Compact Matrix Display

Thomas Kneser
GWDG, Göttingen (FRG)
John S. McCaskill
MPI für biophysikalische Chemie,
Göttingen (FRG)

The following problem arose in the study of the molecular structure of RNA[1]:

One wishes to represent information about the probabilities that various pairs (i, j) from a sequence of length n over a finite alphabet occur. It is important to be able to locate accurately from the display which pair is involved and how probable it is on a logarithmic scale. A similar representation problem arises in displaying the strength of connection between pairs of units of neural networks[2].

For large n, the only compact way to represent the information on a line printer is to encode a different character for each of a finite number of probability levels. The information is then displayed as a matrix. This leads to rather ugly output which is not easily interpretable even if the characters are chosen with the amount of black ink increasing with increasing probability. However, for a first look this is the most efficient way to obtain the information. The problem then is to convert this character output to a high quality image for visual processing.

A succinct way of doing this is by drawing black boxes of varying sizes accurately positioned with lower left corners forming the square matrix of probabilities. TeX provides the opportunity to draw such structures by setting sequences of appropriate `\vrule`s and to merge such plots with additional text and alphanumeric information. This merging leads as a side effect to the obvious advantage that a complete paper can easily be transferred through the networks by transmitting just a single file. Figure 2 shows that part of the input file which defines the matrix, containing line printer style character data. The related graphic output is shown in Figure 3.

Figure 1 shows the structure of one matrix element, with *wo* ranging from about 5*pt* to 10*pt* and *wi* < *wo*. The matrix elements can easily be coded as:

```
\hbox to \wo{\vrule height \wi width \wi
             depth 0pt \hfil}          (V1)
```

V1 is working fine for about $n < 50$ but for larger n problems arise concerning TeX's internal storage ("! TeX capacity exceeded ... "), even if the current page contains nothing other than the matrix and even if *mem_max*† is set to the maximum of $2^{16} - 1$.

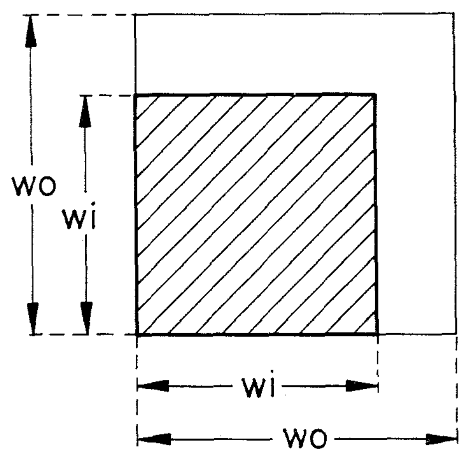

Figure 1

What is the reason for this rapid exhaustion of storage?

Clearly, TeX has to hold all the stuff defining the current page in *mem_array* until the `\shipout` operation is done. Taking this into account we have to ask the following question: how much memory does TeX use for one matrix element?

Analysing V1 following [3], we find that for each matrix element TeX generates the following list of *nodes*:

- 1 box_node
- 1 rule_node
- 1 glue_node

We define σ as the amount of storage TeX needs to allocate one rule matrix element. By summing up TeX's constants *box_node_size*, *rule_node_size* and *glue_spec_size* we get:

$$\sigma_1 = 7 + 4 + 4 = 15 \; memory\_words$$

This requirement is high compared with that for a *character token*, which requires only one *memory_word* to fill about the same area on paper.

Fortunately one finds that there is at least one alternative version with $\sigma < \sigma_1$:

```
\vrule height \wi width \wi depth 0pt
\rest=\wo
\advance\rest by -\wi
\kern\rest                              (V2)
```

† terms in italics with enclosed underline character refer to [3]

```
\centerline{PARTITION FUNCTION PROBABILITIES OF BINDINGS}\medskip
\centerline{log interval =      10.000}
\centerline{decreasing log (base 10) probabilities in order}
\hbox to \hsize{\hss\vbox{\halign{#\quad&\hfil#\cr
\eseq{ @}&    0 to  -0.5\cr\eseq{ *}& -0.5 to  -1.5\cr\eseq{ o}& -1.5 to  -2.5\cr
\eseq{ =}& -2.5 to  -3.5\cr\eseq{ +}& -3.5 to  -4.5\cr\eseq{ -}& -4.5 to  -5.5\cr
\eseq{ ;}& -5.5 to  -6.5\cr\eseq{ :}& -6.5 to  -7.5\cr\eseq{ ,}& -7.5 to  -8.5\cr
\eseq{ .}& -8.5 to  -9.5\cr\eseq{  }&  <  -9.5\cr}}\hss}%
\bigskip
\mrow{           1         2         3         4         5         6         7         8 }
\mrow{  12345678901234567890123456789012345678901234567890123456789012345678901234567890 }
\mrow{ 1G   .:,:,:=o-;;;;:::;-;;;;;,,.,-*+--;;.               ,-:;-;:. :,,.  .,.   .:;;-G}
\mrow{ 2 G   .,,:=o;,,::..;---;;;;,,.-*;,,,,.                  ,   ,,. :,,.         ,:;oG}
\mrow{ 3  G   ,:,o;,,:::..-;;;;;;;;;,*;,,,,..                  . .:;::,,.           ,;=@G}
\mrow{ 4   U   ;=:::::::::;-+====ooo=*-----;::                 :;;=o+;,          ,:-+o+;U}
\mrow{ 5    U   ;,::::::;++====ooo=*o;;---;::                  ;;;+=o;,        .,:-==*o-U}
\mrow{ 6     C   ,,:::::+-+=======*o-;;--;;::                 ,:;;+o+-,       .,:-==+o=;C}
\mrow{ 7      A   ,,:;;::;+oooo===++--;---;::                  ::;oo=;,         ,;--=*=;A}
\mrow{ 8       U   ,;:;;:;=o===oo=++++----;:;                  ::-+*+;.        .:--++==;U}
\mrow{ 9        A   ,:;;:;-==+o+++-+--;;;;;::                  ::;@=+;.        ..:;-o=+:A}
\mrow{10         C   ,::+@;;;;;;;;-*:,,:;;;::                  ::@;-:.        .+=o=o:;;.C}
\mrow{ 1          C   ,:@+;;-;;;;;*-:,::;;;::                  :@;;-:.        +=o=o::;;.C}
\mrow{ 2           U   :-;;;;;;+-*-;:;--++++=:                ;@;:;;:,       +=o=o:.:;;.U}
\mrow{ 3            A   ;--+++++=+-;--+==ooooo;,             :;+,:;;:,      ,;-+-; .:-;,A}
\mrow{ 4             U   --+===oo+--=-+=ooooo;,              ;--::;;:,      :o--;. ,:;;,U}
\mrow{ 5              U   --+==o+--=o--o*oooo;,              :;-::;;-.     .o;:,.  .:;;,U}
\mrow{ 6               C   -+===+;=o---+o====:;             ,,;-::;-:.     o  .    ,:;;:C}
\mrow{ 7                G   +++=+-;--=@---=-;        .   . , :.: .;;:,;,.:o        .:;;;G}
\mrow{ 8                 G   +-++-;;-@=---++-:..    .   ,     .:. :;:;,.:o         .:;;+G}
\mrow{ 9                  C   ---+oo-;-------,,    .   ,      ,:;;;+;::;:.        .:-+=:C}
\mrow{20                   U   -+oo==--------,.   .   ,       .;;:;-;:;--.        ,:-+=-U}
\mrow{ 1                    U   ====+;;;-;--+,.               .;::;-;::;;         .:-+=;U}
\mrow{ 2                     U   ++++;;;;;-++,.    .          .:::;-;::;;         .:-+=;U}
\mrow{ 3                      U   ---;;;;;---,    .           .:::;-;::;;         .:-+=;U}
\mrow{ 4                       A   ;;;;;+==;;.   .            .:::;--::;;         .:-=o;A}
\mrow{ 5                        A   ;:;+==+;;,                .::,;--::--.        .:+oo;A}
\mrow{ 6                         A   ::==+-;;,       .        .:::;++;::;         .:=oo;A}
\mrow{ 7                          G   =+--;+;, .    .         .:::-==-o=-.        ,;+=o+G}
\mrow{ 8                           G   ;;;-+;,                .:::+++o=-,         ,;+oooG}
\mrow{ 9                            A   :;-;;.   .            .:::@+;,.            ,-+o:A}
\mrow{30                 @           C   ::;;.. .             ,:-@;+;:,            -:--:C}
\mrow{ 1                @             C   ::;..               ,:@-;-;:.           -,;--:C}
\mrow{ 2                               U   ;-,               .:*-;---,.          ,.,;---U}
\mrow{ 3                                U   -,.              ,++;-++-,.            ,;+++U}
      .
      .
      .

\mrow{ 8         @                                               G     ,            ,;+*G}
\mrow{ 9        @                                                 U                ,;+=+U}
\mrow{60       @                                                   G    .         ,;+@o+G}
\mrow{ 1                                                            A             .:@=+:A}
\mrow{ 2                                                             C          ,-+@:;;.C}
\mrow{ 3                                                              C        ,-+@=.,. C}
\mrow{ 4                                                               C      ,-+@=-    C}
\mrow{ 5                                                                C     -+@=-,    C}
\mrow{ 6                                                                 C    +@=-,     C}
\mrow{ 7                                                                  C   @=-,      C}
\mrow{ 8                                                                   G            G}
\mrow{ 9                                                                    A           A}
\mrow{70                                                                     A          A}
\mrow{ 1                                                                  @   G         G}
\mrow{ 2                                                                 @     G       .G}
\mrow{ 3                                                                @       G     .-G}
\mrow{ 4                                                               @         G   .:-G}
\mrow{ 5                                                              @           G   :;G}
\mrow{ 6                                                             @             G   :G}
\mrow{ 7                                                            @               U   U}
\mrow{ 8                                                                             U  U}
\mrow{ 9                                                                              U U}
\mrow{80  @                                                                            CC}
\mrow{ 1 @                                                                              C}
\mrow{ 2@                                                                               C}
\mrow{  GGGUUCAUACCUAUUCGGCUUUUAAAGGACCUUUUCACGCGUAGCUAGCUACGCGAGGUGACCCCCGAAGGGGGGUUUC }
```

Figure 2 : Input file SAMPLE1.TEX

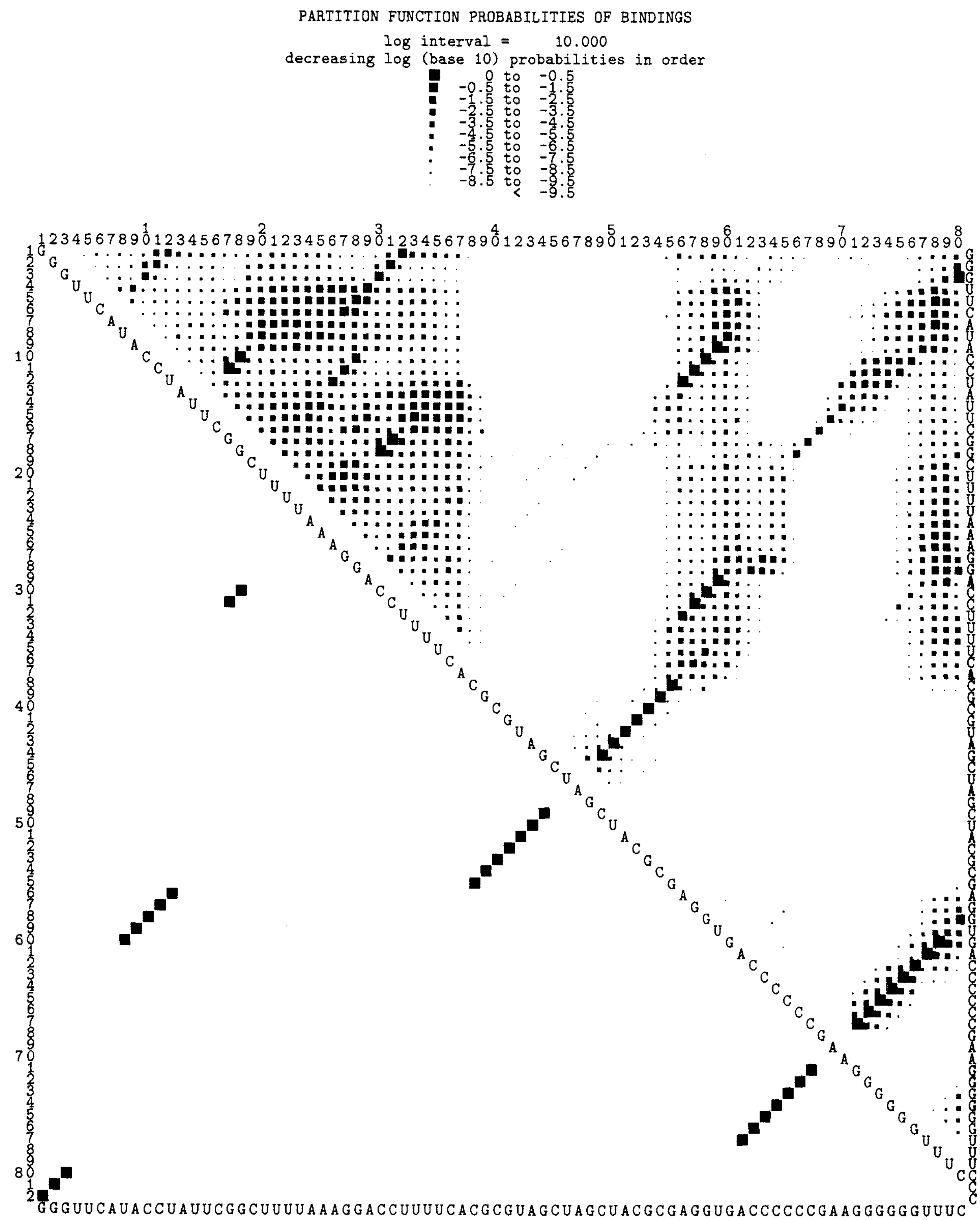

Figure 3 : Output related to SAMPLE1.TEX

Analyzing V2 we find:

- 1 rule_node
- 1 small_node (due to the \kern item)

and:

$$\sigma_2 = 4 + 2 = 6 \ memory\_words$$

since *small_node_size* = 2.

Comparing both σ values we get:

$$\sigma_2 = 0.4\sigma_1$$

Obviously V2 which is formulated more explicitly — and somewhat less elegantly — does the job much more economically. Therefore the macro \setrule in file CMD.TEX (Figure 4) contains V2.

Furthermore, storage requirements are reduced by summing up the space of horizontal sequences of empty matrix elements, using dimension register \accmt. Since the displayed matrices generally are sparse matrices this also leads to substantial memory saving.

Besides storage space, the requirement for CPU time is here also a limiting factor. We define T as the amount of CPU time TEX needs to process a particular matrix. Clearly T depends on the number of processed matrix elements but also heavily on the relation $N_E : N_R : N_O$ (number of empty elements, rules and \otherchars). Since we have developed CMD for an application concerning primarily sparse matrices, CMD should process empty elements faster than elements of the two other types. This is accomplished by the fact that CMD tests for spaces first (refer to Figure 4: the comparison of \next with \spacechar in macro \dodolist). If a space character is identified, CMD needs neither to expand macro \setrule nor to clamber through the \if-chain. However, T is still by a factor of 8 greater than the CPU time needed to set 'normal' running text; for SAMPLE1.TEX with

$$N_E : N_R : N_O = 5271 : 1386 : 420$$

we have measured $T \approx 16\,s$ (VAX 8650, TEX 2.0, Stanford version for VAX/VMS). About $0.3T$ are absorbed by the activities of macro \nonblank; here of course the \if-chain can be replaced by one simple \if...\else...\fi clause if we declare characters '.,;-+=*@' as active (catcode=13) — that is by effectively using indexing technique. But while CPU time tradeoff due to such a modification is small (< 5%), the coding becomes significantly more difficult to maintain and to update.

After these TEXnical considerations let us still mention one possible modification of CMD: for some cases one may want the blacked rule to be shifted from the lower left corner of the element (ref. Figure 3) to the center of the element area. This can easily be accomplished by replacing the body of macro \setrule by:

```
\rest=\wo\advance\rest by -\wi
\rest=0.5\rest
\rulh=\wi\advance\rulh by \rest
\kern\rest
\vrule height\rulh width\wi depth-\rest
\kern\rest                                  (V3)
```

with an additional dimension register \rulh to hold the rule height.

Finally, we hope the proposed TEX coding can serve as an efficient general purpose tool for real discretized matrix display in a large number of different applications.

[1] McCaskill, J.S. "The Equilibrium Partition Function and Base Pair Binding Probabilities for RNA Structure" (1988) in press at Biopolymers

[2] Kohonen, T. "Self-organization and associative memory" Springer, Berlin, Heidelberg, New York (1984)

[3] Knuth, D.E. "TEX: The Program" Addison-Wesley, Reading, Massachusetts (1986)

Figure 4. The file CMD.TEX

```
\message{+++++ CMD Compact Matrix Display , Version 1.4 +++++}
%                                Th. Kneser, John S. McCaskill
\newdimen\wo     % width of elements
\newdimen\wi     % width of \vrule in an element
\newdimen\mtacc  % accumulates length of successive empty elements
\newdimen\rest   % c(\rest) = c(\wo) - c(\wi)
\newdimen\zeropt%
\zeropt=0pt
%
\font\typw=CMTT8% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\typw            % WARNING: the current font should  n o t  be altered
\wo=0.65em       %          after this font related setting
                 % %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
\newdimen\hwaa\hwaa=.05em \newdimen\hwbb\hwbb=.10em
\newdimen\hwcc\hwcc=.15em \newdimen\hwdd\hwdd=.20em
\newdimen\hwee\hwee=.25em \newdimen\hwff\hwff=.30em
\newdimen\hwgg\hwgg=.35em \newdimen\hwhh\hwhh=.40em
\newdimen\hwii\hwii=.50em \newdimen\hwjj\hwjj=.60em
%     \hwXX: heights and widths of \vrules in matrix elements
%
\def\mrow#1{\hbox to \hsize{\hfil\dolist #1\endlist\mtelem\hfil}}%
%               % to set a centered matrix row
\def\eseq#1{\dolist #1\endlist\mtelem}%
%               % to set an inline sequence of matrix elements
\def\dolist{\afterassignment\dodolist\let\next=}
\def\dodolist{\ifx\next\endlist\let\next\relax%
              \else%
                \if\next\spacechar%
                   \global\advance\mtacc by \wo% to accumulate empty elements
                \else%
                   \ifdim\mtacc>\zeropt%
                      \mtelem%  process a sequence of empty elements
                   \fi%
                   \nonblank%   process one nonblank item
                \fi%
                \let\next\dolist%
                \fi%
              \next}%
% \dolist and \dodolist are derived from 'The TeXbook' Ex. 11.5,
% they parse a sequence of tokens (in the actual case the
% argument strings of \eseq and \mrow)
%
\def\nonblank{%  to set nonblank matrix elements
  \if\next .\wi=\hwaa\setrule%
  \else\if\next ,\wi=\hwbb\setrule%
  \else\if\next :\wi=\hwcc\setrule%
  \else\if\next ;\wi=\hwdd\setrule%
  \else\if\next -\wi=\hwee\setrule%
  \else\if\next +\wi=\hwff\setrule%
  \else\if\next =\wi=\hwgg\setrule%
  \else\if\next o\wi=\hwhh\setrule%
  \else\if\next *\wi=\hwii\setrule%
  \else\if\next @\wi=\hwjj\setrule%
  \else\otherchars%
 \fi\fi\fi\fi\fi\fi\fi\fi\fi\fi}% end of \def\nonblank
%
\def\mtelem{\kern\mtacc\global\mtacc=\zeropt}
%               % to set horizontal space equivalent
%               % to a sequence of empty elements
\def\setrule{\vrule height \wi width \wi depth \zeropt%
                \rest=\wo\advance\rest by -\wi%
                \kern\rest}
%               % to set a matrix element containing a rule
\def\otherchars{\hbox to \wo{\next\hss}}%
%               % to set nonblank & nonrule elements
\def\spacechar{ }% to be compared with \next
%
\baselineskip=\wo plus 1pt        % do  n o t  change
\lineskiplimit=0pt\parindent=0pt  % the current font here!
\obeyspaces
\input sample1
\bye
```