Tutorial: SRA Tool kits. By: Bablu Kumar

Installation: SRA Toolkit using conda

1. **Create a conda environment for SRA -Tool kit**

   Command: ***conda create -n sra***

   This command creates a virtual environment named "sra" using Conda, a package manager. It allows isolated Python environments for managing dependencies and projects.

2. **Installation of SRA Toolkit using Conda:** ***conda install -c bioconda sra-tools***

   This command uses conda to install the SRA Toolkit package from the "bioconda" channel. Bioconda is a conda channel that specializes in providing bioinformatics-related packages. The package you're installing is named "sra-tools," which is the official toolkit for working with Sequence Read Archive (SRA) data provided by NCBI.

Let's break down the command:

- *`**conda install**`: This is the command to install packages using Conda.*
- *`**-c bioconda**`: This flag specifies the Conda channel from which to install the package, in this case, the "bioconda" channel.*
- *`**sra-tools**`: This is the name of the package you want to install.*

Download SRA sequences

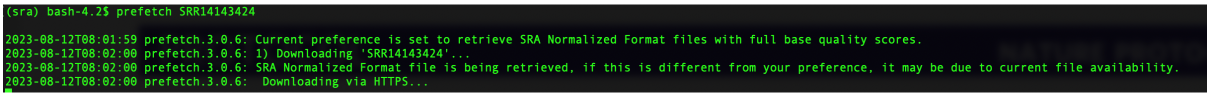
Activate environment: command: "conda activate sra"

This command activates a specific conda environment named "***sra***". When you activate an environment, you're essentially switching to that environment and any subsequent installations or commands will be applied within it.

Method 1: Using *prefetch* and *fasterq-dump*

Step 1: Using prefetch: *prefetch* is a part of the SRA Toolkit that allows you to download sequence data (Runs) from the Sequence Read Archive (SRA) and any additional data needed to convert the downloaded data from SRA format to a more commonly used format (like FASTQ or SAM).

To download a single Run: `***prefetch SRR14143424*** `

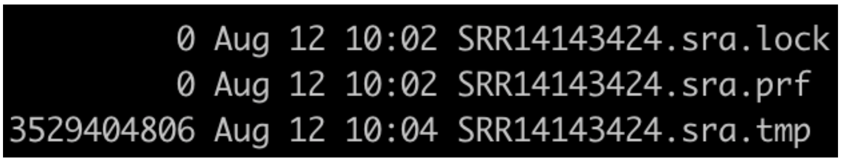


```
(sra) bash-4.2$ prefetch SRR14143424

2023-08-12T08:01:59 prefetch.3.0.6: Current preference is set to retrieve SRA Normalized Format files with full base quality scores.
2023-08-12T08:02:00 prefetch.3.0.6: 1) Downloading 'SRR14143424'...
2023-08-12T08:02:00 prefetch.3.0.6: SRA Normalized Format file is being retrieved, if this is different from your preference, it may be due to current file availability.
2023-08-12T08:02:00 prefetch.3.0.6:  Downloading via HTTPS...
```

Here, ` *SRR14143424* ` is the accession number of the specific Run you want to download.

Temp files: During downloading.

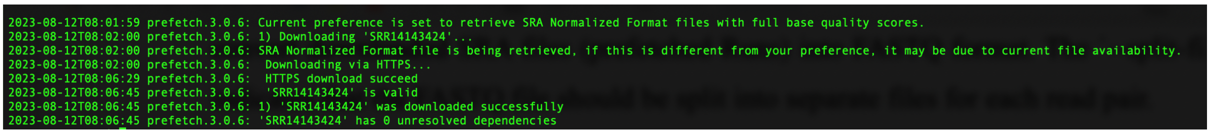


```
         0 Aug 12 10:02 SRR14143424.sra.lock
         0 Aug 12 10:02 SRR14143424.sra.prf
3529404806 Aug 12 10:04 SRR14143424.sra.tmp
```

Note:

1. \`SRR14143424.sra.lock\`: This file might be a lock file used to prevent simultaneous access or modifications to the associated SRA data file by multiple processes.
2. \`SRR14143424.sra.prf\`: This file's purpose is uncertain without more context, but it could potentially contain some metadata or profiling information related to the SRA data.
3. \`SRR14143424.sra.tmp\`: This file appears to be a temporary file, possibly generated during the processing or downloading of the SRA data.

Successfully downloaded file: ***SRR14143424.sra***



```
2023-08-12T08:01:59 prefetch.3.0.6: Current preference is set to retrieve SRA Normalized Format files with full base quality scores.
2023-08-12T08:02:00 prefetch.3.0.6: 1) Downloading 'SRR14143424'...
2023-08-12T08:02:00 prefetch.3.0.6: SRA Normalized Format file is being retrieved, if this is different from your preference, it may be due to current file availability.
2023-08-12T08:02:00 prefetch.3.0.6:  Downloading via HTTPS...
2023-08-12T08:06:29 prefetch.3.0.6:  HTTPS download succeed
2023-08-12T08:06:45 prefetch.3.0.6:  'SRR14143424' is valid
2023-08-12T08:06:45 prefetch.3.0.6: 1) 'SRR14143424' was downloaded successfully
2023-08-12T08:06:45 prefetch.3.0.6: 'SRR14143424' has 0 unresolved dependencies
```

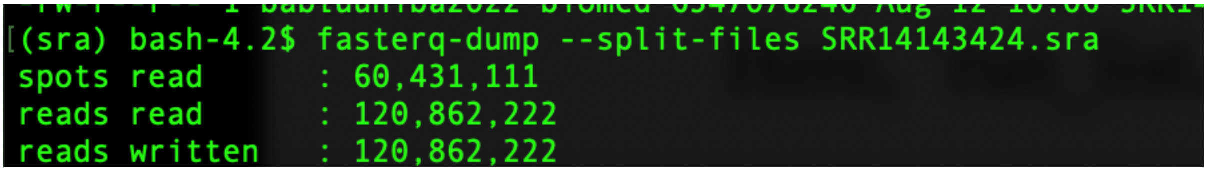
For every accession ID, a corresponding folder is created containing a specific ".sra" file. In this instance, execute the command "**cd SRR14143424**" to access the "**SRR14143424**" folder. Within this folder, there is a ".sra" file named "***SRR14143424.sra***".

Step 2: fasterq-dump:

\`*fasterq-dump*\` is another command from the SRA Toolkit. It's used to convert downloaded SRA files (prefetched Runs) into FASTQ format. The **\`*--split-files*\`** flag indicates that the FASTQ file should be split into separate files for each read pair.

Example command:

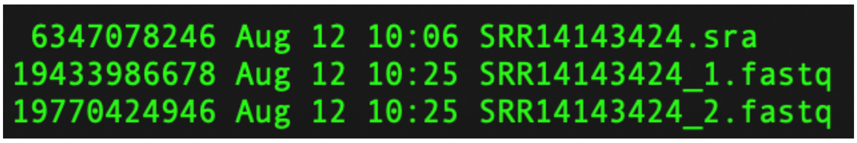
```
***fasterq-dump --split-files SRR14143424.sra***
```



```
[(sra) bash-4.2$ fasterq-dump --split-files SRR14143424.sra
 spots read      : 60,431,111
 reads read      : 120,862,222
 reads written   : 120,862,222
```

Note: These statistics pertain to sequencing data processing: "spots read": This refers to the total number of individual sequences or "spots" that were read from the sequencing run. In this case, there were 60,431,111 such spots. "reads read": This indicates the total count of sequencing reads that were obtained. A read might encompass more than one spot if the sequencing technology produces paired end reads. Here, the count is 120,862,222 reads. "reads written": This signifies the total count of reads that were written or saved after processing. This count is *identical to the "reads read" count, denoting that all the reads read were processed and retained. These statistics provide insights into the quantity of data generated, read, and processed during a sequencing experiment.*

In this example, ` *SRR14143424.sra*` is the downloaded SRA file you want to convert to FASTQ format (SRR14143424_1.fastq, SRR14143424_2.fastq).



```
 6347078246 Aug 12 10:06 SRR14143424.sra
19433986678 Aug 12 10:25 SRR14143424_1.fastq
19770424946 Aug 12 10:25 SRR14143424_2.fastq
```

List of run accession ID

To download a list of Runs from a file: `***prefetch --option-file run_list.txt***`

Here, `run_list.txt` is a text file containing a list of Run accession numbers. `--option-file` specifies that you're providing a file with accession numbers.

Method 2: Using a single fasterq-dump command

You can combine the *prefetch* and conversion steps into one by directly using the Run accession in the conversion command. You don't need to specify the `.sra` extension.

Example:

```
fasterq-dump --split-files SRR14143424
```

In this case, ` SRR14143424` is the Run accession. The command will automatically download the Run and convert it to FASTQ format.

Summary

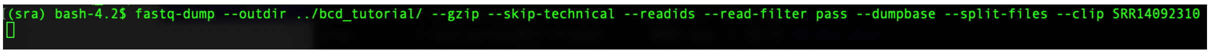
In summary, the SRA Toolkit commands provided allow you to download public sequence data from the SRA, either by first using `prefetch` to download and then converting using `fasterq-dump`, or by directly using the Run accession in the conversion command to combine the download and conversion steps.

fastq_dump

fastq-dump is a tool for downloading sequencing reads from SRA. These sequence reads will be downloaded as FASTQ files.

Command:

```
fastq-dump --outdir ../bcd_tutorial/ --gzip --skip-technical --readids --read-filter pass --dumpbase --split-files --clip SRR14092310
```



```
(sra) bash-4.2$ fastq-dump --outdir ../bcd_tutorial/ --gzip --skip-technical --readids --read-filter pass --dumpbase --split-files --clip SRR14092310
```

Let's go through each option in the provided `fastq-dump` command:

- `--outdir ../bcd_tutorial/`: Specifies the output directory for the resulting FASTQ files. In this case, the output will be saved in the `***bcd_tutorial***` directory located one level up from the current directory (`../` indicates moving one directory level up).
- `--gzip`: This option compresses the output FASTQ files using the gzip compression algorithm. Compressed files have the `.gz` extension.
- `--skip-technical`: Excludes technical reads from the conversion process. Technical reads might contain control or calibration data and are typically not relevant for downstream analysis.
- `--readids`: This option includes read IDs in the output FASTQ files. Read IDs provide information about the sequence's origin and can be important for tracing back to the original data.
- `--read-filter pass`: Filters the reads to include only those marked as "pass". This means that only reads that have passed quality control are included in the output. "Pass" indicates that the reads meet certain quality criteria.
- `--dumpbase`: Instead of providing quality scores, this option outputs the base calls themselves. This is useful when you only need the raw sequence information.
- `--split-files`: Splits paired-end data into separate files for each read. If your data is paired-end (with two reads for each fragment), this option will generate separate files for each read, allowing easier downstream analysis.
- `--clip`: This option might be used to clip adapter sequences or low-quality regions from the reads. Clipping involves trimming parts of the sequence that are of lower quality or are known to be non-biological, such as adapters used in the sequencing process.

- `SRR14092310`: The SRA accession number of the dataset you want to convert to FASTQ format.

Downloaded files

For each accession ID, a dedicated folder is generated, containing specific *"*_pass_1.fastq.gz*" and *"*_pass_2.fastq.gz*" files for paired-end data. However, for single-end data, only a single "pass_1.fastq.gz" file is present. For instance, if you run the command "cd **SRR14092310**," you'll navigate to the "**SRR14092310**" folder. Inside this folder, you'll find "*SRR14092310_pass_1.fastq.gz*" and "*SRR14092310_pass_2.fastq.gz*" files if it's paired-end data.

Summary

In summary, the provided `fastq-dump` command takes the SRA dataset with accession number `SRR14092310`, applies various options to control the conversion process, and generates two FASTQ files (one for each read in paired-end data) in the specified output directory, compressing them using gzip. It ensures that only high-quality, non-technical reads are included, and it might perform base call dumping and clipping if necessary.

Differences between `fastq-dump` and `fasterq-dump`

1. Performance

⇒ `fastq-dump`: This is the standard tool for converting SRA format data to FASTQ format. However, it operates sequentially, which can lead to slower conversion times, especially for large datasets.

⇒ `fasterq-dump`: As the name suggests, this tool is designed for improved performance. It harnesses multi-threading and parallel processing capabilities, leading to significantly faster data conversion, particularly for large datasets.

2. Functionality

⇒ `fastq-dump`: This tool offers various options to customize the conversion process, such as including read IDs, filtering reads, and more. It is versatile and useful when specific customization is required.

⇒ `fasterq-dump`: While `fasterq-dump` prioritizes speed, it sacrifices some customization options available in `fastq-dump`. It focuses on efficiently converting data with a streamlined set of options.

3. Use Cases

⇒ Use `fastq-dump` when you require more control over the conversion process and need to customize the output according to your needs.

⇒ Use `fasterq-dump` when speed is a critical factor, especially when dealing with large datasets. It's ideal for quick conversions without intricate customization.

Resources for reading

1. https://www.ncbi.nlm.nih.gov/sra/docs/sradownload/
2. https://github.com/ncbi/sra-tools/wiki/08.-prefetch-and-fasterq-dump
3. https://rnnh.github.io/bioinfo-notebook/docs/fastq-dump.html