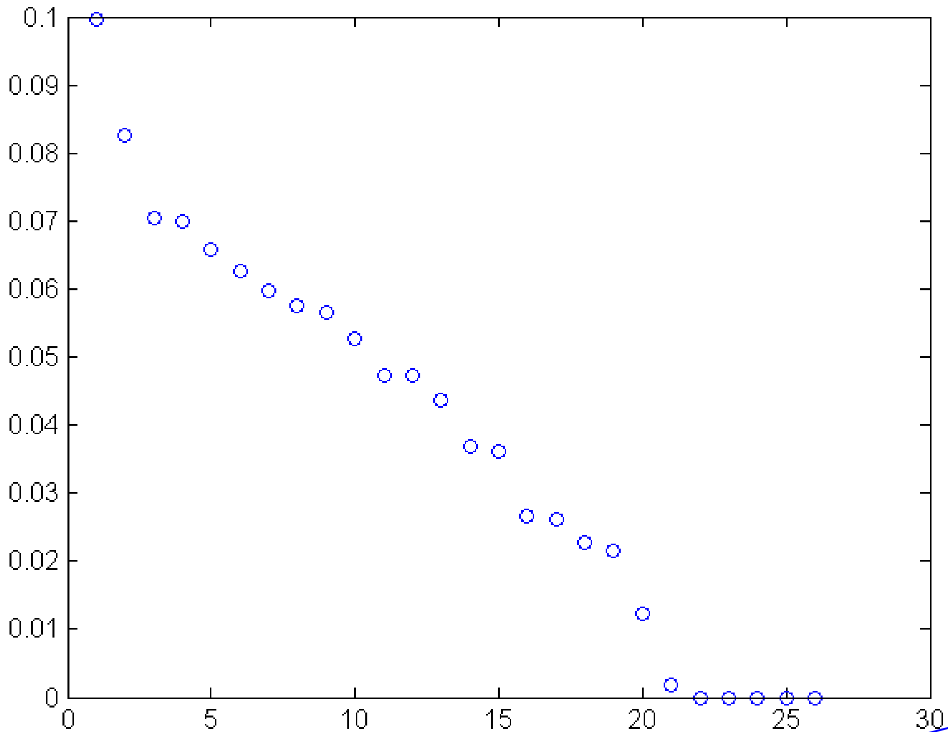
CS395T
Computational Statistics with Application to Bioinformatics

Prof. William H. Press
Spring Term, 2011
The University of Texas at Austin

Lecture 22

Plot distribution in descending order. Also calculate entropy:

```
plot(sort(mono(1:26),'descend'),'ob')
```



Notice that we flatten any structure in x when calculating the entropy.

```
entropy2 = @(x) sum(-x(:).*log(x(:)+1.e-99))/log(2);

h2bound = log(20)/log(2)
h2mono = entropy2(mono)
h2bound =
    4.3219
h2mono =
    4.1908
```

maximum entropy that 20 characters could have

actual (single peptide) entropy of the AA's

Actually, the single peptide ("monographic") entropy is only a bound on the true entropy of proteins, because there can be (and is) multiple symbol nonrandomness.

Standard compression programs bound the entropy, sometimes well, sometimes not:

```
 Directory of  D:\staticbio\prot*

 4/11/08  12:18       9,753,363   ___A_  proteomeHG17.txt
 4/14/08  17:45       5,554,389   ___A_  proteomeHG17.zip
 4/11/08  12:18       5,554,186   ___A_  proteomeHG17_1.txt.gz
```

8 x 5554186 / 9753363 = 4.556 (yuck! not as good as our monographic bound of 4.191)

Let's look at the dipeptide (digraph) and tripeptide (trigraph) distribution.

```
load 'aadist_di.txt';
di = aadist_di ./ sum(aadist_di(:));
h2di = entropy2(di)
h2di =
    8.3542
```

8.3542 / 2 = 4.177

```
load 'aadist_tri.txt';
tri = aadist_tri ./ sum(aadist_tri(:));
h2tri = entropy2(tri)

h2tri =
   12.5026
```

12.5026 / 3 = 4.168

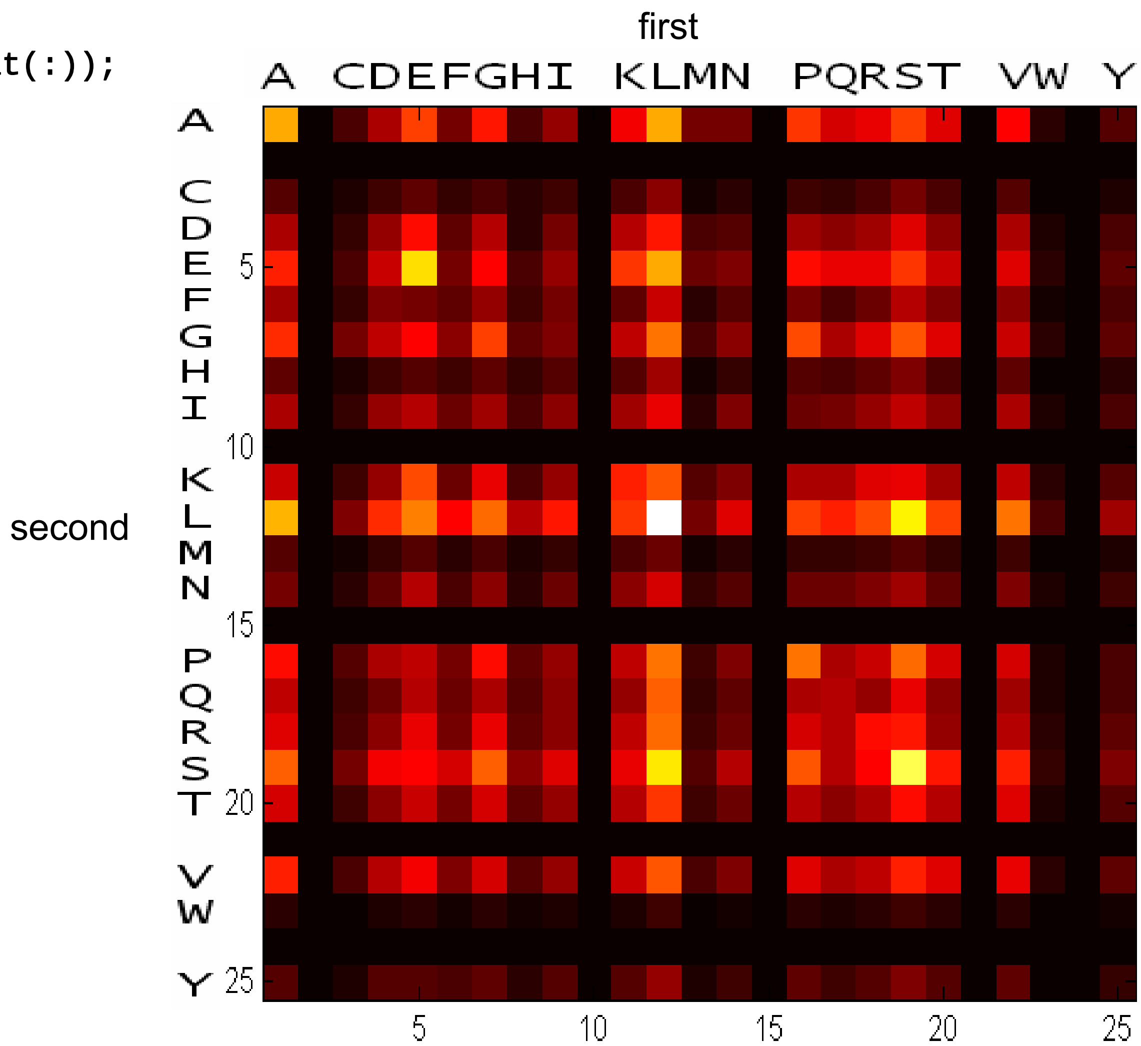
(We'll see in a minute that it's a mathematical theorem that these have to decrease – but they don't have to decrease much!)

Actually it's interesting to look at the dipeptide distribution

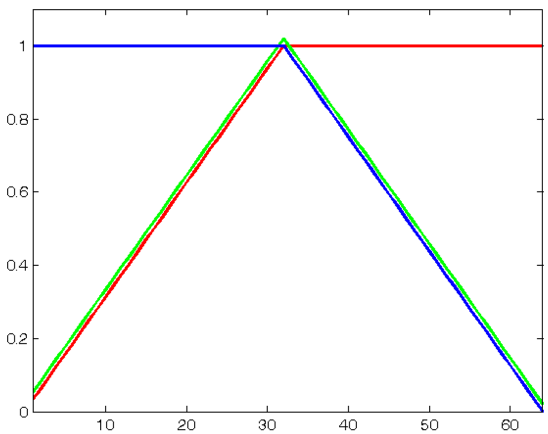
```
dimat = reshape(di,32,32);
image_di = 64*dimat./max(dimat(:));
image(image_di(1:25,1:25))
colormap('hot')
```

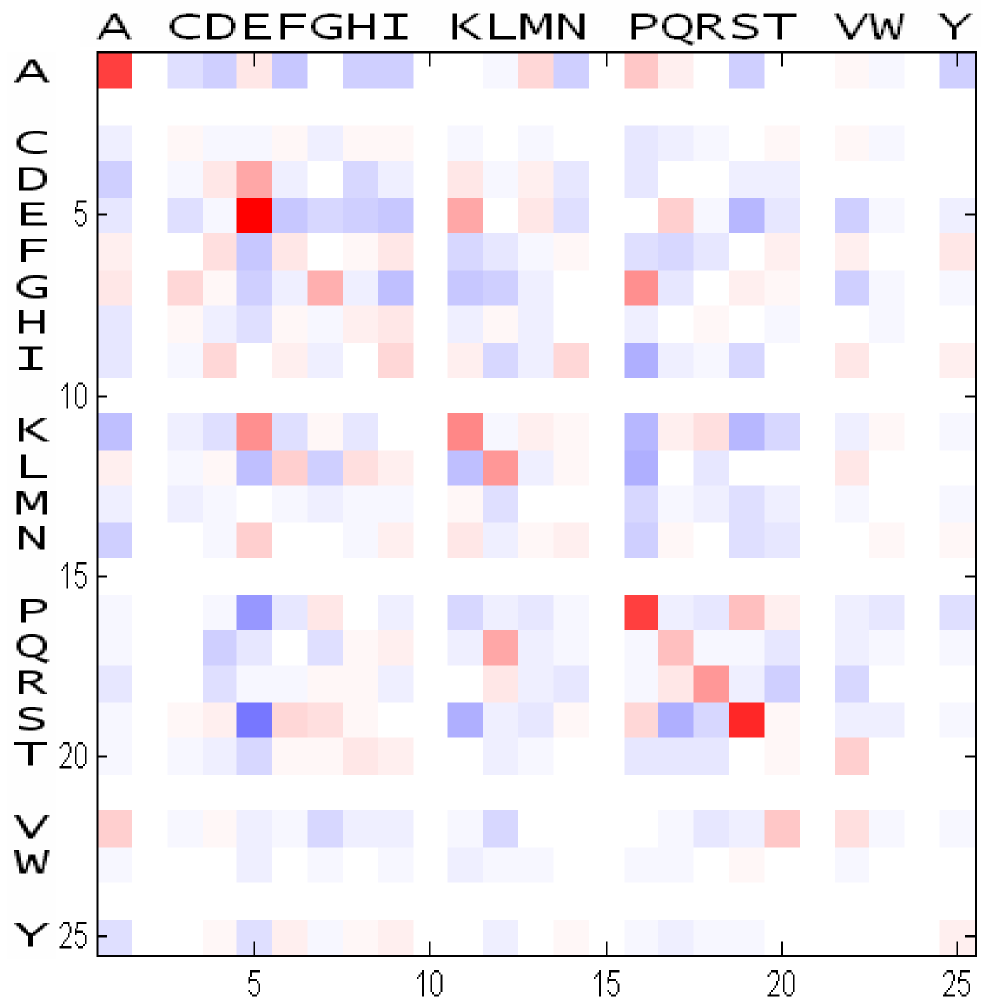
But what we are seeing is mostly just the outer product of the monographic distribution!

A	Alanine
R	Arginine
N	Asparagine
D	Aspartic acid (Aspartate)
C	Cysteine
Q	Glutamine
E	Glutamic acid (Glutamate)
G	Glycine
H	Histidine
I	Isoleucine
L	Leucine
K	Lysine
M	Methionine
F	Phenylalanine
P	Proline
S	Serine
T	Threonine
W	Tryptophan
Y	Tyrosine
V	Valine



```
discrep = dimat - mono * mono';
image_discrep = (32/max(discrep(:)))*discrep+32;
image(image_discrep(1:25,1:25));
genecolormap = [min(1,(1:64)/32); 1-abs(1-(1:64)/32); min(1,(64-(1:64))/32)]';
colormap(genecolormap)
```





Interesting biology: AA's like to repeat. Is this AA chemistry or genomic stuttering? And what's going on among S, E, P, and K?

Is there more we can say about this picture information theoretically?

So far, we have the monographic entropy (H = 4.1908 bits) and the digraph entropy (H = 8.3542 bits).

But the digraph entropy is flattened – doesn't know about rows and columns:

$$H(x,y) = -\sum_{i,j} p_{ij} \ln p_{ij}$$

Let's try to capture something with more structure. The conditional entropy is the expected (average) entropy of the second character, *given* the first:

$$H(y|x) = \underbrace{-\sum_i p_{i\cdot}}_{\text{expectation over rows}} \underbrace{\sum_j \frac{p_{ij}}{p_{i\cdot}} \ln \frac{p_{ij}}{p_{i\cdot}}}_{\text{entropy of one row}} = -\sum_{i,j} p_{ij} \ln \frac{p_{ij}}{p_{i\cdot}}$$

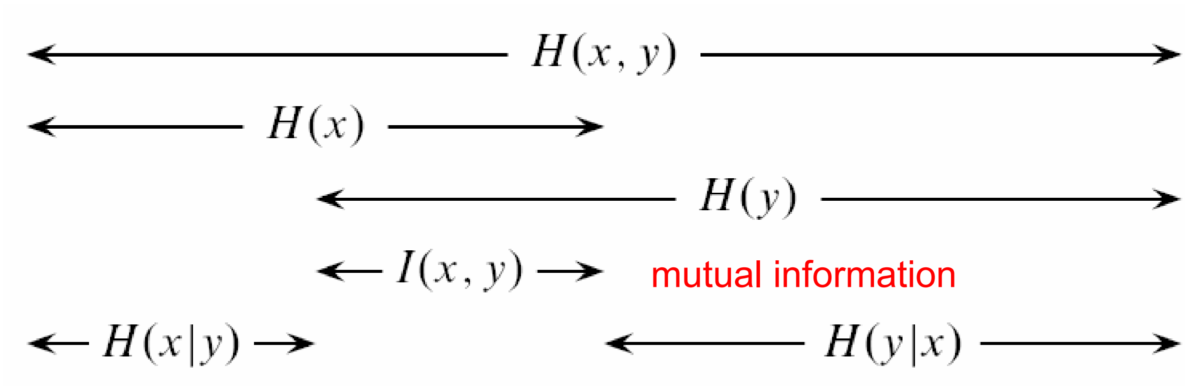
$$= H(x,y) + \sum_i \Big(\sum_j p_{ij}\Big) \ln p_{i\cdot}$$

$$= H(x,y) - H(x)$$

4.1642 bits

So the conditional entropy follows directly from the monographic and digraphic entropies!

In fact there are a bunch of relations, all easy to prove:



$$H(x) - H(x|y) = H(y) - H(y|x) \equiv I(x,y)$$

0.0266 bits

$$I(x,y) = \sum_{i,j} p_{ij} \ln\left(\frac{p_{ij}}{p_{i\cdot}p_{\cdot j}}\right)$$

Proof that mutual information always positive:

$$\begin{aligned} H(y|x) - H(y) &= -\sum_{i,j} p_{ij} \ln \frac{p_{ij}/p_{i\cdot}}{p_{\cdot j}} \\ &= \sum_{i,j} p_{ij} \ln \frac{p_{\cdot j}\,p_{i\cdot}}{p_{ij}} \\ &\le \sum_{i,j} p_{ij} \left(\frac{p_{\cdot j}\,p_{i\cdot}}{p_{ij}} - 1\right) \\ &= \sum_{i,j} p_{i\cdot}p_{\cdot j} - \sum_{i,j} p_{ij} \\ &= 1 - 1 = 0 \end{aligned}$$

Mutual information measures the amount of dependency between two R.V.'s: Given the value of one, how much (measured in bits) do we know about the other.

You might wonder if a quantity as small as 2.7 centibits is ever important. The answer is yes: It is a signal that you could start to detect in 1/.027 ~ 40 characters, and easily detect in ~100.

Mutual information has an interesting interpretation in game theory (or betting)

<u>side information</u>:
Outcome i with probability p_i is what you can bet on at odds $1/p_i$
But you also know the value of another feature j that is partially informative
In other words, you know the matrix p_{ij}
and it's neither diagonal (perfect prediction) nor rank-one (complete independence)

example: i is which horse is running, j is which jockey is riding

What is your best betting strategy?

b_{ij} fraction of assets you bet on i when the side info is j

$$\sum_i b_{ij} = 1, \qquad 0 \le j \le J-1$$

maximize the return on assets per play:

$$W = \left\langle \ln \frac{b_{ij}}{p_{i\cdot}} \right\rangle = \sum_{i,j} p_{ij} \ln \frac{b_{ij}}{p_{i\cdot}}$$

old nag is ridden by:	some unknown	Willie Shoemaker	
Secretariat	0.94	0.01	0.95
old nag	0.04	0.01	0.05
	0.98	0.02	

we can do this by Lagrange multipliers, maximizing the Lagrangian

$$\mathcal{L} = \sum_{i,j} p_{ij} \ln \frac{b_{ij}}{p_{i\cdot}} - \sum_j \lambda_j \left(\sum_i b_{ij} - 1 \right)$$

$$\mathcal{L} = \sum_{i,j} p_{ij} \ln \frac{b_{ij}}{p_{i\cdot}} - \sum_j \lambda_j \Big(\sum_i b_{ij} - 1 \Big)$$

$$0 = \frac{\partial \mathcal{L}}{\partial b_{ij}} = \frac{p_{ij}}{b_{ij}} - \lambda_j$$

$$b_{ij} = \frac{p_{ij}}{\lambda_j} = \frac{p_{ij}}{p_{\cdot j}}$$

This is the famous "proportional betting" formula or "Kelly's formula", first derived by Kelly, a colleague of Shannon, in 1956. You should bet in linear proportion to the probabilities conditioned on any side information.

$$W = \sum_{i,j} p_{ij} \ln \left(\frac{p_{ij}}{p_{i\cdot} p_{\cdot j}} \right) = I(x, y)$$

So your expected gain is the mutual information between the outcome and your side information!

So, e.g., 0.1 nats of mutual information means ≈10% return on capital <u>for each race</u>. You can get rich quickly with that!

Finally, the Kullback-Leibler distance is an information theoretic measure of how different are two distributions ("distance" from one to the other).

A.k.a. "relative entropy".

$$D(\mathbf{p}\|\mathbf{q}) \equiv \sum_i p_i \ln \frac{p_i}{q_i}$$

Notice that it's not symmetric. It also doesn't have a triangle inequality. So it's not a metric in the mathematical sense.

But at least it's always positive!

$$-D(\mathbf{p}\|\mathbf{q}) = \sum_i p_i \ln\left(\frac{q_i}{p_i}\right) \le \sum_i p_i \left(\frac{q_i}{p_i} - 1\right) = 1 - 1 = 0$$

Interpretations:

1. It's the extra length needed to compress **p** with a code designed for **q**

$$-\sum_i p_i \ln q_i = H(\mathbf{p}) + \sum_i p_i \ln \frac{p_i}{q_i} \equiv H(\mathbf{p}) + D(\mathbf{p}\|\mathbf{q})$$

2. It's the average log odds (per character) of rejecting the (false) hypothesis that you are seeing q when you are (actually) seeing p

$$\mathcal{L} = \frac{p(\text{Data}|\mathbf{p})}{p(\text{Data}|\mathbf{q})} = \prod_{\text{data}} \frac{p_i}{q_i}$$

3. It's your expected capital gain when you can estimate the odds of a fair game better than the person offering (fair) odds, and when you bet by Kelly's formula

$$W = \langle \ln(b_i o_i) \rangle = \sum_i p_i \ln(b_i o_i)$$

$$b_i = q_i$$

$$o_i = 1/r_i$$

so

$$W = \langle \ln(b_i o_i) \rangle = \sum_i p_i \ln \frac{q_i}{r_i} = D(\mathbf{p} \| \mathbf{r}) - D(\mathbf{p} \| \mathbf{q})$$

Turns out that if the house keeps a fraction $(1 - f)$, the requirement is

$$D(\mathbf{p} \| \mathbf{r}) - D(\mathbf{p} \| \mathbf{q}) > -\ln f$$

Betting is a competition between you and the bookie on who can more accurately estimate the true odds, as measured by Kullback-Leibler distance.