Uncovering SRA-TOOLS, Prefetch, and Fasterq-dump: Mini Conda Environment

First of all, what are SRA-TOOLS?

Welcome to the field of genomics, where researchers navigate the immense ocean of biological data. The **National Center for Biotechnology Information (NCBI)** created ***SRA-TOOLS***, a potent toolbox for managing **Sequence Read Archive (SRA)** data. But, you wonder, what does that actually mean? Put another way, it's a toolkit that enables you to access, fetch raw datasets, and work with unprocessed genetic data. Consider it your key to discovering the mysteries concealed in the DNA of different species.

Recognizing Prefetch: Your Entryway to Unprocessed Data

Now that you understand what SRA-TOOLS are, let's talk about the star of the show: ***the prefetch command*.** Prefetch is like a magical spell that fetches raw genomic datasets from NCBI's servers and brings them to your local environment. Why does this matter? Genomics data can be enormous, particularly when derived from Next-Generation Sequencing technology. Moving the data closer to your analytical tools saves time, bandwidth, and resources. Instead of having to travel far every time you want to read a book, picture yourself having a library right in your house.

How to Use Prefetch: An Interactive Guide

Let's walk through a simple prefetch scenario:

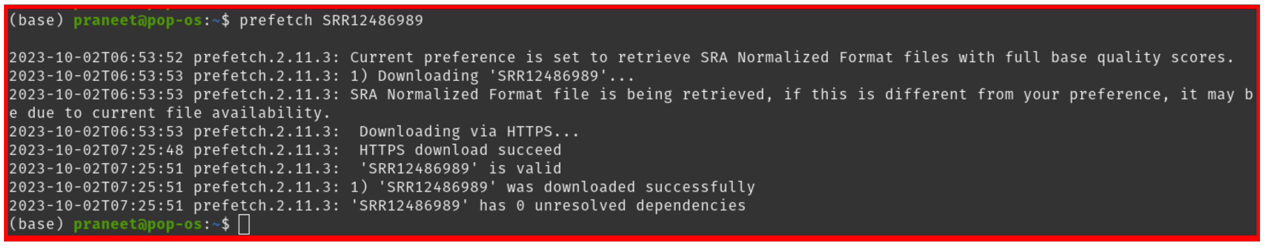
Before this remember that datasets with SRR (Sequence Read Run), ERR (Experiment Run Reference), and DRR (DRA Run Reference) can be fetched. These accession numbers are provided by the **NCBI** for their **SRA datasets**. When you use prefetch with one of these accession numbers, it fetches the associated raw data files and stores them locally for further analysis.

Fetching Your Raw Dataset:

Now comes the exciting part – fetching your dataset.

Let's say you're interested in a dataset with the accession number SRR12486989. In your terminal, type:

Command: prefetch SRR12486989



```
(base) praneet@pop-os:~$ prefetch SRR12486989

2023-10-02T06:53:52 prefetch.2.11.3: Current preference is set to retrieve SRA Normalized Format files with full base quality scores.
2023-10-02T06:53:53 prefetch.2.11.3: 1) Downloading 'SRR12486989'...
2023-10-02T06:53:53 prefetch.2.11.3: SRA Normalized Format file is being retrieved, if this is different from your preference, it may b
e due to current file availability.
2023-10-02T06:53:53 prefetch.2.11.3:  Downloading via HTTPS...
2023-10-02T07:25:48 prefetch.2.11.3:  HTTPS download succeed
2023-10-02T07:25:51 prefetch.2.11.3:  'SRR12486989' is valid
2023-10-02T07:25:51 prefetch.2.11.3: 1) 'SRR12486989' was downloaded successfully
2023-10-02T07:25:51 prefetch.2.11.3: 'SRR12486989' has 0 unresolved dependencies
(base) praneet@pop-os:~$ 
```

Watch as Prefetch goes to work, downloading the raw data and storing it on your local machine.

Let’s understand about temporary files:

Temporary files:

When you use the prefetch command from the SRA-TOOLS toolkit to fetch raw datasets, several temporary files are created. These files aid in managing the download and ensuring data integrity. Here's what each of these temporary files (*.prf, *.tmp, and *.lock) does during the prefetch operation:

1. **Profile files, or .prf files:**

During the ***prefetch*** phase, profile files—which are identified by the ***.prf extension*** —are produced. These files include metadata and details about the downloaded dataset. They support SRA-TOOLS in managing the details of the dataset.

Usage: Local dataset metadata is stored in ***.prf files.*** This metadata contains the file size, accession number, and other pertinent information. Without saving the real sequence data, it only offers a snapshot of the dataset.

Importance: Profile files are essential for verifying the integrity of the downloaded dataset. They serve as a reference point for the prefetch operation and are used during subsequent operations to ensure that the downloaded data matches the expected metadata

2. **Temporary files, or .tmp files:**

The ***.tmp extension*** denotes temporary files that are created during the download process. Temporary or partial data chunks are stored in these files while they are being retrieved from the distant server.

Use: Data chunks are temporarily stored in ***.tmp files*** while they are being fetched. Large datasets might take some time to download, thus these temporary files make sure that any partial data is safely kept before downloading the whole dataset.

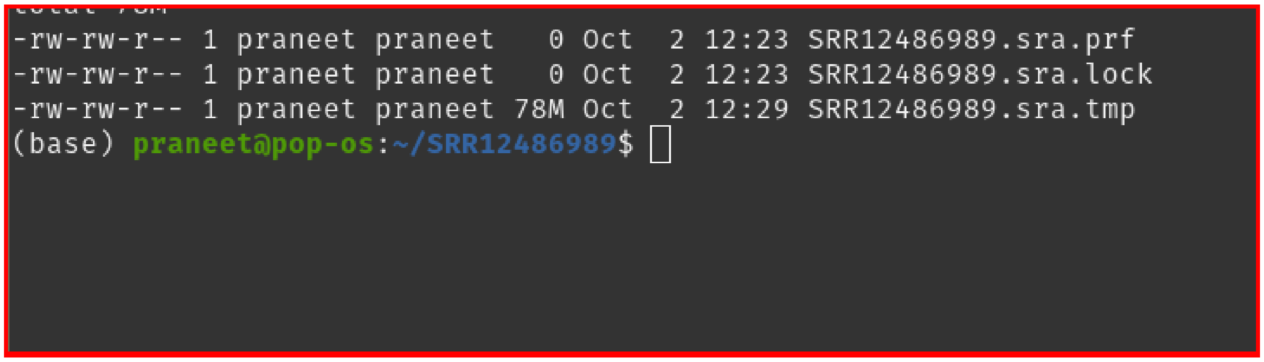
Importance: In the event that the download process is abruptly stopped, temporary files are essential for preventing data loss. The toolkit can restart from these temporary files rather than beginning the download anew in the event that the download is interrupted for any reason.

3. **Lock files, or .lock files, are:**

The ***.lock extension*** is used to identify files that are locked, and their purpose is to stop numerous instances of the same command from interfering with one another. ***Prefetch*** generates a lock file while it's operating to show that a download for a particular dataset is underway.

Usage: Lock files inform other processes that a download for a certain dataset is presently underway. By doing this, competing activities are avoided and only one prefetch command instance downloads the dataset at a time.

Importance: When many instances of the prefetch command try to download the same dataset at once, lock files can help preserve data consistency and avoid possible conflicts. They guarantee that the process of downloading



```
-rw-rw-r-- 1 praneet praneet   0 Oct  2 12:23 SRR12486989.sra.prf
-rw-rw-r-- 1 praneet praneet   0 Oct  2 12:23 SRR12486989.sra.lock
-rw-rw-r-- 1 praneet praneet 78M Oct  2 12:29 SRR12486989.sra.tmp
(base) praneet@pop-os:~/SRR12486989$ 
```

Now that you have understood the temporary files and how to fetch raw datasets using

Prefetch let's quickly understand what is the next process.

Fasterq-dump:

Let's dissect the ***fasterq-dump*** command and become familiar with all of its parts, including the --threads and --progress parameters.

Command: fasterq-dump --threads 2 --progress SRR12486989

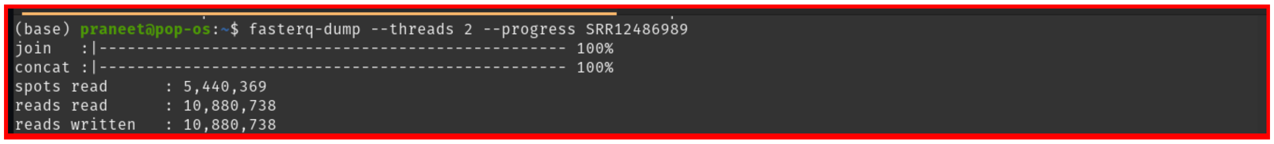
fasterq-dump: This is a command-line tool provided by the **SRA-TOOLS toolkit.** It is used for quickly extracting FASTQ files (which contain sequence data and quality scores) from SRA-formatted datasets.t. It's used to convert downloaded SRA files (prefetched Runs) into FASTQ format. The **`*--split-files*`** flag indicates that the FASTQ file should be split into separate files for each read pair.

--threads 2: This option specifies the number of threads or parallel processes that the fasterq-dump command can use. Modern computers often have multiple processor cores. By using multiple threads, the command can split the workload across these cores, significantly speeding up the data extraction process. In this case, ***--threads 2*** means that the command will use 2 threads for the extraction process.

--progress: When present, this optional flag shows a progress indicator as the extraction operation moves forward. With huge datasets, this can be very helpful since it provides a visual representation of the extent of data extraction that has been finished.

The precise accession number of the dataset you wish to extract is SRR12486989.

Fasterq-dump uses this accession number to determine exactly which dataset to download and convert to FASTQ format when you run the program.



```
(base) praneet@pop-os:~$ fasterq-dump --threads 2 --progress SRR12486989
join   :|-------------------------------------------------- 100%
concat :|-------------------------------------------------- 100%
spots read      : 5,440,369
reads read      : 10,880,738
reads written   : 10,880,738
```

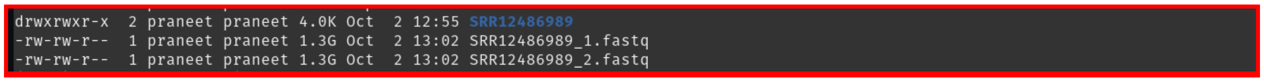
Note: *The following statistics relate to the processing of sequencing data:* ***"spots read":*** *This is the total number of discrete sequences (also known as "spots") that were read during the sequencing run. There were 5,440,369 of these places in this instance****. "reads read":*** *The total number of sequencing reads that were obtained is shown by this. If paired-end reads are produced by the sequencing method, a read may span multiple spots. There are*

10,880,738 readings in this instance. ***"reads written":*** *The total number of reads that were written or saved following processing is indicated by this. This tally is similar to the "reads read" count, indicating that each and every read was handled and saved. These statistics shed light on how much data is produced, accessed, and processed throughout a sequencing experiment.*

In conclusion, *fasterq-dump --threads 2 --progress SRR12486989* effectively extracts the given dataset, leveraging multithreading for quicker processing and offering you a visual progress indication to help you streamline and better manage your bioinformatics workflow.

Recall that prefetch was created especially to retrieve information from the **Sequence Read Archive at the National Cancer Institute.** It could be necessary to employ several tools and techniques for obtaining and processing datasets if you have data from diverse sources or in different formats.

Also remember that one must use both prefetch and faster-q dump, as prefetch only fetches raw dataset, faster-q dump allows to split the files into _1.fastq and _2.fastq which are paired reads ends.



```
drwxrwxr-x  2 praneet praneet 4.0K Oct  2 12:55 SRR12486989
-rw-rw-r--  1 praneet praneet 1.3G Oct  2 13:02 SRR12486989_1.fastq
-rw-rw-r--  1 praneet praneet 1.3G Oct  2 13:02 SRR12486989_2.fastq
```

Conclusion:

Empowering Your Research Journey

In summary, ***SRA-TOOLS and the prefetch*** command are your companions in the vast landscape of genomic exploration. By bringing raw datasets to your fingertips, you empower your research, enabling in-depth analyses and discoveries that can change the way we understand life itself. So, go ahead, fetch those datasets, and embark on a scientific adventure that might just redefine our understanding of the biological world.

Happy researching! 🧪🔬🧬