NAME

rcollate - resize or re-order matrix data

SYNOPSIS

rcollate [**−h[io]**][**−w**][**−f[afdb][N]**][**−t**][**−ic in_col**][**−ir in_row**][**−oc out_col**][**−or out_row**][**−o RxC[xR1xC2..]**] [**input.dat**]

DESCRIPTION

Rcollate reads in a single matrix file (table) and reshapes it to have the number of columns specified by the *−oc* option. The number of rows may be specified with a *−or* option, or may be determined automatically from the size of the input if it is an even multiple of the number of columns (as it should be). Alternatively, both may be specified using a *−o* option with the number of rows and columns separated by an 'x', as in "30x14" for 30 rows by 14 columns. *Rcollate* can also reorder the input into nested blocks by continuing the output size string. For example, "3x10X7x2" would order output data with a 3x10 super-array of 7x2 subblocks. This type of block hierarchy is convenient for visualizing tensor data.

By default, the file is assumed to include an information header, which is copied to the standard output along with the command name. The *−hi* option may be used to turn off the expectation of a header on input. The *−ho* option turns off header output, and *−h* by itself turns off both input and output headers. The *−w* option turns off non-fatal warning messages, such as unexpected EOD.

The input format is assumed to be ASCII, with three white-space separated words (typically numbers) in each record. A different input format may be specified with the *−f* option. The suboptions are *−fa, −ff, −fd,* and *−fb* for ASCII, float, double, and binary, respectively. An optional count may be attached to specify the number of data elements per record, which defaults to 1. Thus, the default setting is *−fa3.* Since *rcollate* does not interpret the fields, all binary options of the same length have the same result. On most architectures, *−ff6, −fd3,* and *−fb24* would all be equivalent. Note that the lack of row separators in binary files means that *rcollate* does not actually do anything for binary files unless the data is being re-ordered.

If an input header is present, it may contain the format, number of components and matrix dimensions. In such cases, the *−ic, −ir* and *−f* options are not required, but will be checked against the header information if provided.

The transpose option, *−t* swaps rows and columns on the input. For binary files with no header information, the user must specify at least one input or output dimension to define the matrix size. For ASCII files, *rcollate* will automatically determine the number of columns based on the position of the first EOL (end-of-line) character, and the number of rows based on the total count of records in the file. The user may override these determinations, allowing the matrix to be resized or re-ordered as well as transposed. If input and output dimensions are given and there is no block re-ordering, the number of input rows must equal the number of output columns, and the number of input columns must equal the number of output rows. If the *−o* option is also given with multiple block levels, the transpose operation will logically precede the re-ordering operation, regardless of their position on the command line.

EXAMPLES

To change put 8760 color triplets per row in a matrix with no header:

```
rcollate -h −oc 8760 input.dat > col8760.dat
```

To transpose a binary file with 145 float triplets per input row:

```
rcollate -ff3 -ic 145 -t orig.flt > transpose.flt
```

To create an appropriate header for a binary float matrix as required by *rmtxop(1)*

```
rcollate -hi -ff3 -or 145 -oc 8760 input.smx | rmtxop dcoef.dmx - > res.txt
```

To visualize a Shirley-Chiu BTDF matrix where the interior resolution is 64x64 and the exterior resolution is 32x32:

```
rcollate -o 64x64X32x32 s-c_bsdf.mtx | rmtxop -fc - > s-c_bsdf.hdr
```

AUTHOR

Greg Ward

NOTES

For large transpose or re-ordering operations on Unix systems, it is most efficient to specify the input file on the command line, rather than reading from the standard input, since *rcollate* can map the file directly into virtual memory.

The *rcollate* command is rather inflexible when it comes to output field and record separators for ASCII data. It accepts any amount of white space between fields on input, but only produces spaces as field separators between words and tabs as record separators on output. Output row separtors will always be an EOL, which may differ between systems.

If no options are given on the command line, or a binary file is specified without a transpose or re-ordering, *rcollate* issues a warning and simply copies its input to its standard output.

SEE ALSO

cnt(1), histo(1), neaten(1), rcalc(1), rlam(1), rmtxop(1), rsplit(1), tabfunc(1), total(1)