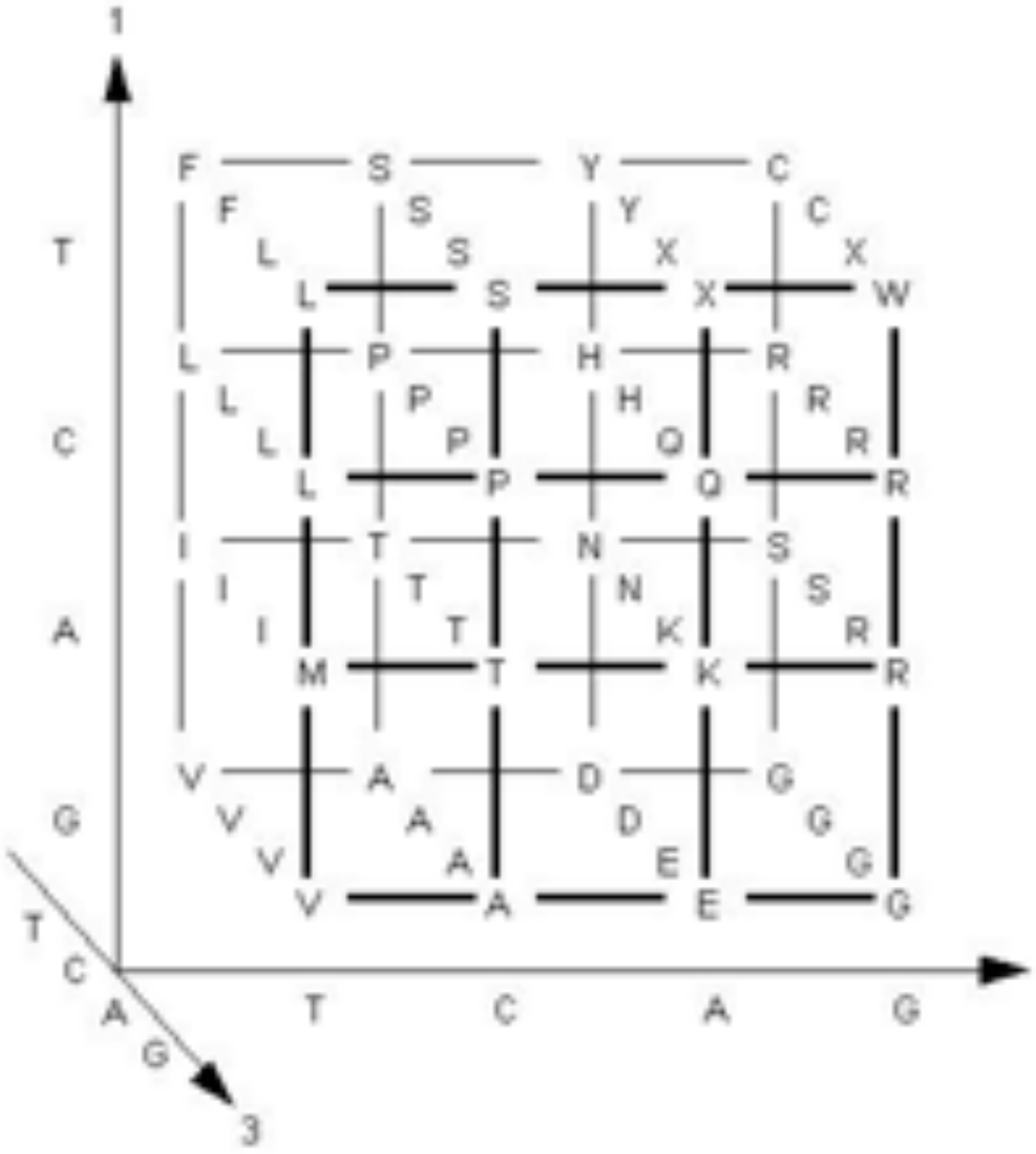
```
(* Genetic code
```




```
*)

(* 64 {a,c,g,t} codons in binary in a,c,g,t order as defined by the following *)
```

```
t =.; g =.; c =.; a =.;
nucgc = Tuples[{a, c, g, t}, 3];
a = {1, 0, 0, 0};
c = {0, 1, 0, 0};
g = {0, 0, 1, 0};
t = {0, 0, 0, 1};

nucgc = Flatten[nucgc];
nucgc = Partition[nucgc, 12];


pivns = PseudoInverse[nucgc]; (*corrected*)


(*

codon triplets entered manually in canonical order by alphabet -
 as above in matrix

  AAA
  AAC
  AAG
  AAT
  ACA
  ACC
  ACG
  ACT
  AGA
  AGC
  AGG
  AGT
  ATA
  ATC
  ATG
  ATT
  CAA
  CAC
  CAG
  CAT
  CCA
  CCC
  CCG
  CCT
  CGA
  CGC
  CGG
  CGT
  CTA
  CTC
```

```
  CTG
  CTT
  GAA
  GAC
  GAG
  GAT
  GCA
  GCC
  GCG
  GCT
  GGA
  GGC
  GGG
  GGT
  GTA
  GTC
  GTG
  GTT
  TAA
  TAC
  TAG
  TAT
  TCA
  TCC
  TCG
  TCT
  TGA
  TGC
  TGG
  TGT
  TTA
  TTC
  TTG
  TTT

  manually copy the triplets from above and replace
  with amino acid single letter code, in identical order,
and make matrix 'aagc' for amino acid genetic code, including 3 stop codons

*)
```

```
(* Map an attribute of the amino acids onto the triplet codons *)

jjt =.

a =.;
c =.;
d =.;
e =.;
f =.;
g =.;
h =.;
i =.;
k =.;
l =.;
m =.;
n =.;
p =.;
q =.;
r =.;
s =.;
t =.;
v =.;
w =.;
x =.; y =.;

jjt = {{k}, {n}, {k}, {n}, {t}, {t}, {t}, {t}, {r}, {s}, {r}, {s}, {i}, {i}, {m}, {i},
    {q}, {h}, {q}, {h}, {p}, {p}, {p}, {p}, {r}, {r}, {r}, {r}, {l}, {l}, {l}, {l},
    {e}, {d}, {e}, {d}, {a}, {a}, {a}, {a}, {g}, {g}, {g}, {g}, {v}, {v}, {v}, {v},
    {x}, {y}, {x}, {y}, {s}, {s}, {s}, {s}, {x}, {c}, {w}, {c}, {l}, {f}, {l}, {f}};

nuchyd =.;

nuchyd = pivns.jjt;
```

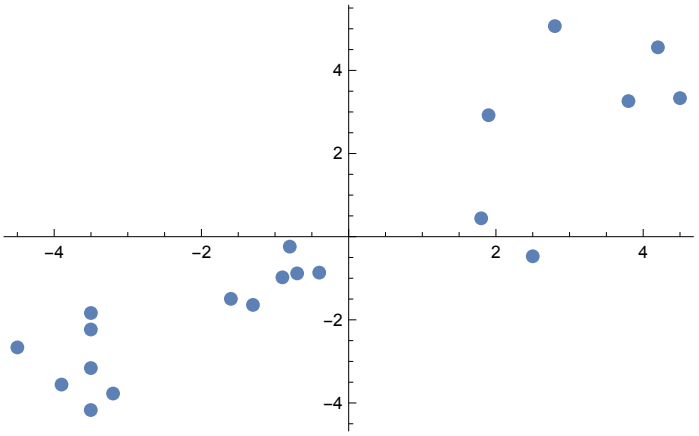
```
(* 20 K&D hydropathy values, padded with 0.0 for stop *)
(* checked against http://
 web.expasy.org/protscale/pscale/Hphob.Doolittle.html *)

kd = {1.8, 2.5, -3.5001, -3.5002, 2.8, -0.4, -3.2, 4.5, -3.9, 3.8, 1.9,
   -3.5003, -1.6, -3.5004, -4.5, -0.8, -0.7, 4.2, -0.9, 0.0, -1.3};


a = kd[[1]];
c = kd[[2]];
d = kd[[3]];
e = kd[[4]] ;
f = kd[[5]] ;
g = kd[[6]] ;
h = kd[[7]] ;
i = kd[[8]] ;
k =  kd[[9]];
l = kd[[10]] ;
m = kd[[11]] ;
n = kd[[12]] ;
p = kd[[13]] ;
q = kd[[14]] ;
r = kd[[15]] ;
s = kd[[16]] ;
t = kd[[17]] ;
v = kd[[18]];
w = kd[[19]];
x = kd[[20]] ;
y = kd[[21]] ;


nuchyd (* numeric with above hydropathy values *)

{{-0.638625}, {-1.25125}, {0.68625}, {0.880375}, {-2.63938}, {-0.163375},
 {-1.46963}, {3.94913}, {-0.169937}, {0.117688}, {-0.388687}, {0.117688}}

(* Reconstruct hydropathy values from nucleotide information
 using dot product of a weighted vector for triplets *)
(* use single letter code for new vectors, precede by the letter 'a',
because variables are global *)

(* Watch out!  Triplets are in  A,C,G,T for use here *)


aa = {0, 0, 1, 0,    0, 1, 0, 0,  .25, .25, .25, .25};
aanuc = Total[aa.nuchyd, 2]

ac = {0, 0, 0, 1,  0, 0, 1, 0,  0, 0.5, 0, 0.5};
```

```
acnuc = Total[ac.nuchyd, 2]

ad = {0, 0, 1, 0,  1, 0, 0, 0,  0, 0.5, 0.0, 0.5};
adnuc = Total[ad.nuchyd, 2]

ae = {0, 0, 1, 0,  1, 0, 0, 0,  0.5, 0, 0.5, 0};
aenuc = Total[ae.nuchyd, 2]

af = {0, 0, 0, 1,  0, 0, 0, 1,  0, 1, 0, 1};
afnuc = Total[af.nuchyd, 2]

ag = {0, 0, 1, 0,  0, 0, 1, 0,  .25, .25, .25, .25};
agnuc = Total[ag.nuchyd, 2]

ah = {0, 1, 0, 0,  1, 0, 0, 0,  0, 0.5, 0, 0.5};
ahnuc = Total[ah.nuchyd, 2]

ai = {1, 0, 0, 0,  0, 0, 0, 1,  1 / 3, 1 / 3, 0, 1 / 3};
ainuc = Total[ai.nuchyd, 2]

ak = {1, 0, 0, 0,  1, 0, 0, 0,  0.5, 0, 0.5, 0};
aknuc = Total[ak.nuchyd, 2]

al = {0, 2 / 3, 0, 1 / 3,  0, 0, 0, 1,  2 / 6, 1 / 6, 2 / 6, 1 / 6};
alnuc = Total[al.nuchyd, 2]

am = {1, 0, 0, 0,  0, 0, 0, 1,  0, 0, 1, 0};
amnuc = Total[am.nuchyd, 2]

an = {1, 0, 0, 0,  1, 0, 0, 0,  0, 0.5, 0, 0.5};
annuc = Total[an.nuchyd, 2]

ap = {0, 1, 0, 0,  0, 1, 0, 0,  .25, .25, .25, .25};
apnuc = Total[ap.nuchyd, 2]

aq = {0, 1, 0, 0,  1, 0, 0, 0,  0.5, 0, 0.5, 0};
aqnuc = Total[aq.nuchyd, 2]

ar = {1 / 3, 2 / 3, 0, 0,  0, 0, 1, 0,  1 / 3, 1 / 6, 1 / 3, 1 / 6};
arnuc = Total[ar.nuchyd, 2]

as = {1 / 3, 0, 0, 2 / 3,  0, 2 / 3, 1 / 3, 0,  1 / 6, 1 / 3, 1 / 6, 1 / 3};
asnuc = Total[as.nuchyd, 2]

at = {1, 0, 0, 0,  0, 1, 0, 0,  .25, .25, .25, .25};
atnuc = Total[at.nuchyd, 2]

av = {0, 0, 1, 0,  0, 0, 0, 1,  .25, .25, .25, .25};
avnuc = Total[av.nuchyd, 2]
```

```
aw = {0, 0, 0, 1,  0, 0, 1, 0,  0, 0, 1, 0};
awnuc = Total[aw.nuchyd, 2]

ax = {0, 0, 0, 1,  2 / 3, 0, 1 / 3, 0,  2 / 3, 0, 1 / 3, 0};
axnuc = Total[ax.nuchyd, 2]

ay = {0, 0, 0, 1,  1, 0, 0, 0,  0, 1 / 2, 0, 1 / 2};
aynuc = Total[ay.nuchyd, 2]
```

```
0.442063

-0.471563

-1.83544

-2.23244

5.06488

-0.864187

-3.77294

3.33231

-3.55731

3.26144

2.92181

-3.16031

-1.49544

-4.16994

-2.66365

-0.239396

-0.882813

4.55456

-0.977938

-1.61194

-1.64131
```

```
(* Fill empirical and SVP values for hydropathy into a 20 x 2 matrix *)

paired = {{a, aanuc}, {c, acnuc}, {d, adnuc}, {e, aenuc},
    {f, afnuc}, {g, agnuc}, {h, ahnuc}, {i, ainuc}, {k, aknuc},
    {l, alnuc}, {m, amnuc}, {n, annuc}, {p, apnuc}, {q, aqnuc}, {r, arnuc},
    {s, asnuc}, {t, atnuc}, {v, avnuc}, {w, awnuc}, {y, aynuc}};

ListPlot[paired, PlotStyle → PointSize[0.02]]
FindFit[paired, intercept + slope * xaxis, {intercept, slope}, xaxis]
Correlation[Take[paired, All, {1}], Take[paired, All, {2}]]
```




```
{intercept → 0.00335099, slope → 0.862013}

{{0.914202}}
```

```
(*
Reaction Centers of Photosynthetic Bacteria (1990)Michel-Beyerle M.ed.pp.209-218,
Springer-Verlag Berlin.  Genetic Coding Algorithms for
  Engineering Membrane Proteins.  Yang M.M.,Coleman,W.J.,& Youvan,
D.C.Massachusetts Institute of Technology,Department of Chemistry,
Cambridge 02139.  Abstract available at
  http://
 www.kairos-scientific.com/searchable/abstracts/Yang1990.htm and quoted here:

    A solution to the problem of relating the
     physicochemical properties of the amino acids to their codon
     sequences has been achieved by treating the genetic code as a
     system of linear equations and applying the numerical method,
Singular Value Decomposition (SVD). For example,
hydropathy
 and molar volume,
which are important deteminants of protein
 structure and function,
can be quantitatively related to the nucleotide sequence.The
  20 hydropathy values of the amino acid residues were
  remapped to 12 nucleotide-determined values which,
in turn,were used to predict structural aspects the
 photosynthetic reaction center protein,
without DNA→protein translation.
*)
```

```
(* hotmap *)

hotmap = Table[{
     Piecewise[{
       {1, 1 <= i < 42},
       {1, 42 <= i < 84},
       {Floor[5.8 * (i - 83)], 84 <= i < 128},
       {256, 128 <= i < 170},
       {256, 170 <= i < 212},
       {256, 212 <= i <= 256}
      }],
     Piecewise[{
       {1, 1 <= i < 42},
       {(i - 41) * 6, 42 <= i < 84},
       {256, 84 <= i < 128},
       {256 - (i - 128) * 6, 128 <= i < 170},
       {1, 170 <= i < 212},
       {Floor[(i - 211) * 5.7], 212 <= i <= 256}
      }],
     Piecewise[{
       {Floor[i * 5.8], 1 <= i < 42},
       {256 - ((i - 42) * 6), 42 <= i < 84},
       {1, 84 <= i < 128},
       {1, 128 <= i < 170},
       {(i - 169) * 6, 170 <= i < 212},
       {256, 212 <= i <= 256}
      }]
    },
    {i, 1, 256, 1}];

hotmap[[1, All]] = 1 ; (* This changes 1,1,
5 to 1,1,1 at i=1 for the blue channel *)

(* Pseudocolor 61 x 12 pivns matrix *)

(* get scaling factors *)

graymat = pivns;
graymin = Min[graymat]
graymax = Max[graymat]
delta = graymax - graymin
```

$$-\frac{1}{96}$$

$$\frac{5}{96}$$

$$\frac{1}{16}$$

```
(* rescale graymat according to min max to get grayscales of 1 to 256 *)

For[x = 1, x ≤ 12, x++,
  For[y = 1, y ≤ 61, y++,
   graymat[[x, y]] = Floor[(graymat[[x, y]] - (graymin)) * (255 / delta) + 1] ]];

colormat = Table[{x, y}, {x, 12}, {y, 61}];
colormat[[All, All]] = {0, 0, 0};
For[x = 1, x ≤ 12, x++,
  For[y = 1, y ≤ 61, y++,
   colormat[[x, y]] = hotmap[[graymat[[x, y]] ]] / 256
  ]];

gmat = Graphics[RasterArray[Apply[RGBColor, colormat, {2}]],
    ImageSize → {10 * 61, 10 * 12}, AspectRatio → Automatic];
Show[
 gmat]
```

Part::partw : Part 17387547842683 of
{{1, 1, 1}, {1, 1, 11}, {1, 1, 17}, {1, 1, 23}, {1, 1, 29}, {1, 1, 34}, {1, 1, 40}, {1, 1, 46}, {1, 1, 52}, {1, 1, 58}, {1, 1, 63}, {1, 1, 69}, {1, 1, 75}, {1, 1, 81}, {1, 1, 87}, {1, 1, 92}, {1, 1, 98}, {1, 1, 104}, {1, 1, 110}, {1, 1, 116}, {1, 1, 121}, ≪9≫, {1, 1, 179}, {1, 1, 185}, {1, 1, 191}, {1, 1, 197}, {1, 1, 203}, {1, 1, 208}, {1, 1, 214}, {1, 1, 220}, {1, 1, 226}, {1, 1, 232}, {1, 1, 237}, {1, 6, 256}, {1, 12, 250}, {1, 18, 244}, {1, 24, 238}, {1, 30, 232}, {1, 36, 226}, {1, 42, 220}, {1, 48, 214}, {1, 54, 208}, ≪206≫} does not exist. ≫

Part::partw : Part 17387547842683 of
{{1, 1, 1}, {1, 1, 11}, {1, 1, 17}, {1, 1, 23}, {1, 1, 29}, {1, 1, 34}, {1, 1, 40}, {1, 1, 46}, {1, 1, 52}, {1, 1, 58}, {1, 1, 63}, {1, 1, 69}, {1, 1, 75}, {1, 1, 81}, {1, 1, 87}, {1, 1, 92}, {1, 1, 98}, {1, 1, 104}, {1, 1, 110}, {1, 1, 116}, {1, 1, 121}, ≪9≫, {1, 1, 179}, {1, 1, 185}, {1, 1, 191}, {1, 1, 197}, {1, 1, 203}, {1, 1, 208}, {1, 1, 214}, {1, 1, 220}, {1, 1, 226}, {1, 1, 232}, {1, 1, 237}, {1, 6, 256}, {1, 12, 250}, {1, 18, 244}, {1, 24, 238}, {1, 30, 232}, {1, 36, 226}, {1, 42, 220}, {1, 48, 214}, {1, 54, 208}, ≪206≫} does not exist. ≫

Part::partw : Part 17387547842683 of
{{1, 1, 1}, {1, 1, 11}, {1, 1, 17}, {1, 1, 23}, {1, 1, 29}, {1, 1, 34}, {1, 1, 40}, {1, 1, 46}, {1, 1, 52}, {1, 1, 58}, {1, 1, 63}, {1, 1, 69}, {1, 1, 75}, {1, 1, 81}, {1, 1, 87}, {1, 1, 92}, {1, 1, 98}, {1, 1, 104}, {1, 1, 110}, {1, 1, 116}, {1, 1, 121}, ≪9≫, {1, 1, 179}, {1, 1, 185}, {1, 1, 191}, {1, 1, 197}, {1, 1, 203}, {1, 1, 208}, {1, 1, 214}, {1, 1, 220}, {1, 1, 226}, {1, 1, 232}, {1, 1, 237}, {1, 6, 256}, {1, 12, 250}, {1, 18, 244}, {1, 24, 238}, {1, 30, 232}, {1, 36, 226}, {1, 42, 220}, {1, 48, 214}, {1, 54, 208}, ≪206≫} does not exist. ≫

General::stop : Further output of Part::partw will be suppressed during this calculation. ≫

RasterArray::obs : RasterArray is obsolete. Translating to Raster. ≫