ngs-tools Documentation

version 0.1

Roland Schmucki

May 03, 2023

Contents

Welcome to ngs-tools's documentation!

Introduction

Collection of helper tools written in C (Roche's BIOS/Bioinfo-C library), Bash, Python, and R for next-generation sequencing data analysis.

For the Docker image, the base image is from the bios-to-go repository

Usage

How to use with Singularity

In order to re-use the Docker image and run it, for example on the HPC with Singularity, first, create a personal access token with "read_registry" rights in the repo and then export this token in the terminal where the Singularity container should be run by the following

```
export SINGULARITY_DOCKER_USERNAME=<username>
export SINGULARITY_DOCKER_PASSWORD=<read_registry token>

singularity run docker://ghcr.io/bedapub/ngs-tools:main make_cls -h
```

See also Singularity documentation

Installation

We provide a Docker image where all tools are already installed. This image can be used for Docker and Singularity. See Usage.

Tools

annotate_loci

```
Description:

This tool takes one input file with genomic coordinates in its first column and
an additional file with gene locus information. It then tries to annoate all
genomic regions from the first file, line by line,
with the annotations from the second file by overlapping the coordinates.

Usage: annotate_loci -i FILE -loci FILE -format gct|topTable

        -i                    input file with loci information (required), ie first column
                              must contain a coordinate string
                              CHR:BEGIN-END , ie separated by colon and dash. If the input
                              format is gct or topTable then all subsequent columns sent to
        -loci                 input file with loci information (required), tab-delimited for
                              CHR   BEGIN   END   STRAND   GENE   SYMBOL  DESCRIPTION
        -format gct|topTable  input file -i is in gct|topTable format (optional)
        -verbose              show more information (optional)

Report bugs and feedback to roland.schmucki@roche.com
```

count2tpm

```
Description:

Calculate normalized read counts for input GCT file.
Source:
https://haroldpimentel.wordpress.com/2014/05/08/what-the-fpkm-a-review-rna-seq-expression-un
Note that NaN are output as zero 0.

Usage: count2tpm -i GCT-file -l Length-file [-cpm|rpkm|tpm] [-log2|log10] [-col INT] [-digit


Mandatory input parameters:

        -g     GCT file with read counts per gene (unique gene identifier in 1st column):
        -l     tab-delimited file with gene identifier in 1st and gene length in
               2nd columns, respectively.
               These files can be found in the corresponding genome annotation folders,
               e.g. for human in folder /<path to genomes folder>/hg38/gtf/refseq/


Optional input parameters:

        -tpm     transcript per million (default)
        -rpkm    reads per kilobase of exon per million reads mapped
        -cpm     counts per million mapped reads
        -log2    log2 transform output (adding 0.01)
        -log10   log10 transform output (adding 0.01)
        -col     if input length file contains several columns, then specify
                 the column number with this index (default last column)
        -digits  number of digits after comma for output (default 3)


 Report bugs and feedback to roland.schmucki@roche.com
```

expression2gct

```
Description:

Convert biokit expression gene count files into GCT format.

Usage: expression2gct -infile='list of files' -outfile-prefix STRING

Mandatory input parameters:

        -infile: either a file containing the paths to input expression files OR
                 list of space/comma-separated files, e.g.
                 -infile='sample1.expression,sample2.expression'

        -outfile-prefix: there are 2 - 4 output files, already existing files of same
                         name will be overwritten:
                         STRING_rpkm.gct
                         STRING_count.gct

Optional input parameters:

       -use-unique-counts  (use the unique rpkm/read counts, default is multiple)

       -old-biokit-format  (use if expression file was generated with Biokit v3.8 or
                           earlier; the annotation is in column #7 instead of #8)


Report bugs and feedback to roland.schmucki@roche.com
```

extract_sequence

```
Description:

Extract from an input fasta or fastq file sequences by ids from another input file.

Usage: extract_sequence [-verbose] [-delimiter='. TAB'] [-useEntireIdLine]
          [-quick] [-not] -ids ids_file -fasta|fastq fasta_file

Mandatory parameters:

          -ids                file name containing sequence id's
          -fasta|fastq        file name containing the fasta or fastq sequences

Optional parameters:

          -delimiter          delimiter on the sequence id line
          -useEntireIdLine    use the entire line as id and not split line by
                              -delimiter into fields
          -quick              stop search after the first match
          -not                inverse the search, ie output sequences that are
                              not in the ids file
          -verbose            output additional information


Report bugs and feedback to roland.schmucki@roche.com
```

make_cls

```
Description:

Create a phenotype CLS file from a given GCT and annotation file.         The CLS format is
https://software.broadinstitute.org/cancer/software/gsea/wiki/index.php/Data_formats#Phenoty

Usage: make_cls -gct FILE  -i FILE

Mandatory input parameters:

          -gct GCT_FILE         input file in GCT format
          -i   ANNOTATION_FILE  input file with sample annotations

        The annotation file is a 2 column tab-delimited file (comments or header mark with #
          column 1: sample name as given in input GCT file
          column 2: sample group


Report bugs and feedback to roland.schmucki@roche.com
```

make_design_contrast_matrix

```
Description:

Usage: make_design_contrast_matrix [-prefix STRING] -gct FILE -i FILE

Mandatory parameters:

        -gct GCT_FILE         input file in GCT format
        -i   ANNOTATION_FILE  input file with sample annotations with 2 columns (see below)

Optional parameters:

        -prefix STRING        a string for the output prefix

        The annotation file is a 2 column tab-delimited file (comments or header mark with #
          column 1: sample name as given in input GCT file
          column 2: sample group


Report bugs and feedback to roland.schmucki@roche.com
```

mean

```
Description:

Calculate means per sample conditions.

Usage: mean [-skip INT] [-gzip]  -i INFILE  -s SAMPLE_ANNOTATIONS

Mandatory parameters:
          -i FILE  inptu file with sample data, e.g. read counts
          -s FILE  input file with sample annotations

The SAMPLE_ANNOTATIONS file is tab-delimited input file with at least 2 columns:
         column 1: sample name
         column 2: sample condition

The read count from INFILE are averaged (mean) for each sample condition.
The INFILE headers should match with the sample names specified in the SAMPLE_ANNOTATIONS fi
Note that the header line must begin with '#' or with 'ID'

Optional parameters:

          -skip INT     denotes how many columns from the INFILE should be skipped and
                        not used for calculation, e.g. skip ID or description columns, defau
          -gzip         use if INFILE is gzipped

Report bugs and feedback to roland.schmucki@roche.com
```

merge_fastq

```
Description:

Merge reads from several fastq files into one fastq file.
IMPORTANT: input files for mate R1 and R2 reads must be in the same ORDER.

Usage: merge_fastq [-sbatch] [-t INT] [-old-version] [-script-prefix STR] [-bsub-path STR] -

Mandatory parameters:

  -i  input_file  tab-delimited file with 2 columns: input gzipped fastq file,
                  output gzipped fastq file


Optional parameters:

  -sbatch         use "sbatch" for submitting to queue, default is "bsub"
  -t  integer     number of minutes for queuing system, default 360 = 6 hours, only for "sba
  -old-version    use old version which is much slower
  -script-prefix  prefix for temp scripts, e.g. path, default ./merge_fastq
  -bsub-path      path to bsub command on the shpc, default bsub


Report bugs and feedback to roland.schmucki@roche.com
```

merge_gct

```
Usage: merge_gct [-h] FILE1 FILE2 [FILE3 ...]

Merge GCT files

Optional arguments

  -h   display this help and exit

 Contact roland.schmucki@roche.com
```

minmax_gct

```
Filter away all features from a GCT file if the row
MIN or MAX is lower/greater/lower equal (MIN-EQUAL)/greater equal (MAX-EQUAL)
than a user given threshold. Use MIN-/MAX-REVERSE to output reversed comparison.
Results are redirected to the standard output.

3 input arguments required:

  1. input GCT file
  2. threshold value (real number)
  3. MIN or MAX or MIN-EQUAL or MAX-EQUAL or MIN-REVERSE or MAX-REVERSE

Contact roland.schmucki@roche.com
```

reorder_gct

```
Usage: reorder_gct [-h] -g GCT_FILE -s SAMPLE_FILE

Re-order samples (columns) in the GCT file by
the names given in the SAMPLE file.

  -g   input GCT file
  -s   input SAMPLE file with re-ordered sample names (file must contain exactly one colum

Optional arguments

  -h   display this help and exit

 Contact roland.schmucki@roche.com
```

replace_header_gct

```
Usage: replace_header_gct [-h] -g GCT_FILE -s SAMPLE_FILE

Replace the sample names in the GCT file by
the names given in the SAMPLE file.

  -g   input GCT file
  -s   input SAMPLE file
       2 columns required:
         1st: sample names in input GCT file
         2nd: sample names in output GCT file

Optional arguments

  -h   display this help and exit

 Contact roland.schmucki@roche.com
```

sort_gct

```
Usage: sort_gct [-h] -g GCT_FILE [-c 1|2] [-n] [-r]

Sorts input GCT file by column 1 (default) or 2 in numeric or alphabetic (default) order


  -g   input GCT file
  -c   column 1 or 2 (default is 1)
  -n   order numerically or alphabetically (default)
  -r   reverse order

Optional arguments

  -h   display this help and exit

 Contact roland.schmucki@roche.com
```

subset_gct

```
Usage: subset_gct [-h] -g GCT_FILE -k KEYS_FILE -s SAMPLES_FILE

Creates a subset of the input GCT file

  -g   input GCT file
  -k   input KEYS file
       1 column required:
         1st: keys (e.g. Genes) in input GCT file
  -s   input SAMPLES file
       1 column required:
         1st: samples names in input GCT file to output
Optional arguments

  -h   display this help and exit

 Contact roland.schmucki@roche.com
```

Contributing

Any contribution, feedback and bug report highly welcome. For major changes, please open an issue first to discuss what you would like to change. Thank you!

License

GNU GPLv3

Indices and tables

- **`genindex`**
- **`modindex`**
- **`search`**