alphanumeric string that encodes the plate, region and raster address of the read. The plate name is an encoding of a timestamp plus one character hash value that is virtually unique. The region is a two place decimal indicating the gasket division (there is always at least one gasket). The raster coordinate indicates the x and y coordinates on the plate modulus 4096 in base 36 encoding. Paired end data are presented in the orientation in which they will be aligned to a reference (5'-3'-5'-3'), which is the same orientation in which they were sequenced.

```
<plate> = [A-Z0-9]{7}
<region> = [0-9]{2}
    <xy> = [A..Z0..9]{5}
   BASES = [ACGTN]+
QUALITIES = [0-9]+ | <quality>\s[0-9]+
 READNAME = <plate><region><xy>
```

Reads are sorted in order of READNAME. Records are variable length. Files are analogous to FASTA/QUAL (described above), and should be specified as 'fastq' in SRA submissions.

The grammar for the 454 sequence file:

```
>READNAME
BASES
```

The grammar for the 454 qualities file:

```
>READNAME
QUALITIES
```

Helicos FASTQ

Bindings for Helicos FASTQ are:

```
<flowcell> ::= VHE-[0-9]+
<channel> ::= 1-25
<field> ::= 1-1100
<camera> :: [1234]
<position> :: 1-100000
<sep> ::= [-]
READNAME::= <flowcell><sep><channel><sep><field><sep><camera><sep><position>
QUALITIES ::= [!-I]+
```

A single record grammar is:

```
@READNAME
BASES
+
QUALITIES
```

For example,

```
@VHE-232481681003-9-1100-3-7971
ATCATTAAACATAAGTTTCAATCAACACTAATCATCAAC
+
//////////////////////////////////////
```

Helicos reads can be variable in length, but the number of BASES and QUALITIES must be the same for a given read.

SOLiD native

SOLiD users may submit CSFASTA and QUAL files as SOLiD native data. Primary analysis output of the SOLiD system is in color space. Paired end data are presented in the same orientation in which they were sequenced (5'-3'-5'-3').

Specific bindings for the ABI SOLiD System are:

```
<flowcell> = [a-zA-Z0-9_-:]{2}+
   <slide> = 0..1
   <panel> = 1..4096
       <X> = 1..4096
       <Y> = 1..4096
     BASES =  [TtGg][0123\.]+
 QUALITIES = [0-9]+ | <quality>\s[0-9]+
     <sep> = [_]
  READNAME = <flowcell><sep><slide><sep><panel><sep><x>sep><y>
   TAGNAME = <panel><sep><x><sep><y><sep><tag>
```

The interpretation of the separator (<sep>) is right associative. Reads are sorted in panel order within a given set of related files. All SOLiD data are fixed length.

The files have an optional header that is identified by lines that begin with the hash/pound/number sign (#). The HEADER can be defined as:

```
# <date> <path> [--flag]* --tag <tag> --minlength=<length> --prefix=<prefix> <path>
# Cwd: <path>
# Title: <flowcell>
```

The grammar for the CSFASTA file is:

```
#HEADER (multiple lines)
>TAGNAME
BASES
```

The grammar for the QUAL file is:

```
#HEADER (multiple lines)
>TAGNAME
QUALITIES
```

As with FASTA/QUAL pairs, there are several rules for pairs of CSFASTA/QUAL files. TAGNAME must be identical between files for a given read, and QUALITIES are generally in whitespace or tab-separated decimal values. Note the following guidelines for CSFASTA/QUAL pairs of files:

- In a given pair of files, there must be the same number of reads in both.
- For a given read, there must be the same number of color space digits and QUALITIES, i.e., the BASES line is typically 1 character longer than the number of QUALITIES (due to the color space indexing base that begins each BASES string).

External Documents and Links

Applied Biosystems documentation on 2 base encoding (PDF): https://www3.appliedbiosystems.com/cms/groups/mcb_marketing/documents/generaldocuments/cms_058265.pdf

Complete Genomics (CG) native

The SRA is able to process Complete Genomics data, though these data require a unique workflow for submission: the SRA pulls the data in its native directory structure directly from S3. Please contact the SRA for details (sra@ncbi.nlm.nih.gov).

External Documents and Links

Complete Genomics documentation on formats: http://www.completegenomics.com/customer-support/documentation/100357139.html

Analysis File Types

SRA accepts most common data analysis file types.

Analysis File Types	Description
Comma separated values (.csv)	A comma delimited text file that can be viewed as a spreadsheet. The first line should contain column headers.
Tab separated values (.tsv)	A tab delimited text file that can be viewed as a spreadsheet. The first line should contain column headers.
ACE	Multiple alignment file output from the phred assembler and similar programs. See README for a description of the ACE file format.
WIG	The wiggle (WIG) format allows display of continuous-valued data in track format. This display type is useful for GC percent, probability scores, and transcriptome data. See http://genome.ucsc.edu/goldenPath/help/wiggle.html for a description of the Wiggle Track format.
BED	BED format provides a flexible way to define the data lines that are displayed in an annotation track. See http://genome.ucsc.edu/FAQ/FAQformat#format1 for a description of the BED format.
The Variant Call format (VCF)	VCF is format for storing DNA polymorphism data with annotation.
Mutation Annotation Format (MAF)	MAF format is a tab delimited file that stores information for mutations.
General Feature Format (GFF)	GFF format is used for annotation of biological sequences.

Legacy formats

Sequence Read Format (SRF)

SRF is a community standard developed by James Bonfield and Asim Siddiqui. It has been used to contain large amounts of Illumina and SOLiD data for deposit and have served as a backing storage format. Several implementations exist. Io_lib based implementations maintained as part of the Staden package.

External Documents and Links

Sequence Read Format (SRF) homepage: http://srf.sourceforge.net

Illumina native

Submitters may submit native data from the primary analysis output of the Illumina GA. The filetype is “Illumina_native” and constituent files for a run should be tarred together into a single tar file.

Illumina GA readname can be defined as follows:

```
<flowcell> = [a-zA-Z0-9_-]{2}+
    <lane> = 1..8
    <tile> = 1..1024
       <X> = 0..4096
       <Y> = 0..4096
<sep> ::= [_:\t]
READNAME ::= [<flowcell><sep> | s_]<lane><sep><tile><sep><x>sep><y>
```

The interpretation of the separator (<sep>) is right associative. Within a related set of files, reads are grouped by tile. Reads should be fixed length, and the number of quality scores and bases is the same in each.

Allowed characters:

```
    BASES = [AaCcTtGgNn\.]+
QUALITIES = \!\"\#\$\%\&\'\(\)\*\+,\-\.\/0-9:;<=>\?\@A-I]+
            or
            \@A-Z\[\\\]\^_`a-h]+
```

qseq

The basecalling program Bustard emits a *_qseq.txt* file for each lane (two files for mate pairs). Paired end data are presented in the orientation in which they were sequenced (5’-3’-3’-5’).

Each read is contained on a single line with tab separators in the following format:

- Machine name: unique identifier of the sequencer.
- Run number: unique number to identify the run on the sequencer.
- Lane number: positive integer (currently 1-8).
- Tile number: positive integer.
- X: x coordinate of the spot. Integer (can be negative).
- Y: y coordinate of the spot. Integer (can be negative).
- Index: positive integer. No indexing should have a value of 1.
- Read Number: 1 for single reads; 1 or 2 for paired ends.
- Sequence (BASES)
- Quality: the calibrated quality string. (QUALITIES)
- Filter: Did the read pass filtering? 0 - No, 1 - Yes.

seq, prb, int

The _seq.txt, _prb.txt, and _int.txt files are emitted by Bustard, the primary analysis program. In Illumina pipeline versions 1.3 and earlier produced tab files in the following formats first defined in the 1.1 version of the GA pipeline:

The sequence text files (_seq.txt) have this format:

```
<READNAME>\t<BASES>
```

The qualities text files have four scores per base call (_prb.txt) in this format:

```
<READNAME>\t{%d %d %d %d}+  with value range [-40,40]
```

The intensity text files have four scores per base call (_int.txt) in this format:

```
<READNAME>\t{%5.1f %5.1f %5.1f %5.1f}+  with value range [-16384.0,16383.0]
```

Each of these files was either presented tile by tile, or in one file per lane. The number of reads must be equal between the input files for a lane. Illumina pipeline versions 1.4 and later could only produce these files by running Bustard under non-default conditions.

Illumina scarf

Another text file output by Gerald analysis stage is a single colon separated file with one record per line containing read name, sequence, and quality.

Overview of SRA output formats

SRA native format (VDB)

SRA files do not have a fixed format, but are actually portable database files (VDB; "vertical database") with embedded schema. The schema is recorded on a per-object basis, allowing us to change schema over time while ensuring that older databases remain accessible. The database-like structure of SRA data files allows relatively simply interconversion between multiple different formats. The various 'dump' utilities of the SRA Toolkit are specifically designed to provide this conversion. The utility 'vdb-dump' can be used to interrogate the native SRA data format directly.

SAM

The Toolkit utility 'sam-dump' can be used to output any SRA data file into SAM format. Note that only data submitted with alignment data (e.g., submitted as aligned BAM) will output aligned SAM. All other datasets will output unaligned, un-headered SAM.

FASTQ

All SRA data can be converted to FASTQ format using 'fastq-dump'. Since SRA data are stored in a concatenated form, it is important to note that specific options may have to be invoked in order for paired-end fastq to be formatted correctly during output. It is

recommended that new users review fastq-dump documentation to ensure proper output formatting before committing to large dataset extractions.

SFF

Only those datasets submitted as SFF are suitable for conversion back into SFF format. All other submitted data formats lack the information required to generate SFF. Consequently, the utility 'sff-dump' will provide a clear error message if a given dataset cannot be converted to SFF.

SOLiD native (CSFASTA/QUAL)

All SRA data can be output into color space data. The utility 'abi-dump' can be used to output CSFASTA and QUAL data files (with appropriate options, fastq-dump can be used to output "CSFASTQ" format).

Illumina native formats

All SRA data can be output into Illumina native format, as it is functionally similar to FASTQ. The Toolkit utility 'illumina-dump' can be used to output data into "standard" Illumina native, or qseq depending on the options invoked.

Analysis Submission Guide

Created: April 25, 2010; Updated: October 19, 2011.

1. Overview

This document reviews submission procedures and guidelines for SRA analysis objects, including

- De novo assemblies (to be specified in a future version of this document)
- Reference alignments
- Sequence annotations (to be specified in a future version of this document)
- Abundance measurements (to be specified in a future version of this document)

In keeping with developing NIH policy, this document also shows how to submit primary sequencing data as a part of the analysis object.

1.1. History

Guidelines for SRA analysis submission were developed in conjunction with two NIH roadmap initiatives: The Cancer Genome Atlas (TCGA), and the Human Microbiome Project (HMP). The TCGA established early requirements to allow submission of all needed primary data through the BAM file format. The HMP pioneered requirements for annotation of raw sequencing data from metagenome projects where assembly into higher constructs is difficult.

1.2. Goals

1. Meet the needs of users by providing a home somewhere in the data model for all desired properties.
2. Distinguish where in the data model each desired property should reside.
3. Define processing directives that might be important to interpreting the sequencing/alignment data and loading it into an archive database.
4. Eliminate dependence on spreadsheets and filenames to convey metadata.
5. Provide searchable metadata that can be used by query writers in the public database.
6. Provide query source for programmatic construction of component descriptions that users of protected data will see inside the dbGaP authorized access download interface.

1.3. Scope

In its current revision, this document describes metadata needs for BAM file submission. It does not describe the submission modalities. Higher level analysis types and other analysis types are not described. Some BAM files are submitted using preexisting SRA data, other BAM files will be submitted containing de novo sequencing data as part of its payload. This document does not describe archive requirements for the BAM file read

placement records, which may have additional requirements in order to be loaded into the NCBI alignment database. These requirements need further development.

1.4. Revision History

Drafts A-E created 2010-09-14 to 2010-10-08. Document released with draft status 20 Oct 2010.

1.5. Related Documents

Elements of TCGA project requirements have been incorporated into this document [Tim Fennell. *BAM File Format for TCGA Submissions.* Draft v2, July 9, 2009.]

Submitters should also consult the established SRA submission documentation:

Quick Start Guide:

http://www.ncbi.nlm.nih.gov/books/NBK47529/

Aspera Transfer Guide:

http://www.ncbi.nlm.nih.gov/books/NBK242625/

Here is the released SRA XML Schema:

http://trace.ncbi.nlm.nih.gov/Traces/sra/sra.cgi?view=xml_schemas

For details on the SAM/BAM specification please reference:

http://samtools.sourceforge.net/SAM1.pdf

A BAM file validator utility is available here:
http://picard.sourceforge.net/command-line-overview.shtml – ValidateSamFile

2. Data Model

NCBI Object	Accession	Sequencer Production Unit	BAM Component
Submission envelope	SRA	n/a	n/a
Analysis	SRZ	n/a	BAM file
Study	SRP	n/a	n/a
Experiment	SRX	n/a	Library (LB)
Sample	SRS	n/a	Sample (SM)
Run	SRR	Lane/slide/plate	Read Group (RG)
Reference Sequence	NC_ and others	n/a	Sequence Dictionary (SQ)
Probe set	Pr	capture array	n/a

2.1. Submission Metadata

The submission metadata pertains the submission "package" or "envelope" conveying the data to the archive.

Submitter id/alias – Submitter's name or alias for the submission.

Submission date – ISO 8601 date for the date of transmission of the file to NCBI.

Submitter contact – name and email address of the submitter contact(s).

Center name – NCBI short name for the submitting center.

2.2. Analysis Metadata

Analysis alias – Submitter's name or alias for the analysis object.

Analysis title – The title string that will be presented to users of the public archive when this record is retrieved in a search result. Please limit this string to 80 characters.

Analysis type – DE_NOVO_ASSEMBLY | REFERENCE_ALIGNMENT | SEQUENCE_ANNOTATION | ABUNDANCE_MEASUREMENT

Analysis Description – A free form description of the analysis product and the process by which it was produced.

Analysis date – ISO 8601 date when the analysis was completed and the BAM file written.

Analysis center – NCBI short name for center that performed the analysis

Analysis Files and Checksums – Each analysis file and its MD5 checksum.

2.2.1. Reference Alignment Metadata

This section enumerates metadata components that are specific to reference alignment analysis objects.

Standard Assembly – Controlled name for the reference assembly or set of reference sequences used in the alignment. The following table shows a catalog of standard assemblies that are supported by NCBI. Other SRAs may define and support different assemblies. A set of cross referenced sequences may also be specified as the reference assembly.

short_name	Description	source
GRCh37	GRCh37 is the Genome Reference Consortium Human Reference 37 released 24-FEB-2009, and includes haploid and alternative loci sequences.	http://www.ncbi.nlm.nih.gov/projects/genome/assembly/grc/human/index.shtml

Table continues on next page...

Table continued from previous page.

short_name	Description	source
	This reference can also be specified in the NAME field (db="gencoll", accession="GCA_000001405.1")	
GRCh37-lite	GRCh37-lite is a subset of the full GRCh37 human genome assembly plus the human mitochondrial genome reference sequence (the "rCRS") from Mitomap.org. This set of sequences excludes all the alternate loci scaffolds of the full GRCh37 assembly, and has the pseudo-autosomal regions (PARs) on chromosome Y masked with Ns. This haploid representation of the genome is provided as a convenience for use in alignment pipelines that cannot handle the multiple placements expected in the PARs and in regions of the genome that are represented by the alternate loci.	http://www.ncbi.nlm.nih.gov/projects/genome/assembly/grc/human/index.shtml http://www.mitomap.org/MITOMAP
HG18	The March 2006 human reference sequence (NCBI Build 36.1) was produced by the International Human Genome Sequencing Consortium and is distributed by UCSC.	http://genome.ucsc.edu/cgi-bin/hgGateway?db=hg18
NCBI36	NCBI Build 36.3 released 24 March 2008. This build consists of a reference assembly for the whole genome, alternate assemblies for the whole genome produced by Celera and by JCVI, plus alternate assemblies for some parts of the genome.	ftp://ftp.ncbi.nlm.nih.gov/genomes/H_sapiens/ARCHIVE/BUILD.36.3/
NCBI36-HG18_Broad_variant	Broad Institute variant of Build 36/HG 18.	ftp://ftp.ncbi.nlm.nih.gov/genomes/H_sapiens/ARCHIVE/BUILD.36.3/special_requests/assembly_variants/NCBI36-HG18_Broad_variant.README
NCBI36_BCCAGSC_variant	British Columbia Cancer Agency Genome Sequencing Center variant of Build 36/HG 18.	ftp://ftp.ncbi.nlm.nih.gov/genomes/H_sapiens/ARCHIVE/BUILD.36.3/

Table continues on next page...

Table continued from previous page.

short_name	Description	source
		special_requests/assembly_variants/ NCBI36_BCCAGSC_variant.README
NCBI36_BCM_variant	Baylor College of Medicine variant of Build 36/HG 18.	ftp://ftp.ncbi.nlm.nih.gov/genomes/ H_sapiens/ARCHIVE/BUILD.36.3/ special_requests/assembly_variants/ NCBI36_BCM_variant.README
NCBI36_WUGSC_variant	Washington University variant of Build 36/HG 18.	ftp://ftp.ncbi.nlm.nih.gov/genomes/ H_sapiens/ARCHIVE/BUILD.36.3/ special_requests/assembly_variants/ NCBI36_WUGSC_variant.README

Custom assembly – It is possible to specify a list of contigs including de novo assemblies of unmapped reads that together comprises the reference sequence. More development is needed to define the business rules that would apply to this kind of reference specification.

Processing pipeline – The sequence of processes/tools/operations and their versions can be specified for the alignment process.

Processing directives – certain specific instructions to the data loading software, or properties that users of the data should be aware of:

- **alignment_includes_unaligned_reads** - Whether unaligned reads are provided in the alignment, and what to do with them
- **alignment_marks_duplicate_reads** - Whether duplicates are removed from the alignment
- **alignment_includes_failed_reads** - Whether non-PF filtered reads have been included in the read groups

2.3. Study Metadata

For open SRA submissions, the submitter must create or reference a SRA data producing study (SRP).

For protected SRA submissions, the submitter must reference an existing dbGaP authorized access study (phs). Reference can be made to the study handle with refcenter=”NCBI”. Submitters should NOT create these records.

2.4. Sample Metadata

For open SRA submissions, the submitter must create or reference a SRA sample or BioSample (SRS).

For protected SRA submissions, the submitter must reference an existing BioSample record (SRS). Reference can be made to the submitted sample name with refcenter set to the original repository short name. Submitters should NOT create these records.

Open SRA samples or Biosamples have diverse attributes and information content.

Protected SRA samples are exported from dbGaP and make visible a standard subset of attributes, including at the time of this writing:

Title – Brief yet unique headline returned with the record as part of a search result.

Identifiers – SRS accession, dbGaP sample accession

Organism – Target organism {human}

Original_repository – Namespace for sample set {TCGA}

Submitted_sample_id – Sample name {TCGA aliquot id}

Submitted_subject_id – Subject name {TCGA subject id, substring of the aliquot id}

Sex – {male, female, unknown}

Sample_type – Project specific sample type {TCGA: normal, primary tumor, etc}

Is_tumor – {0,1}

Histological_type – Sample diagnosis {TCGA: Serous Cystadenocarcinoma, etc}

Analyte_type – {DNA, RNA, etc}

Study_name – Short name for the parent study {TCGA}

Description – Free form text describing the sample.

Links – Includes link to parent dbGaP authorized access study homepage

An example of a TCGA record that has this information:

http://www.ncbi.nlm.nih.gov/biosample/limits?term=TCGA-13-0725-01A-01D-0359-05

2.5. Library Metadata

Each library mentioned in the BAM will map to a new or existing SRA experiment. The SRA experiment contains the following data:

Experiment title – The title string that will be presented to users of the public archive when this record is retrieved in a search result. Please limit this string to 80 characters.

Experiment description – Description of the library and its sequencing.

Library Name – Controlled vocabulary of terms describing overall strategy of the library.
Library Strategy – Controlled vocabulary of terms describing overall strategy of the library. Terms used by TCGA include {WGS, WXS, RNA-Seq}.

Library Source – Controlled vocabulary of terms describing starting material from the sample. Terms used by TCGA include {GENOMIC, TRANSCRIPTOMIC*}.

Library Selection method – Controlled vocabulary of terms describing selection or reduction method use in library construction. Terms used by TCGA include {Random, Hybrid Selection}.

Library Layout – Specification of the layout: fragment/paired, and if paired, the nominal insert size and standard deviation.

Library Protocol description – Description of the library construction protocol, or reference to a standard protocol.

Targeted loci* - Set of loci to be selected for sequencing {16S RNA, exome} and associated probes.

Platform – Controlled vocabulary of platform type {Illumina, LS454, AB_SOLID, CompleteGenomics}

Instrument model – Controlled vocabulary of instrument models {Illumina Genome Analyzer II, etc}

Expected sequence length – Number of raw bases or color space calls expected for the read (includes both mate pairs and all technical portions).

Sequence processing software and version – Name and version of sequencing processing software used.

2.6. Run Metadata

Each read group will map to exactly one new or existing SRA run.

Run name – Production flowcell/slide/plate name

Run date – ISO 8601 date the run was produced

Run center – NCBI center short name where the run was produced (useful if different from the submitter).

Run file info – Information about the run data file(s). If BAM, then this is the BAM file name and its checksum.

Processing directives – certain specific instructions to the data loading software, encoded as tag-value attributes, including:

- Actual raw sequence length, including both mate pairs and all technical portions.
- Quality scoring system {phred, log-odds}
- Quality basis character {! or @}

Submission Maintenance Guide

Created: March 30, 2012; Updated: May 15, 2013.

Overview

This document is intended to review how to update and maintain a submission thought XML modification.

Goals

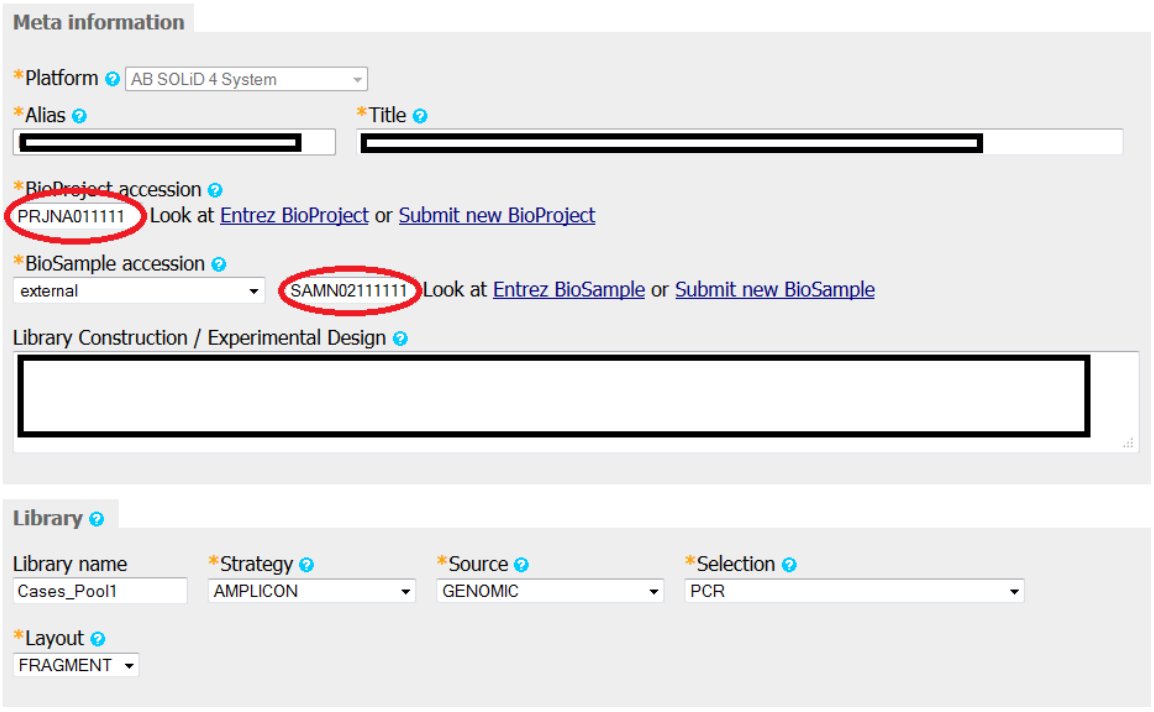
This document will address the following:

- Updating a Submission using the interactive interface
- Proper XML format for modification/update of a submission
- Address the XML tags that are blocked for updates

Interactive Update and Maintenance of a Submission

Updating an Experiment

It is possible to update the Experiment even after the Run is loaded. It is possible to update any of the text fields except the "Alias". It is also possible to move the Experiment from one Study to another and it is possible to assign a new Sample to the Experiment as well. You would update the two drop boxes circled in red. Click "Save" to save your changes.



Cannot Move Runs Between Experiments

If a Run was created and the data was loaded for the wrong Experiment and the Run needs to be moved, please contact SRA at sra@ncbi.nlm.nih.gov.

Adding Links to your Submission

Maintenance through XML

Setting up the Submission.xml File

The submission.xml file is a file that is designed to identify what files are being sent and what object is going to be added or modified.

A simple submission.xml file that is intended to add SRA objects to an existing submission will look like this:

```
<SUBMISSION xmlns:xsi="http://www.w3.org/2001/XMLSchemainstance"
xmlns="" alias=" Modification" lab_name="" center_name="NCBI"
accession="SRA049652"><CONTACTS>
        <CONTACT name="" />
    </CONTACTS>
     <ACTIONS>
         <ACTION>
            <ADD source="study3.xml" schema="study" notes="study
descriptor" />
        </ACTION>
         <ACTION>
            < ADD source="experiment3.xml" schema="experiment"
notes="experiment descriptor" />
        </ACTION>
         <ACTION>
            < ADD source="run2.xml" schema="run" notes="run
descriptor" />
        </ACTION>
         <ACTION>
            < ADD source="sample3.xml" schema="sample" notes="sample
descriptor" />
         </ACTION>
    </ACTIONS>
    <FILES />
</SUBMISSION>
```

A simple submission.xml file that is intended to update all parts of the submission will look like this:

```
<SUBMISSION xmlns:xsi="http://www.w3.org/2001/XMLSchemainstance"
xmlns="" alias=" Modification" lab_name="" center_name="NCBI"
accession="SRA049652"><CONTACTS>
        <CONTACT name="" />
    </CONTACTS>
```

```
     <ACTIONS>
         <ACTION>
            <MODIFY source="study3.xml" schema="study" notes="study
descriptor" />
        </ACTION>
         <ACTION>
            <MODIFY source="experiment3.xml" schema="experiment"
notes="experiment descriptor" />
        </ACTION>
         <ACTION>
            <MODIFY source="run2.xml" schema="run" notes="run
descriptor" />
        </ACTION>
         <ACTION>
            <MODIFY source="sample3.xml" schema="sample" notes="sample
descriptor" />
         </ACTION>
    </ACTIONS>
    <FILES />
</SUBMISSION>
```

It is possible to add and update objects in an existing submission using a single submission.xml file. You should not make a submission set because we do not process submission sets. You would need to make sure that the ADD and MODIFY commands are applied to different files like the following:

```
…..
<ACTIONS>
         <ACTION>
            <ADD source="study3.xml" schema="study" notes="study
descriptor" />
        </ACTION>
         <ACTION>
            < ADD source="experiment3.xml" schema="experiment"
notes="experiment descriptor" />
        </ACTION>
         <ACTION>
            < ADD source="run3.xml" schema="run" notes="run
descriptor" />
        </ACTION>
         <ACTION>
            < ADD source="sample3.xml" schema="sample" notes="sample
descriptor" />
         </ACTION>
         <ACTION>
            <MODIFY source="study1.xml" schema="study" notes="study
descriptor" />
        </ACTION>
         <ACTION>
            < MODIFY source="experiment1.xml" schema="experiment"
notes="experiment descriptor" />
        </ACTION>
```

```
            <ACTION>
               < MODIFY source="run1.xml" schema="run" notes="run
        descriptor" />
              </ACTION>
               <ACTION>
               < MODIFY source="sample1.xml" schema="sample" notes="sample
        descriptor" />
               </ACTION>
           </ACTIONS>
        …….
```

Setting up the Other XML Files

Setting up the other XML files for the SRA Study, Experiment, Sample, and Run will be very similar. It is possible to update or add many SRA objects at once by using sets. For instance it is possible to update several Experiments at once by using the EXPERIMENT_SET tag:

```
<EXPERIMENT_SET>
    <EXPERIMENT xmlns="" alias="Experiment" center_name="NCBI " accession="SRX117934">
        <TITLE>454</TITLE>
        <STUDY_REF accession="SRP010597"></STUDY_REF>
        <DESIGN>
            <DESIGN_DESCRIPTION>454 Library</DESIGN_DESCRIPTION>
            <SAMPLE_DESCRIPTOR accession="SRS290404"></SAMPLE_DESCRIPTOR>
…….
        <PROCESSING></PROCESSING>
    </EXPERIMENT>
    <EXPERIMENT xmlns="" alias="Experiment2" center_name="NCBI " >
        <TITLE>Illumina</TITLE>
        <STUDY_REF accession="SRP010597"></STUDY_REF>
        <DESIGN>
            <DESIGN_DESCRIPTION>Illumina library</DESIGN_DESCRIPTION>
            <SAMPLE_DESCRIPTOR accession="SRS290404"></SAMPLE_DESCRIPTOR>
            <LIBRARY_DESCRIPTOR>
                <LIBRARY_NAME>Human</LIBRARY_NAME>
                <LIBRARY_STRATEGY>WGS</LIBRARY_STRATEGY>
                <LIBRARY_SOURCE>GENOMIC</LIBRARY_SOURCE>
                <LIBRARY_SELECTION>RANDOM</LIBRARY_SELECTION>
                <LIBRARY_LAYOUT>
                    <SINGLE></SINGLE>
                </LIBRARY_LAYOUT>
                …….
    </EXPERIMENT>
</EXPERIMENT_SET>
```

There are SAMPLE_SET, STUDY_SET, and RUN_SET tags as well. This makes it very easy to update large submissions using a single file. It is possible to update any attributes and free text fields through XML before the submission is released and afterwards as well.

It is possible to specify a SRA object using an alias, an accession or both when writing the XML for an update.

Adding Pubmed link to your SRA Study

To add a Pubmed link to your study you would use the following XML code in your study XML.

```
<STUDY_LINKS>
    <STUDY_LINK>
        <ENTREZ_LINK>
            <DB>pubmed</DB>
            <ID>xxxxxxxx</ID>
        </ENTREZ_LINK>
    </STUDY_LINK>
</STUDY_LINKS>
```

Where you would enter a valid Pubmed ID into the tag “ID” and submit the new XML with the action “MODIFY”.

Adding Links to your Home page

It can be beneficial to add links to the webpage for your projects or to another website that is relevant to the data. Below is an XML code to add a URL link. Please substitute the information in the LABEL tag and the URL tag. The URL address must be full, including the “http://”.

```
<STUDY_LINKS>
    <STUDY_LINK>
        <URL_LINK>
            <LABEL>Human Sequencing Project</LABEL>
            <URL>http://www.HSP.ord/HPS</URL>
        </URL_LINK>
    </STUDY_LINK>
</STUDY_LINKS>
```

Parts of the Schema that are Blocked from Modification

Certain parts of the Experiment schema must be blocked from modification because they are used in file processing.

Before a Run is loaded

There are no parts of the schema that are locked for editing through XML if a Run is not loaded.

After a Run is loaded

1. SPOT_DESCRIPTOR
2. Sample POOL
3. PLATFORM
4. DATA_BLOCK

If your Experiment contains multiple Runs and one Run has loaded and the others have error messages. It is possible to update the Run itself with a correct PLATFORM, SPOT_DESCRITOR and POOL.

Common Errors and Methods to Fix Them

Run status is "wait" 24 hours after submitting data files

If you have uploaded your data files using FTP or Aspera and 24 hours later your Run status is set to "wait" that indicates that we were not able to match the information in the Run to the data file you uploaded. Usually this means that the md5sum did not match because the file was corrupted during transfer. Another possibility is that the file name did not match, please make sure you have entered the full name of the file including the extension (.fastq, .sff, .srf, .fq, etc).

If the file name is correct, please just re-upload the data and check back in 24 hours again.

If the file name is incorrect, it is possible to fix this through the interactive interface or by submitting an update XML for the Run(s) with the correct file name.

Run status is "ERROR:data out of range: converting quality"

This is an error that is associated with fastq files where the qualities for the sequences are encoded differently that our default. It is best to contact SRA at sra@ncbi.nlm.nih.gov. This error can only be fixed through XML modification.

Run status is "data inconsistent: length of reads in file(s): 101 is greater than spot length declared in experiment: 98"

This error indicates that in the Experiment the read length was entered as 98, but the read lengths in the file(s) is 101. Please update the Experiment with the correct read length. The update can be done through the interactive interface or through XML.

Run status is "data inconsistent: cumulative length of reads data in file(s): 49 is less than spot length declared in experiment: 98, most probably mate-pair is absent in spot 'FC81B6AABXX:6:1101:1205:2049'"

This problem arises when a mate paired fastq file has missing mates. Please check your mate pairs in the fastq file(s) and make sure that every read has a mate with the correct header.

Another possible solution can be to make sure that the data files were not separated. In the interactive interface please make sure that all the fields in the Run are the same for a pair of fastq files. If any of the fields are different the files will not be processed together and cause this error message to appear.

Run status is "no assembly information is provided" when you submit BAM files

This error indicates that the BAM file does not contain a reference sequences and you did not enter a FastA file into the Run which contains the reference sequence. This problem can be solved through the interactive interface where you would add the FastA file of the reference sequence into the Run. You may also send an update XML with the FastA file included in the same DATA_BLOCK as the BAM file.

Run status is "bam files are loaded whole. no multiple references allowed"

If your Run has the above error message it is best to contact SRA at sra@ncbi.nlm.nih.gov.

SRA XML Writer's Guide

SRA Glossary

Created: May 12, 2011; Updated: May 20, 2011.

1. Overview

This document lists terms and usage for SRA XML. Each set of terms is associated with a "descriptor," which packages like concepts together into computable text data that can be expressed as XML.

2. Sample Descriptor

The sample descriptor captures information that would allow the user of the SRA to map the sequencing data to a single sample, or a sample pool member.

See Table 1 for sample descriptor terms.

Table 1 – Sample Descriptor Terms

tag	Description
/SAMPLE_DESCRIPTOR:	The SAMPLE_DESCRIPTOR specifies how to decode the individual reads of interest from the monolithic spot sequence. The spot descriptor contains aspects of the experimental design, platform, and processing information. There will be two methods of specification: one will be an index into a table of typical decodings, the other being an exact specification.
/SAMPLE_DESCRIPTOR/ @accession:	Identifies a record by its accession. The scope of resolution is the entire Archive.
/SAMPLE_DESCRIPTOR/ @refcenter:	The center namespace of the attribute "refname". When absent, the namespace is assumed to be the current submission.
/SAMPLE_DESCRIPTOR/ @refname:	Identifies a record by name that is known within the namespace defined by attribute "refcenter" Use this field when referencing an object for which an accession has not yet been issued.
/SAMPLE_DESCRIPTOR/POOL:	Identifies a list of group/pool/multiplex sample members. This implies that this sample record is a group, pool, or multiplex, but is continues to receive its own accession and can be referenced by an experiment. By default if no match to any of the listed members can be determined, then the default sampel reference is used.
/SAMPLE_DESCRIPTOR/ POOL/MEMBER:	Impementation of lookup table between Sample Pool member and identified read_group_tags for a given READ_LABEL
/SAMPLE_DESCRIPTOR/ POOL/MEMBER/@accession:	Identifies a record by its accession. The scope of resolution is the entire Archive.
/SAMPLE_DESCRIPTOR/ POOL/MEMBER/ @member_name:	Label a sample within a scope of the pool

Table 1 continues on next page...

Table 1 continued from previous page.

tag	Description
/SAMPLE_DESCRIPTOR/ POOL/MEMBER/@proportion:	Proportion of this sample (in percent) that was included in sample pool.
/SAMPLE_DESCRIPTOR/ POOL/MEMBER/@refcenter:	The center namespace of the attribute "refname". When absent, the namespace is assumed to be the current submission.
/SAMPLE_DESCRIPTOR/ POOL/MEMBER/@refname:	Identifies a record by name that is known within the namespace defined by attribute "refcenter" Use this field when referencing an object for which an accession has not yet been issued.
/SAMPLE_DESCRIPTOR/ POOL/MEMBER/ READ_LABEL/@read_group_tag:	Assignment of read_group_tag to decoded read

3. Library Descriptor

The library descriptor captures information that would allow the user of the SRA to interpret the sequencing data's origin and preparation.

See Table 2 for library descriptor terms.

Table 2 - Library Descriptor Terms

tag	**Description**	**link**
/LIBRARY_DESCRIPTOR:	The LIBRARY_DESCRIPTOR specifies the origin of the material being sequenced and any treatments that the material might have undergone that affect the sequencing result. This specification is needed even if the platform does not require a library construction step per se.	
/LIBRARY_DESCRIPTOR/ LIBRARY_CONSTRUCTION_PROTOCOL:	Free form text describing the protocol by which the sequencing library was constructed.	
/LIBRARY_DESCRIPTOR/LIBRARY_LAYOUT:	LIBRARY_LAYOUT specifies whether to expect single, paired, or other configuration of reads. In the case of paired reads, information about the	

Table 2 continues on next page...

Table 2 continued from previous page.

	relative distance and orientation is specified.	
/LIBRARY_DESCRIPTOR/LIBRARY_LAYOUT/PAIRED/@NOMINAL_LENGTH:		
/LIBRARY_DESCRIPTOR/LIBRARY_LAYOUT/PAIRED/@NOMINAL_SDEV:		
/LIBRARY_DESCRIPTOR/LIBRARY_LAYOUT/PAIRED/@ORIENTATION:		
/LIBRARY_DESCRIPTOR/LIBRARY_LAYOUT/SINGLE:	Reads are unpaired (usual case).	
/LIBRARY_DESCRIPTOR/LIBRARY_NAME:	The submitter's name for this library.	
/LIBRARY_DESCRIPTOR/LIBRARY_SELECTION:	Whether any method was used to select for or against, enrich, or screen the material being sequenced.	
/LIBRARY_DESCRIPTOR/LIBRARY_SELECTION[5-methylcytidine antibody]:	Selection of methylated DNA fragments using an antibody raised against 5-methylcytosine or 5-methylcytidine (m5C).	
/LIBRARY_DESCRIPTOR/LIBRARY_SELECTION[CAGE]:	Cap-analysis gene expression.	<XREF_LINK><DB>pubmed</DB><ID>14663149</ID> </XREF_LINK> http://www.ncbi.nlm.nih.gov/pubmed?term=14663149[uid]
/LIBRARY_DESCRIPTOR/LIBRARY_SELECTION[CF-H]:	Cot-filtered highly repetitive genomic DNA	
/LIBRARY_DESCRIPTOR/LIBRARY_SELECTION[CF-M]:	Cot-filtered moderately repetitive genomic DNA	
/LIBRARY_DESCRIPTOR/LIBRARY_SELECTION[CF-S]:	Cot-filtered single/low-copy genomic DNA	
/LIBRARY_DESCRIPTOR/LIBRARY_SELECTION[CF-T]:	Cot-filtered theoretical single-copy genomic DNA	
/LIBRARY_DESCRIPTOR/LIBRARY_SELECTION[ChIP]:	Chromatin immunoprecipitation	
/LIBRARY_DESCRIPTOR/LIBRARY_SELECTION[DNAse]:	Deoxyribonuclease (MNase) digestion	

Table 2 continues on next page...

Table 2 continued from previous page.

/LIBRARY_DESCRIPTOR/ LIBRARY_SELECTION[HMPR]:	Hypo-methylated partial restriction digest	
/LIBRARY_DESCRIPTOR/ LIBRARY_SELECTION[Hybrid Selection]:	Selection by hybridization in array or solution.	
/LIBRARY_DESCRIPTOR/ LIBRARY_SELECTION[MBD2 protein methyl-CpG binding domain]:	Enrichment by methyl-CpG binding domain.	
/LIBRARY_DESCRIPTOR/ LIBRARY_SELECTION[MF]:	Methyl Filtrated	
/LIBRARY_DESCRIPTOR/ LIBRARY_SELECTION[MNase]:	Micrococcal Nuclease (MNase) digestion	
/LIBRARY_DESCRIPTOR/ LIBRARY_SELECTION[MSLL]:	Methylation Spanning Linking Library	
/LIBRARY_DESCRIPTOR/ LIBRARY_SELECTION[PCR]:	Source material was selected by designed primers.	
/LIBRARY_DESCRIPTOR/ LIBRARY_SELECTION[RACE]:	Rapid Amplification of cDNA Ends.	<XREF_LINK> <DB>pubmed</DB> <ID>2461560</ID> </ XREF_LINK> http://www.ncbi.nlm.nih.gov/pubmed?term=2461560[uid]
/LIBRARY_DESCRIPTOR/ LIBRARY_SELECTION[RANDOM PCR]:	Source material was selected by randomly generated primers.	
/LIBRARY_DESCRIPTOR/ LIBRARY_SELECTION[RANDOM]:	Random selection by shearing or other method.	
/LIBRARY_DESCRIPTOR/ LIBRARY_SELECTION[RT-PCR]:	Source material was selected by reverse transcription PCR	
/LIBRARY_DESCRIPTOR/ LIBRARY_SELECTION[Reduced Representation]:	Reproducible genomic subsets, often generated by restriction fragment size selection, containing a manageable number of loci to facilitate re-sampling.	
/LIBRARY_DESCRIPTOR/ LIBRARY_SELECTION[Restriction Digest]:	DNA fractionation using restriction enzymes.	
/LIBRARY_DESCRIPTOR/ LIBRARY_SELECTION[cDNA]:	PolyA selection or enrichment for messenger	

Table 2 continues on next page...

Table 2 continued from previous page.

	RNA (mRNA). complementary DNA.	
/LIBRARY_DESCRIPTOR/ LIBRARY_SELECTION[other]:	Other library enrichment, screening, or selection process.	
/LIBRARY_DESCRIPTOR/ LIBRARY_SELECTION[size fractionation]:	Physical selection of size appropriate targets.	
/LIBRARY_DESCRIPTOR/ LIBRARY_SELECTION[unspecified]:	Library enrichment, screening, or selection is not specified.	
/LIBRARY_DESCRIPTOR/LIBRARY_SOURCE:	The LIBRARY_SOURCE specifies the type of source material that is being sequenced.	
/LIBRARY_DESCRIPTOR/ LIBRARY_SOURCE[GENOMIC]:	Genomic DNA (includes PCR products from genomic DNA).	
/LIBRARY_DESCRIPTOR/ LIBRARY_SOURCE[METAGENOMIC]:	Mixed material from metagenome.	
/LIBRARY_DESCRIPTOR/ LIBRARY_SOURCE[METATRANSCRIPTOMIC]:	Transcription products from community targets	
/LIBRARY_DESCRIPTOR/ LIBRARY_SOURCE[OTHER]:	Other, unspecified, or unknown library source material.	
/LIBRARY_DESCRIPTOR/ LIBRARY_SOURCE[SYNTHETIC]:	Synthetic DNA.	
/LIBRARY_DESCRIPTOR/ LIBRARY_SOURCE[TRANSCRIPTOMIC]:	Transcription products or non genomic DNA (EST, cDNA, RT-PCR, screened libraries).	
/LIBRARY_DESCRIPTOR/ LIBRARY_SOURCE[VIRAL RNA]:	Viral RNA.	
/LIBRARY_DESCRIPTOR/LIBRARY_STRATEGY:	Sequencing technique intended for this library.	
/LIBRARY_DESCRIPTOR/ LIBRARY_STRATEGY[AMPLICON]:	Sequencing of overlapping or distinct PCR or RT-PCR products. For example, metagenomic community profiling using SSU rRNA .	<XREF_LINK> <DB>pubmed</DB> <ID>19023400</ID> </ XREF_LINK> http://www.ncbi.nlm.nih.gov/pubmed?term=19023400[uid]

Table 2 continues on next page...

Table 2 continued from previous page.

/LIBRARY_DESCRIPTOR/ LIBRARY_STRATEGY[Bisulfite-Seq]:	MethylC-seq. Sequencing following treatment of DNA with bisulfite to convert cytosine residues to uracil depending on methylation status.	<XREF_LINK> <DB>pubmed</DB> <ID>19829295</ID> </ XREF_LINK> http:// www.ncbi.nlm.nih.gov/ pubmed?term= 19829295[uid]
/LIBRARY_DESCRIPTOR/ LIBRARY_STRATEGY[CLONEEND]:	Clone end (5', 3', or both) sequencing.	<XREF_LINK> <DB>pubmed</DB> <ID>18836033</ID> </ XREF_LINK> http:// www.ncbi.nlm.nih.gov/ pubmed?term= 18836033[uid]
/LIBRARY_DESCRIPTOR/ LIBRARY_STRATEGY[CLONE]:	Genomic clone based (hierarchical) sequencing.	
/LIBRARY_DESCRIPTOR/ LIBRARY_STRATEGY[CTS]:	Concatenated Tag Sequencing	
/LIBRARY_DESCRIPTOR/ LIBRARY_STRATEGY[ChIP-Seq]:	Direct sequencing of chromatin immunoprecipitates.	<XREF_LINK> <DB>pubmed</DB> <ID>18684996</ID> </ XREF_LINK> http:// www.ncbi.nlm.nih.gov/ pubmed?term= 18684996[uid]
/LIBRARY_DESCRIPTOR/ LIBRARY_STRATEGY[DNase-Hypersensitivity]:	Sequencing of hypersensitive sites, or segments of open chromatin that are more readily cleaved by DNaseI.	
/LIBRARY_DESCRIPTOR/ LIBRARY_STRATEGY[EST]:	Single pass sequencing of cDNA templates	
/LIBRARY_DESCRIPTOR/ LIBRARY_STRATEGY[FINISHING]:	Sequencing intended to finish (close) gaps in existing coverage.	
/LIBRARY_DESCRIPTOR/ LIBRARY_STRATEGY[FL-cDNA]:	Full-length sequencing of cDNA templates	
/LIBRARY_DESCRIPTOR/ LIBRARY_STRATEGY[MBD-Seq]:	Direct sequencing of methylated fractions sequencing strategy.	

Table 2 continues on next page...

Table 2 continued from previous page.

/LIBRARY_DESCRIPTOR/ LIBRARY_STRATEGY[MNase-Seq]:	Direct sequencing following MNase digestion.	
/LIBRARY_DESCRIPTOR/ LIBRARY_STRATEGY[MRE-Seq]:	Methylation-Sensitive Restriction Enzyme Sequencing strategy.	
/LIBRARY_DESCRIPTOR/ LIBRARY_STRATEGY[MeDIP-Seq]:	Methylated DNA Immunoprecipitation Sequencing strategy.	
/LIBRARY_DESCRIPTOR/ LIBRARY_STRATEGY[OTHER]:	Library strategy not listed.	
/LIBRARY_DESCRIPTOR/ LIBRARY_STRATEGY[POOLCLONE]:	Shotgun of pooled clones (usually BACs and Fosmids).	
/LIBRARY_DESCRIPTOR/ LIBRARY_STRATEGY[RNA-Seq]:	Random sequencing of whole transcriptome.	<XREF_LINK> <DB>pubmed</DB> <ID>18611170</ID> </ XREF_LINK> http:// www.ncbi.nlm.nih.gov/ pubmed?term= 18611170[uid]
/LIBRARY_DESCRIPTOR/ LIBRARY_STRATEGY[WCS]:	Random sequencing of a whole chromosome or other replicon isolated from a genome.	
/LIBRARY_DESCRIPTOR/ LIBRARY_STRATEGY[WGS]:	Random sequencing of the whole genome.	
/LIBRARY_DESCRIPTOR/ LIBRARY_STRATEGY[WXS]:	Random sequencing of exonic regions selected from the genome.	<XREF_LINK> <DB>pubmed</DB> <ID>20111037</ID> </ XREF_LINK> http:// www.ncbi.nlm.nih.gov/ pubmed?term= 20111037[uid]
/LIBRARY_DESCRIPTOR/ POOLING_STRATEGY:	The optional pooling strategy indicates how the library or libraries are organized if multiple samples are involved.	
/LIBRARY_DESCRIPTOR/ POOLING_STRATEGY[multiplexed libraries]:	Multiple libraries were prepared each of which can be distinguished in the	

Table 2 continues on next page...

Table 2 continued from previous page.

	sequencing result through a molecular barcode or other indicator. Each library may be made from the same or different samples. This option is expected when the libraries are part of the same study.	
/LIBRARY_DESCRIPTOR/ POOLING_STRATEGY[multiplexed samples]:	A library was prepared of multiplexed samples each of which can be distinguished in the sequencing result through a molecular barcode or other indicator.	
/LIBRARY_DESCRIPTOR/ POOLING_STRATEGY[none]:	There is a one-to-one correspondence with sample and library (normal case).	
/LIBRARY_DESCRIPTOR/ POOLING_STRATEGY[simple pool]:	The sequencing is done on a pool of identified samples which cannot be distinguished in the sequencing result.	
/LIBRARY_DESCRIPTOR/ POOLING_STRATEGY[spiked library]:	One library is prepared with an oligonucleotide sequence included that when sequenced can help provide quality control for the library.	
/LIBRARY_DESCRIPTOR/TARGETED_LOCI:	Names the gene(s) or locus(loci) or other genomic feature(s) targeted by the sequence.	
/LIBRARY_DESCRIPTOR/TARGETED_LOCI/ LOCUS/@locus_name[16S rRNA]:	Bacterial ribosomal RNA hypervariable region(s).	
/LIBRARY_DESCRIPTOR/TARGETED_LOCI/ LOCUS/@locus_name[exome]:	All exonic regions of the genome.	
/LIBRARY_DESCRIPTOR/TARGETED_LOCI/ LOCUS/@locus_name[other]:	Other locus, please describe.	
/LIBRARY_DESCRIPTOR/TARGETED_LOCI/ LOCUS/PROBE_SET:	Reference to an archived primer or probe set. Example: dbProbe	

Table 2 continues on next page...

Table 2 continued from previous page.

/LIBRARY_DESCRIPTOR/TARGETED_LOCI/LOCUS/PROBE_SET/DB:	INSDC controlled vocabulary of permitted cross references. Please see http://www.insdc.org/db_xref.html . For example, FLYBASE.	
/LIBRARY_DESCRIPTOR/TARGETED_LOCI/LOCUS/PROBE_SET/ID:	Accession in the referenced database. For example, FBtr0080008 (in FLYBASE).	

4. Spot Descriptor

The spot descriptor captures information that would allow the user of the SRA to interpret the sequencing data and differentiate between technical and application extents in the read. Reads that are mate pairs are concatenated into a single monolithic "spot" sequence.

See Table 3 for spot descriptor terms.

Table 3 - Spot Descriptor Terms

Tag	description
/SPOT_DESCRIPTOR:	The SPOT_DESCRIPTOR specifies how to decode the individual reads of interest from the monolithic spot sequence. The spot descriptor contains aspects of the experimental design, platform, and processing information. There will be two methods of specification: one will be an index into a table of typical decodings, the other being an exact specification.
/SPOT_DESCRIPTOR/SPOT_DECODE_SPEC/ADAPTER_SPEC:	Some technologies will require knowledge of the sequencing adapter or the last base of the adapter in order to decode the spot.
/SPOT_DESCRIPTOR/SPOT_DECODE_SPEC/READ_SPEC/BASE_COORD:	The location of the read start in terms of base count (1 is beginning of spot).
/SPOT_DESCRIPTOR/SPOT_DECODE_SPEC/READ_SPEC/CYCLE_COORD:	The location of the read start in terms of cycle count (1 is beginning of spot).
/SPOT_DESCRIPTOR/SPOT_DECODE_SPEC/READ_SPEC/EXPECTED_BASECALL:	An expected basecall for a current read. Read will be zero-length if basecall is not present. Users of this facility should start migrating to EXPECTED_BASECALL_TABLE, as this field will be phased out.
/SPOT_DESCRIPTOR/SPOT_DECODE_SPEC/READ_SPEC/EXPECTED_BASECALL/@base_coord:	Specify an optional starting point for tag (base offset from 1).

Table 3 continues on next page...

Table 3 continued from previous page.

Tag	description
/SPOT_DESCRIPTOR/SPOT_DECODE_SPEC/ READ_SPEC/EXPECTED_BASECALL/ @default_length:	Specify whether the spot should have a default length for this tag if the expected base cannot be matched.
/SPOT_DESCRIPTOR/SPOT_DECODE_SPEC/ READ_SPEC/EXPECTED_BASECALL_TABLE:	A set of choices of expected basecalls for a current read. Read will be zero-length if none is found.
/SPOT_DESCRIPTOR/SPOT_DECODE_SPEC/ READ_SPEC/EXPECTED_BASECALL_TABLE/ @base_coord:	Specify an optional starting point for tag (base offset from 1).
/SPOT_DESCRIPTOR/SPOT_DECODE_SPEC/ READ_SPEC/EXPECTED_BASECALL_TABLE/ @default_length:	Specify whether the spot should have a default length for this tag if the expected base cannot be matched.
/SPOT_DESCRIPTOR/SPOT_DECODE_SPEC/ READ_SPEC/EXPECTED_BASECALL_TABLE/ BASECALL:	Element's body contains a basecall, attribute provide description of this read meaning as well as matching rules.
/SPOT_DESCRIPTOR/SPOT_DECODE_SPEC/ READ_SPEC/EXPECTED_BASECALL_TABLE/ BASECALL/@match_edge:	Where the match should occur. Changes the rules on how min_match and max_mismatch are counted.
/SPOT_DESCRIPTOR/SPOT_DECODE_SPEC/ READ_SPEC/EXPECTED_BASECALL_TABLE/ BASECALL/@match_edge[end]:	Both matches and mismatches are counted. When @max_mismatch is exceeded - it is not a match. When @min_match is reached - match is declared.
/SPOT_DESCRIPTOR/SPOT_DECODE_SPEC/ READ_SPEC/EXPECTED_BASECALL_TABLE/ BASECALL/@match_edge[full]:	Only @max_mismatch influences matching process
/SPOT_DESCRIPTOR/SPOT_DECODE_SPEC/ READ_SPEC/EXPECTED_BASECALL_TABLE/ BASECALL/@match_edge[start]:	Both matches and mismatches are counted. When @max_mismatch is exceeded - it is not a match. When @min_match is reached - match is declared.
/SPOT_DESCRIPTOR/SPOT_DECODE_SPEC/ READ_SPEC/EXPECTED_BASECALL_TABLE/ BASECALL/@max_mismatch:	Maximum number of mismatches
/SPOT_DESCRIPTOR/SPOT_DECODE_SPEC/ READ_SPEC/EXPECTED_BASECALL_TABLE/ BASECALL/@min_match:	Minimum number of matches to trigger identification.
/SPOT_DESCRIPTOR/SPOT_DECODE_SPEC/ READ_SPEC/EXPECTED_BASECALL_TABLE/ BASECALL/@read_group_tag:	When match occurs, the read will be tagged with this group membership
/SPOT_DESCRIPTOR/SPOT_DECODE_SPEC/ READ_SPEC/READ_INDEX:	READ_INDEX starts at 0 and is incrementally increased for each sequential READ_SPEC within a SPOT_DECODE_SPEC
/SPOT_DESCRIPTOR/SPOT_DECODE_SPEC/ READ_SPEC/READ_LABEL:	READ_LABEL is a name for this tag, and can be used to on output to determine read name, for example F or R.

Table 3 continues on next page...

Table 3 continued from previous page.

Tag	description
/SPOT_DESCRIPTOR/SPOT_DECODE_SPEC/READ_SPEC/RELATIVE_ORDER:	The read is located beginning at the offset or cycle relative to another read. This choice is appropriate for example when specifying a read that follows a variable length expected sequence(s).
/SPOT_DESCRIPTOR/SPOT_DECODE_SPEC/READ_SPEC/RELATIVE_ORDER/@follows_read_index:	Specify the read index that precedes this read.
/SPOT_DESCRIPTOR/SPOT_DECODE_SPEC/READ_SPEC/RELATIVE_ORDER/@precedes_read_index:	Specify the read index that follows this read.
/SPOT_DESCRIPTOR/SPOT_DECODE_SPEC/SPOT_LENGTH:	Number of base/color calls, cycles, or flows per spot (raw sequence length or flow length including all application and technical tags and mate pairs, but not including gap lengths). This value will be platform dependent, library dependent, and possibly run dependent. Variable length platforms will still have a constant flow/cycle length.

5. Gap Descriptor

The gap descriptor captures information that would allow the user of the SRA to map the sequencing data to a putative reference genome substrate and to understand mate pairing statistics.

See Table 4 for gap descriptor terms.

Table 4 - Gap Descriptor Terms

tag	description
/GAP_DESCRIPTOR:	The GAP_DESCRIPTOR specifies how the tags that comprise a spot are to be spaced against a notional reference sequence.
/GAP_DESCRIPTOR/GAP/GAP_SPEC/@link3:	Specify the read label at the 3' end of the gap, or NULL if it's the last tag.
/GAP_DESCRIPTOR/GAP/GAP_SPEC/@link5:	Specify the read label at the 5' end of the gap, or NULL if it's the first tag.
/GAP_DESCRIPTOR/GAP/GAP_SPEC/histogram:	Frequency distribution of gap length values.
/GAP_DESCRIPTOR/GAP/GAP_SPEC/histogram/bin:	Length value bin
/GAP_DESCRIPTOR/GAP/GAP_SPEC/histogram/bin/@ktile:	k-tile where k is the k-th bin

Table 4 continues on next page...

Table 4 continued from previous page.

tag	description
/GAP_DESCRIPTOR/GAP/GAP_SPEC/ histogram/bin/@value:	Frequency count or 0.
/GAP_DESCRIPTOR/GAP/GAP_SPEC/interval:	Specify the read label at the 5' end of the gap, or NULL if it's the first tag.
/GAP_DESCRIPTOR/GAP/GAP_SPEC/interval/ @max_length:	Minimum length in base pairs of the interval.
/GAP_DESCRIPTOR/GAP/GAP_SPEC/interval/ @min_length:	Minimum length in base pairs of the interval.
/GAP_DESCRIPTOR/GAP/GAP_SPEC/statistic:	Specify the read label at the 5' end of the gap, or NULL if it's the first tag.
/GAP_DESCRIPTOR/GAP/GAP_SPEC/statistic/ @mean:	Mean length in base pairs of the interval.
/GAP_DESCRIPTOR/GAP/GAP_SPEC/statistic/ @stdev:	Standard deviation of length in base pairs of the interval.
/GAP_DESCRIPTOR/GAP/GAP_TYPE:	Specifies the gap type and parameters.
/GAP_DESCRIPTOR/GAP/GAP_TYPE/MatePair:	Mated tags with predicted separation and orientation.
/GAP_DESCRIPTOR/GAP/GAP_TYPE/MatePair/ @orientation[anti-normal]:	Tags are facing unidirectionally on opposite strand.
/GAP_DESCRIPTOR/GAP/GAP_TYPE/MatePair/ @orientation[innie]:	Tags are facing towards each other.
/GAP_DESCRIPTOR/GAP/GAP_TYPE/MatePair/ @orientation[normal]:	Tags are facing unidiretionally.
/GAP_DESCRIPTOR/GAP/GAP_TYPE/MatePair/ @orientation[outie]:	Tags are facing away from each other.
/GAP_DESCRIPTOR/GAP/GAP_TYPE/PairedEnd:	Mated tags sequenced from two ends of a physical extent of genomic material.
/GAP_DESCRIPTOR/GAP/GAP_TYPE/Tandem:	Tandem gaps between ligands.

6. Platform Descriptor

The platform descriptor captures information about the instrument that produced the sequencing data.

See Table 5 for platform descriptor terms.

Table 5 - Platform Descriptor Terms

Tag	description	link
/PLATFORM:	The PLATFORM record selects which sequencing platform and platform-specific runtime parameters. This will be determined by the Center.	
/PLATFORM/ABI_SOLID:	ABI is 4-channel flowgram with 1-to-1 mapping between basecalls and flows	
/PLATFORM/ABI_SOLID/ INSTRUMENT_MODEL[AB SOLiD 4 System]:	AB SOLiD System 4.x system	
/PLATFORM/ABI_SOLID/ INSTRUMENT_MODEL[AB SOLiD 4hq System]:	AB SOLiD System 4hq system	
/PLATFORM/ABI_SOLID/ INSTRUMENT_MODEL[AB SOLiD 5500]:	AB SOLiD System 5500 system	
/PLATFORM/ABI_SOLID/ INSTRUMENT_MODEL[AB SOLiD 5500xl]:	AB SOLiD System 5500xl system	
/PLATFORM/ABI_SOLID/ INSTRUMENT_MODEL[AB SOLiD PI System]:	AB SOLiD System PI system	
/PLATFORM/ABI_SOLID/ INSTRUMENT_MODEL[AB SOLiD System 2.0]:	AB SOLiD System 2.0 system	
/PLATFORM/ABI_SOLID/ INSTRUMENT_MODEL[AB SOLiD System 3.0]:	AB SOLiD System 3.0 system	
/PLATFORM/ABI_SOLID/ INSTRUMENT_MODEL[AB SOLiD System]:	Unspecified early AB SOLiD system	
/PLATFORM/ABI_SOLID/ SEQUENCE_LENGTH:	The fixed number of bases expected in each raw sequence, including both mate pairs and any technical reads. This is optional in the schema now but will be required by business rules and future schema versions.	
/PLATFORM/COMPLETE_GENOMICS:	CompleteGenomics platform type. At present there is no instrument model.	
/PLATFORM/COMPLETE_GENOMICS/ INSTRUMENT_MODEL[Complete Genomics]:	Unspecified Complete Genomics system	
/PLATFORM/HELICOS:	Helicos is similar to 454 technology - uses 1-color sequential flows	
/PLATFORM/HELICOS/FLOW_COUNT:	The number of flows of challenge bases. This is a constraint on maximum read length, but not equivalent. This is optional in the schema now but will be required by business rules and future schema versions.	

Table 5 continues on next page...

Table 5 continued from previous page.

Tag	description	link
/PLATFORM/HELICOS/FLOW_SEQUENCE:	The fixed sequence of challenge bases that flow across the flowcell. This is optional in the schema now but will be required by business rules and future schema versions.	
/PLATFORM/HELICOS/ INSTRUMENT_MODEL[Helicos HeliScope]:	Helicos HeliScope system	
/PLATFORM/ILLUMINA:	Illumina is 4-channel flowgram with 1-to-1 mapping between basecalls and flows	
/PLATFORM/ILLUMINA/ INSTRUMENT_MODEL[Illumina Genome Analyzer II]:	Illumina Genome Analyzer II system	
/PLATFORM/ILLUMINA/ INSTRUMENT_MODEL[Illumina Genome Analyzer IIx]:	Illumina Genome Analyzer IIx system	
/PLATFORM/ILLUMINA/ INSTRUMENT_MODEL[Illumina Genome Analyzer]:	Illumina Genome Analyzer system	
/PLATFORM/ILLUMINA/ INSTRUMENT_MODEL[Illumina HiSeq 1000]:	Illumina HiSeq 1000 system	
/PLATFORM/ILLUMINA/ INSTRUMENT_MODEL[Illumina HiSeq 2000]:	Illumina HiSeq 2000 system	
/PLATFORM/ILLUMINA/ INSTRUMENT_MODEL[Illumina MiSeq]:	Illumina MiSeq system	
/PLATFORM/ILLUMINA/ SEQUENCE_LENGTH:	The fixed number of bases expected in each raw sequence, including both mate pairs and any technical reads.	
/PLATFORM/ION_TORRENT:	Ion Torrent Personal Genome Machine (PGM) from Life Technologies.	
/PLATFORM/ION_TORRENT/ INSTRUMENT_MODEL[Ion Torrent PGM]:	Ion Torrent PGM system	
/PLATFORM/LS454:	454 technology use 1-color sequential flows	
/PLATFORM/LS454/FLOW_COUNT:	The number of flows of challenge bases. This is a constraint on maximum read length, but not equivalent. This is optional in the schema now but will be required by business rules and future schema versions.	
/PLATFORM/LS454/FLOW_SEQUENCE:	The fixed sequence of challenge bases that flow across the picotiter plate. This is optional in the schema now but will be required by business rules and future schema versions.	

Table 5 continues on next page...

Table 5 continued from previous page.

Tag	description	link
/PLATFORM/LS454/ INSTRUMENT_MODEL[454 GS 20]:	454 GS 20 system	
/PLATFORM/LS454/ INSTRUMENT_MODEL[454 GS FLX Titanium]:	454 GS FLX Titanium system	
/PLATFORM/LS454/ INSTRUMENT_MODEL[454 GS FLX]:	454 GS FLX system	
/PLATFORM/LS454/ INSTRUMENT_MODEL[454 GS Junior]:	454 GS Junior system	
/PLATFORM/LS454/ INSTRUMENT_MODEL[454 GS]:	Unspecified early 454 GS system	
/PLATFORM/LS454/KEY_SEQUENCE:	The first bases that are expected to be produced by the challenge bases. This is optional in the schema now but will be required by business rules and future schema versions.	
/PLATFORM/PACBIO_SMRT:	PacificBiosciences platform type for the single molecule real time (SMRT) technology.	
/PLATFORM/PACBIO_SMRT/ INSTRUMENT_MODEL[PacBio RS]:	PacBio RS system	

7. Processing Descriptor

The processing descriptor captures information about the treatment of the data after the sequencing instrument produced it, prior to submission to the SRA.

See Table 6 for processing descriptor terms.

Table 6 - Processing Descriptor Terms

Tag	description
/PROCESSING/DIRECTIVES/SAMPLE_DEMUX_DIRECTIVE:	Tells the Archive who will execute the sample demultiplexing operation..
/PROCESSING/DIRECTIVES/ SAMPLE_DEMUX_DIRECTIVE[leave_as_pool]:	There shall be no sample de-multiplexing at the level of assiging individual reads to sample pool members.
/PROCESSING/DIRECTIVES/ SAMPLE_DEMUX_DIRECTIVE[submitter_demultiplexed]:	The submitter has assigned individual reads to sample pool members by providing individual files containing reads with the same member assignment.

Table 6 continues on next page...

Table 6 continued from previous page.

Tag	description
/PROCESSING/PIPELINE/PIPE_SECTION/@section_name:	Name of the processing pipeline section.
/PROCESSING/PIPELINE/PIPE_SECTION/NOTES:	Notes about the program or process for primary analysis.
/PROCESSING/PIPELINE/PIPE_SECTION/PREV_STEP_INDEX:	STEP_INDEX of the previous step in the workflow. Set toNIL if the first pipe section.
/PROCESSING/PIPELINE/PIPE_SECTION/PROGRAM:	Name of the program or process for primary analysis. This may include a test or condition that leads to branching in the workflow.
/PROCESSING/PIPELINE/PIPE_SECTION/STEP_INDEX:	Lexically ordered value that allows for the pipe section to be hierarchically ordered. The float primitive data type is used to allow for pipe sections to be inserted later on.
/PROCESSING/PIPELINE/PIPE_SECTION/VERSION:	Version of the program or process for primary analysis.

Using the SRA Data Block Descriptor

Created: October 21, 2009; Updated: September 16, 2010.

1. Overview

The SRA schema version 1.1 supports the constructs to help describe the assignment of run file objects to SRA run data blocks. Run DATA_BLOCKs are specifications for the archive loader. Once loaded into the archive, the parameters are no longer needed in order to interpret the data that was archived.

This content will eventually join the SRA XML Writer's Guide.

1.1. Overview of Data Block Descriptor Usage

The SRA Run DATA_BLOCK is intended for use to convey information to archive loaders. Once the data have been loaded into the Archive and converted into an SRA native object, the information in the DATA_BLOCK descriptor is no longer relevant to users of the data.

The DATA_BLOCK is optional in the schema, but is required for all RUN XML documents used for submission. This is so that when RUN XML documents are returned to users of the Archive, or mirrored between Archives, the DATA_BLOCK section can be redacted.

There are two classes of parameters: DATA_BLOCK descriptor attributes and FILE attributes.

1.2. Related Documents

- SRA File Formats Guide (under development)
- SRA XML Specification Release SRA_1-1 Change Notice

2. DATA_BLOCK Descriptor Attributes

2.1. Multiple Data Blocks

The XML schema allows you to specify multiple data blocks in sequence order, but you are not guaranteed to emit the blocks in any order. Use the new **DATA_BLOCK.serial** attribute to impose a total ordering on the data blocks so that they will get loaded in the order specified.

Example: One SOLiD run broken into 95 pieces for ease of transmission:

```
<DATA_BLOCK name="VAB_Florence_20080709_1_1000G_10" serial="1">
     <FILES>
     <FILE filename="Florence_20080709_1_1000G_10.0001.0001_0025.srf"
         filetype="srf">
     </FILE>
```

```
        </FILES>
    </DATA_BLOCK>
    <DATA_BLOCK name="VAB_Florence_20080709_1_1000G_10" serial="26">
        <FILES>
        <FILE filename="Florence_20080709_1_1000G_10.0002.0026_0050.srf"
            filetype="srf">
        </FILE>
        </FILES>
    </DATA_BLOCK>
```

and so on.

2.2. Multiple Samples, User De-multiplexed

The XML schema now allows you to specify multiple data blocks per run each of which is assigned to a subset of the sequencing that is associated with a particular sample. In this case the submitter has de-multiplexed the sequencing run and submitted separate files. A default file may be used to contain the reads that did not get assigned to a particular sample. The **DATA_BLOCK.member** attribute records the pool member name that the reads should be assigned to.

```
    <DATA_BLOCK
      serial = "1"
      name = "FMSX0OV"
      region = "1"
      member_name = "default"
    >
      <FILES>
        <FILE filename="default.sff"
filetype="sff"
checksum_method="MD5"
checksum="4026fc6b91ed2ffbef374a665e02802b" />
      </FILES>
    </DATA_BLOCK>
    <DATA_BLOCK
      serial = "2"
      name = "FMSX0OV"
      region = "1"
      member_name = "R27Cecum"
    >
     <FILES>
        <FILE filename="R27Cecum.sff"
filetype="sff"
checksum_method="MD5"
checksum="7f7ba170dbc6a25409a5eb6d845da88f" />
      </FILES>
    </DATA_BLOCK>
```

3. FILE Descriptor Attributes

Here is a quick guide for how to use the FILE descriptor attributes for text data. More details follow in the sections below.

Table 1 - File input use cases and DATA_BLOCK programming settings

Expected File Forms	**Library Type (F=Frag, P=paired)**	**Number of files**	**Filetype**	**DATA_BLOCK.name**	**READ_LABEL(s) Bracket indicate array of choices for one file.**	**DATA_SERIES_LABEL(s) Bracket indicates an array of choices for one file.**	**quality_scoring_system**	**quality_encoding**	**ascii_offset**
csfasta file qual file	F	2	SOLiD_native_csfasta SOLiD_native_qual	Flowcell/slide	F3	INSDC:read, INSDC:quality	phred	decimal	
csfasta file qual file	P	4	SOLiD_native	Flowcell/slide	F3, R3	INSDC:read, INSDC:quality	phred	decimal	
qseq file	F	1	Illumina_native_qseq	Flowcell	F	[INSDC:read, INSDC:quality]	log-odds	ascii	@
fastq file with barcode	F	2	Illumina_native_fastq	Flowcell	[F, B]	[INSDC:read, INSDC:quality]	log-odds	ascii	@
qseq file	P	2	Illumina_native_qseq	Flowcell	[F, R]	[INSDC:read, INSDC:quality]	log-odds	ascii	@
fastq files with barcode	P	3	Illumina_native_fastq	Flowcell	[F,R,B]	[INSDC:read, INSDC:quality]	log-odds	ascii	@
qseq int	F	2	Illumina_native_qseq Illumina_native_int	Flowcell	F, I	[INSDC:read, INSDC:quality] INSDC:intensity	log-odds	ascii	@
qseq int	P	3	Illumina_native_qseq Illumina_native_int	Flowcell	[F, R], I	[INSDC:read, INSDC:quality] INSDC:intensity	log-odds	ascii	@
seq, prb	F/P	2	Illumina_native_seq Illumina_native_prb	Flowcell	F,R	INSDC:read, INSDC:quality	log-odds	decimal	
seq, prb, int	F/P	3	Illumina_native_seq Illumina_native_prb Illumina_native_int	Flowcell	F,R	INSDC:read, INSDC:quality INSDC:intensity	log-odds	decimal	
seq, fna qual	F/P	2	454_native_seq 454_native_qual	Plate		INSDC:read, INSDC:quality	phred	decimal	
fastq	F	1	Helicos_native	Flowcell		[INSDC:read, INSDC:quality]	phred	ascii	@
fastq with decimal quality scores	F/P	1	fastq	Not used		[INSDC:read, INSDC:quality]	phred	decimal	
Fastq with character quality scores	F/P	1	fastq	Not used		[INSDC:read, INSDC:quality]	log-odds	ascii	@ or !

3.1. Multiple Segments

The submitter may present different parts of the spot sequence in distinct files. The records must exist in both files and be in the same order. The **DATA_BLOCK.FILES.FILE.READ_LABEL** connects the file with the named read in a spot descriptor.

For a certain spot descriptor:

```
<SPOT_DESCRIPTOR>
    <SPOT_DECODE_SPEC>
        <NUMBER_OF_READS_PER_SPOT>2</NUMBER_OF_READS_PER_SPOT>
        <READ_SPEC>
            <READ_INDEX>1</READ_INDEX>
            <READ_LABEL>forward</READ_LABEL>
            <READ_CLASS>Application Read</READ_CLASS>
            <READ_TYPE>Forward</READ_TYPE>
            <BASE_COORD>1</BASE_COORD>
        </READ_SPEC>
        <READ_SPEC>
            <READ_INDEX>2</READ_INDEX>
            <READ_LABEL>reverse</READ_LABEL>
            <READ_CLASS>Application Read</READ_CLASS>
            <READ_TYPE>Reverse</READ_TYPE>
            <BASE_COORD>37</BASE_COORD>
        </READ_SPEC>
    </SPOT_DECODE_SPEC>
</SPOT_DESCRIPTOR>
```

can have the associated RUN code:

```
<DATA_BLOCK name = "HWX170-FC8080_1000" sector="1">
    <FILES>
        <FILE filename="HWX170-FC8080_1000_1_1_qseq.txt"
            filetype="Illumina_native_qseq"
            checksum_method="MD5"
            checksum="d41d8cd98f00b204e9800998ecf8427e">
            <READ_LABEL>F</READ_LABEL>
            <DATA_SERIES_LABEL>INSDC:read</DATA_SERIES_LABEL>
            <DATA_SERIES_LABEL>INSDC:quality</DATA_SERIES_LABEL>
        </FILE>
        <FILE filename="HWX170-FC8080_1000_1_2_qseq.txt"
            filetype="Illumina_native_qseq"
            checksum_method="MD5"
            checksum="204e9800998ecf8427ed41d8cd98f00b">
            <READ_LABEL>R</READ_LABEL>
            <DATA_SERIES_LABEL>INSDC:read</DATA_SERIES_LABEL>
            <DATA_SERIES_LABEL>INSDC:quality</DATA_SERIES_LABEL>
        </FILE>
    </FILES>
</DATA_BLOCK>
```

3.2. Multiple Data Series

A native format submission may consist of a single data block containing multiple data series (columns) each represented by a distinct file. The **DATA_BLOCK.FILES.FILE.DATA_SERIES_LABEL** can be used to define a precise mapping between components and columns.

```
<DATA_BLOCK>
     <FILES>
         <FILE filename='Solid0044_20081126_2_F3.csfasta'
             filetype="SOLiD_native_csfasta"
             checksum_method="MD5"
             checksum="d41d8cd98f00b204e9800998ecf8427e" >
             <DATA_SERIES_LABEL>INSDC:read</DATA_SERIES_LABEL>
         </FILE>

         <FILE filename='Solid0044_20081126_2_F3_QV.qual'
             filetype="SOLiD_native_qual"
             checksum_method="MD5"
             checksum="9800998ecf8427ed41d8cd98f00b204e">
             <DATA_SERIES_LABEL>INSDC:quality</DATA_SERIES_LABEL>
         </FILE>
     </FILES>
 </DATA_BLOCK>
```

3.3. Combining Segments and Data Series

The two parameters **DATA_BLOCK.FILES.FILE.READ_LABEL** and **DATA_BLOCK.FILES.FILE.DATA_SERIES_LABEL** can be combined into a two dimensional specification of files to segments and columns.

```
<DATA_BLOCK>
     <FILES>
         <FILE filename='Solid0044_20081126_2_F3.csfasta'
             filetype="SOLiD_native_csfasta"
             checksum_method="MD5"
             checksum="d41d8cd98f00b204e9800998ecf8427e">
             <READ_LABEL>F3</READ_LABEL>
             <DATA_SERIES_LABEL>INSDC:read</DATA_SERIES_LABEL>
         </FILE>
         <FILE filename='Solid0044_20081126_2_F3_QV.qual'
             filetype="SOLiD_native_qual"
             checksum_method="MD5"
             checksum="9800998ecf8427ed41d8cd98f00b204e">
             <READ_LABEL>F3</READ_LABEL>
             <DATA_SERIES_LABEL>INSDC:quality</DATA_SERIES_LABEL>
         </FILE>
         <FILE filename='Solid0044_20081126_2_R3.csfasta'
             filetype="SOLiD_native_csfasta"
             checksum_method="MD5"
             checksum="4d1d8cd98f00b204e9800998ecf8427e" >
             <READ_LABEL>R3</READ_LABEL>
```

```
            <DATA_SERIES_LABEL>INSDC:read</DATA_SERIES_LABEL>
        </FILE>
        <FILE filename='Solid0044_20081126_2_R3_QV.qual'
            filetype="SOLiD_native_qual"
            checksum_method="MD5"
            checksum="8900998ecf8427ed41d8cd98f00b204e">
            <READ_LABEL>R3</READ_LABEL>
            <DATA_SERIES_LABEL>INSDC:quality</DATA_SERIES_LABEL>
        </FILE>
    </FILES>
</DATA_BLOCK>
```

3.4. Specifying Qualities

Quality forms are particularly problematic as their formats are not well constrained. To better support this form of submission certain DATA_BLOCK parameters can be used to reduce the ambiguity of the input data.

The **DATA_BLOCK.FILES.FILE.quality_scoring_system** parameter can be used to specify whether the quality scores encountered in the fastq file are phred scale or log-odds scale. The SRA will convert log-odds into phred, but to do this properly the loader must know whether the log-odds scale is being used. For example:

```
<DATA_BLOCK name="KN-930" sector="1">
    <FILES>
        <FILE filename="KN-930_1.fastq"
            filetype="fastq"
            quality_scoring_system="log-odds"
            quality_encoding="ascii"
            ascii_offset="@">
            <DATA_SERIES_LABEL>INSDC:read</DATA_SERIES_LABEL>
            <DATA_SERIES_LABEL>INSDC:quality</DATA_SERIES_LABEL>
        </FILE>
    </FILES>
</DATA_BLOCK>
```

The **DATA_BLOCK.FILES.FILE.quality_encoding** parameter can tell whether the quality string in the fastq or native file is an ASCII character based string or an array of decimal values.

The **DATA_BLOCK.FILES.FILE.ascii_offset** tells which character is used as the basis (the zero) for the quality scores (choices are ascii 33 (!) or ascii 64(@)). Note that values can be negative. Negative values may be valid if the **DATA_BLOCK.FILES.FILE.quality_scoring_system** parameter is set to “log-odds”.

For example, consider the following sequencing data files :

```
gizmo2> sffinfo -s EAY20JP03.fna | head -n 2

>EAY20JP03GX7O6
GGGGGGGGGTAGGGGATGATGCCTTTGCAGTCAGTGCGGTGTCTGACAGCAACAGTGAGA
```

```
gizmo2> sffinfo -q EAY20JP03.qual | head -n 2

>EAY20JP03GX7O6
27 18 13 10 7 5 3 1 1 20 25 41 34 21 9 28 24 28 24 28 28 35 25 38 31 14 28 28 28 28
 27 28 28 27 28 28 28 35 26 28 27 28 28 28 28 27 28 25 27 28 35 25 28 28 28 25 28
 25 28 25
```

These can be represented with the following XML:

```
<DATA_BLOCK>
     <FILES>
         <FILE filename="EAY20JP03.fna"
             filetype="454_native_seq"
             checksum_method="MD5"
             checksum="d41d8cd98f00b204e9800998ecf8427e">
             <DATA_SERIES_LABEL>INSDC:read</DATA_SERIES_LABEL>
         </FILE>
         <FILE filename="EAY20JP03.qual"
             filetype="454_native_qual"
             checksum_method="MD5"
             checksum="9800998ecf8427ed41d8cd98f00b204e"
             quality_encoding="decimal">
             <DATA_SERIES_LABEL>INSDC:quality</DATA_SERIES_LABEL>
         </FILE>
     </FILES>
 </DATA_BLOCK>
```

Another example :

```
<DATA_BLOCK>
    <FILES>
        <FILE filename="s_7_sequence.fastq"
            filetype="fastq"
            checksum_method="MD5"
            checksum="d41d8cd98f00b204e9800998ecf8427e"
            quality_scoring_system="phred"
            quality_encoding="ascii"
            ascii_offset="!">
            <DATA_SERIES_LABEL>INSDC:read</DATA_SERIES_LABEL>
            <DATA_SERIES_LABEL>INSDC:quality</DATA_SERIES_LABEL>
        </FILE>
    </FILES>
</DATA_BLOCK>
```

Note that even with the offset of @, negative values (down to -5) may be generated by the decoding.

3.5. Using Filename and Checksum Attributes

New FILE attributes of checksum_method and checksum have been introduced in order to provide the loader a specification of what files to actually load. This separates concerns of verifying that a transmission of data arrived intact at NCBI, and the need to direct the loader's activities to individual components within that transmission. The combination of

filename and checksum are used to verify file identity and integrity in both cases. Consequently, the **RUN.DATA_BLOCK.FILES.FILE.filename** and **RUN.DATA_BLOCK.FILES.FILE.checksum** can, but do not have to be the same values entered into the **SUBMISSION.FILES.FILE.filename** and **SUBMISSION.FILES.FILE.checksum**.

4. Implications for Loader Design

The changes in this document imply changes to both the Toolkit and the submission pipeline.

a. SRA, SRF, SFF file types are determined by their magic number (file command). If this fails, they are regarded as "Text" files. Text files are either one of the "native" platform types, or generic "fastq".
b. Filenames are NOT relied upon in order to decide how to process the content of containers and of generic fastq. The XML must specify the necessary information. This protects against the tendency for instrument manufacturers to change the file names of their standard output files. Native file formats do have constrained filenames as defined by the manufacturer.
c. The final archival representation of single dimensional quality scores is always the phred scoring system. The DATA_BLOCK settings exist in order to tell the loader how to interpret the input data. Log-odds representation is converted to phred representation as part of the loading process. The original log-odds scores are NOT preserved in the SRA.
d. "Native" loaders use grammars specified in the SRA File Formats Guide. If a loader is asked to parse files according to a certain native file model (for example, "Illumina_native", it uses a limited set of grammars to determine the filetype of each input file. If an input file fails to match one of the grammars, the load should fail. The loader should indicate on which line of input the failure occurred.
e. The "Fastq" loader uses a set of grammars specified in the SRA File Formats Guide. If no grammar matches the input file, the load fails. The loader should indicate on which line of input the failure occurred.
f. Read names are specified as spot addresses according to the naming rule for each "native" grammar. For native loaders, read names must be unique in the file, be of the same order if found split between multiple files, and be within reasonable ranges determined by the vendor. Read names of runs successfully loaded with "native" loaders are indexed. -0 is rounded to 0.
g. Read names are not interpreted, indexed, or tested for uniqueness in files with filetype "fastq". Read names from "fastq" type input are NOT preserved in the SRA. Consequently, secondary analysis depending on fastq input cannot be processed by NCBI, because of the inability to link read names before accessioning.
h. The "Fastq" filetype does NOT support mate pairs, multiply segmented reads (each of which is in a different file), or data series that are found in different files.

i. If an input file cannot be interpreted according to a solution in **Error! Reference source not found.** then the input is rejected and the load fails. The user visible error message should indicate "Submitted filetype or format is not supported."
j. The loader should fail to load the run if any data block among several data blocks fails to load, and indicate which data block failed to load.
k. Loading may terminate with exception at the first instance of error without the need to determine further errors in that load attempt. The loader should indicate which file caused the exception.
l. The loader should be repeatable given the same input and the initial starting state. There should be a method in the loading pipeline to reinitialize a load once an exception has been thrown and loading has stopped.
m. A 454_native loader should be developed.
n. A Helicos_native loader should be developed.
   The loaders that limit the number of data blocks that can be loaded in one run need to be changed to accommodate the serial attribute that orders the load of data blocks.
p. A new column in the SRA corresponding to member_name should be added in order to store the member assignment from a user-demultiplexed bar code run.
q. The data series types defined in this document should be added to the Toolkit:
   - INSDC:read
   - INSDC:read_filter
   - INSDC:quality
   - INSDC:intensity
   - INSDC:signal
   - INSDC:noise
   - INSDC:position
   - INSDC:clip_quality_left
   - INSDC:clip_quality_right
   - INSDC:tab
   - INSDC:readname
   - INSDC:read_seg

SRA Barcoding Guide

Created: March 10, 2010; Updated: September 24, 2010.

1. Overview

This document reviews the features and submission requirements for SRA barcoded experiments and resulting sequencing data.

2. Use Cases

2.1. Default

The default SRA submission use case is for each experiment to have exactly one sample.

```
<EXPERIMENT>
  <DESIGN>
    <SAMPLE_DESCRIPTOR refname="Sample_1" refcenter="XYZ" />
```

where *Sample_1* is a SAMPLE record defined using the *SRA.sample.xsd* schema.

2.2. Sample Pool

In this use case one sequencing library is prepared from a pool of samples. The samples can be identified in the pool but the resulting sequencing data cannot distinguish the pool members except by secondary analysis such as alignment, which would occur outside of the SRA.

```
<EXPERIMENT>
  <DESIGN>
    <SAMPLE_DESCRIPTOR>
       <POOL>
          <MEMBER> member_name="BAC_1" accession="SRS000001"/>
          <MEMBER> member_name="BAC_2" accession="SRS000002"/>
           ...
```

Each member is defined by referencing its sample record. The member name has scope within the experiment and its child runs.

2.3. Sample Pool with Barcodes

A sample pool with barcodes is set up as follows:

1. Define the sample pool within the Sample Descriptor block. The SAMPLE_DESCRIPTOR attributes (accession or refname) can be used to define the default sample (the one that reads are assigned to if their barcode values cannot be decoded because of sequencing error or some other artifact).

```
<EXPERIMENT>
  <DESIGN>
    <SAMPLE_DESCRIPTOR refname="unassigned_bacs" refcenter="XYZ" >
```

```
<POOL>
   <MEMBER> member_name="BAC_1" accession="SRS000001">
      <READ_LABEL read_group_tag="ACTGTT">barcode_tag</READ_LABEL>
    </MEMBER>
   <MEMBER> member_name="BAC_2" accession="SRS000002">
      <READ_LABEL read_group_tag="TAGTTG">barcode_tag</READ_LABEL>
    </MEMBER>
    ...
```

2 Next, define the SPOT_DESCRIPTOR to include a barcode tag as one of the "technical reads". In this example, the barcode tag appears at the end of the read, and is decoded by substring matching.

```
<SPOT_DESCRIPTOR>
  <SPOT_DECODE_SPEC>
    <READ_SPEC>
      <READ_INDEX>0</READ_INDEX>
      <READ_CLASS>Application Read</READ_CLASS>
      <READ_TYPE>Forward</READ_TYPE>
      <BASE_COORD>1</BASE_COORD>
    </READ_SPEC>
    <READ_SPEC>
      <READ_INDEX>1</READ_INDEX>
      <READ_LABEL>barcode_tag</READ_LABEL>
      <READ_CLASS>Technical Read</READ_CLASS>
      <READ_TYPE>BarCode</READ_TYPE>
      <EXPECTED_BASECALL_TABLE
        ...
      />
  </SPOT_DECODE_SPEC>
</SPOT_DESCRIPTOR>
```

3 Next, define the lookup table that will associate a bar code pattern match with a member of the sample pool:

```
<EXPECTED_BASECALL_TABLE>
  <BASECALL read_group_tag="ACTGTT" min_match="6"
            max_mismatch="0" match_edge="full" >ACTGTT</BASECALL>
  <BASECALL read_group_tag="TAGTGG" min_match="6"
            max_mismatch="0" match_edge="full" >TAGTGG</BASECALL>
  ...
</EXPECTED_BASECALL_TABLE>
```

2.3.1. Sample pool with barcodes de-multiplexed by the submitter

The submitter takes care to split the reads within each run and reconstitute the submission container files in such a way that all the reads associated with a given member are contiguous and receive that member's reference.

1 The SRA Run must be configured so that the SRA Loader will associate individual reads with members of the sample pool. This is done in the DATA_BLOCK/ member attribute.

```
    <DATA_BLOCK
      name = "FMSX0OV"
      region = "1"
      member_name = "BAC_1"
    >
      <FILES>
        <FILE filename="BAC_1.sff"
filetype="sff"
checksum_method="MD5"
checksum="4026fc6b91ed2ffbef374a665e02802b" />
...
      </FILES>
    </DATA_BLOCK>
    <DATA_BLOCK
      name = "FMSX0OV"
      region = "1"
      member_name = "BAC_2"
    >
     <FILES>
        <FILE filename="BAC_2.sff"
filetype="sff"
checksum_method="MD5"
checksum="7f7ba170dbc6a25409a5eb6d845da88f" />
...
      </FILES>
    </DATA_BLOCK>
```

2. The EXPECTED_BASECALL_TABLE serves to document what was done in order to split up the run, but is not used to load the run. Barcode design is crucial to the success of the experiment and its processing, so users of the archive data may wish to repeat the de-multiplexing step. Therefore, include the EXPECTED_BASECALL_TABLE in the experiment's SPOT_DESCRIPTOR.
3. An additional file containing auxiliary tag location information necessary to the loading of the data may be required. This tab delimited text file contains information about how to locate the barcode(s) and other technical tags within each raw spot sequence.

Fields are:

INSDC:read_name: String value used to join with native read name in run data file. For example, EQYRFS112HPIGW

INSDC:read_seg: A vector of start-stop coordinates (basis 1, inclusive) that partitions a particular raw spot sequence among the tags defined for this run. The read_seg vector is expressed like this: [1-4],[5-12],[13-]. This tells the SRA loader that first tag has start coordinate 1 and end coordinate 4, and the tag starts at coordinate 13 and goes to the end

of the raw spot sequence. The expression [0] indicates that the tag is not present in the sequence.

Here are some example entries:

```
INSDC:read_name    INSDC:read_seg
EQYRFS112HPIGW    [1-4][5-12][13-]
EQYRFS112HPIHA    [1-4][0][5-]
EQYRFS112HPIMX    [1-4][0][5-]
```

The FILE block must contain an additional entry that specifies the submission of the auxiliary read segments file:

```
    <DATA_BLOCK
      name = "FMSX0OV"
      region = "1"
      member_name = "BAC_1"
    >
      <FILES>
        <FILE filename="BAC_1.sff"
filetype="sff"
checksum_method="MD5"
checksum="4026fc6b91ed2ffbef374a665e02802b"
</FILE>
    <FILE filename="BAC_1.readseg.tab"
filetype="tab"
checksum_method="MD5"
checksum="fc6b91ed2ffbef374a665e02802b4026"
<DATA_SERIES_LABEL>INSDC:read_seg</DATA_SERIES_LABEL>
        </FILE>
      </FILES>
    </DATA_BLOCK>
```

4. An addition file containing auxiliary clipping information necessary to the annotation of the data may be required for submitter-loads. This tab delimited text file contains information about how to locate the barcode(s) and other technical reads within each raw spot sequence.

Fields are:

INSDC:read_name: String value used to join with native read name in run data file. For example, EQYRFS112HPIGW

INSDC:clip_quality_left: A coordinate (basis 1, inclusive) indicating the start of good quality biological sequence.

INSDC:clip_quality_right: A coordinate (basis 1, inclusive) indicating the end of good quality biological sequence.

Here are some example entries:

```
INSDC:read_name    INSDC:clip_quality_left    INSDC:clip_quality_right
EQYRFS112HPIGW    13                        278
```

```
EQYRFS112HPIHA    13                          280
EQYRFS112HPIMX    5                          277
```

The FILE block must contain an additional entry that specifies the submission of the auxiliary clips file:

```
    <DATA_BLOCK
      name = "FMSX0OV"
      region = "1"
      member_name = "BAC_1"
    >
      <FILES>
        <FILE filename="BAC_1.sff"
filetype="sff"
checksum_method="MD5"
checksum="4026fc6b91ed2ffbef374a665e02802b"
</FILE>
    <FILE filename="BAC_1.clips.tab"
filetype="tab"
checksum_method="MD5"
checksum="fc6b91ed2ffbef374a665e02802b4026"
<DATA_SERIES_LABEL>INSDC:clip_quality_left</DATA_SERIES_LABEL>
<DATA_SERIES_LABEL>INSDC:clip_quality_right</DATA_SERIES_LABEL>

        </FILE>
      </FILES>
    </DATA_BLOCK>
```

5. Steps 3, 4 may be combined into one auxiliary data file.

2.3.2. Sample pool with barcodes de-multiplexed by the Archive

Submission is simpler in the case where the Archive is asked to perform the demultiplexing according to the instructions in the SPOT_DESCRIPTOR.

1 First, enter one file or set of files per data block. Do NOT use a member_name attribute:

```
    <DATA_BLOCK
      name = "FMSX0OV"
      region = "1"
    >
     <FILES>
        <FILE filename="all.bacs.sff"
filetype="sff"
checksum_method="MD5"
checksum="7f7ba170dbc6a25409a5eb6d845da88f" />
      </FILES>
    </DATA_BLOCK>
```

2 Create the EXPECTED_BASECALL_TABLE to tell the loader how to recognize each barcode tag and assign individual reads to a sample pool member.

2.4. Study Pool

The pooling of studies and experiments in a single run is not currently supported. The sequencing center is expected to de-multiplex the data for each study and return the appropriate subset to each investigator. From there it will be possible to make wholly distinct submissions. The SRA_LINK can be used to identify the relationship of several SRA runs to a single production run.

3. Features

3.1. Multi-dimensional Barcodes

More than one dimension of barcoding can be used with each tuple decoded to yield a member.

```
<EXPERIMENT>
  <DESIGN>
    <SAMPLE_DESCRIPTOR refname="unassigned_bacs" refcenter="XYZ" >
       <POOL>
          <MEMBER> member_name="site1_fraction1" accession="SRS000001">
             <READ_LABEL read_group_tag="site1">barcode_tag_a</READ_LABEL>
             <READ_LABEL read_group_tag="fraction1">barcode_tag_b</READ_LABEL>
           </MEMBER>
          <MEMBER> member_name="site1_fraction2" accession="SRS000002">
             <READ_LABEL read_group_tag="site1">barcode_tag_a</READ_LABEL>
             <READ_LABEL read_group_tag="fraction2">barcode_tag_b</READ_LABEL>
           </MEMBER>
          <MEMBER> member_name="site2_fraction1" accession="SRS000003">
             <READ_LABEL read_group_tag="site2">barcode_tag_a</READ_LABEL>
             <READ_LABEL read_group_tag="fraction1">barcode_tag_b</READ_LABEL>
           </MEMBER>
          <MEMBER> member_name="site2_fraction1" accession="SRS000004">
             <READ_LABEL read_group_tag="site2">barcode_tag_a</READ_LABEL>
             <READ_LABEL read_group_tag="fraction2">barcode_tag_b</READ_LABEL>
           </MEMBER>


           ...

      <SPOT_DESCRIPTOR>
        <SPOT_DECODE_SPEC>
          <READ_SPEC>
            <READ_INDEX>0</READ_INDEX>
            <READ_CLASS>Application Read</READ_CLASS>
            <READ_TYPE>Forward</READ_TYPE>
            <BASE_COORD>1</BASE_COORD>
          </READ_SPEC>
          <READ_SPEC>
            <READ_INDEX>1</READ_INDEX>
            <READ_LABEL>barcode_tag_a</READ_LABEL>
            <READ_CLASS>Technical Read</READ_CLASS>
```

```
            <READ_TYPE>BarCode</READ_TYPE>
            <EXPECTED_BASECALL_TABLE
              ...
            />
          <READ_SPEC>
            <READ_INDEX>2</READ_INDEX>
            <READ_LABEL>barcode_tag_b</READ_LABEL>
            <READ_CLASS>Technical Read</READ_CLASS>
            <READ_TYPE>BarCode</READ_TYPE>
            <EXPECTED_BASECALL_TABLE
              ...
            />

        </SPOT_DECODE_SPEC>
      </SPOT_DESCRIPTOR>
```

with the EXPECTED_BASECALL_TABLE for *barcode_tag_a* and for *barcode_tag_b* set up in the SPOT_DESCRIPTOR.

A toy example,

```
          <READ_SPEC>
            <READ_INDEX>1</READ_INDEX>
            <READ_LABEL>barcode_tag_a</READ_LABEL>
            <READ_CLASS>Technical Read</READ_CLASS>
            <READ_TYPE>BarCode</READ_TYPE>
            <EXPECTED_BASECALL_TABLE
<BASECALL read_group_tag="site1" min_match="4" max_mismatch="0"
match_edge="full" >GCAT</BASECALL>
<BASECALL read_group_tag="site2" min_match="4" max_mismatch="0"
match_edge="full" >CTTG</BASECALL>
            />
          <READ_SPEC>
            <READ_INDEX>2</READ_INDEX>
            <READ_LABEL>barcode_tag_b</READ_LABEL>
            <READ_CLASS>Technical Read</READ_CLASS>
            <READ_TYPE>BarCode</READ_TYPE>
            <EXPECTED_BASECALL_TABLE
<BASECALL read_group_tag="fraction1" min_match="4" max_mismatch="0"
match_edge="full" >AACA</BASECALL>
<BASECALL read_group_tag="fraction2" min_match="4" max_mismatch="0"
match_edge="full" >GTAT</BASECALL>
            />
```

3.2. Overloading Barcodes

More than one barcode can be used to resolve a particular member:

```
            <EXPECTED_BASECALL_TABLE>
              <BASECALL read_group_tag="site_1" min_match="6"
                        max_mismatch="0" match_edge="full" >ACTGTT</BASECALL>
              <BASECALL read_group_tag="site_1" min_match="6"
                        max_mismatch="0" match_edge="full" >TAGTGG</BASECALL>
```

```
    ...
</EXPECTED_BASECALL_TABLE>
```

3.3. Primer-Barcode Pairs

Using the same mechanism of subsequence matching, a combination of primer and barcode tags can be defined for each spot. These definitions are made in the SPOT_DESCRIPTOR block. Then, a table of tag value pairs can be defined that maps each combination of primer and barcode to a particular sample pool member.

3.4. Overlapping Pools

Two sample pools may contain the same sample(s). Each sample pool will have one library constructed for it. A distinct SRA Experiment should be defined for each pool.

4. Data Preparation

4.1. 454 Barcode Libraries

You should download the *sfftools* toolkit from Roche Diagnostics Corporation (license required). This will allow you to work with SFF files to dump their contents and the partition and recombine the files.

4.1.1. Submitter Demultiplexing

Using the utility *sfffinfo*, obtain the list of reads and their sequences from each plate's worth of run data.

```
sffinfo -s EQYRFS1.sff > EQYRFS1.fasta
```

Use a substring alignment program to match against a set of barcode sequences. One such program that works well for short, nearly exact matches is the MUMmer package (www.mummer.sourceforge.net). For example, these commands work well to find instances of barcode exact matches in the sequencing data.

```
nucmer -maxmatch -g 0 -c 12 -l 12 EQYRFS1reads.fasta barcodes.fasta -p EQYRFS1-barcodes
 show-coords -cTH EQYRFS1-barcodes > EQYRFS1-barcodes.coords
```

The latter file can be used to generate the set of individual read-barcode hits as well as the auxiliary readseg tab file.

```
grep barcode01 EQYRFS1-barcodes.coords | awk '{ print $(NF-1) }'
 > barcode01.seqs
```

taking care to identify 100% matches, eliminate duplicate hits, and choose between multiple barcode hits.

Finally, each of the hit files can be applied to the master SFF file to extract the subset of records associated with a certain bar code.

```
sfffile -i barcode01.seqs -o EQYRFS1-barcode01 EQYRFS1.sff
...
```

The command *sfffile -t <filename>* can also be used to reset the quality_clip_left parameter. If only one mapping tag (barcode/primer etc) is being used, then this is sufficient for the loader to recognize the barcode tag boundaries and the auxiliary readseg tab file is not needed.

Using the SRA Identifier Block

Created: August 3, 2012; Updated: September 18, 2012.

Overview

The purpose of the SRA Identifier block is to capture in one place all keys that are used as IDs. An ID can identify exactly one record within a context. A record may have multiple IDs. A record's ID must be unique within a context, and all objects in a context must have an ID. These properties do not hold for "names" or other monikers.

The SRA Identifier block contains the following identifiers:

- PRIMARY_ID – Primary key to an INSDC database.
- SECONDARY_ID – Secondary key or defunct primary key to an INSDC database
- EXTERNAL_ID – Identifier from another database qualified by a namespace.
- SUBMITTER_ID – Local identifier qualified by a namespace.
- UUID – Universally unique identifier which requires no namespace.

The identifiers block contains a set of identifiers. The identifiers may occur in any order or combination so long as exactly one PRIMARY_ID is present.

Goals

- Consolidate use of identifiers for each SRA document
- Distinguish between accessions and named IDs
- Add support for UUIDs
- Improve flexibility for submitter identification of records

Features

- Tracks archived assigned ids
- Tracks submitter assigned ids
- Supports tracking of object mergers and replacements via secondary ids and aliases
- Tracks alternate or external ids assigned by different databases or archives
- Support for 3rd party assigned ids including catalog ids
- Explicit support for UUIDs

Design

The IdentifiersType is defined in the SRA.common.xsd schema, please look in the following location(s):

- http://www.ncbi.nlm.nih.gov/viewvc/v1/trunk/sra/doc/SRA_1-4/SRA.common.xsd?view=co

Here is the relevant type code from SRA.common.xsd:

```
    <xsd:complexType name="IdentifierNodeType">
        <xsd:simpleContent>
            <xsd:extension base="xs:string">
                <xsd:attribute name="label" use="optional"
type="xs:string"/>
            </xsd:extension>
        </xsd:simpleContent>
    </xsd:complexType>

    <xsd:complexType name="AccessionType">
        <xsd:simpleContent>
            <xsd:extension base="com:IdentifierNodeType"/>
        </xsd:simpleContent>
    </xsd:complexType>

    <xsd:complexType name="NameType">
        <xsd:simpleContent>
            <xsd:extension base="com:IdentifierNodeType">
                <xsd:attribute name="namespace" use="required"
type="xs:string"/>
            </xsd:extension>
        </xsd:simpleContent>
    </xsd:complexType>

    <xsd:complexType name="UUIDType">
        <xsd:simpleContent>
            <xsd:extension base="com:IdentifierNodeType"/>
        </xsd:simpleContent>
    </xsd:complexType>

    <xsd:complexType name="IdentifierType">
        <xsd:sequence>
            <xsd:element name="PRIMARY_ID" type="com:AccessionType"
minOccurs="0" maxOccurs="1"/>
            <xsd:element name="SECONDARY_ID" type="com:AccessionType"
minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="EXTERNAL_ID" type="com:NameType"
minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="SUBMITTER_ID" type="com:NameType"
minOccurs="0" maxOccurs="unbounded"/>
            <xsd:element name="UUID" type="com:UUIDType" minOccurs="0"
maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
```

Data Types

The IdentifierNodeType abstract type extends xs:string with the following attributes:

- **label** – whether and how to display a tag string.

Four concrete types subclass IdentiferType in order to suggest the business use of the identifier:

AccessionType – A key in an INSDC primary database.

NameType –A key in an external database.

The following attributes are required:

- **namespace** – Namespace (database) of the external name

UUIDType – A key that is universally unique and requires no namespace.

Data Structure

PRIMARY_ID – A primary identifier, or key, in the INSDC primary database (accession). Example: SRR330090. Exactly one primary identifier is required in every IDENTIFIER block. This value is equivalent to the document/@accession attribute.

SECONDARY_ID – A foreign key in the INSDC primary database (accession), or a defunct primary key in the INSDC primary database. Example: SRR330091. Any number of secondary identifiers may be present.

EXTERNAL_ID – A key in an external database qualified by the name of the database. Example: Coriell NA12878. Any number of external names may be present.

SUBMITTER_ID – A key that resolves within the submitter's namespace. Exactly one local name must be present on submission. Local names are not needed for data download or exchange between archives. This value is equivalent to the (document/@alias, document/@center_name) attribute tuple.

UUID – A key that is universally unique and needs no namespace. UUIDs are not used by the Archive but rather are provided as part of the SRA xml schema to serve downstream applications, including non-INSDC SRA mirrors.

Compatibility

The existing NameGroup and RefNameGroup attribute groups currently used to identify records will continue in use.

Semantics

The IdentifierType is implemented by each SRA archive with additional business rules governing use of namespaces and scope of identifiers.

Replacement tracking

The IdentifierType can be used to name record(s) replaced (taken over) by the current record. The transitive closure of these replacing relations is a set of currently active

records with replaced descendants. The converse relation (replaced by) can be computed from this forest so it is not tracked explicitly.

Persistence

One goal of the IDENTIFIERS block is to document data migration, replacement, and equivalency relationships independently of the life cycle of the record, so that Archive users who form dependencies on a certain SRA record can always recover the relationship to other records even if the record has been suppressed.

Use Cases

Data Migration

The IDENTIFIERS block can be used to manage the transition of metadata from one record to another and provide a trackback mechanism to recover previous incarnations. This would include:

- Tracking a record in the archive (or prior to archiving) with a submitter supplied identifier.
- Tracking a record's identifier before and after a data migration.
- Tracking a record's identifier before and after a data consolidation.
- Tracking a changes in an identifier used for a dependency

Data Replacement

The IDENTIFIERS block can be used to indicate that the content has been replaced, and identify the previous record that represented the content. A run may have been mis-loaded due to errors in the original load process or a misrepresentation of the metadata that caused the data to be interpreted differently. If the result of the mis-load is an SRA archive image that is substantially different then the run's accession will be replaced. Another example is where duplicate runs have been discovered, and each run can be mapped to its duplicates although only one of them is retained in the archive.

Data Equivalency

The IDENTIFIERS block can be used to point to records that are equivalent and can be used interchangeably. An example is the BioProject and SRA study identifiers, which for a time will both be active identifiers of a study record (until migration from SRA study to BioProject is completed). Another example is where equivalent records have been discovered in multiple SRA instances. This would happen when a submitter has sent the same submission to both NCBI and EBI, for example. Over time, the INSDC may elect to retain one instance and suppress the other one, but the ID block can be used to maintain the equivalence relation.

Examples

SRA document identifiers

The document can contain IDENTIFIERS block in co-existence with existing NameGroup attribute group :

```
<RUN xmlnamespace="" run_center="BI" run_date="2011-08-04T04:00:00Z"
instrument_name="SL-HAC">
    <IDENTIFIERS>
      <PRIMARY_ID>SRR354028</PRIMARY_ID>
      <SUBMITTER_ID  namespace="BI" >BI.PE.110804_SL-
HAC_0370_BFCB02H8ACXX.6.UNMATCHED.srf</SUBMITTER_ID>
    </IDENTIFIERS>
```

**IDENTIFIER values must agree with NameGroup values

The document can contain IDENTIFIERS block in lieu of existing NameGroup attribute group:

```
<RUN>
    <IDENTIFIERS>
      <SUBMITTER_ID  namespace="BI" >BI.PE.110804_SL-
HAC_0370_BFCB02H8ACXX.6.UNMATCHED.srf</SUBMITTER_ID>
      <PRIMARY_ID>SRR354028</PRIMARY_ID>
    </IDENTIFIERS>
```

This gives a migration path for adoption of Identifier block in place of the name group attributes group, or a method for reverse construction of the NameGroup attributes from the ID block.

SRA document references

Document dependency references to other documents can be encoded with or without the RefNameGroup attributes.

SRA Study / BioProject / dbGaP study Reference

```
SRA Study:
Refname group example (same as previous schema versions):
  Reference by local names:
    <STUDY_REF refcenter="BI" refname="Ceratotherium_simum_simum_WGS"/>

    Is equivalent to:
    </STUDY_REF>
            <SUBMITTER_ID namespace="BI" >Ceratotherium_simum_simum_WGS</SUBMITTER_ID>
    <STUDY_REF>
  Reference by accession
<STUDY_REF accession ="SRPxxxxxx"/>

   Is equivalent to:
    <STUDY_REF>
          <IDENTIFIERS>
```

```
            <PRIMARY_ID>SRPxxxxxx</PRIMARY_ID>
          </IDENTIFIERS>
    </STUDY_REF>
BioProject:
    <STUDY_REF refcenter="BioProject" refname="PRJNA74583"/>
   Is equivalent to:
    <STUDY_REF>
          <IDENTIFIERS>
            <EXTERNAL_ID namespace="BioProject">PRJNA74583</SUBMITTER_ID>
          </IDENTIFIERS>
    </STUDY_REF>

dbGaP:
    <STUDY_REF refcenter="dbgap" refname="phsxxxxxx"/>
  Is equivalent to:
    <STUDY_REF>
          <IDENTIFIERS>
            <EXTERNAL_ID namespace="dbgap">phsxxxxxx</SUBMITTER_ID>
          </IDENTIFIERS>
    </STUDY_REF>
```

Sample Reference

Sample reference using the identifiers block should contain only 1 identifier: either a primary_id, an external_ id or a submitter_id

```
experiment-to-sample and experiment-to-BioSample reference
Referencing SRA samples by accession
        <SAMPLE_DESCRIPTOR>
          <IDENTIFIERS>
            <PRIMARY_ID>SRSxxxxxx</PRIMARY_ID>
          </IDENTIFIERS>
        </SAMPLE_DESCRIPTOR>
Referencing SRA samples by submitter_id/alias
        <SAMPLE_DESCRIPTOR>
          <IDENTIFIERS>
            <SUBMITTER_ID namespace="JGI">10908</SUBMITTER_ID>
          </IDENTIFIERS>
        </SAMPLE_DESCRIPTOR>
Referencing BioSamples by accession
        <SAMPLE_DESCRIPTOR>
          <IDENTIFIERS>
            <EXTERNAL_ID namespace="BioSample">SAMNxxxxxx</EXTERNAL_ID>
          </IDENTIFIERS>
        </SAMPLE_DESCRIPTOR>
Referencing dbgap samples
        <SAMPLE_DESCRIPTOR>
          <IDENTIFIERS>
            <EXTERNAL_ID namespace="phsxxxxxx">submitted_sample_id</EXTERNAL_ID>
          </IDENTIFIERS>
        </SAMPLE_DESCRIPTOR>
```

Replaced Record

The information that a certain record has been replaced is not indicated in the IDENTIFIERS block, but is tracked in the SRA database and livelist.

```
<RUN run_date="2008-11-24T23:08:44Z" instrument_name="GA-5">
    <IDENTIFIERS>
      <PRIMARY_ID>SRR292241</PRIMARY_ID>
    </IDENTIFIERS>
```

Replacer Record

This example shows how a record, SRR390728, replaces a predecessor SRR292241:

```
<RUN run_date="2008-11-24T23:08:44Z" instrument_name="GA-5">
    <IDENTIFIERS>
      <PRIMARY_ID>SRR390728</PRIMARY_ID>
      <SECONDARY_ID>SRR292241</SECONDARY_ID>
    </IDENTIFIERS>
```

Elected Record

This example shows how one record, SRR351940, has replaced 9 others (elected as successor), as in the case where several 'readgroup' runs provisionally reference the same bam file, one is selected for cSRA loading and the remaining runs are suppressed.

```
<RUN>
    <IDENTIFIERS>
      <PRIMARY_ID>SRR351940</PRIMARY_ID>
      <SECONDARY_ID>SRR351941</SECONDARY_ID>
      <SECONDARY_ID>SRR351942</SECONDARY_ID>
      <SECONDARY_ID>SRR351943</SECONDARY_ID>
      <SECONDARY_ID>SRR351944</SECONDARY_ID>
      <SECONDARY_ID>SRR351945</SECONDARY_ID>
      <SECONDARY_ID>SRR351946</SECONDARY_ID>
      <SECONDARY_ID>SRR351947</SECONDARY_ID>
      <SECONDARY_ID>SRR351948</SECONDARY_ID>
      <SECONDARY_ID>SRR351949</SECONDARY_ID>
   </IDENTIFIERS>
```

Successor Record

This example shows how one record, SRR351940, has replaced another kind of record, analysis object SRZ019522.

```
<RUN>
    <IDENTIFIERS>
      <PRIMARY_ID>SRR351940</PRIMARY_ID>
      <SECONDARY_ID>SRZ019522</SECONDARY_ID>
   </IDENTIFIERS>
```

Submitter alternate identifiers

Submitted records can retain their alternate identifiers and these can be treated as identifiers rather than attributes of the record. The label attribute calls out the display field.

```
<RUN center_name="BI" alias="70291ABXX110301.7.tagged_393.bam"
run_center="BI" run_date="2011-03-01T05:00:00Z" instrument_name="SL-
HBZ" accession="SRR404010">
  <IDENTIFIERS>
    <PRIMARY_ID>SRR404010</PRIMARY_ID>
    <SUBMITTER_ID namespace="BI">70291ABXX110301.7.tagged_393.bam</
SUBMITTER_ID>
    <SUBMITTER_ID namespace="BI" label="read group platform unit"
>70291ABXX110301.7.CCAGTTAG</SUBMITTER_ID>
  </IDENTIFIERS>
```

Submitter replaced identifiers

Submitters can replace an identifier with a new one without disturbing the linkage to existing SRA accessions. However, the primary identifier must be supplied and the defunct identifier must be removed by an update submission.

```
existing…
<RUN alias="454_O.mykiss_GD3412001" accession="SRR090454" center_name="INRA">
<IDENTIFIERS>
    <PRIMARY_ID>SRR090454</PRIMARY_ID>
    <SUBMITTER_ID namespace="INRA">454_O.mykiss_GD3412001</SUBMITTER_ID>
  </IDENTIFIERS>
…

updated…

<RUN alias="454_O.mykiss_GD3412001" accession="SRR090454" center_name="INRA">
<IDENTIFIERS>
    <PRIMARY_ID>SRR090454</PRIMARY_ID>
    <SUBMITTER_ID namespace="INRA">454_O.mykiss_GD3412001</SUBMITTER_ID>
    <SUBMITTER_ID namespace="INRA">454_O.mykiss_GB5RBPX02 </SUBMITTER_ID>
  </IDENTIFIERS>
…
```

Commonly used external identifiers

In lieu of a local identifier, a submitter can use a supported external identifier. A good example is a cell line DNA isolate sample from one of the Coriell NA12878:

```
<SAMPLE>
<IDENTIFIERS>
    <PRIMARY_ID > SRS000090</PRIMARY_ID>
    <EXTERNAL_ID namespace="Coriell" label="Catalog ID">NA12878 </EXTERNAL_ID>
    <EXTERNAL_ID namespace="Coriell" label="Catalog ID">GM12878 </EXTERNAL_ID>
  </IDENTIFIERS>
…
```

Universally unique identifiers

A downstream user of SRA xml data could annotate it with a universally unique identifier. This requires no namespace because it is universally unique (according to the generation method). The INSDC SRAs do not use UUIDs and these are ignored on submission.

```
<RUN alias="68b329da9893e34099c7d8ad5cb9c940" accession="SRR090454" center_name="">
<IDENTIFIERS>
    <PRIMARY_ID>SRR090454</PRIMARY_ID>
    <UUID> 68b329da9893e34099c7d8ad5cb9c940 </UUID>
  </IDENTIFIERS>
…
```

Submission Considerations

The PRIMARY_ID will appear in each record obtained from the SRA, but clearly it is unknown at the time of submission. Consequently, for submitted records the PRIMARY_ID should be omitted:

```
<EXPERIMENT>
  <IDENTIFIERS>
    <SUBMITTER_ID namespace="BI">658005.WR28289.HS_PF_Sen_42.C0DJUACXX120207.P</SUBMITTER_ID>
  </IDENTIFIERS>
```

For modify submissions, the submitter can either use the assigned PRIMARY_ID or reuse the original SUBMITTER_ID if it successfully identifies the record to modify.

NCBI will continue to support submissions without IDENTIFIER blocks. The existing NameGroup and RefNameGroup attribute groups will be processed and identifiers will be derived from these attributes.

Exchange Considerations

For mirror archives, NCBI will supply SRA records with the IDENTIFIER block and the NameGroup and RefNameGroup attributes. The attributes will be derived from the appropriate content in the IDENTIFIER block.

Mirror archives may choose to continue relying on the NameGroup and RefNameGroup attributes and not support the IDENTIFIER block. In this case the IDENTIFIER block will be written as follows:

PRIMARY_ID := @accession

SECONDARY_ID @namespace=@center_name := @alias

or for individual submitters that do not have a center_name,

SECONDARY_ID @namespace=null := @alias

The IDENTIFIER block can be dropped from the mirror.