**SRA( Sequence Read Archive):** The **Sequence Read Archive** (**SRA**, previously known as the **Short Read Archive**) is a bioinformatics database that provides a public repository for DNA sequencing data, especially the "short reads" generated by high-throughput sequencing, which are typically less than 1,000 base pairs in length.

### [Frequently Used Tools:](https://trace.ncbi.nlm.nih.gov/Traces/sra/sra.cgi?view=toolkit_doc#header-global)

[fastq-dump](https://trace.ncbi.nlm.nih.gov/Traces/sra/sra.cgi?view=toolkit_doc&f=fastq-dump): Convert SRA data into fastq format

fastq-dump --split-files accession\_number.sra (convert the prefetched Runs from compressed SRA format to fastq or sam format.)

--split files: Dump each read into separate file. Files will receive suffix corresponding to read number

fastq-dump –split-files accession\_number (To avoid the prefetch step and download and convert the Run in one step)

[prefetch](https://trace.ncbi.nlm.nih.gov/Traces/sra/sra.cgi?view=toolkit_doc&f=prefetch): This program downloads Runs (sequence files in the compressed SRA format) and all the additional data necessary to convert the Run from the SRA format to a more commonly used format.

prefetch accession number (downloads the data stored as the given accession number.)

[sam-dump](https://trace.ncbi.nlm.nih.gov/Traces/sra/sra.cgi?view=toolkit_doc&f=sam-dump): Convert SRA data to sam format

[sra-pileup](https://trace.ncbi.nlm.nih.gov/Traces/sra/sra.cgi?view=toolkit_doc&f=sra-pileup): Generate pileup statistics on aligned SRA data

[vdb-config](https://trace.ncbi.nlm.nih.gov/Traces/sra/sra.cgi?view=toolkit_doc&f=vdb-config): Display and modify VDB configuration information

[vdb-decrypt](https://trace.ncbi.nlm.nih.gov/Traces/sra/sra.cgi?view=toolkit_doc&f=vdb-decrypt): Decrypt non-SRA dbGaP data ("phenotype data")

### [Additional Tools:](https://trace.ncbi.nlm.nih.gov/Traces/sra/sra.cgi?view=toolkit_doc#header-global)

[abi-dump](https://trace.ncbi.nlm.nih.gov/Traces/sra/sra.cgi?view=toolkit_doc&f=abi-dump): Convert SRA data into ABI format (csfasta / qual)

[illumina-dump](https://trace.ncbi.nlm.nih.gov/Traces/sra/sra.cgi?view=toolkit_doc&f=illumina-dump): Convert SRA data into Illumina native formats (qseq, etc.)

[sff-dump](https://trace.ncbi.nlm.nih.gov/Traces/sra/sra.cgi?view=toolkit_doc&f=sff-dump): Convert SRA data to sff format

[sra-stat](https://trace.ncbi.nlm.nih.gov/Traces/sra/sra.cgi?view=toolkit_doc&f=sra-stat): Generate statistics about SRA data (quality distribution, etc.)

[vdb-dump](https://trace.ncbi.nlm.nih.gov/Traces/sra/sra.cgi?view=toolkit_doc&f=vdb-dump): Output the native VDB format of SRA data.

[vdb-encrypt](https://trace.ncbi.nlm.nih.gov/Traces/sra/sra.cgi?view=toolkit_doc&f=vdb-encrypt): Encrypt non-SRA dbGaP data ("phenotype data")

[vdb-validate](https://trace.ncbi.nlm.nih.gov/Traces/sra/sra.cgi?view=toolkit_doc&f=vdb-validate): Validate the integrity of downloaded SRA data

**NOTE: ./research\_proxy.sh & (**To run the research proxy in the background)

**Burrows-Wheeler Aligner:** BWA is a software package for mapping low-divergent sequences against a large reference genome, such as the human genome. It consists of three algorithms: BWA-backtrack, BWA-SW and BWA-MEM.  BWA-MEM, which is the latest, is generally recommended for high-quality queries as it is faster and more accurate.

For 70bp or longer Illumina, 454, Ion Torrent and Sanger reads, assembly contigs and BAC sequences, BWA-MEM is usually the preferred algorithm.

BWA outputs alignment in the new standard SAM (Sequence Alignment/Map) format.

**COMMANDS AND OPTIONS:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **index** | bwa index [-p prefix] [-a algoType] <in.db.fasta>  Index database sequences in the FASTA format.  **OPTIONS:**   |  | | --- | |  | |  | | **-p***STR* | Prefix of the output database [same as db filename] | | **-a***STR* | Algorithm for constructing BWT index. Available options are:   |  | | --- | |  | |  | | **is** | IS linear-time algorithm for constructing suffix array. It requires 5.37N memory where N is the size of the database. IS is moderately fast, but does not work with database larger than 2GB. IS is the default algorithm due to its simplicity. The current codes for IS algorithm are reimplemented by Yuta Mori. | | **bwtsw** | Algorithm implemented in BWT-SW. This method works with the whole human genome. | | |

|  |  |
| --- | --- |
| **mem** | **bwa mem** [**-aCHMpP**] [**-t** *nThreads*] [**-k** *minSeedLen*] [**-w** *bandWidth*] [**-d** *zDropoff*] [**-r** *seedSplitRatio*] [**-c** *maxOcc*] [**-A** *matchScore*] [**-B** *mmPenalty*] [**-O** *gapOpenPen*] [**-E** *gapExtPen*] [**-L** *clipPen*] [**-U** *unpairPen*] [**-R** *RGline*] [**-v** *verboseLevel*] *db.prefix* *reads.fq* [*mates.fq*] |

Briefly, the algorithm works by seeding alignments with maximal exact matches (MEMs) and then extending seeds with the affine-gap Smith-Waterman algorithm (SW).

|  |  |
| --- | --- |
| **-t***INT* | Number of threads |
| **-k INT** | Minimum seed length. Matches shorter than INT will be missed. The alignment speed is usually insensitive to this value unless it significantly deviates 20. |
| **-w INT** | Band width. Essentially, gaps longer than INT will not be found. Note that the maximum gap length is also affected by the scoring matrix and the hit length, not solely determined by this option. |
|  |  |

bwa index command gives five files output that are listed below:

.ann file: text file, to record reference sequence, name, length etc.

.pac file: packed reference sequence (binary). 00-> A, 01-> C, 10-> G, 11-> T.

.sa file: Binary, suffix array index (half of total genome size ~ 1.5G).

.amb file: text file to record appearance of N in reference fasta.

.fai file: fasta index file.

.bwt file: binary file.

**BWA-MEM2:** The index size on the disk is down by 8 times (.fai file) and in memory by 4 times due to moving to only one type of FM-index (compressed full-text substring index, 2bit.64) and 8x compression of suffix array (.sa file).

**Indexing of Reference sequence:**

1. git clone <https://github.com/bwa-mem2/bwa-mem2.git>
2. cd bwa-mem2
3. make CXX=icpc (using intel C/C++ compiler)  
   or make (using gcc compiler)
4. bwa-mem2 index <ref.fa>
5. bwa-mem2 mem [-t <#threads>] <ref.fa> <in\_1.fastq> [<in\_2.fastq>] > <output.sam>

bwa-mem2 index ~/mitochondria\_assembly/raw/index/GrCh38.primary\_assembly.genome.fa (This will generate the index files for the reference genome. We will be getting six file formats as described above.)

**SAMTOOLS:** Samtools is a set of utilities that manipulate alignments in the BAM format. It imports from and exports to the SAM format, does sorting, merging and indexing, and allows to retrieve reads in any regions.

It regards an input file as the standard input (stdin) and an output file as the standard output (stdout). Several commands can thus be combined with unix pipes.

Commands: samtools view : With no options or regions specified, prints all alignments in the specified input alignment file (SAM,BAM or CRAM) to standard output in SAM format.

Options->

**-b** Output in the BAM format.

**-C**

Output in the CRAM format (requires -T).

**-1**

Enable fast BAM compression (implies -b).

**-u**

Output uncompressed BAM. This option saves time spent on compression/decompression and is thus preferred when the output is piped to another samtools command.

**-h**

Include the header in the output.

**-H**

Output the header only.

**-c**

Instead of printing the alignments, only count them and print the total number. All filter options, such as **-f**, **-F**, and **-q**, are taken into account.

**-?**

Output long help and exit immediately.

**-o***FILE*

Output to *FILE [stdout].*

**-U***FILE*

Write alignments that are *not* selected by the various filter options to *FILE*. When this option is used, all alignments (or all alignments intersecting the *regions* specified) are written to either the output file or this file, but never both.

**-t***FILE*

A tab-delimited *FILE*. Each line must contain the reference name in the first column and the length of the reference in the second column, with one line for each distinct reference. Any additional fields beyond the second column are ignored. This file also defines the order of the reference sequences in sorting. If you run: `samtools faidx <ref.fa>', the resulting index file *<ref.fa>.fai* can be used as this *FILE*.

**-T***FILE*

A FASTA format reference *FILE*, optionally compressed by **bgzip** and ideally indexed by **samtools** **faidx**. If an index is not present, one will be generated for you.

**-L***FILE*

Only output alignments overlapping the input BED *FILE* [null].

**-M**

Use the multi-region iterator on the union of a BED file and command-line region arguments. This avoids re-reading the same regions of files so can sometimes be much faster. Note this also removes duplicate sequences. Without this a sequence that overlaps multiple regions specified on the command line will be reported multiple times. The usage of a BED file is optional and its path has to be preceded by **-L** option.

**-r***STR*

Output alignments in read group *STR* [null]. Note that records with no **RG** tag will also be output when using this option. This behaviour may change in a future release.

**-R***FILE*

Output alignments in read groups listed in *FILE* [null]. Note that records with no **RG** tag will also be output when using this option. This behaviour may change in a future release.

**-d***STR:STR*

Only output alignments with tag *STR* and associated value *STR* [null].

**-D***STR:FILE*

Only output alignments with tag *STR* and associated values listed in *FILE* [null].

**-q***INT*

Skip alignments with MAPQ smaller than *INT* [0].

**-l***STR*

Only output alignments in library *STR* [null].

**-m***INT*

Only output alignments with number of CIGAR bases consuming query sequence ≥ *INT* [0]

**-f***INT*

Only output alignments with all bits set in *INT* present in the FLAG field. *INT* can be specified in hex by beginning with `0x' (i.e. /^0x[0-9A-F]+/) or in octal by beginning with `0' (i.e. /^0[0-7]+/) [0].

**-F***INT*

Do not output alignments with any bits set in *INT* present in the FLAG field. *INT* can be specified in hex by beginning with `0x' (i.e. /^0x[0-9A-F]+/) or in octal by beginning with `0' (i.e. /^0[0-7]+/) [0].

**-G***INT*

Do not output alignments with all bits set in *INT* present in the FLAG field. This is the opposite of *-f* such that *-f12 -G12* is the same as no filtering at all. *INT* can be specified in hex by beginning with `0x' (i.e. /^0x[0-9A-F]+/) or in octal by beginning with `0' (i.e. /^0[0-7]+/) [0].

**-x***STR*

Read tag to exclude from output (repeatable) [null]

**-B**

Collapse the backward CIGAR operation.

**-s***FLOAT*

Output only a proportion of the input alignments. This subsampling acts in the same way on all of the alignment records in the same template or read pair, so it never keeps a read but not its mate.

The integer and fractional parts of the **-s***INT***.***FRAC* option are used separately: the part after the decimal point sets the fraction of templates/pairs to be kept, while the integer part is used as a seed that influences *which* subset of reads is kept.

When subsampling data that has previously been subsampled, be sure to use a different seed value from those used previously; otherwise more reads will be retained than expected.

**-@***INT*

Number of BAM compression threads to use in addition to main thread [0].

**-S**

Ignored for compatibility with previous samtools versions. Previously this option was required if input was in SAM format, but now the correct format is automatically detected by examining the first few characters of input.

**-X**

Include customized index file as a part of arugments. See **EXAMPLES** section for sample of useage.

**--no-PG**

Do not add a @PG line to the header of the output file.

samtools sort : This command sort alignments by leftmost coordinates, or by read name when -n is used. The sorted output is written to standard output by default, or to the specified file when -o is used.

Options:

**-l***INT*

Set the desired compression level for the final output file, ranging from 0 (uncompressed) or 1 (fastest but minimal compression) to 9 (best compression but slowest to write), similarly to **gzip**(1)'s compression level setting.

If **-l** is not used, the default compression level will apply.

**-m***INT*

Approximately the maximum required memory per thread, specified either in bytes or with a **K**, **M**, or **G** suffix. [768 MiB]

To prevent sort from creating a huge number of temporary files, it enforces a minimum value of 1M for this setting.

**-n**

Sort by read names (i.e., the **QNAME** field) rather than by chromosomal coordinates.

**-t***TAG*

Sort first by the value in the alignment tag TAG, then by position or name (if also using **-n**).

**-o***FILE*

Write the final sorted output to *FILE*, rather than to standard output.

**-O***FORMAT*

Write the final output as **sam**, **bam**, or **cram**.

By default, samtools tries to select a format based on the **-o** filename extension; if output is to standard output or no format can be deduced, **bam** is selected.

**-T***PREFIX*

Write temporary files to *PREFIX***.***nnnn***.bam,** or if the specified *PREFIX* is an existing directory, to *PREFIX***/samtools.***mmm***.***mmm***.tmp.***nnnn***.bam,** where *mmm* is unique to this invocation of the **sort** command.

By default, any temporary files are written alongside the output file, as *out.bam***.tmp.***nnnn***.bam,** or if output is to standard output, in the current directory as **samtools.***mmm***.***mmm***.tmp.***nnnn***.bam.**

**-@***INT*

Set number of sorting and compression threads. By default, operation is single-threaded.

**--no-PG**

Do not add a @PG line to the header of the output file.

samtools index: This command index a coordinate-sorted BGZIP-compressed SAM, VAM or CRAM file for fast random access. For SAM this only works if the file has been BGZF compressed first. If output filename is given, the index file will be written to out.index. Otherwise, for a CRAM file aln.cram, index file aln.cram**.crai** will be created; for a BAM file aln.bam, either aln.bam**.bai** or aln.bam**.csi** will be created; and for a compressed SAM file aln.sam.gz, either aln.sam.gz**.bai** or aln.sam.gz**.csi** will be created, depending on the index format selected.

Options:

**-b**

Create a BAI index. This is currently the default when no format options are used.

**-c**

Create a CSI index. By default, the minimum interval size for the index is 2^14, which is the same as the fixed value used by the BAI format.

**-m***INT*

Create a CSI index, with a minimum interval size of 2^INT.

**-@, --threads***INT*

Number of input/output compression threads to use in addition to main thread [0].

samtools depth: Computes the depth at each position or region.

Options:

**-a**

**-a**

Output all positions (including those with zero depth)

**-a -a, -aa**

Output absolutely all positions, including unused reference sequences. Note that when used in conjunction with a BED file the -a option may sometimes operate as if -aa was specified if the reference sequence has coverage outside of the region specified in the BED file.

**-b***FILE*

Compute depth at list of positions or regions in specified BED *FILE.* []

**-f***FILE*

Use the BAM files specified in the *FILE* (a file of filenames, one file per line) []

**-H**

Write a comment line showing column names at the beginning of the output. The names are CHROM, POS, and then the input file name for each depth column. If one of the inputs came from stdin, the name “-” will be used for the corresponding column.

**-l***INT*

Ignore reads shorter than *INT*

**-m, -d***INT*

At a position, read at most *INT* reads per input file. This means figures greater than *INT* may be reported in the output.

Setting this limit reduces the amount of memory and time needed to process regions with very high coverage. Passing zero for this option sets it to the highest possible value, effectively removing the depth limit. [8000]

Note that up to release 1.8, samtools would enforce a minimum value for this option. This no longer happens and the limit is set exactly as specified.

**-o***FILE*

Write output to *FILE*. Using “-” for *FILE* will send the output to stdout (also the default if this option is not used).

**-q***INT*

Only count reads with base quality greater than *INT*

**-Q***INT*

Only count reads with mapping quality greater than *INT*

**-r***CHR***:***FROM***-***TO*

Only report depth in specified region.

**-X**

If this option is set, it will allows user to specify customized index file location(s) if the data folder does not contain any index file. Example usage: samtools depth [options] -X /data\_folder/in1.bam [/data\_folder/in2.bam [...]] /index\_folder/index1.bai [/index\_folder/index2.bai [...]]

**-g***FLAGS*

By default, reads that have any of the flags UNMAP, SECONDARY, QCFAIL, or DUP set are skipped. To include these reads back in the analysis, use this option together with the desired flag or flag combination. *FLAGS* can be specified in hex by beginning with `0x' (i.e. /^0x[0-9A-F]+/), in octal by beginning with `0' (i.e. /^0[0-7]+/), as a decimal number not beginning with '0' or as a comma-separated list of flag names. [0]

For a list of flag names see *samtools-flags*(1).

**-G***FLAGS*

Discard any read that has any of the flags specified by *FLAGS* set. FLAGS are specified as for the **-g** option. [UNMAP,SECONDARY,QCFAIL,DUP]

samtools faidx: This command index reference sequence in the FASTA format or extract subsequence from indexed reference sequence. If no region is specified, faidx will index the file and create <ref.fasta>.fai on the disk. If regions are specified, the subsequences will be retrieved and printed to stdout in the fasta format.

The input file can be compressed in the **BGZF** format.

Options:

-o, --output: FILE Write FASTA to file rather than to stdout.

-n, --length: Length of FASTA sequence line.

-f, --fastq: Read FASTQ files and output extracted sequences in FASTQ format.

samtools fqidx: This command index reference sequence in the FASTQ format or extract subsequence from indexed reference sequence. If no region is specified, fqidx will index the file and create <ref.fastq>.fai on the disk.

The input file can be compressed in the BGZF format. This command should only be used on fastq files with a small number of entries.

Options:

**-o, --output***FILE*

Write FASTQ to file rather than to stdout.

**-n, --length***INT*

Length of FASTQ sequence line. [60]

**-c, --continue**

Continue working if a non-existent region is requested.

**-r , --region-file***FILE*

Read regions from a file. Format is chr:from-to, one per line .

**-i, --reverse-complement**

Output the sequence as the reverse complement. When this option is used, “/rc” will be appended to the sequence names. To turn this off or change the string appended, use the **--mark-strand** option.

**--mark-strand TYPE**

Append strand indicator to sequence name. TYPE can be one of:

**rc**

Append '/rc' when writing the reverse complement. This is the default.

**no**

Do not append anything.

**sign**

Append '(+)' for forward strand or '(-)' for reverse complement. This matches the output of “bedtools getfasta -s”.

**custom,<pos>,<neg>**

Append string <pos> to names when writing the forward strand and <neg> when writing the reverse strand. Spaces are preserved, so it is possible to move the indicator into the comment part of the description line by including a leading space in the strings <pos> and <neg>.

**-h, --help**

Print help message and exit.