Faster, Better, Cheaper

Biology used to be a hypothesis driven science...

But one of the major themes driving contemporary genomics research is the idea of doing experiments to capture raw data on "complete" state of a particular system.

Examples include the Human Genome Project and the Proteomics project.

Obtaining such massive amounts of data requires increases in automation and parallelism over traditional laboratory experiments.

A variety of laboratory techniques exist to enable biologists to measure the *expression* of a single gene under certain conditions.

Microarray technology enables one to do such experiments on a vastly larger scale.

Using arrays

Suppose you want to do millions of different chemical reactions simultaneously.

You can't do them in single test tube, because how do you know where to look for the results of any given reaction?

To exploit parallelism, you need to be able to associate each intended reaction with a location or address.

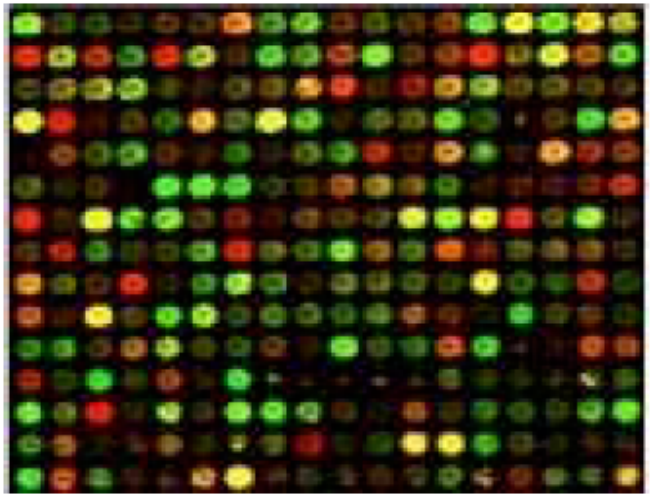
The traditional solution is to do experiments in 96-well or 384-well *plates*, where each chamber is kept separate from each other.

Coordinating the results from many plates involves careful labeling, e.g. with barcodes.

Using arrays

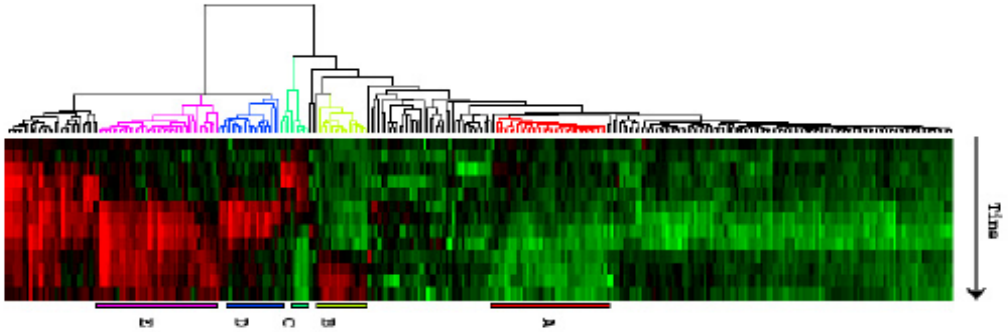
Certain technologies have been developed where different compounds are anchored to tiny *beads*, so reacting beads can be eye-balled, isolated, and identified.

But the best solution is to attach distinct compounds to different regions of a solid substrate so you know *where* they are.



What are arrays good for?

Identification of genes involved in the cell division cycle for yeast:



Sequencing variants of a *known* genome, for detecting single nucleotide polymorphisms (SNPs) or identifying a specific strain of virus (e.g. the Affymetrix HIV-1 array).

Measuring differential expression of all genes in tumor and normal cells, to determine which genes may cause/cure cancer, or identify which treatment a specific tumor should respond best to.

Measuring differential expression of all genes in different tissue types, to determine what makes one cell type different than another.

Microarray technology

Single stranded DNA/RNA molecules are anchored by one end to the plate/substrate. These molecules will seek to hybridize with complementary strands floating in solution.

The target molecules are fluorescently labeled, so that the spots on the *chip/array* where hybridization occurs can be identified.

The strength of the detected signal somewhat reflects the amount of stuff which binds to it, and thus the amount of the target in solution. Such *quantitative* expression data is not very reliable, however.

More accurate data comes from comparing the *relative* amounts of expressed DNA/RNA in two related samples, each of which is colored with a different fluorophore. The ratio of green/red tells us about the relative expression in both samples.

Sources of errors

There are several factors which lead to errors in microarray hybridization data, beyond obvious manufacturing defects and experimental errors.

The strength of the bond formed between two single stranded DNA/RNA molecules is a function of (1) the length of the bonded molecules, (2) the base composition of the molecules, since A/T and C/G bond with different energies, (3) the number and location of base mismatches, since end mismatches cause less trouble.

Cross hybridization is a source of many false positive errors, where a closely related DNA sequence binds at the probe in the absence of the desired target.

Heat breaks these bonds, so the *stringency* of hybridization can be effected by changing the temperature and other conditions.

Self hybridization occurs when probe molecules fold and hybridize with themselves, thus rendering them less effective at hybridizing with the target. This occurs particularly in *self-palindromic* probes.

Spotted microarrays

There are two distinct but important types of DNA/RNA array technology.

Spotted microarrays, pioneered by Pat Brown at Stanford, lay down rows of tiny drops from racks of previously prepared DNA/RNA samples.

Extensive automation makes it possible to build small glass slides containing tens of thousands of different probe spots.

Any given DNA/RNA probe can be hundreds of bases long, and in principle made from any DNA/RNA sample.

Thus reasonable spotted arrays might contain all genes in a given organism, or all EST fragments from a library.

Affymetrix Arrays

Affymetrix arrays are *synthesized* by using the same light mask technology that silicon chips are manufactured.

They exploit *photo-sensitive* reactions to (1) remove a blocking group at the end of a molecule, and (2) to extend the molecule with a given blocked base.

Thus *any* subset of DNA k-mers can be built up using at most $4k$ reactions.

Affymetrix arrays are typically limited to oligos of $k \approx 25$, but can contain hundreds of probes.

It costs about $500,000 to fabricate the masks for a new array design, so they win only if you can make many copies.

Thus their primary target is disease diagnosis by determining the genetic makeup of a patient's tissue.

Other array technologies

Agilent Technologies, a spin-off from HP, is manufacturing array makers using ink-jet printer technology.

These mimic the Affymetrix synthesis approach of growing another base at the end of a molecule by (1) deprotecting the end, and (2) attaching a new base with a protected end to avoid duplication.

These are exciting because it becomes practical to synthesize only one instance of a given array design.

Other companies, such as Lynx Corporation are developing similar technologies on addressable *beads* to get around patent issues concerning arrays of oligonucleotides.

Image processing issues

Each array image contains so much information it must be read by a computer, not a person.

The first step of image processing is *gridding*, identifying the mapping between the image and the locations of the spots on the array.

The careful mechanical construction of these arrays makes this a tractable problem.

Determining the hybridization level of a spot is not just a function of intensity, but a statistical analysis of *control probes* and *redundant* probes for the specific target.

But your data analysis has only begun once you have the hybridization level for each spot. . .

New Application: Linkage Analysis

Many genetic diseases (e.g. Huntington's Disease, Tay-Sachs) are caused by inheriting defective versions (allele) of genes from one or both parents, who in turn inherited them from their parents...

Trying to figure out which genes actually cause the disease is complicated by several factors, including (1) the phenotype may not show up unless you get some combination of bad alleles, (2) about 1/4 of your genes come from each grandparent, so there are many possible candidates and (3) how do you tell which alleles you got from whom?

Through recombination, each chromosome you inherit from Mom is a random mixture of the corresponding chromosomes of her parents.

In this mixture, alternations between parents occurs relatively rarely, so if you inherit given gene from grandma, likely the neighboring gene also came from grandma.

New Application: Linkage Analysis

Linkage analysis mapped diseases to approximate locations on chromosomes bases on (1) pedigree analysis (i.e. family trees annotated with the presence or absence of disease), and (2) experimental data describing which flavor of genetic marker you have at each of a few hundred positions.

This analysis is done through very computationally-intensive linkage analysis programs.

Specially-designed linkage analysis microarrays with hundreds of thousands of markers will enable researchers to gather very accurate measurements of exactly where the crossovers were and what you inherited from each parent.

Microarray software

Affymetrics supplies image analysis and data analysis software for customers of its microarrays.

Several third-party bioinformatics companies (e.g. Sc-analytics, Silicon Genetics, Compugen Lion Bioscience) supply such software applicable for spotted microarray systems.

Data mining software uses clustering and other statistical methods to identify interesting features in microarray and heterogeneous data sets.

Websites such as Netaffx (www.affymetrix.com) provide links between various public databases, enabling you to click on genes that look interesting in your microarray data and find out what is known about them.

Cluster/Treeview

Developed by Michael Eisen, this is the original microarray clustering software.

Provides a computational and graphical environment for analyzing data from DNA micorarray experiments

Cluster organizes and analyzes the data in a number of ways, clustering either genes or experiments by similarity.

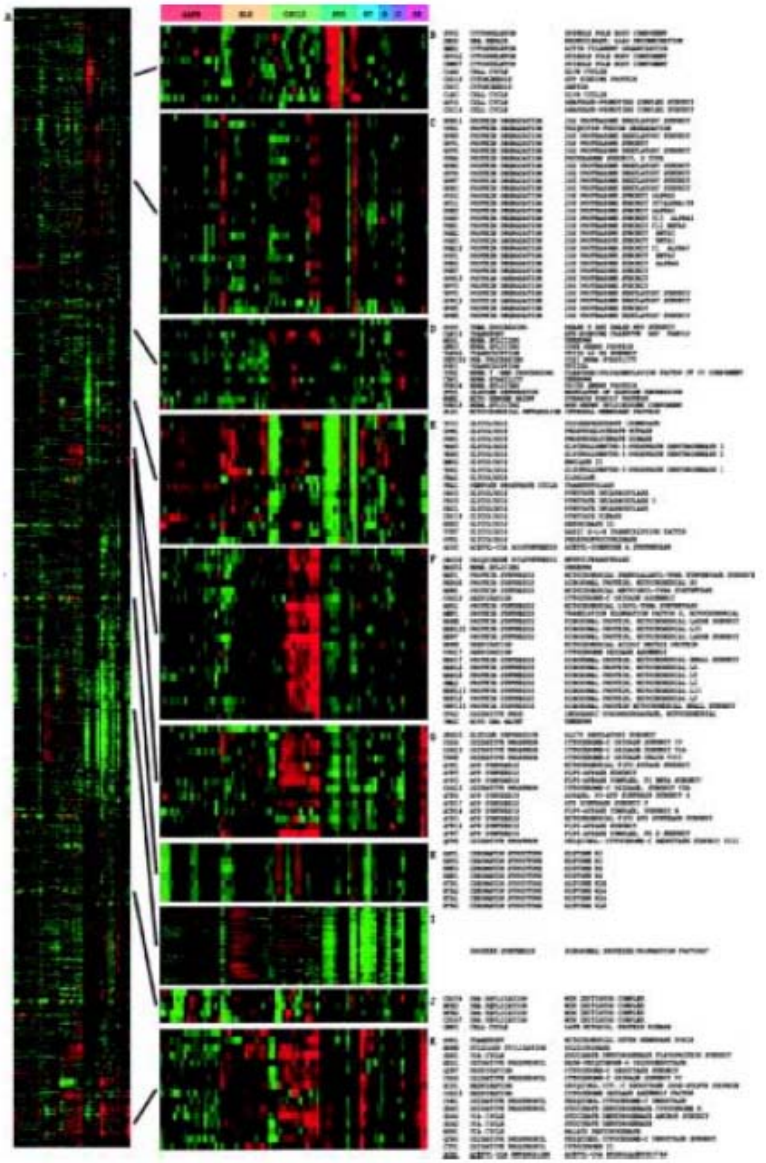
TreeView allows interactive graphical analysis of the results from Cluster.

Cluster uses the correlation coefficient (-1 to 1) of *logarithmically normalized* primary data for each gene. Since this primary data is a ratio of red/green, this is zero for equal expression, positive for up-regulated expression, and negative for down-regulated expression.

$$corr(X,Y) = \frac{\sum_i (x_i - \bar{X})(y_i - \bar{Y})}{\sqrt{\sum_i (x_i - \bar{X})^2}\sqrt{\sum_i (y_i - \bar{Y})^2}}$$

The denominators are the standard deviations of the two sequences, to normalize them. Positive correlation is achieved when both measurements on the same size of their respective means.

Cluster/Treeview



Algorithmics of Cluster/Treeview

This agglomerative clustering algorithm uses the average-linkage method of Sokal and Michener. Each cluster is assigned a gene-expression profile by, for each probe averaging all of the values of genes in its cluster. The profile of a merged cluster is determined by averaging the two component clusters weighted by the number of non-missing values in each cluster.

These clusters are displayed by permuting the rows of the matrix to reflect the structure of the tree/dendogram. A heuristic solution for this TSP-like problem is used to order the genes so as to help visualize the clusters.

Note that there are 2^{n-1} ways to permute the n leaves of any binary tree while leaving a non-intersecting drawing. A polynomial-time dynamic programming can be used to find the best possible ordering:

Let $C[r, i, j]$ be the cost of the cheapest way to permute the subtree with root r such that gene i is on the left and gene j is on the right. Then

$$C[r, i, j] = \min_{k,l} C[child_1, i, k] + C[child_2, l, j] + cost(k, l)$$