Notes on the public genome paper

Overview: For the purposes of Bioinformatics I, our immediate goal in studying the genome paper published in *Nature* is to identify some major problems in computational sequence analysis and to see how they are currently being solved. As we'll see, much of the reported results were found by programs written just for the paper; there's no doubt that being able to write your own programs (or get them quickly written for you) is extremely useful in many aspects of genome analysis. On the other hand, a number of general-purpose programs were critical, and we'll identify some of them.

Generating and evaluating the sequence data (pp. 863-875). Fig. 2 on p. 863 summarizes the approach taken by the public consortium. (Celera by-passed the BAC library and clone mapping steps.) Most of the software used to generate the sequence data is of interest only to the large sequencing centers. The most critical computational problem is assembling the sequence "reads" (roughly 500 bp with a 1% error rate) into an accurate genome sequence. The consortium used the PHRED/PHRAP software package to do this. Assembly of whole-genome shotgun data (e.g. by Celera and with the mouse and rat genome sequences) requires different assembly programs. We won't cover them in Bioinformatics I.

Variation in GC content (pp. 876-877). See Fig. 12, p. 876. Looking at big intervals of the genome, one sees that the percent of G and C nucleotides varies more than can be explained by random fluctuations in a uniform process. The genome average is 42% G+C, but there are regions well above 50%. I don't see any need for general-purpose software here. Mike could write a program to produce Fig. 12 in a few hours. (I'll use the name "Mike" for a hypothetical undergraduate who is a good programmer.)

CpG islands (pp. 877-878). One needs a precise definition of a "CpG island". A typical one goes something like "a region of at least 200 bp with at most 50% G+C where the number of CpG dinucleotides is at least 0.6 of what would be expected by chance". Of course, there are additional subtleties about precisely where a CpG island starts and stops. Fortunately, the list of CpG islands that one finds is not very sensitive to the precise definition. There programs available to find CpG islands, or Mike could write one in a day or two.

Recombination rates (pp. 878-879). I won't go there.

Interspersed repeats (pp. 879-888). An absolutely wonderful analysis, mostly by Arian Smit. Incidentally, the Celera paper says almost nothing about repeats. Look especially at Fig. 18 (p. 881) and Fig. 23 (p. 884). Arian has produced a great (though slow) program for identifying interspersed repeats in (mostly mammalian) genomic sequences. Everyone uses it, including you. Repeats of a given family can be compared to predict a "consensus" ancestral sequence, to which the individual elements are related by substitution, insertion and deletion events over evolutionary time. By comparing the copy with the consensus sequence, RepeatMasker determines a percent divergence, which is proportional to the age of the insertion event. Clearly, a number of special-purpose programs were needed to produce the information reported here.

Simple sequence repeats (pp. 888-889). A few percent of the human genome (more in mouse) consists of tandem repeats of very short sequences (e.g., CACACACA···). RepeatMasker finds some of these, and there are programs and Web servers designed specifically to find tandem repeats. We probably won't say any more about this topic.

Segmental duplication (pp. 889-892). One of the "news flashes" of the genome paper is that about 5% of the human genome consists of long and highly similar (i.e., very recent) duplications. Programs that find such regions can be simple and extremely fast. Two such programs (SSAHA and BLAT) have been published in the last year; might make a good term project for a computer-type student.

Non-coding RNAs (pp. 892-896). A very hot topic in both biology and bioinformatics. Finding these genes in genomic sequence is a completely different computational problem from finding protein-coding genes. This is a great topic for a semester report by a team consisting of a biologist and a computer scientist.

Protein-coding genes (pp. 896-901). There are two main classes of computational tools for gene prediction. First, there are "ab initio" methods, which use only properties of the genomic sequences, such a specific short patterns associated with genes and longer range differences between coding and non-coding DNA (e.g., GC content). The clear favorite tool is GenScan, though many others have been written. Second, the genomic sequence is compared with databases of known DNA and protein sequences; regions similar to a known protein are almost certainly a gene (or pseudo-gene), and moreover the newly discovered gene can be hypothesized to share any known properties of the database sequence. The most popular database searching program is called Blast, with FastA a not-too-distant second. Other programs (the public consortium used Spidey and Celera used Sim4) do a significantly better job at precisely aligning an EST or gene (cDNA) sequence with the genome. Bioinformatics I devotes much time to ab initio gene prediction and database searching.

Proteome analysis (pp. 901-908). Perhaps the most common computation in bioinformatics is to compare a protein sequence with every other known protein, most frequently using Blastp or PSI-Blast (both discussed in Bioinformatics I). Much of "protein space" consists of "domains" that perform some common function and are shared among many proteins. Many groups have made a database of these domains and produced a Web site where a submitted protein sequence is analyzed to find any domains that it might contain. Bioinformatics I will briefly cover this topic.

Comparison of the human and mouse genomes (pp. 909-910). One computational approach for making sense of the human genome is to compare it with the mouse sequence. Naturally, this topic is much more central to the mouse genome project, since the mouse wasn't sequenced when this paper was written (not to mention our species-centricity). The class with cover this subject.