



```

0 0 1 0 0 0 0 1 0 0 0 1
0 0 1 0 0 0 0 1 0 0 1 0
0 0 1 0 0 0 0 1 0 1 0 0
0 0 1 0 0 0 0 1 1 0 0 0
0 0 1 0 0 0 1 0 0 0 0 1
0 0 1 0 0 0 1 0 0 0 1 0
0 0 1 0 0 0 1 0 0 1 0 0
0 0 1 0 0 0 1 0 1 0 0 0
0 0 1 0 0 1 0 0 0 0 0 1
0 0 1 0 0 1 0 0 0 0 1 0
0 0 1 0 0 1 0 0 0 1 0 0
0 0 1 0 0 1 0 0 1 0 0 0
0 0 1 0 1 0 0 0 0 0 0 1
0 0 1 0 1 0 0 0 0 0 1 0
0 0 1 0 1 0 0 0 0 1 0 0
0 0 1 0 1 0 0 0 1 0 0 0
0 1 0 0 0 0 0 1 0 0 0 1
0 1 0 0 0 0 0 1 0 0 1 0
0 1 0 0 0 0 0 1 0 1 0 0
0 1 0 0 0 0 0 1 1 0 0 0
0 1 0 0 0 0 1 0 0 0 0 1
0 1 0 0 0 0 1 0 0 0 1 0
0 1 0 0 0 0 1 0 0 0 1 0
0 1 0 0 0 0 1 0 1 0 0 0
0 1 0 0 0 1 0 0 0 0 0 1
0 1 0 0 0 1 0 0 0 0 1 0
0 1 0 0 0 1 0 0 0 1 0 0
0 1 0 0 0 1 0 0 0 1 0 0
1 0 0 0 0 0 0 1 0 0 0 1
1 0 0 0 0 0 0 1 0 0 1 0
1 0 0 0 0 0 0 1 0 1 0 0
1 0 0 0 0 0 0 1 1 0 0 0
1 0 0 0 0 0 1 0 0 0 0 1
1 0 0 0 0 0 1 0 0 0 1 0
1 0 0 0 0 0 1 0 0 1 0 0
1 0 0 0 0 1 0 0 0 0 0 1
1 0 0 0 0 1 0 0 0 0 1 0
1 0 0 0 0 1 0 0 0 1 0 0
1 0 0 0 1 0 0 0 0 0 1 0
1 0 0 0 1 0 0 0 0 1 0 0
1 0 0 0 1 0 0 0 1 0 0 0

```

(\* drop stop codon rows \*)

```

nucgcns = Drop[nucgc, {57}];
nucgcns = Drop[nucgcns, {51}];
nucgcns = Drop[nucgcns, {49}];

```

(\* take pseudoinverse ~ 2 hrs computaton time on Sony VAIO PCG-K64S,  
then hardcode result and comment out PseudoInverse \*)



163	179	37	163	37	25	25	25	97
16416	16416	4104	16416	4104	1824	1824	1824	8208
11	97	11	7	11	25	97	11	97
864	8208	864	513	864	1824	8208	864	8208
35	179	97	179	179	37	163	37	
2736	16416	8208	16416	16416	4104	16416	4104	
97	163	89	163	847	439	863	439	35
8208	16416	8208	16416	16416	8208	16416	8208	2736
179	97	179	179	37	163	37	97	
16416	8208	16416	16416	4104	16416	4104	8208	
163	89	163	847	439	863	439	35	179
16416	8208	16416	16416	8208	16416	8208	2736	16416
97	179	179	37	163	37	97	163	
8208	16416	16416	4104	16416	4104	8208	16416	
89	163	847	439	863	439	25	25	25
8208	16416	16416	8208	16416	8208	1824	1824	1824
97	11	97	11	7	11	89	26	43
8208	864	8208	864	513	864	1824	513	864
13	739	389	739	965	193	1429	193	193
1368	57456	28728	57456	114912	16416	114912	16416	
1663	3133	221	3133	965	193	1429	193	13
28728	57456	4104	57456	114912	16416	114912	16416	1368
739	389	739	965	193	1429	193	1663	
57456	28728	57456	114912	16416	114912	16416	28728	
3133	221	3133	965	193	1429	193	13	739
57456	4104	57456	114912	16416	114912	16416	1368	57456
389	739	965	193	1429	193	1663	3133	221
28728	57456	114912	16416	114912	16416	28728	57456	4104
3133	965	193	1429	193	65	65	73	149
57456	114912	16416	114912	16416	6384	6384	12768	16416
59	149	173	29	173	73	149	59	149
6048	16416	3024	513	3024	12768	16416	6048	16416
35	179	97	179	847	439	863	439	97
2736	16416	8208	16416	16416	8208	16416	8208	8208
163	89	163	179	37	163	37	35	
16416	8208	16416	16416	4104	16416	4104	2736	
179	97	179	847	439	863	439	97	163
16416	8208	16416	16416	8208	16416	8208	8208	16416
89	163	179	37	163	37	35	179	
8208	16416	16416	4104	16416	4104	2736	16416	
97	179	847	439	863	439	97	163	89
8208	16416	16416	8208	16416	8208	8208	16416	8208
163	179	37	163	37	25	25	89	
16416	16416	4104	16416	4104	1824	1824	1824	
26	43	26	11	7	11	25	97	11
513	864	513	864	513	864	1824	8208	864
83	3245	1747	3245	1205	241	1189	241	305
1368	57456	28728	57456	114912	16416	114912	16416	28728
851	43	851	1205	241	1189	241	83	3245
57456	4104	57456	114912	16416	114912	16416	1368	57456
1747	3245	1205	241	1189	241	305	851	
28728	57456	114912	16416	114912	16416	28728	57456	
43	851	1205	241	1189	241	83	3245	1747
4104	57456	114912	16416	114912	16416	1368	57456	28728
3245	1205	241	1189	241	305	851	43	
57456	114912	16416	114912	16416	28728	57456	4104	
851	1205	241	1189	241	415	415	25	101
57456	114912	16416	114912	16416	6384	6384	12768	16416

11	101	19	1	19	25	101	11	101		
6048	16416	3024	513	3024	12768	16416	6048	16416	}	
35	179	97	847	179	37	163	439	97	163	
2736	16416	8208	16416	16416	4104	16416	8208	8208	16416	
89	863	179	37	163	439	35	179	97	847	
8208	16416	16416	4104	16416	8208	2736	16416	8208	16416	
179	37	163	439	97	163	89	863	179	37	
16416	4104	16416	8208	8208	16416	8208	16416	16416	4104	
163	439	35	179	97	847	179	37	163	439	
16416	8208	2736	16416	8208	16416	16416	4104	16416	8208	
97	163	89	863	179	37	163	439	25	89	
8208	16416	8208	16416	16416	4104	16416	8208	1824	1824	
25	97	11	26	11	7	43	25	97	11	26
1824	8208	864	513	864	513	864	1824	8208	864	513
13	965	1663	965	739	193	3133	193	}		
1368	114912	28728	114912	57456	16416	57456	16416			
389	1429	221	1429	739	193	3133	193	13		
28728	114912	4104	114912	57456	16416	57456	16416	1368		
965	1663	965	739	193	3133	193	389			
114912	28728	114912	57456	16416	57456	16416	28728			
1429	221	1429	739	193	3133	193	13	965		
114912	4104	114912	57456	16416	57456	16416	1368	114912		
1663	965	739	193	3133	193	389	1429			
28728	114912	57456	16416	57456	16416	28728	114912			
221	1429	739	193	3133	193	73	73	65		
4104	114912	57456	16416	57456	16416	12768	12768	6384		
149	173	149	59	29	59	65	149	173	149	
16416	3024	16416	6048	513	6048	6384	16416	3024	16416	
35	847	97	179	179	439	163	37	97	863	
2736	16416	8208	16416	16416	8208	16416	4104	8208	16416	
89	163	179	439	163	37	35	847	97	179	
8208	16416	16416	8208	16416	4104	2736	16416	8208	16416	
179	439	163	37	97	863	89	163	179	439	
16416	8208	16416	4104	8208	16416	8208	16416	16416	8208	
163	37	35	847	97	179	179	439	163	37	
16416	4104	2736	16416	8208	16416	16416	8208	16416	4104	
97	863	89	163	179	439	163	37	89	25	
8208	16416	8208	16416	16416	8208	16416	4104	1824	1824	
25	26	11	97	43	7	11	25	26	11	97
1824	513	864	8208	864	513	864	1824	513	864	8208
83	1205	305	1205	3245	241	851	241	}		
1368	114912	28728	114912	57456	16416	57456	16416			
1747	1189	43	1189	3245	241	851	241	83		
28728	114912	4104	114912	57456	16416	57456	16416	1368		
1205	305	1205	3245	241	851	241	1747			
114912	28728	114912	57456	16416	57456	16416	28728			
1189	43	1189	3245	241	851	241	83	1205		
114912	4104	114912	57456	16416	57456	16416	1368	114912		
305	1205	3245	241	851	241	1747	1189	43		
28728	114912	57456	16416	57456	16416	28728	114912	4104		
1189	3245	241	851	241	25	25	415	101		
114912	57456	16416	57456	16416	12768	12768	6384	16416		
19	101	11	1	11	415	101	19	101		
3024	16416	6048	513	6048	6384	16416	3024	16416	}};	

(\* codon triplets entered manually in canonical order by alphabet -  
as above in nucgc 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 'aagcns' for amino acid genetic code, delete 3 stop codons \*)

aagcns =

{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,  
y,  
y,  
s,  
s,  
s,  
s,  
c,  
w,  
c,  
l,  
f,  
l,  
f);
```

```

(* 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

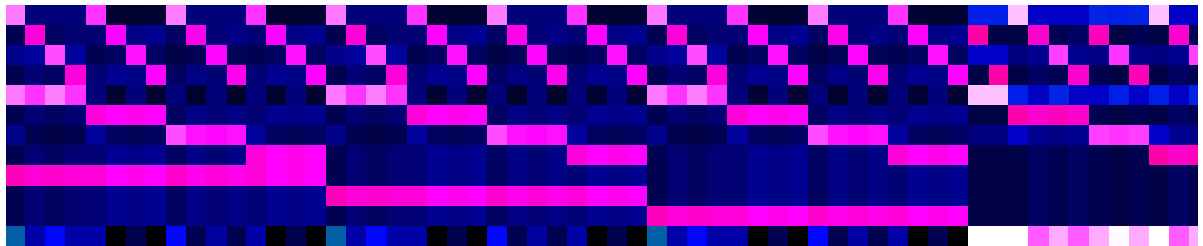
- 143
- 8208

125
1824

1411
16416

```

```
(* 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];
```



```
(* 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 =.; 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}, {y}, {y}, {s}, {s}, {s}, {s}, {c}, {w}, {c}, {l}, {f}, {l}, {f}};
```

```
nuchyd =.
```

```
(* symbolic *)
```

```
nuchyd = pivns.jjt;
```

```
MatrixForm[nuchyd]
```

$$\begin{pmatrix} -\frac{457a}{8208} + \frac{55c}{432} - \frac{131d}{8208} - \frac{e}{513} + \frac{485f}{4104} - \frac{307g}{8208} - \frac{131h}{8208} - \frac{37i}{864} - \frac{k}{513} + \frac{157l}{2052} - \frac{211m}{16416} - \frac{131n}{8208} - \frac{457p}{8208} - \frac{q}{513} - \frac{4}{8} \\ \frac{1733a}{8208} - \frac{11c}{432} + \frac{847d}{8208} + \frac{103e}{1026} - \frac{97f}{4104} + \frac{1703g}{8208} - \frac{179h}{8208} - \frac{25i}{864} - \frac{101k}{4104} - \frac{67l}{1026} - \frac{163m}{16416} - \frac{179n}{8208} - \frac{319p}{8208} - \frac{101q}{4104} \\ -\frac{319a}{8208} - \frac{11c}{432} - \frac{179d}{8208} - \frac{101e}{4104} - \frac{97f}{4104} - \frac{349g}{8208} + \frac{847h}{8208} - \frac{25i}{864} - \frac{101k}{4104} + \frac{379l}{2052} - \frac{163m}{16416} - \frac{179n}{8208} + \frac{1733p}{8208} + \frac{103q}{1026} \\ -\frac{319a}{8208} - \frac{11c}{432} - \frac{179d}{8208} - \frac{101e}{4104} - \frac{97f}{4104} - \frac{349g}{8208} - \frac{179h}{8208} + \frac{137i}{864} + \frac{103k}{1026} - \frac{67l}{1026} + \frac{863m}{16416} + \frac{847n}{8208} - \frac{319p}{8208} - \frac{101q}{4104} \\ -\frac{319a}{8208} - \frac{11c}{432} - \frac{179d}{8208} - \frac{101e}{4104} + \frac{52f}{513} - \frac{349g}{8208} - \frac{179h}{8208} + \frac{137i}{864} - \frac{101k}{4104} + \frac{1271l}{4104} + \frac{863m}{16416} - \frac{179n}{8208} - \frac{319p}{8208} - \frac{101q}{4104} \\ -\frac{91a}{2052} + \frac{173c}{1512} - \frac{739d}{28728} - \frac{331e}{14364} - \frac{149f}{8208} + \frac{6343g}{28728} - \frac{739h}{28728} - \frac{193i}{6048} - \frac{331k}{14364} - \frac{491l}{8208} - \frac{1429m}{114912} - \frac{739n}{28728} - \frac{91p}{2052} - \\ \frac{1733a}{8208} - \frac{11c}{432} - \frac{179d}{8208} - \frac{101e}{4104} - \frac{97f}{4104} - \frac{349g}{8208} - \frac{179h}{8208} - \frac{25i}{864} - \frac{101k}{4104} - \frac{67l}{1026} - \frac{163m}{16416} - \frac{179n}{8208} + \frac{1733p}{8208} - \frac{101q}{4104} \\ -\frac{103a}{2052} - \frac{19c}{1512} + \frac{3245d}{28728} + \frac{1745e}{14364} - \frac{101f}{8208} - \frac{1457g}{28728} + \frac{3245h}{28728} - \frac{241i}{6048} + \frac{1745k}{14364} - \frac{443l}{8208} - \frac{1189m}{114912} + \frac{3245n}{28728} - \frac{103p}{2052} \\ \frac{97a}{4104} + \frac{c}{27} + \frac{167d}{4104} - \frac{101e}{4104} + \frac{319f}{8208} + \frac{41g}{2052} + \frac{167h}{4104} + \frac{29i}{864} - \frac{101k}{4104} - \frac{23l}{8208} - \frac{163m}{16416} + \frac{167n}{4104} + \frac{97p}{4104} - \frac{101q}{4104} - \frac{1}{41} \\ \frac{149a}{8208} - \frac{59c}{3024} - \frac{965d}{57456} + \frac{695e}{14364} - \frac{149f}{8208} + \frac{887g}{57456} - \frac{965h}{57456} - \frac{55i}{1512} + \frac{695k}{14364} + \frac{535l}{8208} + \frac{3133m}{57456} - \frac{965n}{57456} + \frac{149p}{8208} + \frac{69}{14} \\ \frac{97a}{4104} + \frac{c}{27} + \frac{167d}{4104} - \frac{101e}{4104} + \frac{319f}{8208} + \frac{41g}{2052} + \frac{167h}{4104} + \frac{29i}{864} - \frac{101k}{4104} - \frac{23l}{8208} - \frac{163m}{16416} + \frac{167n}{4104} + \frac{97p}{4104} - \frac{101q}{4104} - \frac{1}{41} \\ \frac{101a}{8208} - \frac{11c}{3024} - \frac{1205d}{57456} + \frac{719e}{14364} - \frac{101f}{8208} + \frac{1703g}{57456} - \frac{1205h}{57456} + \frac{41i}{1512} + \frac{719k}{14364} + \frac{583l}{8208} - \frac{851m}{57456} - \frac{1205n}{57456} + \frac{101p}{8208} + \frac{1}{41} \end{pmatrix}$$

```
(* 20 empirical observed values for propensity to be in a membrane,  
i.e., hydropathy values*)
```

```
a = 1.37;  
c = 1.12;  
d = 0.17;  
e = 0.16;  
f = 1.93;  
g = 1.03;  
h = 0.74;  
i = 2.20;  
k = 0.19;  
l = 1.78;  
m = 1.39;  
n = 0.43;  
p = 0.51;  
q = 0.35;  
r = 0.30;  
s = 0.83;  
t = 0.89;  
v = 1.81;  
w = 1.56;  
y = 1.01;
```

```
nuchyd (* numeric with above hydropathy values *)
```

```
{{0.495109}, {0.431103}, {0.121103}, {0.277978}, {1.19798}, {0.10284},  
{0.237353}, {-0.212878}, {0.396728}, {0.261992}, {0.396728}, {0.269845}}
```

```
(* 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 *)
```

```
(* a=  
{0,1,0,0, 0,0,1,0, 1,0,0,0},  
{0,1,0,0, 0,0,1,0, 0,1,0,0},  
{0,1,0,0, 0,0,1,0, 0,0,1,0},  
{0,1,0,0, 0,0,1,0, 0,0,0,1}}; *)
```

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

```
0.99978
```

```
(* c=
  {{1,0,0,0, 0,1,0,0, 1,0,0,0},
   {1,0,0,0, 0,1,0,0, 0,0,1,0}}; *)
```

```
ac = {1, 0, 0, 0, 0, 1, 0, 0, 0.5, 0, 0.5, 0};
acnuc = Total[ac.nuchyd, 2]
```

0.994678

```
(* d=
  {{0,1,0,0, 0,0,0,1, 0,0,1,0},
   {0,1,0,0, 0,0,0,1, 1,0,0,0}}; *)
```

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

```
adnuc = Total[ad.nuchyd, 2]
```

0.614953

```
(* e={{0,1,0,0, 0,0,0,1, 0,0,0,1},
        {0,1,0,0, 0,0,0,1, 0,1,0,0}}; *)
```

```
ae = {0, 1, 0, 0, 0, 0, 0, 1, 0, .5, 0, .5};
```

```
aenuc = Total[ae.nuchyd, 2]
```

0.484144

```
(* f={{1,0,0,0, 1,0,0,0, 1,0,0,0},
        {1,0,0,0, 1,0,0,0, 0,0,1,0}}; *)
```

```
af = {1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0};
```

```
afnuc = Total[af.nuchyd, 2]
```

2.48654

```
(*g=
  {{0,1,0,0, 0,1,0,0, 1,0,0,0},
   {0,1,0,0, 0,1,0,0, 0,1,0,0},
   {0,1,0,0, 0,1,0,0, 0,0,1,0},
   {0,1,0,0, 0,1,0,0, 0,0,0,1}}; *)
```

```
ag = {0, 1, 0, 0, 0, 1, 0, 0, .25, .25, .25, .25};
```

```
agnuc = Total[ag.nuchyd, 2]
```

0.865267

```

(* h=
  {{0,0,1,0,    0,0,0,1,    1,0,0,0},
   {0,0,1,0,    0,0,0,1,    0,0,1,0}}; *)

ah = {0, 0, 1, 0,  0, 0, 0, 1,  .5, 0, .5, 0};
ahnuc = Total[ah.nuchyd, 2]

0.304953

(* i=
  {{0,0,0,1,    1,0,0,0,    1,0,0,0},
   {0,0,0,1,    1,0,0,0,    0,0,1,0},
   {0,0,0,1,    1,0,0,0,    0,0,0,1}}; *)

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

1.83039

(* k=
  {{0,0,0,1,    0,0,0,1,    0,1,0,0},
   {0,0,0,1,    0,0,0,1,    0,0,0,1}}; *)

ak = {0, 0, 0, 1,  0, 0, 0, 1,  0, .5, 0, .5};
aknuc = Total[ak.nuchyd, 2]

0.331019

(* l=
  {{0,0,1,0,    1,0,0,0,    1,0,0,0},
   {0,0,1,0,    1,0,0,0,    0,1,0,0},
   {0,0,1,0,    1,0,0,0,    0,0,1,0},
   {0,0,1,0,    1,0,0,0,    0,0,0,1},

   {1,0,0,0,    1,0,0,0,    0,1,0,0},
   {1,0,0,0,    1,0,0,0,    0,0,0,1}}; *)

al = {.33, 0, .66, 0,  1, 0, 0, 0,  1/6, 2/6, 1/6, 2/6};
alnuc = Total[al.nuchyd, 2]

1.75081

(* m=
  {{0,0,0,1,    1,0,0,0,    0,1,0,0}}; *)

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

1.73795

```

```
(* n=
  {{0,0,0,1,    0,0,0,1,    1,0,0,0},
   {0,0,0,1,    0,0,0,1,    0,0,1,0}}; *)
```

```
an = {0, 0, 0, 1,  0, 0, 0, 1,  .5, 0, .5, 0};
annuc = Total[an.nuchyd, 2]
```

0.461828

```
(*
  p=
  {{0,0,1,0,    0,0,1,0,    1,0,0,0},
   {0,0,1,0,    0,0,1,0,    0,1,0,0},
   {0,0,1,0,    0,0,1,0,    0,0,1,0},
   {0,0,1,0,    0,0,1,0,    0,0,0,1}}; *)
```

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

0.68978

```
(* q=
  {{0,0,1,0,    0,0,0,1,    0,1,0,0},
   {0,0,1,0,    0,0,0,1,    0,0,0,1}}; *)
```

```
aq = {0, 0, 1, 0,  0, 0, 0, 1,  0, .5, 0, .5};
aqnuc = Total[aq.nuchyd, 2]
```

0.174144

```
(* r=
  {{0,0,1,0,    0,1,0,0,    1,0,0,0},
   {0,0,1,0,    0,1,0,0,    0,1,0,0},
   {0,0,1,0,    0,1,0,0,    0,0,1,0},
   {0,0,1,0,    0,1,0,0,    0,0,0,1},

   {0,0,0,1,    0,1,0,0,    0,1,0,0},
   {0,0,0,1,    0,1,0,0,    0,0,0,1}}; *)
```

```
ar = {0, 0, 4/6, 2/6,  0, 1, 0, 0,  1/6, 2/6, 1/6, 2/6};
arnuc = Total[ar.nuchyd, 2]
```

0.585757

```
(* s=
  {{1,0,0,0,    0,0,1,0,    1,0,0,0},
   {1,0,0,0,    0,0,1,0,    0,1,0,0},
   {1,0,0,0,    0,0,1,0,    0,0,1,0},
   {1,0,0,0,    0,0,1,0,    0,0,0,1}},
  {{0,0,0,1,    0,1,0,0,    1,0,0,0},
   {0,0,0,1,    0,1,0,0,    0,0,1,0}}; *)
```

```
as = {4/6, 0, 0, 2/6, 0, 2/6, 4/6, 0, 2/6, 1/6, 2/6, 1/6};
asnuc = Total[as.nuchyd, 2]
```

```
0.968373
```

```
(* t=
  {{0,0,0,1,    0,0,1,0,    1,0,0,0},
   {0,0,0,1,    0,0,1,0,    0,1,0,0},
   {0,0,0,1,    0,0,1,0,    0,0,1,0},
   {0,0,0,1,    0,0,1,0,    0,0,0,1}}; *)
```

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

```
0.846655
```

```
(* v=
  {{0,1,0,0,    1,0,0,0,    1,0,0,0},
   {0,1,0,0,    1,0,0,0,    0,1,0,0},
   {0,1,0,0,    1,0,0,0,    0,0,1,0},
   {0,1,0,0,    1,0,0,0,    0,0,0,1}}; *)
```

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

```
1.9604
```

```
(* w=
  {{1,0,0,0,    0,1,0,0,    0,1,0,0}}; *)
```

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

```
0.859942
```

```
(* (*x=
  {{1,0,0,0,    0,0,0,1,    0,0,0,1},
   {1,0,0,0,    0,0,0,1,    0,1,0,0},
   {1,0,0,0,    0,1,0,0,    0,0,0,1}}; *)
```

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

```
(* y=
  {{1,0,0,0,    0,0,0,1,    1,0,0,0},
   {1,0,0,0,    0,0,0,1,    0,0,1,0}}; *)

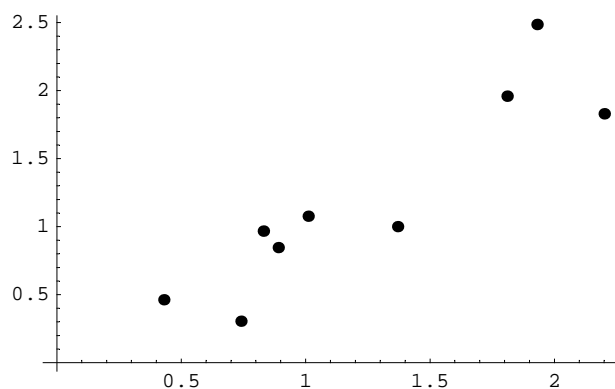
ay = {1, 0, 0, 0,  0, 0, 0, 1,  1, 0, 1, 0};
aynuc = Total[ay.nuchyd, 2]

1.07569
```

```
(* 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]
```



```
{intercept -> 0.150414, slope -> 0.860636}
```

```
(*
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 determinants 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 of the photosynthetic reaction center protein, without DNA→protein translation.

```
*)
```