Problem Set 3

1. Population Genetics (22 pts)

I) **Mutations** (4 pts): define the following terms having to do with mutations in 15 words or less:
1 point per definition. The definition are <15 words each, so it's rather loose.

- Deletion

Loss of a segment of DNA from a chromosome.

- Nucleotide transversion

Nucleotide substitution mutation where a purine is replaced with a pyrimidine, or vice versa.

- Missense mutation

Mutation in DNA coding sequence that causes an amino acid substitution in the resulting polypetide.

- Frameshift mutation

Mutation caused by insertions or deletions resulting a shift in reading frame of subsequent codons.

II) **Mutagenesis** (2 pts): what is a mutagen (in ten words or less)? Name two types of mutagens and briefly (in 20 words or less) explain how they cause genetic mutations.

1 point for each type of mutagen (agent capable of increasing the rate of mutations). There can be many examples of the individual types, so use your judgment to accept them or not.

- **Base analogues—like 5-bromouracil, an analog of thymine that is incorporated in thymine's place and pairs preferentially with guanine. Simply—molecule that is incorporated into DNA in place of a normal base.**
- **Nucleotide analogues—like base analogues, can be incorporated in place of a normal nucleotide. Examples are ddNTPs, 3'-Azido-2',3'-deoxythymidine (AZT), etc.**
- **Chemical agents—can react with DNA and change the hydrogen-bonding properties of the bases. Examples: nitrous acid (HNO_2 deaminates A, C, and G) and alkylating agents like EMS(ethyl methanesulfonate).**
- **Intercalating agents—can insert between adjacent base pairs of DNA and cause misalignment between template and daughter strands of DNA. Examples—EtBr, proflavin.**
- **Ultraviolet radiation—can cause pyrimidine dimers that block transcription and lead to inaccuracy of DNA replication.**
- **Ionizing radiation—large doses increases overall rate of mutations.**

III) **Gene pool** (6 pts): a transversion in the second codon position for the sixth amino acid in the β-globin chain of hemoglobin is the recessive mutation responsible for sickle cell anemia. When the mutation is homozygous, it is lethal. However, people heterozygous for the sickle cell allele are protected from infection by the protozoan *Plasmodium falciparum*, which causes malaria.

a) Define the terms "allele fixation" and "heterozygote superiority" in less than 20 words. Relate them to this case—in areas where malaria is a threat, would either allele become fixed?

2 points total, 1 for the definitions and 1 for the case specific question.

- **Allele fixation: when an allele's frequency in a population reaches 1.**
- **Heterozygote superiority: when the fitness(viability and/or fertility) of the heterozygote is greater than that of both the homozygotes.**

In the case of heterozygote superiority, fixation doesn't occur because it's beneficial to keep both alleles in the gene pool.

b) Let HbS^+ denote the dominant allele and HbS^S the sickle cell allele. If in a certain population we find the following genotypic breakdown:

Number of HbS^+/HbS^+ individuals = 3915
Number of HbS^+/HbS^S individuals = 585
Number of HbS^S/HbS^S individuals = 0

What are the genotypic and allelic frequencies of the population in question?
4 points total, 2 points per answer.

Genotypic frequencies: $HbS^+/HbS^+ = 0.87$, $HbS^+/HbS^S = 0.13$, $HbS^S/HbS^S = 0.00$

Allelic frequencies: $HbS^+ = \frac{3915\times 2+585}{9000} = 0.935$, $HbS^S = (585/9000) = 0.065$

IV) **Hardy-Weinberg** (10 pts): a 32 bp deletion in the gene coding for the human chemokine receptor CCR5, termed Δ32, is found to offer some AIDS resistance. The mutation of the membrane bound receptor hinders HIV infection of T cells. Here let's denote the normal CCR5 allele as "A" and the Δ32 allele as "a." Let's say a 1000 person population was genotyped and we obtain 795 AA, 190 Aa, and 15 aa.

a) There are around half a dozens of assumptions upon which the Hardy-Weinberg principle depends in order to have predictive value. Name three of these.

5 points total here, grade on the quality of combined answers.

- **Mating is random; there are no subpopulations that differ in allele frequency.**
- **Allele frequencies are the same in males and females**
- **All the genotypes are equal in viability and fertility (selection does not operate).**
- **Mutation does not occur.**
- **Migration into the population is absent.**
- **The population is sufficiently large that the frequencies of alleles do not change from generation to generation merely because of chance (no drift).**

b) Use the χ^2 test to determine whether the frequencies provided agree with those predicted by the Hardy-Weinberg law.

5 points here, 2 points for the χ^2 value, 2 point of the p-value, 1 point for explanation of significance, partial credit if they show work.

The allele frequencies are A=0.89, a=0.11. So using the HW principle, the predicted genotype frequencies should be AA = $(0.89)^2$ = 0.7921, Aa = 2(0.89)(0.11) = 0.1958, aa =

$(0.11)^2$ = 0.0121. Out of 100 people then you would expect 792.1 AA, 195.8 Aa, and 12.1 aa.

$$\div^2 = \frac{(795-792.1)^2}{792.1} + \frac{(190-195.8)^2}{195.8} + \frac{(15-12.1)^2}{12.1} = 0.877$$

The degrees of freedom is 2, so the p-value is 0.644. Hence, it is not a significant deviation from what's predicated by the HW principle.

2. Monte Carlo Genetic Drift Simulation with Mathematica (10 pts)

Consider a population of sexually-reproducing organisms with an allele A that has an initial frequency of 50%. In the absence of any randomness or other factors, the frequency would remain constant in successive generations. Here, you'll use Mathematica to simulate what happens in a small population of organisms when gene inheritance is random.

In each successive generation, each individual offspring will inherit the A allele if Mathematica's Random[] function returns less than or equal to the frequency of the previous generation. Keep track of how the frequency changes from generation to generation, and output your results with the ListPlot[] function, graphing frequency versus time (in generations). Stop iterating when the frequency of A reaches 0 or 1.

Start with an initial population of 20 organisms. Run your experiment three times, and attach your output. Try modifying the program to use much larger numbers of organisms – what do you observe then?

Provide both your code as well as output with your answers.

8 pts given for a complete Mathematica notebook. 2 pts for observing that the number of generations tends to *increases* as the number of organisms is increased. If the notebook didn't work but the student understands the overall concept, give partial credit.

The following solution is based on a set of Mathematica population genetics algorithms from http://fig.cox.miami.edu/Faculty/Tom/bil358b/popgen.html.

```
Clear[freq];
Clear lessThanFreqA;
organisms = 20;

lessThanFreqA[frequencyA_] :=
    Sum[If[Random[] < frequencyA, 1, 0],
        {i, 1, organisms}]/organisms;

freq[i_] := freq[i] = lessThanFreqA[freq[i - 1]];
freq[0] = .5;

freqlist = {};
generations = 0;
For[i = 0, freq[i] < 1 && freq[i] > 0, ++i,
      AppendTo[freqlist, freq[i]];
      ++generations];

ListPlot[freqlist,
  PlotRange -> {{0, generations}, {0, 1}},
  PlotJoined -> True]
```

3. Genome Sequencing (6 pts)

a) List the major steps involved in modern chain termination sequencing (2 pts)

The chain termination sequencing method, sometimes called the Sanger method, allows a sequence to be read by encoding sequence information as DNA size which can be easily read out by gel electrophoresis. For example, to determine the position of all A's in sequence one carries out an enzymatic reaction in which DNA is synthesized in the presence of both regular A's (deoxy-ATP, or dATP) and "chain terminator" A's (dideoxy-ATP, or ddATP). The latter stops the reaction ONLY at A positions. Because we force the reaction to always begin at the same position (by using a short DNA primer) the result is a collection of DNA fragments which have sizes corresponding to the position of A in the chain. By carrying out the analgous reaction for the other 3 bases we can deduce the complete sequence.

In the original sequencing technique, the DNA fragments were labeled radioactively and the resulting fragments were separated on a slab polyacrylimide gel, with one "letter" reaction per lane. The gel was then visualized by exposing it to film in a process known as autoradiography and the film was read off as sequence by eye. In the current ABI high-throughput sequencers, DNA fragments are separated by capillary electrophoresis in which a "gel" is present in a tube and can be automatically flushed out and "repoured" after a run is complete. (Pouring slab gels is a laborious and hard-to-automate process.) All 4 bases can be separated in a single gel run because each chain terminator has a unique fluorescent probe which is read out as a fluorescent peak as it comes out of the capillary tube. By knowing just the size of the DNA fragment and the "color" of the chain terminator we can deduce the sequence.

b) Explain what is meant by shotgun sequence assembly (2 pts)

Shotgun assembly is when you build up a master sequence directly from the short sequences obtained from individual sequencing experiments, simply by examining in the sequences for overlaps. It does not require any prior knowledge of the genome and so can be carried out in the absence of a genetic or physical map. [Brown, 1999]

c) Is shotgun sequencing always the best choice for eukaryotic organisms? Why or why not? (1 pt)

Shotgun sequencing relies on the ability of computers to piece together, or "assemble", short sequencing reads (300-500 bp) into a complete genome sequence by matching overlapping sequences. This bypasses the need for a detailed and laborious mapping effort. Sequence assembly algorithms are very computationally intensive when applied to genome sequences but are made possible by the power of modern computing. Assembling larger genomes (e.g. 3 Gbp, the size of the human genome) is now possible with state-of-the-art computers. Also, the ability to successfully assemble the sequence without mapping is heavily dependent on the amount of repetitive sequence. The more repetitive the genome sequence, the less likely it is that it can be successfully assembled after shotgun sequncing without additional, independent information. Thus, shotgun sequencing is used for microbes because they tend to have relatively small genomes and a small amount of repetitive sequence.

In the case of the human genome, the large size, but more crucially the tremendous number of repetitive elements (>50% of the sequence), created a fierce and continuing debate about whether it can be (or was) assembled from shotgun sequencing alone.

d) As part of an epidemiological study group, you have been asked to sequence the genome of Salmonella typhi, the causative agent in typhoid fever. What method would you use, and why? (1 pt)

See the above answer. You would want to use shotgun sequence assembly.

4. Sequence Analysis (32 pts)

Download the *S. typhi* genome from: ftp://ftp.sanger.ac.uk/pub/pathogens/st/St.dna

Write a perl program which identifies all possible open reading frames (ORFs). For the sake of this exercise, an ORF starts with an ATG (which is the furthest 5', or "left-most", if there are many). The ORF ends when hitting a stop codon (TAA, TGA, TAG). Design your program to search both strands. Include your well-annotated code at the end of your problem set. **Hint:** You may find code from the two previous problem sets useful for this task.

```
#!/usr/local/bin/perl

#-------------------------------------------------------------------------------
# Program:   ORF_Analyzer.pl
# Author:    Jon Radoff (jradoff@charter.net)
# Date:      October 15, 2002
#
# This program predicts ORFs in an input file containing genome data and
# groups them by size.
#-------------------------------------------------------------------------------

#-------------------------------------------------------------------------------
# Subroutine:  Reverse_Complement( $DNA_String )
#
# Produce the antisense version of $string by copying it into $anti
# and then replacing each nucleotide with the appropriate antisense
# nucleotide, using a case sensitive search and replacing with a
# lowercase character to inhibit re-replacing certain nucleotides.
#
# (Note that a this is considerably faster than switching to an intermediate
# set of symbols, e.g., numerals, and then reconverting back to ACGT)
#-------------------------------------------------------------------------------

sub Reverse_Complement
{

       my $anti   = uc $_[0];
       $anti =~ tr/ATGC/TACG/;
       $anti = reverse( uc $anti) ;

       return $anti;
}

#-------------------------------------------------------------------------------
# Subroutine:  Translate_DNA_Codon( $DNA_codon )
#
# Returns the single-character amino acid code for the DNA codon provided.
# Note that this translates DNA sequences containing T's (and -not- RNA
# sequences containing U's).  While not directly reflecting the biological
# process, it is more computationally efficient to skip the transcription
# phase.
#-------------------------------------------------------------------------------

sub Translate_DNA_Codon {
       if ($_[0] =~ /GC./i) { return A; }
       elsif ($_[0] =~ /TGC|TGT/i) { return C; }
       elsif ($_[0] =~ /GAC|GAT/i) {return D; }
       elsif ($_[0] =~ /GAA|GAG/i) {return E;}
       elsif ($_[0] =~ /TTC|TTT/i) {return F;}
       elsif ($_[0] =~ /GG./i) {return G;}
       elsif ($_[0] =~ /CAC|CAT/i) {return H;}
       elsif ($_[0] =~ /ATA|ATC|ATT/i) {return I;}
       elsif ($_[0] =~ /AAA|AAG/i) {return K;}
       elsif ($_[0] =~ /TTA|TTG|CT./i) {return L;}
       elsif ($_[0] =~ /ATG/i) {return M;}
       elsif ($_[0] =~ /AAC|AAT/i) {return N;}
       elsif ($_[0] =~ /CC./i) {return P;}
       elsif ($_[0] =~ /CAA|CAG/i) {return Q;}
       elsif ($_[0] =~ /AGA|AGG|CG./i) {return R;}
       elsif ($_[0] =~ /AGC|AGT|TC./i) {return S;}
       elsif ($_[0] =~ /AC./i) {return T;}
       elsif ($_[0] =~ /GT./i) {return V;}
       elsif ($_[0] =~ /TGG/i) {return W;}
       elsif ($_[0] =~ /TAC|TAT/i) {return Y;}
```

```
	elsif ($_[0] =~ /TAA|TGA|TAG/i) {return "Z";} # Stop

	return "X"; # undetermined
}

#-------------------------------------------------------------------------
# Subroutine:  Translate_DNA( $DNA_codon, $Position )
#
# Steps through a DNA sequence, starting from a given position, and
# translates codons into the appropriate amino acid.  The resulting
# amino acid sequence is returned.
#-------------------------------------------------------------------------

sub Translate_DNA {

	my $DNA_Sequence = $_[0];
	my $Position = $_[1];
	my $Amino_Acid_Sequence;

	while (substr $DNA_Sequence,$Position,3) {
		$codon = (substr $DNA_Sequence,$Position,3);
		$Amino_Acid_Sequence .= Translate_DNA_Codon($codon);
		$Position = $Position + 3;
	}

	return $Amino_Acid_Sequence;

}


#-------------------------------------------------------------------------
#
# Main program
#
#-------------------------------------------------------------------------

# First, we slurp the entire input file into a string called $text, remove
# the data contained before the ">" and newline character, and remove
# all the newline characters.
undef $/;
$text = <>;
$text =~ s/\>.+?\n//;
$text =~ s/\n//g;
$text = uc $text;

$Strand = 1;

$Min_Length = 75;    # Specifies the smallest ORF we want to look for
$Max_Length = 100;   # Specifies the largest ORF we want to look for

print "\nPredicting ORFs between $Min_Length and $Max_Length\n\n";

# Loop for both strands:
for ($Strand=1; $Strand <= 2; $Strand++) {
	print "\nComputing strand #$Strand:\n";
	$ORF_Count = 0;
	if($Strand==2) {
		$text = Reverse_Complement($text);
	}
	# Loop for three reading frames:
	for ($Pos = 0; $Pos < 3; $Pos++) {

		print "\nTranslating frame position $Pos:\n";

		$Amino = Translate_DNA($text,$Pos);
		$Last_Pos = 0;

		# Now that we have the complete amino acid sequence, we
		# just need extract the ORFs.  Since we define an
		# ORF as any occurence of ATG (equivalent to Methionine, M)
		# we just need to extract all the strings that span M to Z.
```

```
                @ORF = ($Amino =~ /M.*?Z/gi);

                foreach $ORF (@ORF) {
                        # Make sure this ORF is of the size we care about
                        $ORF =~ s/Z//i; # Remove the stop codon
                        $Len = length $ORF;
                        if( ($Len <= $Max_Length) && ($Len >=$Min_Length) )
                        {
                                $ORF_Count++;
                                # Calculate position in the genome.  This is
                                # simply the ORF's location in the overall
                                # amino acid sequence multiplied by 3, plus
                                # the frame offset
                                $Amino_Pos = index($Amino,$ORF,$Last_Pos);
                                  $Last_Pos = $Amino_Pos;
                                $Genome_Position = $Pos + ($Amino_Pos*3);
                                print "$Genome_Position ORF=$ORF\n";
                        }
                }

        }

        $ORF_OnStrand[$Strand] = $ORF_OnStrand[$Strand] + $ORF_Count;
        $Total_ORF = $Total_ORF + $ORF_Count;

}

print "\nSense strand = $ORF_OnStrand[1] ORFs\n";
print "Antisense strand = $ORF_OnStrand[2] ORFs\n";
print "Total = $Total_ORF ORFs\n";
```

a) Finding small ORFs (8 pts): Most gene-finding programs do not predict genes that are shorter than 100 amino acids. These small, potential genes are not included in most collections of predicted genes. How many such ORFs did your program find that code for proteins between 75 and 100 amino acids (including 75 and 100)? How many were found on each strand of the genome?

2799 total: 1398 on the sense strand, 1401 on the antisense.

b) Checking small ORFs (10 pts): Use BLASTp (http://crobar.med.harvard.edu or http://www2.ebi.ac.uk/blast2/) to compare the first 3 small ORFs (from the first reading frame) that you found to the Swissprot database. For each of these 3 ORFs, list its genome position, amino acid sequence, and the top BLAST match in each case? (Please use single letter annotation for the amino acids.) Are they statistically significant?

Genome Position	Amino Acid Sequence	Top BLAST match	E-Value
13770	MRKNAQPTISMVTPRLNKAGWAADLAA ALMAALISVISLVTFLAISLAAGVVANVR RVGLICVITWISPWKKRCVA	Hypothetical protein lmo2357	0.056
69705	MKIIKDALAGTLESSDVMIRIGPSSEPGIR LELESLVKQQFGAAIEQVVRETLAKLGVE RALVSVDDKGALECILRARVQAAALRAA EQTEIQWSAL	Putative citrate lyase acyl carrier protein (Gamma chain).	2.5e-43
84786	MVLYAGRLVYSSVPYWPKPERWAMTSF GLCSHSTSSVIPRRAKAGPLLASMKISVVS ISFRNASRPSSLKIFSDRACRLRRESCG	Hypothetical protein APE2145	0.17

The only statistically significant result is the sequence beginning at 69705, which has a very low E-Value (2.5e-43).

c) G+C content and ORFs (8 pts): Download the *E. coli* genome (if you haven't already) from http://www.courses.fas.harvard.edu/~bphys101/problemsets/Ecoli_K12.txt. What are the G+C contents of the *S. typhi* and the *E. coli* genomes? Run your program on the *E.coli* genome. How many small (75-100 aa) ORFs do you find per kb of genome sequence for each of these genomes? How does the average ORF size one expects to find at random change as a function of G+C content? Why?

Genome	G+C content	Total small ORFs	Small ORFs/kb
S. typhi	52.05%	2799	2799/4809037=5.82e-4
E. coli	50.79%	2599	2599/4639221=5.60e-4

Since stop codons include TAA, TGA, TAG, C's and G's are underrepresented relative to T and A. A similar situation exists with the start codon ATG. At random, one would expect to find that ORFs are longer on average for genomes with high G+C content.

d) Optimal oligo design (6 pts): You decide to design an oligonucleotide microarray based on the sequences of predicted ORFs greater than 100 amino acids. Name at least 3 criteria that you would use to select sequences to be used as probes on the array.

Criteria for sequence selection:

- *Make sure the probe is specific enough so that it will only hybridize to the intended target*
- *Consider melting temperature based on the sequence, and group sequences into separate experiments based on uniform melting temperatures if necessary*
- *Consider secondary structure of the oligonucleotides and avoid sequences that fold back on themselves, form helixes, etc.*

5. DNA Microarrays (30 pts)

S. cerevisiae, or baker's yeast, is commonly studied using microarrays. Yeast has a powerful genetic system and its 6,220+ ORFs are sequenced and well characterized, hence it is a great candidate for whole-genome profiling using DNA microarrays.

I) **Chip construction** (6 pts): after the amplification of DNA from specifically prepared libraries of ORFs, the amplified DNA is arrayed(printed) on slides to create microarray chips used in further experimentation.

a) Why are the slides usually coated with compounds like polylysine before DNA is printed onto them? (Hint: polylysine coating gives the slide an overall positive charge)

***2 points*: DNA has an overall negative charge, so the coating holds the DNA on the slide by electrostatic interactions.**

b) A microarray containing the entire yeast genome has 6,220 spots of DNA. If the printed area of the slides are only 20mm by 20mm, and there needs to be around 100µm of space between the edges of each spot, then what's the expected average diameter of each spot?

***4 points*: so there must be ~80 spots on one side, 20mm / 80 ~= 250ìm, with the spacing then the expected diameter is ~150ìm. Accept anything in this vicinity. Actual values go anywhere from 100ìm to 200ìm**

II) **From RNA purification to Hybridization** (6 pts): the next step of the microarray experiment involves harvesting different populations of cells and purifying their RNA, which is then reverse-transcribed into cDNA. The cDNA probes from different populations are then purified and labeled (in various ways) with different fluorochromes (Cy3/Cy5). The probes are then applied to the slide for hybridization to occur.

a) While certain researchers extract total RNA from cells to study, others like to extract only messenger RNA. In 20 words or less, why might solely extracting messenger RNA be preferred?

***2 points*: there is also tRNA and ribosomal RNA. By isolating mRNA you only look at what's transcribed and hence more closely associated with gene expression.**

b) Why are poly-thymine compounds of 12-17 bp often used to isolate mRNA? (Hint: think of a common feature at the 3' end of eukaryotic mRNA)

***2 points*: eukaryotic mRNA transcripts are polyadenylated by poly(A) polymerase post-transcriptionally; adding a "poly-A" tail of up to 200 residues. Complementarity to the poly-thymine compounds will sequester mRNA specifically.**

c) Why is it important to use RT(reverse transcriptase) to convert the RNA to cDNA?

***2 points*: this is important because cDNA is much more stable (it will not fold back on itself and form secondary structure) and binds more easily and accurately to the DNA printed on the microarray.**

III) **Data collection/analysis** (6 pts): after hybridization, the slide is run under an automated scanner which detects the fluorescent intensity of the two channels corresponding to cy3 and cy5.

a) Explain what excitation and emission wavelength of fluorochromes are in less than 20 words and find these wavelengths for cy3 and cy5. Why can't rhodamine (another fluorescent dye), with absorption frequency of 570 and emission wavelength of 590, be used as a substitute for Cy5 opposite Cy3?

***1 point (this one is obvious)*: excitation wavelength is the wavelength of light which the fluorochrome absorbs, when exposed to light at this wavelength, the fluorochrome emits another, higher wavelength light.**

***2 points (allow some keeway as these wavelengths don't have to be exact)*: Cy3 absorbs at ~550nm and emits at ~570nm, while Cy5 absorbs at ~649nm and emits at ~670nm.**

***1 points*: rhodamine can't be used opposite Cy3 because the emission wavelength for Cy3 falls in the excitation range for rhodamine. This would cause resonant fluorescence and make measurements inaccurate.**

b) Cy5 labeled RNA is purified from a population of *S. cerevisiae* grown a medium where galactose is the only sugar source. Cy3 labeled RNA is purified from the same strain growing in a medium where glucose is the only sugar source. Equal amounts are made into cDNA and competitively hybridized onto a microarray. What color would you expect spots corresponding to genes associated with the glucose metabolism pathway to appear on the computer?

***2 points*: due to catabolite repression, the Cy3 labeled yeast will express glucose metabolism genes at a much higher rate. Hence the spots will appear yellow-green, the emission of Cy3.**

IV) **Error Analysis** (12 pts): because of the numerous steps involved and the high level of automation in microarray experiments, there are numerous possible sources of error, both random and systematic.

a) Fluorescence tends to stick non-specifically to the surface of the microarray slides around the DNA spots; this causes a "background" fluorescence that can confound the data. You measure the mean and standard deviation of this background from two arrays:

	Array 1	**Array 2**
Average background	800 units	1000 units
s of background	40	30

In which case would 60 units above background be more significant?

***2 points*: even though 60 units seems more significant compared to 800 units than compared to 1000 units, it would actually be more significant in array two, because it's 2 (60/30) standard deviations away from the mean as opposed to 1.5 (60/40).**

b) Contrast random and systematic errors in 20 words or less.

***2 points*: Generally speaking, random errors are due to factors confounding experimental data in a random manner, it's most easily eliminated by repetition. Systematic errors occur systematically to confound measurements across different samples; it is most easily removed by normalization.**

c) For the following errors, indicate whether they are random or systematic and **briefly** explain your rationale.

1 point each, 4 points total.

- Error in RNA purification. **Random, doesn't happen all the time.**
- Cross-hybridization of probes. **Random, the occurrence is closer to a random event, it cannot be attributed systematically.**
- Uneven printing, scanning, or hybridization. **Random, it doesn't have to happen every time.**
- Spatial(sector) bias by the scanner. **Systematic, will produce same error every time.**

d) What are *housekeeping genes*? Why would spotting housekeeping genes in every sector of the slide help normalize for spatial bias of the scanner?

***2 points*: housekeeping genes include essential metabolic enzymes that are present constitutively in low levels in all cells. Because they are always present in a steady amount, fluorescence measurement of these spots can be used to normalize measurements.**

e) Define cross-hybridization in relation to microarrays.

***2 points*: when small regions of sequence similarity found among repetitive elements in gene family cause non-specific hybridization of cDNA probes.**

(2 bonus points) Does the completed sequence of the yeast genome make it possible to limit cross-hybridization at the microarray design step? How can this be done?

Yes because you can design the oligos spotted onto the chip so that they don't include sequences likely to cause cross-hybridization.