

NAME

vm_slice - Perform matrix slicing

VERSION

Version 0.2.466

SYNOPSIS

vm_slice [input_matrix] [output_matrix] [-b] [-c *columns*] [-h] [-n *notice_level*] [-q] [-r *rows*] [-s *rand_percent rest_file*] [-t] [-z *mod_num*]

Parameters with optional arguments should be placed after mandatory commandline arguments.

DESCRIPTION

Perform matrix slicing. The input file and output file are optional. If only one is specified, it is assumed to be the input file. If the input file or output file not specified, STDIN or STDOUT is used to allow for piping.

PARAMETERS

-b Output the matrix in binary format.

-c *columns*

Only output a selected set of *columns* specified in the string.

-h Print out the man page for help

-n *notice_level*

Set the degree of program output. Use:

-n 0 No output

-n 10 Normal program output

-n 20 Parameters useful for reproducing the results

-n 30 All output

-q Transpose the matrix.

-r *rows*

Only output a selected set *rows* specified in the string.

-s *rand_percent rest_file*

Output a random subset of *rows*. *rand_percent* is the fraction of the matrix to output and must be between 0 and 1. The random selection is taken after any other slicing. If *rest_file* is not NO_OUTPUT, the remaining slice is output to the specified filename.

-t Output the matrix in 'pretty' format.

-z mod_num

Reorder the matrix *rows* using the supplied modulus number. For a *mod_num* of 3, the first row in the new matrix is 0, followed by 3, 6, 9, ..., 1, 4, 7, ...

MATRIX FILE FORMATS

Five file formats are available for matrix and vector I/O depending on the type of matrix and format: full, symmetric, sparse, diagonal, pretty, and binary format.

The format for the full matrix is:

```
VecMat <rows> <columns>
<data at row1, col1>
<data at row2, col1>
<data at row3, col1>
...
<data at row2, col_n>
<data at row1, col2>
...
```

The format for a symmetric matrix is:

```
VecMatSym <rows>
<data at row1, col1>
<data at row2, col1>
<data at row2, col2>
<data at row3, col1>
<data at row3, col2>
<data at row3, col3>
...
```

for the lower triangle. The format for a diagonal matrix is:

```
VecMatDiag <rows>
<data at row1, col1>
<data at row2, col2>
<data at row3, col3>
...
```

for the diagonal. The format for a sparse matrix is:

```
VecMatSparse <rows> <columns>
<col_number*rows+row_number> <value>
<col_number*rows+row_number> <value>
...
END
```

The format for a pretty file is:

```
<data row1, col1> <data row1, col2> ... <data row1,col_n>
<data row2, col1> <data row2, col2> ... <data row2,col_n>
...
```

Binary files can also be loaded and saved, but are not architecture independent.

AUTHORS

W. Michael Brown