Bioinformatic Tools Catalog

This is a guide for some of the Bioinformatic tools available in Cluster Kabré. We aim to help the user identifying the main purposes of each tool so we provide a brief description of the functionality, an example of the command line and a reference for the manual. We strongly suggest to search all the tools available, using the command module avail, once you have enter in the cluster

COPYING FILES INTO CLUSTER KABRÉ

If you want to upload and store files into your user inside the cluster, you could use the bash syntax **scp**. This command allows files to be copied securely between different hosts.

1. If your files are located in your local host (your computer) and you want to copy them into a specific directory in your cluster user. This command must be executed in your local host

```
scp file.txt your_username@cluster.cenat.ac.cr:/some/remote/directory
```

2. If you want to move files between a different remote host to Kabré:

```
$ scp your_username@rh1.edu:/some/remote/directory/foobar.txt
\your_username@cluster.cenat.ac.cr:/some/remote/directory/
```

You could copy multiple files at the same time by adding all the file names in the previous commands separated by a space. Also if you want to copy multiple files with some pattern in their names you could use * to represent any sequence of characters. For example, if i have the files file_1.txt, file_2.txt and file_n.txt, the command could be modify to copy all my **.txt** files at the same time:

```
scp *.txt your_username@cluster.cenat.ac.cr:/some/remote/directory
```

ADN / PROTEIN SEQUENCE PROCESSING

SRA-TOOLS: This tool stores raw sequences from next generation technologies into the cluster storage using the accession numbers from databases of the International Nucleotide Sequence Database Collaboration (INSDC)

- Module: `sra-tools/git`
- Example for storing fastq files: `fastq-dump -Z SSR000 >> genome_1.fast`

  -Z = joins all the output split data into a single stream
  -SSR000 = genome accession number for EBI database
  >> genome_1.fastq = file generated with the genome.
- Manual: https://ncbi.github.io/sra-tools/fastq-dump.html

EMBOSS: This tool is a comprehensive sequence analisis, composed of 150 different programs with many functions. For the purpose of this catalog, we present some of their basic data processing commands, however we advise to explore all the different options it offers.

- Manual: http://manuals.bioinformatics.ucr.edu/home/emboss

1. Seqret: Reads a sequence or a set of sequences and writes them out displaying sequences, reformatting sequences, producing the reverse complement of a sequence, extracting fragments of a sequence, sequence case conversion, or a combination of these tasks.

   - Module: `emboss/6.6.0` `emboss/gcc`
   - Example for changing a sequence format :`>seqret -sequence file0_1 -outseq file0_1_sq.fastq -sformat1 abi -osformat2 fastq-sanger`

     -sequence= input sequence name
     -outseq= output sequence name
     -sformat1= source format
     -osformat= output sequence format
   - Example to write the reverse-complement of a sequence :`>seqret -srev -sequence nnmt.fasta -outseq nnmtR.fasta`

-srev= performs the reverse complement of a sequence
-sequence= input sequence name
-outseq= output sequence name

- Manual: http://emboss.sourceforge.net/apps/release/6.2/emboss/apps/seqret.html

2. Transeq: It generates a translation from a nucleotide sequence

- Module: `emboss/6.6.0`
  `emboss/gcc`
- Example: `transeq -sequence file.fna -outseq file_translate`

-sequence = Input DNA sequence in fasta format
-outseq = name of the translated file

- Manual: http://emboss.sourceforge.net/apps/release/6.2/emboss/apps/transeq.html

TRIMMOMATIC: It is used to trim and crop FASTQ data, and to remove adapters from Illumina sequencing for both pair end and single end sequencing

- Module: `trimmomatic/0.36`
- Example (single end): `java -jar trimmomatic-0.36.jar SE -phred33 input.fq.gz output.fq.gz ILLUMINACLIP:TruSeq3-SE:2:30:10 LEADING:3 TRAILING:3 SLIDINGWINDOW:4:15 MINLEN:36`

-SE = single end
-phred33 = refers to the phred score used
-input.fq.gz = input file in FASTQ compressed format
-output.fq.gz = outout file in FASTQ compressed format

- Example (paird end): `java -jar trimmomatic-0.30.jar PE 1_sequence.txt.gz 2_sequence.txt.gz lane1_forward_paired.fq.gz lane1_forward_unpaired.fq.gz lane1_reverse_paired.fq.gz lane1_reverse_unpaired.fq.gz`

```
ILLUMINACLIP:TruSeq3-PE.fa:2:30:10   LEADING:3   TRAILING:3
SLIDINGWINDOW:4:15 MINLEN:36
```

-PE = Pair end
-phred33 = refers to the phred score used
-input.fq.gz = input file in FASTQ compressed format
-output.fq.gz = outout file in FASTQ compressed format

- Manual:http://www.usadellab.org/cms/uploads/supplementary/Trimmomatic/TrimmomaticManual_V0.32.pdf

CD-HIT: It is used for clustering and comparing nucleotide or protein sequences and can handle very large databases.

- Module: `cdhit/git`
- Example: `cd-hit-est -i allseqs.fasta -o allseqsTc1 -c 1 -n 10 -p 1`

cd-hit-est= specific command for nucleotide sequences
i= FASTA file with all the sequences (repeated data)
o= name of the output file
c= threshold for sequences identity, 1=100% indicates that the sequences must be exactly the same to be merged
n= word size, 5 is the default value
p= 1 prints the matching region between representative sequences and each sequence in the custer

This program generates two files: a FASTA file with non repeated sequences and a .clstr file that specifies which sequences were grouped.

- Manual:http://www.usadellab.org/cms/uploads/supplementary/Trimmomatic/TrimmomaticManual_V0.32.pdf

GENE IDENTIFICATION AND SEQUENCE ANALYSIS

BLAST: It finds similarities between sequences of nucleotides or protein. The program compares the sequences to sequences in databases and gives a score of the accuracy of the matches

- Module: `blast+/2.2.29`

- Example: `blastn -db nt -query nt.fsa -out results.out`
  -query = FASTA file input
  -out = Output result file
  -db = makes the BLAST against a database
- Manual: https://www.ncbi.nlm.nih.gov/books/NBK279670/

HMMR: Is used to search sequence against databases references for homologous of DNA sequences or Protein, and for making sequence alignments. This tool uses Markov models as probabilistic models.

- Module: `hmmer/3.1b2(default:3.1)`
- Eaxmple:

- Manual:http://eddylab.org/software/hhttp://eddylab.org/Markovmodelssoftware/hmmer3/3.1b2/Userguide.pdfmmer3/3.1b2/Userguide.pdf

SEQUENCE ASSEMBLY

MEGAHIT: it assembles large and complex sequencing reads from Next Generation Sequencing using succinct Brujin graph for low memory use.

- Module: `megahit/git`

- Example: `megahit -1 File_1.fastq.gz -2 File_2.fastq.gz -o File.megahit_asm`

This tool supports Single en and Pair End reads in FASTA or FASTQ formats for pairends

-1 = Forward read input file.
-2 = Reverse read input file.
-r = Input for single end reads
--12 = Input Interleaved Pair End reads

- Manual:https://github.com/voutcn/megahit/

VELVET: makes genome assemblies by generating contigs and the scaffolds from very short reads produced by Next Generation Sequencing technologies. It is useful for data of new organism with no prior genome assemblies or for the determination of the origin of unmapped reads. Velvet always work with two programs *velveth* and *velvetg*

- Module: `velvet/1.2.10`

- Example: Velvet works with two steps: hashing that uses the executable `velveth` and graph building which uses `velvetg`

    1. Velvet hashing: `velveth directory.assembly 31 ~/fastq.gz -shortPaired ~/R1.fastq.gz ~/R2.fastq.gz`

       directory.assembly = The output directory name
       31= directory hash length.
       ~/fastq.gz = file format
       -shortPaired = Indicates the file category of the sequences used
       ~/R1.fastq.gz ~/R2.fastq.gz = files used

    2. Velvet graph building: `velvetg directory.assembly/ exp_cov 1 -min_contig_lgth 1000`

       directory.assembly= Directory created with velvet hashing
       exp_cov= assembly optimization parametre
       -min_contig_lgth = minimum size required for the contigs to be established

- Manual: https://www.ebi.ac.uk/~zerbino/velvet/Manual.pdf
  https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2952100/

SPADES: Is an assembler that supports Illumina or Ion Torrent reads. Also it can generate hibrid assemblies using PacBio, Oxford Nanopore and Sanger reads. SPAdes is designed for small genomes as bacterial or fungal genomes.

- Module: `module load SPAdes/3.11.0`

- Example: `spades.py -1 ~/R1 -2 ~/R2 -o assembly_spades/`

  -1 <filename> = file with forward paired-end reads
  -2 <filename> = file with reverse paired-end reads
  ALSO MAY BE:
  --12 <filename> = file with interlaced forward and reverse paired-end reads
  -s <filename> = file with unpaired reads

  -o = output directory

- Manual: http://cab.spbu.ru/files/release3.12.0/manual.html

PHYLOGENETIC ANALYSIS

BAMTOOLS: it is a toolkit for reading, writing, and manipulating genome alignment files in BAM format. It can change the bam format to other, count the alignments, filter the BAM files, create an index for the alignments, merge multiple bam files into one, and many other

- Module: `bamtools/2.5.1`
  `bamtools/git`

- Example for format conversion:

```
bamtools convert -format [bed|fasta|fastq|json|pileup|sam|yaml] -in input_alignments.bam -out output_reads.[bed|fasta|fastq|json|pileup|sam|yaml]
```

-convert = indicates to change the format of a .bam file
-format = specify to which format the file will be changed (any of the above)
-in = the input alignment file
-out = the output file in the new format

- Example for merging multiple files:

```
bamtools merge -in input_alignments_1.bam -in input_alignments_2.bam -in input_alignments_3.bam -out output_alignments_merged.bam
```

-in = indicates each BAM file to be merged
-out = indicates the name of the output file with all the BAM files merged

- Manual: https://github.com/pezmaster31/bamtools/wiki/Using-the-toolkit

SAMTOOLS: it's a set of tools for manipulating BAM alignments files, including sorting, merging and indexing, and to retrieve reads in any regions swiftly.

- Manual: http://www.htslib.org/doc/samtools-1.2.html

- Module: `samtools/1.9`
  `samtools/git`

- Example for alignment sorting :

```
samtools sort -n input_alignments.bam output_alignments_sorted
```

-n = it sorts by read names rather than by chromosomal coordinates
- input = bam files used as an input
-output = the name of the output file with alignments sorted

- Example for obtaining statistical parameters of each alignment :

```
samtools idxstats input_alignments_sorted.bam
```

-idxstats = Performs the statistical analysis
- name of the input BAM file