Model-Based Multifactor Dimensionality Reduction

MBMDR-4.1.0 is a software that is able to detect multiple sets of significant gene-gene and/or gene-environment interactions in relation to a trait of interest, while efficiently controlling type I error rates. The trait can be expressed either on a binary or a continuous scale, or as a censored trait. To see the command line help, type

mbmdr.out --help

The instructions to run MBMDR-4.1.0 are (depending on the data type) as follows:

```
mbmdr.out --binary [options] 'mbmdrFile'
mbmdr.out --continuous [options] 'mbmdrFile'
mbmdr.out --survival [options] 'mbmdrFile'
```

If your data is expressed on a binary or continuous scale, then the 'mbmdrFile' must be represented using the following structure (for censored trait see *--help --survival*)

Tr S1 S2 ... Sm
X1 Y11 Y12 ... Y1m
... ...
Xk Yk1 Yk2 ... Ykm

The first line is a title line: Tr is the name of the trait and the Sj's are the names of the markers (SNPs or environment variables).

The first column contains the trait values: in the binary case, Xi is 1 if the i^{th} subject is a case and 0 if it is a control ; in the continuous case Xi is a continuous value representing the state of the ith subject. The other columns contain the markers values:

- if Sj is a SNP: Yij is 0 if the i^{th} subject is homozygous for the first allele, 1 if heterozygous and 2 if homozygous for the second allele.
- if Sj is an environment variable: the X different possible values of the environment variables should be coded 0, 1, ..., X-1.

Missingness: a missing Yij value must be coded -9. Missing Xi values are not accepted (the program will ignore the subject and generate a warning)

If your dataset is in PLINK format, you can first use the following command line to create the 'mbmdrFile' (replace --binary by --continuous or --survival depending on your trait)

```
mbmdr.out --plink2mbmdr --binary -ped 'pedFile' -map 'mapFile' -o 'mbmdrFile' -tr 'trFile'
```

This command will also generate a translation file ‘trFile’, giving the chosen label for each genotype of each SNP. The ‘pedFile’ must contain a title line.

The different options of the program are: (the options between square brackets are not mandatory)

OPTION	DESCRIPTION	DEFAULT
[-n INT]	Number of p-values to compute	1000
[-p INT]	Permutation amount for multiple-testing	999
[-r INT]	Random seed parameter	Random value
[-m INT]	Minimum group size to be statistically relevant	10
[-x DOUBLE]	Cutoff value for the chi-square test	0.1
[-mt STRING]	Multiple-testing correction algorithm: NONE, MAXT, MINP, RAWP, STRAT1, STRAT2 or speedMAXT.	MAXT
[-o STRING]	Output file name	'infileprefix'_output.txt
[-a STRING]	Adjust method to be used: CODOMINANT, ADDITIVE or NONE	CODOMINANT
[-d STRING]	Dimension of interactions: 1D, 2D or 3D.	2D
[-v STRING]	Verbose: NONE, SHORT, MEDIUM or LONG	NONE
[-pb STRING]	Progress bar: NONE or NORMAL	NORMAL
[-e LIST]	Erase markers (LIST: comma-separated list of marker names)	
[-E FILE]	Erase markers (FILE: file composed of one marker name per line)	
[-f LIST]	Filter: analyse only the pairs composed of exactly one marker (for instance an environment variable) from the comma-separated list of markers names	
[-F FILE]	Filter: analyse only the interactions composed of exactly one marker from the given file (in MBMDR format) and one marker from the input file.	
[-if STRING]	Input Format : MBMDR of MDR	MBMDR
[-rt STRING]	Rank transformation (continuous trait only): NONE or RANK_TRANSFORM	RANK_TRANSFORM

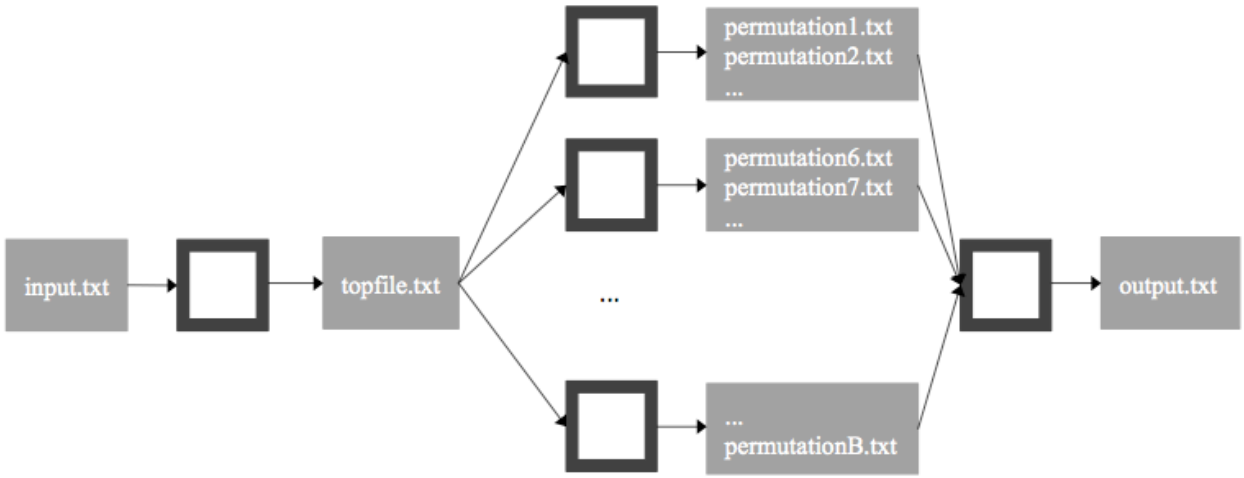
Parallel Workflows

We have developed two parallel workflows: for the MAXT and speedMAXT algorithms respectively. To see the command line help, type *mbmdr.out --help --parallel*

(WARNING: please use the same set op options at each step)

MAXT Workflow

This workflow is composed of three steps. The first step prepares the parallel work, the second step does the parallel work and the last step sums up the results.



STEP 1: compute the top vector on one CPU

```
mbmdr.out --continuous --pstep1 [options] 'mbmdrFile'
```

OPTIONS
[-t STRING] *sets the top file name (default: topFile.txt)*

You can also use the -n, -m, -a, -f, -v and -pb options

STEP 2: compute the permutations on N CPUs (1, 2, ..., N)

```
mbmdr.out --continuous --pstep2 -p INT -o STRING [options] 'mbmdrFile'
```

OPTIONS
-p INT *sets the permutation amount to be run on the current CPU*
-o STRING *sets the output file name (all CPUs must use 'xxxi.txt'*
where xxx is a common prefix and i the CPU id)
[-t STRING] *sets the top file name (default: topFile.txt)*

You can also use the -r, -m, -a, -f, -pb options

STEP 3: create the final output file on one CPU

```
mbmdr.out --continuous --pstep3 -c STRING -q INT [options] 'mbmdrFile'
```

OPTIONS

-c STRING	*sets the common prefix 'xxx' of the files generated at step 2*
-q INT	*sets the quantity of files generated at step 2*
[-p INT]	*sets the permutation amount (default: 999)*
[-o STRING]	*sets the output file name (default: 'inputprefix'_output.txt the file will be created in the directory of the input file)*
[-t STRING]	*sets the top file name (default: topFile.txt)*

You can also use the -r optionStep 1 (on one CPU)

speedMAXT Workflow

This workflow is composed of four steps. This time, the computation of the top file is also parallelized.

STEP 1: compute partial top vectors on N CPUs (1, 2, ..., N)

```
mbmdr.out --continuous --speedstep1 -i INT -N INT [options] 'mbmdrFile'
```

-i INT	sets the current CPU id
-N INT	sets the total amount of CPUs
[-t STRING]	sets the top file name (default: topFile.txt)
[-ti STRING]	sets the prefix of the temporary top files (default: top)

You can also use the -n, -m, -a, -v and -pb options

STEP 2: create the final top vector on one CPU

```
mbmdr.out --continuous --speedstep2 -N INT 'mbmdrFile'
```

-N INT	sets the total amount of CPUs
[-t STRING]	sets the top file name (default: topFile.txt)
[-ti STRING]	sets the prefix of the temporary top files (default: top)

STEP 3: compute the permutations on N CPUs (1, 2, ..., N)

```
mbmdr.out --continuous --speedstep3 -p INT -o STRING [options] 'mbmdrFile'
```

OPTIONS

-p INT	sets the permutation amount to be run on the current CPU
-o STRING	sets the output file name (all CPUs must use 'xxxi.txt' where xxx is a common prefix and i the CPU id)
[-t STRING]	sets the top file name (default: topFile.txt)

You can also use the -r, -m, -a, -pb options

STEP 4: create the final output file on one CPU

```
mbmdr.out --continuous --speedstep4 -c STRING -q INT [options] 'mbmdrFile'
```

OPTIONS

-c STRING	sets the common prefix 'xxx' of the files generated at step 3
-q INT	sets the quantity of files generated at step 3
[-p INT]	sets the permutation amount (default: 999)
[-o STRING]	sets the output file name (default: 'inputprefix'_output.txt the file will be created in the directory of the input file)
[-t STRING]	sets the top file name (default: topFile.txt)

You can also use the -r option

Remark: a much more detailed manual, giving the implementation details of the different algorithms, is under construction.