Programming Exercise

One of the typical bioinformatics tasks is to find the intersection of a list of genomic features. These might be genes, SNPs, or peaks of some kind. The task is quite simple. Where do the features in list A overlap the features in list B. Each feature is designated by begin and end coordinates as well as a chromosome. There are several file formats for features. Let's use the 3 field BED format (chromosome, begin, end).

```
A 120 130
A 2500 2673
etc
```

Let's build and test a variety of solutions to the feature intersection problem.

- File-based linear search
- Array-based linear search
- Double array-based linear search
- Chromosome-indexed linear search
- Approximate location hashing
- Sorted list binary search
- Double sorted list

Ultimately, we want to create a table that compares the efficiency of various solutions. In the table below, CPU is the sum of user and system time from the unix `time` command. CMP/s is the number of comparisons per second. Mem is the amount of memory used. Coding is approximately how long it took to program and debug the program.

Method	File1	File2	CPU	CMP/s	Mem	Coding
File						
Array						

Feature Generator

Before we start coding solutions, we need to build a program that generates random features. The number of chromosomes, length of the chromosomes, and distance between begin and end coordinates (max, average, or absolute is up to you) should be parameters, as should be the total number of features generated.

Library

To make sure that each program operates similarly, all programs should import the same feature comparison function from a shared library.

```
package FeatureComparison;
sub overlap {
    my ($f1, $f2) = @_;
    etc.
}
1;
```

A feature is simply a hash reference.

```
my $feature = {
    chrom => 'A',
    beg => 100,
    end => 200,
};
```

File-based linear search

Open up one file and grab the first feature. Open up a second file and compare the feature to all other features. Repeat until done.

Array-based linear search

Read one file into an array. Open the other file and compare each feature to the array of features.

Double array-based linear search

Read both files into arrays. You might make a subroutine for that. Compare all features to each other.

Chromosome-indexed linear search

Read both files into hashes where the hash key is the chromosome and the value is a reference to an array of features. Compare all features on the same chromosomes.

Approximate location hashing

This is similar to the above except that not only is the chromosome indexed, but also the approximate location. For example, you could break up a chromosome into 100 segments or into 10kb pieces.

Sorted list binary search

Read a list of features into an array and sort it. Use a binary search to compare features.

Double sorted list

Sort both lists...