PETER PARK: For today, I'll just talk a little bit more generally at the beginning about a few observations that I've had. Perhaps a little bit about reliability microarray studies. I'll talk about classification problem in general. And then, I'll talk more about phenotypes. And then review some literature that are well cited.

So I bet you've had some exposure to that, right? Did anyone-- maybe Marco, or Zack, talk about lymphoma studies? Do you remember any other papers that you've covered? Yeah?

AUDIENCE: [INAUDIBLE]

PETER PARK: Anything else you remember? All right, so--

AUDIENCE: [INAUDIBLE]

PETER PARK: Yeah, right, so maybe this stuff that I have is new. OK, good.

So you probably know all this. It's actually-- so the cDNA arrays are cheaper. So they are used a lot of biology labs. The ones that I work with, for example, in the new building, they have their own libraries. They print their own.

Affy arrays are-- I think the consensus that the Affy arrays are probably the most accurate at this point. But it's still $8 to $1,000 per array.

A lot of groups are experimenting with these printed oligonucleotide arrays. So Agilent has-- so there are just whole variety now-- Agilent has one type of platform where they actually give you the chips, and you have to basically buy their whole setup. And it's cheaper. People seem to like it, but who knows what's going to happen in the future.

I mean, it's all unclear at this point exactly what platform is the best information per dollar spent. But there are other companies that will actually make these oligonucleotides. And then you just buy those and then you print it yourself. So you just have those spray printers and then print those.

I think those will run at this point, maybe a third of the Affy cost. So that's a good option for the future. But it's not clear how good those are. So it's cheaper at this point, but not as good. We are actually trying to print that at the [? Harvard ?] Partner Center ourselves.

And so if we can get that to work-- we're doing it for human and mouse at this point. If we can get that to work, that would be very good for our investigators.

SAGE Is a very accurate way of doing expression data analysis. It basically involves sequencing every little tag. So advantage is that you get a very-- even for a very small copy number of transcripts, you can get them accurately. The downside is that it's expensive. But there are a bunch of groups around here that have large libraries.

So I'm personally interested in the second point, which is, nowadays, called systems biology-- so that is integrating a lot of high throughput technologies in genomics and proteomics, and putting that together.

So microarray is where everything started. Now that all these other technologies are getting more mature, we're all interested in combining all of those. And there are a lot of public and commercial tools, tools, and databases that you should be aware of.

So I think the problem is that even the best biologists don't know what's out there. So the challenge for me and a lot of other people in a similar situation is that you have to, instead of assuming that they know what's out there, you have to tell them what to do. Tell the biologists that, you should try such and such, because they're just unaware of what's going on in mathematics, generally.

So in terms of data integration, initially people did a lot of expression data studies. There have been some nice programs that linked literature. For example, we, at the center, just got a commercial license to something called Ingenuity. You basically put your expression data into the software and it'll generate nice networks.

And you can-- well, there are many ways of being this data. But you can, for example, click on links between genes, and it will bring up all literature that site those genes. I mean, there are lots of programs too that do that, but this particular program does it really well.

So something like that is enormously helpful, if you know that such things exist and if you have access to it. There are some free ones out there, but they're just not as good. I mean, this software, they have tons of people-- not tons, a lot of people, just sitting there, curating literature by hand. And so they've accumulated this huge database.

But I think that's-- eventually, this is, I think, how a lot of us are going to do research. So these tools will get smarter and smarter, so you don't have to go through PubMed, yourself.

There are other types of data. I don't if you've actually covered any of these in class, but there's a lot of protein DNA binding data. So the most famous data set is from the Whitehead, from Rick Young's lab, where for yeast, they basically took all the transcription factors, about a little more than 100 for yeast, and then they found where in the genome these transcription factors actually bind. So in vivo.

So they can't locate-- you don't have enough resolutions to pinpoint exactly what base pair they are binding, but you know which genes are actually using these transcription factors for their regulation. So that data is enormous useful.

I think they're trying to do that for human genome, but I don't know how that's going.

Protein microarrays-- so that has not been as successful as people initially, thought it would be. There are just a lot of difficulties in just manufacturing. But for example, at the center, Harvard Partner Center, there are some contracts with some companies that make these protein arrays with about 50 or 60 molecules sitting there, then I guess these are 50 or 60 really common antibody detection systems.

So I think that could be very helpful. But I think the technology is still not quite there. At least not for the price that's reasonable for most investigators.

Mass spec-- that's an enormous area that has taken off in the past, I'd say, at least in terms of informatics, maybe just past a year or two. There are enormous amounts of data and a lot of informatics challenges. And so another set of data that could be integrated into your studies.

So when people talk about systems biology, people mean different things. But generally, they refer to the fact that you're combining all these different types to generate some coherent picture. And it's actually really hard. And I think some of the reasons will be clear in the next few slides.

OK, so let me just skip that. So just to give you a funny story on the necessity for a good analysis with all this data. There was a well-cited paper in *Nature Genetics* in 2001 called Transcriptional Regulation and Function During the Human Cell Cycle. So I think it actually came from the group that did the yeast cell cycle a couple of years back, prior to that.

And in that paper, they claimed that there are large number of mammalian genes, about 700, to have a cell cycle specific patterns. So I think this was done on Affy data, actually.

Anyway, the next year, there's a paper in PNAS called "Analysis of cell cycle specific gene expression in human cells as determined by microarrays and double diamidine block synchronization. So from the title, you can't really tell what the subject is. But basically what they did in the 2002 paper is go back and reanalyze the data from 2001 paper.

And so, if you look at the text here, the original microarray data presented to support the existence of cyclic gene expression in human cells is now re-examined with a statistical approach, we find that there is internal evidence implying that the original microarray data do not support the proposed patterns of gene expression.

So one of these-- I think it was the first author that's a statistician. So they claimed that to study the cell cycle, you need to synchronize at the beginning, using one of many methods. And they look at this data and say, well after randomizing data, we still see a bunch of genes that have cyclic behavior.

In fact, if you study it carefully, cells are not even synchronized. So they claim that the first experiment was just junk. And the 700 genes that they observed was something that just could have come by chance. Because I mean, because you have so many genes, you're going to see some genes that go in cycles.

So it came out in PNAS. And so since then, at least, there were just exchanges back and forth. So the first group has a website that details this is what we did, this is what the PNAS paper did, and this is why they're wrong. It's just a whole website. And I think the second guys has also had a rebuttal to that website.

So I mean, this is an important thing to settle, right? Because it came out in *Nature Genetics,* a lot of people are using this data, but the other group says, this is all nonsense. And I think-- I mean, so I read a little bit of that. I think what happened was-- I mean, as usual the truth is somewhere in between.

I mean, the first group was so interested in showing that there are lots of genes that are cycling, that they just used methods that will be advantageous to them. And the second paper, they were so interested in showing that the data was wrong, that they defined things in such a way that there would be no cycling genes.

AUDIENCE: [INAUDIBLE] if you found the same genes in another [INAUDIBLE].

PETER PARK: Maybe, I don't know. Yeah. I mean, I bet, depending on how you do it. I would think that they would find some overlap with the first one, but not 700.

So I guess the lesson is that you should make sure you do things well before you publish. At least in such a way that no one can attack you like this. And this is just one example. There are a lot of problems with early studies. And I'll actually tell you about another problem that has been very common.

So one thing that I've been interested in with a student, who actually published a paper on this before anyone else, so there are substantial differences in different technologies for microarray platforms. So cDNAs-- they're very different from Affy arrays. So one of the common questions is, what platform should I use?

If something's cheaper, does it work as well? Do they agree? So which platform is the best?

And so there are a bunch of papers on this, some of which are cited in the article, but I mean, that's just a very small subset. And most of them are conflicting, except that they agree that they do not agree very well. So they're conflicting in terms of which one is better, how accurate are they.

But I think overall, the overlap is surprisingly small. So if you do your experiment with cDNA arrays, find genes that are different with [INAUDIBLE]. So you repeat the experiment with Affy arrays, you get a very different result.

So that's a big problem, right? I mean, for one thing, you're not sure if the results that you get are correct, or not. And another thing is that eventually, we'd like to combine data from different people. So if someone else, in another school, has done very similar experiment, and you want to see how your experiment compares with that, then you don't want to have to repeat their experiment. You'd like to just do your experiment on whatever platform you have and compare it against what someone else has done, which may not be on the same platform.

And so this is a big issue. And it's far away from being settled-- just there are so many issues and problems, so that I think it'll be a while before these things get sorted out.

So as I mentioned, Affy arrays appear to have the most reproducible experiment. I think most people that have tried multiple privatizations will say that they agree with correlation of coin 98, 99. But one thing I've done with Steve Greenberg, whom you've met, I think, couple weeks ago, is that we've studied how reproducible there are between two generations of Affy arrays.

So there is a very popular set of arrays-- the first one was U95-- that's 95 refers to the version of gene database that they use to collect the information about the probes. So that, I think, people used for a couple of years-- lots of chips, I mean, many thousands of chips.

And then, you U133 is the newer one. So U95 actually came in five chips, and U133 came in two chips. So they changed a bunch of things so they were able to fit into two chips. Now, there is just a single chip with the whole human genes. So from five arrays, three or four years ago, now you have a single array.

And so a lot of experiments have been done. These are human arrays.

Just to give you a sense of how many arrays are out there, when I was talking to someone from Aventis pharmaceutical, they have a proprietary database with-- I think they said about I can't remember, it was either 20,000 or 30,000 chips.

AUDIENCE: Of their own?

PETER PARK: Or their own, right. So their internal database. So 20,000, let's say, time's $1,000, that's $20 million, right? That's a lot of money.

So anyway, there's a lot of data using these. Actually, I don't if these arrays are not the one-- they didn't have 20,000 of these arrays. But they have a lot of Affy arrays. These are 20,000 Affy arrays. And that's just one company.

Anyway, so what Steve and I wanted to do is look at how the results would vary depending on which platform you use. And he actually had done the experiment on U95, and then he had a new grant, he had more samples, so he want to do uniformity again, and so he did it on U133.

And it's not because he had so much money that he just wanted to try this. So what he did was he hybridized 14 muscle life samples from with patients with inflammatory myopathies, which is his specialty, then they were hybridized to both chips.

So I looked at this data-- we submitted the paper just recently. But what we looked at was, what happens when we grouped the arrays using hierarchical clustering, which you know by now, and when we find differentially expressed genes in each case.

So you can't read the labels from there, I bet, if you look at the top left one-- so we basically pulled all the data together and said, OK, just can I cluster them? So what you should see is, same sample, hybridized to 95 and 133 should be next to each other, and that's going to be the case for all your samples.

In this clustering, everything on the left is U133, and everything on the right is U95. Yeah. So if you cluster, it's not the sample characteristics, but the array type that is the distinguishing feature.

AUDIENCE: What did he [INAUDIBLE]?

PETER PARK: Well, in this case, we used Pearson correlations. So any linear normalization would not affect the result. But no matter what you try, it's going to turn out this way. And so this is disturbing, right? So there are some ways to fix this. So in this particular case, we normalized the genes in a particular way, and then we were able to actually get this to line up correctly.

But what we did in this case, in general, doesn't work. So we could only do it here. But in general, it doesn't work. So remember, this is one platform, just different generations. So if you were to compare different platforms, I mean, things are just much harder.

And so, that's just an idea of-- that should give you some idea as to how different they are. Another way that we looked at was, from each platform, we looked at-- so the samples were naturally divided into two groups. So we looked at, for each platform, what genes are differentially expressed.

So from the U95 samples, we got a long list, from 133, we got a long list. How much overlap is there? That was the question. Does that make sense?

So if you look at the dotted line, if you look at--

AUDIENCE: [INAUDIBLE]

PETER PARK: Right, so we had 14. And there were basically five of one type, and the rest could be considered as another type. So using those five and nine chips, we looked at the different questions. So we did that separately for each one.

So I mean, they should overlap, right? Exactly same samples. We did a lot of analysis to make sure that there wasn't a lot of degradation of RNA, and such. But if you look at this-- so forget the solid line for now-- if you look at the dotted dashed line, out of top 100, there are about 20%, less than 20% that overlap. So about 20 genes will show up in both, but 80 genes are not in common.

So that's somewhat disturbing, right? I mean, if you look at top 50-- and usually, when investigators do these experiments, they can't really look at more than top 20, 30, 50, it's about 15% that are in common. So 15%, that's like seven or eight genes out of 50. So that's somewhat disturbing.

So for the paper, we actually found some way of fixing this problem in the general case. So we could raise the percentages a lot. But most people who will not go through this trouble of what we did should be aware that their list is not all that robust. And in some sense, this is not surprising. Although the degree to which that they disagree is surprising.

Generally, you have several hundred, or maybe even thousands of genes, that are differentially expressed. And there are different genes expressed at a very similar level. So any perturbation that you make to the data, things are going to just shuffle, and they're generally not very stable.

So it's not surprising that they don't agree very much. But in this case, because it was so tightly controlled, I mean, because they get the same RNA labeled at the same time, they were hybridized in the same facility, we expected this to be a little higher. But it's not.

So this is a challenge, I think. And it should be a caution for people doing experiments.

One thing that-- so one thing that people should pay more attention to, therefore, is not so much what genes are expressed, but what kind of genes are expressed. So people are now thinking of more robust way of looking at the list. I mean, for example, what kind of goal category genes are represented at the top, that kind of thing.

AUDIENCE: [INAUDIBLE]

PETER PARK: Yeah, so that's a good question. So we looked at that in detail. So it turns out that if the probes-- so you all know how Affy arrays work. So if the probes are exactly the same, the sequences are exactly the same. They are very reproducible. So they give very good results. But if they are different, I mean, I don't have pictures here to show you, but no relationship.

Even though they are supposed to detect the same transcript. So basically, I mean, the lesson is that slight changes in the sequence information results in huge changes in your expression.

AUDIENCE: [INAUDIBLE]

PETER PARK: No, the length is the same. The length that-- those are 25 mers. And the reason that they're-- so there is a debate as to what the optimal length is. And so people have-- so the commercial suppliers that I listed earlier, they have different lengths. So there are some 30 mers, 50 mers, a lot of people think that the longer nucleotides, oligonucleotides usually have 70 mers, or that size, are the best. But that's still a bit of an open question.

But Affy would probably like to have longer probes, but they're limited by technology. So they are built by-- it's the same technology that you use to build semiconductor chips, and so you can't tile so many nucleotides.

So what they did-- so the reason that Affy got better, is that this-- it's so hard to figure out exactly which sequences would give you the best result. So basically, what Affy has done over the years is that, throughout the generations, they figured out, just through trial and error, which ones give bad results. And so next time the product comes out, they get rid of all the bad ones and just try new ones.

So by that trial and error process, in my view, they've gotten pretty good. So if you look at two or three years ago, if you look at the old chip, just how much variation there is among the probes that are supposed to detect the same gene, you'd see just things all over the place. It's just maybe half the probes may give you completely different numbers.

But now, it is much better. Things have certainly improved. And I think Affy, the platform itself, has some advantages, because they are using some multiple probes. Other platforms, they don't have the luxury of doing that. So they have a single, longer probe, but they have no idea whether it's correct or not.

And I've looked at, actually, commercial arrays, not Affy, and compared that against Affy, and you just get disturbing results. So you probably have heard a lot of good things about microarrays, and that's probably true overall. But there are some issues that you need to be careful about.

So yeah, I was a little surprised about this. And I guess the recommendation is that if you have different platforms, at least, unless you're going to go through, unless you are able to go through what we went through, not to combine the arrays. But then, that's not a very satisfactory solution.

I mean, if you have a database with all these steps that everyone else has done, what use is it if you can't use them. Other than just each data set in its own state.

So OK, I don't think we need to talk about how many chips we need. I don't if anyone is planning to do array experiments. Your array experiments-- oh, OK.

So generally, I think for most platforms that have been around for a while, most people would agree that biological variability dominates technical measurement errors. So for Affy, you don't want to do many replicates of the same tissue or sample.

In terms of exactly how many you need, how many chips you need, that's a hard question. Only because you don't know how much variability that you have in your samples. So if someone can tell me, I have this much variability in my samples, I want this false positive rate, this false negative rate, I can give you numbers based on some model.

But usually people can't give me those numbers. So there's no point in talking about this. Some, generally speaking, if you have data in cancer data, where you have a lot of variability, I think people would say-- some studies have said, having samples like 10 to 15 per group for a tumor comparison might give you something like 75% false positive rate. And for detecting genes that are three-fold variable.

So yeah, there can be some estimates like that for given parameters. But in general, it's hard. So yeah, let's leave it at that.

I think I'm going to skip most of these. I'm sure you've seen these before.

Let's talk about p-values, multiple testing. OK, so maybe not. So just to give you one example. So suppose you flip a coin 10 times and get all heads. This is like a statistics 101 problem question. What's the probability that you get all heads?

And it turns out it's about 0.001-- it's like 1 of our 2 to the 10th. So yeah, it's about 0.1%. So if the question is, is it biased? You'd say, probably, right? Pretty unlikely.

But what if there are 10,000 people flipping coins, and one person gets 10 heads. So should you be surprised that someone got 10 heads? Probably not, right?

So you need to adjust for this. This is the same problem that I mentioned earlier with the *Nature Genetics* paper. You need to adjust for the fact that there are so many genes. In some sense, people find it somewhat difficult to understand, because I'm interested in one gene, but if my probability depends on what kind of chip I use-- for example, if I use a chip with 10,000 genes, versus a chip with 30,000 genes, my probability will be different for the gene that I'm interested in.

So people find it sometimes hard to understand. But some correction has to be made, I think. So it's often referred to as multiple testing adjustments. I think there is some consensus now on how to do this. So maybe we won't talk about this here.

But people generally-- and I agree with this, as well, there's something called false discovery, that you may have heard of. I think that's a very reasonable approach. Exactly how to calculate that, it's not so simple. But it can be done.

All right, so let's talk a little more about the subject that I'm interested in, in terms of application.

So eventually, for people that are interested in applications, clinical application of microarray, or any other type of high throughput data, is that you like to use this in your clinical setting. That is treatment related things. So the challenge is to stop doing these simple methodological problems at some point and move on to real problems.

And that means incorporating large amount of phenotypic data. So in a clinical setting, typically, you have lots of other types of data listed here. And we like to find relationships between genomic data and phenotype data-- phenotypic data. So questions could be, what genes are variable or correlated with particular phenotype? What should we use as predictors?

So I'm just introducing the framework at a basic level. And then we'll look at some papers.

So the most simple case that people have spent a lot of effort on is the binary case. So the phenotypic data, in this case, is just a label-- zero, one, yes, or no, disease, versus normal. And there are basically every method out there that could be applied to this, has been applied, and there's tons of papers on this-- how to do this better. How do you pick out genes that are related to the label? And how do you do the prediction?

Have guys talked about prediction? Like how to-- do you know leave-one-out cross validation is? Right, so good. And then, people did multiple subclasses, not just two, but multiple cases. And then, in this case, it's non-ordered-- so you have different subtypes.

You could have an ordered subclasses-- that is, if you have a rating for severity of disease, maybe you have one through 5 as your phenotype.

Continuous-- for some reason, this hasn't really-- the last two haven't been done as well as they should have been, or they should be. Yeah, as well as they should be. But it's certainly in an area where a lot of progress needs to be made.

So for example, I did one collaboration where the phenotype was some invasive ability of cells. So what genes predict the ability of these cancer cells? And so how to apply through all this finding genes, and then finding predictions-- prediction algorithms, and then making predictions, how to do that optimally, it's not completely settled.

And finally, the sensor data type is something that I'm very interested in. And I'll talk a little more about that later. So it turns out that a lot of these phenotypes can be reduced to the binary type. So if you have multiple subclasses, and combine them into two. If you have continuous data you can say, well, low invasive versus high, you have severity of disease, you can divide into two.

So you can always turn into a simpler problem. And that's what people have done in many cases. But you lose a lot of information. So you'd like to preserve that and do that-- carry the same procedure out.

So I mean, you probably have seen this before, there are lots of questions that could be asked, and have been asked.

So just to give you a general framework, so because there are so many genes in data, usually there are some type of dimensionality reduction. Even when you have a very good method, you still need to reduce to the data set size. Sometimes unless, you have a nice computer, things may not even fit on your data unless you do some of reduction.

So this problem is called feature selection in computer science. So there are lots of different ways of filtering genes you know threshold value from expression, variation filtering, and so on. There are lots of tools for doing the dimensionality reduction.

So typically, either you reduce dimension by just cutting down the number of genes, or you could actually find some of minor combinations of genes. And then, use those as the reduced dimension. So if you do like principal components, or a single value composition, you reduce the dimension, but the dimensions that you have in the end are not genes. Those are combinations of genes.

So there are different ways of doing this. And with disadvantages and advantages for each one. So I won't go through this, but t-test you probably have heard of.

So let me just spend maybe two or three slides on just one thing that a lot of people have noticed. This problem has been fixed in most cases, say, starting a year or two ago. But if you go back to earlier papers, they've done a lot of suboptimal things, or in this case, just some wrong thing in their study.

So there are a lot of papers on classifying disease categories. And this has been the primary, or one of the primary applications, of Affy technology, or cDNA technology.

Of course, the performance depends on the methodology used. A lot of times you see these machine learning algorithms, like singular-- not singular, support vector machines. Neural networks, they appear to give pretty good results, often.

And because you probably know, if there aren't enough samples, you want to use set training and testing sets to assess your accuracy. If not, use leave-one-out cross-validation.

It turns out that a lot of papers in good journals have made mistakes from prediction accuracy. And what they state are overestimates of the real accuracy. So this wasn't discussed by other people, right?

AUDIENCE: Talked about a paper that [INAUDIBLE].

PETER PARK: Oh, he did? In breast cancer? Right, so let me very briefly then-- so let me just talk about this one simulation study. Did he talk about this? OK, all right, so this is Rick's-- Rex Simon at NIH, he's a statistician who's done a lot of good work. He published this paper-- actually I think the date is-- it may have been 2002. Anyway, *Journal of National Cancer Institute.*

So he did a simulation study just to show you what kind of mistakes could be made. So he generated simulated data-- 20 expression profiles, 10 randomly assigned to one class, the rest of the other.

So in this case, because the data are all random, there's no true underlying differences. And that means whatever prediction you should make, the error should be about half, right? And he averaged this procedure over 2,000 data sets.

So resubstitution here means-- OK, so you have 20 samples. So you built your model, prediction model, based on all 20, so you have your model. And then, you test the model on each one.

So if you do that, 98.2% of the time, you make no error. You always get the labels correct. Cancer or not, or whatever labels you might be in a particular study, that's the label.

And that's clearly a wrong thing to do, right? You can't build your model based on your data, and then predict the data. But what a lot of people have done in good journals is to remove the test sample after gene selection. So I have my 20 samples, and then I filter the genes, and then I pick the genes that are informative of the classes.

And then, to predict the model, I'll leave one out, or to predict the label, I leave one out, build my model, but based on the genes that I've already selected. And then predict the one that's left out. Then you leave another one out, build a new model to the prediction, and so on.

So you think, well, that should be OK, right? But it's OK if the gene set that you get is the same every time you-- if they were the same every time you delete the sample and recalculate. But I mean, as we saw earlier, the gene not very stable. So if you do that, there's a lot of bias.

So if you do that in this simulated data set, 90% of the time, on a random data set, you have 90% accuracy. So generally, the difference is not this big. But it could be substantial, depending on, say, the size of your data set.

Now, if you do it correctly, that is, you remove the data set before selection, before selecting the genes, you reselect the genes and you do the prediction, and so on, then you get median at 11 misclassification. So you should get 10, so you get something close to something close to the same.

Of course, no one really does it exactly the way it should be done. So the way it should be done is, delete one sample, and then start everything from scratch. Like you renormalize, you do everything. But no one really does that. So they just actually normalize using all the data, and then, nowadays, you move on, and then you move one to the gene selection, and so on.

So I think people probably suspected that there was a problem with this, but not as big as it is in certain cases. So nowadays, like if you see a paper in good journals, it will have the supplementary information, they'll talk about what the bias is. But they often report the better rate in the paper. So not good.

OK, so let's talk about survival times now. So how many of you are familiar with my censor data? OK, so I won't really talk about the basics too much then.

I mean, the censoring rate often is in the order of-- it could be like 50%. That's not unusual. So you want to make sure that you take care of your censoring correctly. And generally, we talk about the right censoring-- you have the patient, a study's terminated, [INAUDIBLE] patients die, your patient drops out of a study.

OK, so as n the other phenotypes, the simplest thing you can do is to use standard univariate approach. That is, I look at my genes one at a times, see as this gene relates to survival. So there are many ways of doing that because, I mean, this problem has been around for decades.

And so there are good methods for doing this. For example, log ranked tests will give you some p-value as to how that gene is correlated with your survival data. And so a lot of people will do that, some sort of univariate approach, considering [INAUDIBLE] time, and then they put together at the end, some type of voting method.

So to do a prediction of the patient, you say, well, what does my gene say about this patient. What about gene two? What about 100? They add up. Sometimes weighted differently, depending on how good the gene is in predicting-- or in its correlation to the phenotype.

So as I'll talk a little more later, this is often not optimal. So I mean, it's how people do it because it's easy, but it's not optimal.

So I'll skip that. So let's just do some-- take a look at some examples. So eventually, you like to do your study and then correlate that with your phenotype. And then actually go on to show that whatever-- you don't want to just show that it's related to the phenotype. You want to show that it's better than what people use currently.

So this wasn't done until, let's say, 2001, 2002. So people just said, oh, my expression profiles are related to my phenotype. But it's actually better. So it's actually-- I guess this is an interesting question for debate. I mean, people have published all these papers for two or three years.

But I actually don't know of any hospital that will do the testing and then do prediction on the patient. I mean, there was some report of a plan to do this in Netherlands, but this hasn't really come about.

So I mean, I actually had a good friend in the pathology department who's actually done a lot of arrays himself, and so now that he finished his PhD, he's in a pathology fellowship, he said, OK, maybe I'll try to get this thing to the clinic. And it turns out, there's just a lot of problems.

For one thing, a lot of these studies have not shown that, compared to the cost involved, that it's actually a good deal. So apparently, there are very simple tests for a lot of, say, cancer-related diagnoses. There are very simple tests that are very cheap, and very easy to do. Take very little time.

So he said, it's not clear in the clinic whether those tests are actually less accurate than this. It's much simpler, people already do this. And so there isn't a great incentive to switch to this.

And probably the bigger reason, though, is money. So insurance companies these days are paid per patient-- so I forget the terminology for this. But if the hospital wants to do an extra procedure like this test, basically they don't get reimbursed by the insurance company unless it can be shown to the insurance company that this is a necessary procedure to be done on all these patients.

And that's a difficult thing. So my friend looked into this. And it's not easy-- a hospital, or insurance companies are not willing to shell out money unless it's proven so in a reasonably good way. And of course, if I were to, for example, show them the earlier results on how [? useful ?] these are, no one's going to, say, pay $1,000 per patient.

So I don't know at what point this is going to really be done in the clinic. But I think as more people publish better papers, and do a more comprehensive cost benefit analysis, maybe it will happen.

I think initially, people are very excited that this is what you're going to do, right? You walk in the clinic, they do this, and give you all these diagnoses. But who knows when this is going to happen.

So yeah, those are some practical issues that are not so easy to resolve. In any case, this is a paper that came out in *Nature.* This is, I think, a group from the Netherlands, where I think they are more-- at least from a couple articles that I read, they are more closer to actually doing this in the clinic.

And one article actually cited the results of this paper as a basis for doing this in clinic. So one problem is that I don't think what was done in this paper is that great. So it's a bit of an issue. But at least in this paper, they claim that this gene expression profile will outperform all currently used clinical parameters in predicting disease outcome in breast cancer.

So is anyone familiar with this paper?

AUDIENCE: [INAUDIBLE]

PETER PARK: All right, so did he talk about the methodology, as well?

AUDIENCE: [INAUDIBLE]

PETER PARK: Oh, OK. I must have forgotten. Did he talk about this briefly, as well? It's also a breast cancer--

AUDIENCE: [INAUDIBLE]

PETER PARK: OK, but so you talked about this paper, but not--

OK, I apologize. I can't remember from last year what papers were covered. And I knew that the lymphoma papers were covered by Zack, or somebody. But then I forgot that Steve might have covered this. Anyway, so actually, there isn't that much time so it's good that he covered this.

So basically, the approach is you build some sort of classifier, you pick out the genes through some classifier. And then, so the way that these papers prove that things are good, or the expression profiles are better, is that you do some of Kaplan-Meier plot for each one, and then you do a log rank test, or some tests like that, to show that there's a big difference.

So they do this for a variety of stratified groups. So you might take patients that were assigned one category of disease by your typical prognosis, and then you show that, within that group, there's enough variability. So somehow, there is some information in the expression data that's not captured by this.

And the next step that are done increasingly now is after you get this classifier, you put that together with all the other data. So in this case-- in this case, they have, basically out of all the expression data that they have, they just come up with one signature. So yes, it looks good, versus no, it doesn't.

And so that becomes just a one variable in your multivariate model that you have. So typically, most studies, without expression data, will just have a multivariate model, Cox model, sometimes. So Cox model is just a multiple linear regression, sort of done differently for censor data.

So for Cox model, you fit everything. And then if you say, if you see that the p-value is small for that, then you say, well it's a new parameter, or new variable that you should take into account in your study. So I think this is a fairly reasonable approach.

There is another study-- you may have seen this, as well-- where they do something they do something better, I think. Which is, after you get all your expression data, they actually manually go through the top genes, and then try to come up with new variables. So I think this good signature-- which is batch signature-- is too crude. It doesn't really give you as much information from the data.

I think you could get, I think, a lot more information from your expression data. So in this example, they actually have not yes or no score, but they actually have a prediction score. So that's a continuous variable.

And then, they actually classify their genes, depending on their expression profiles, and their known and annotations into these groups. So in this case they have five groups, so proliferation signature value-- this is a set of genes in P6-- that's just one gene that just happened to be different from all the other profiles.

And then they have these signature values. So I think this is a fairly reasonable approach. And I think it gives you a lot more information. Plus these coefficients, I think, give you a more robust view of how that expression, that group of genes, is impacting your survival in this case.

So you get a prediction score. And then, I guess from here, you could do-- you could put other things, other variables in this setting, in the regression setting. Actually, there's a paper that came out this past week-- this week. And-- you're smiling?

AUDIENCE: I know this one.

PETER PARK: Oh, you know this one. OK. So I mean, I feel like at some point, someone can write a software where you add the data set, and it'll just do all the things that are done. Because it's, I think, fairly similar in all the papers. Except the algorithms-- and basically, the choice of algorithms used is like who the investigator knows, who knows how to do these things.

So in this case, there's Tef Srini is a statistician at Stanford, and so this paper uses the methods that he developed, which I think is a fairly good method. So just to give you a rough idea of what they do here is, so they make predictions in the end-- so they select genes by the method that Tef Srini, which is actually-- it's a little complicated to explain.

But it basically does a search through all your expression of genes, but you can also actually look for combinations of genes. So if you remember, basically all the typical methods will look for a gene that's related to the phenotype. So if there is any interaction between genes, you basically lose them all.

So there is some fancy method that he developed where you basically search through all your genes-- it's like regression, stepwise regression type, where you basically search through your genes, you keep the one that gives you the best information. And then, you try next set, and then you try combinations, and that kind of thing.

So they identify a bunch of genes. And then they do a prediction. So they had 116 adults, you have one group-- a training set with 59, you do a prediction on 57. The way they do the prediction is an algorithm that was published in TNAS a year or two ago.

Basically, you take a new sample, and then see which of the clusters the profile is closest to. But they do this in a slightly clever way. So instead of just looking at correlations, they do something a little bit fancier. But I mean, the underlying idea is the same.

And then, in the end, they do the multivariate analysis gene expression predictor-- is strongly independent prognostic factor. So this is a figure in the paper. I guess at this point, we'll skip this.

So for the next actually about five, 10 minutes, let me talk about something that I did. So most studies so far are used survival curves as a way of verifying the results. So you basically do the clustering, unsupervised clustering, and then you define groups and say, are they really different? So if they're really different as verified by these curves, then you say, oh, that's great.

But in some sense, this is an indirect way of doing things. I mean, if maybe there is a different way of clustering that gives you a better result, who knows. This is just checking to see if your clustering was done correctly.

So in another way, another problem with this is that-- yeah, well, I mean, so that's, I think, one way of doing things.

Another way, actually, to do the censoring is as was done in a lot of the other papers, is to turn survival times into binary indicator, as I mentioned. So something that we were interested in was whether there is some combination. So maybe it's not really gene A that's predictive, but it's gene A plus half of gene B, plus 2 times gene C-- maybe that's really the most predictive of survival time.

So otherwise, you don't know exactly how to combine your genes. If you do some of [? boding ?] method, for example, you are combining information from all your genes, but you're not doing it in an optimal way. So we actually thought about this problem, and then came up with what I think is a good solution.

Of course, the difficulty is that it's too complicated, and so it's hard to write a software that someone can use to just press a button and get a good result. And we can't just do analysis for other people. So as in many bioinformatics algorithms, it's kind of there, but we haven't really done as much as we could have done with it.

So just to give you a brief overview of this-- so the basic problem is that you have too many variables. So we use some method, it's called partial least squares. It's actually becoming very popular now. And it turns out that it's a compromise between just doing least squares, like your regression, versus PCA.

So principle component analysis is good in that you try to maximize information contained in your few components, but it doesn't really have anything to do with the phenotype. So if you pick variables based on PCA, you do it to measure reduction in some optimal way. But it may not be related to your phenotype.

If you do regression, you are picking something that's really tightly correlated with your phenotype. But you're not really doing any dimension reduction. Or you can't fit it in an optimal way. So there's something called partial least squares that's a compromise between the two, and it appears to work really well in most cases.

And it turns out that someone had worked this out. So they figured out how to do partial least squares for different types of phenotypes continuous and binary phenotype. So it gets a little complicated, as soon as you have a phenotype that's not binary, or as soon as it's not continuous but it has been worked out.

But when you have censoring, well, again, if you have a small number of genes, you could use something like Cox model to correlate gene expression to censoring. But when you have too many genes, you can't do that. Your standard methods fail. So the question was, can we do this? Can we apply a method that's well known to work for regular, just other simpler phenotype, but can we work-- can we get this to work for censor data?

So it turns out that we came up with a fairly good solution. It turns out that this was worked out many years ago, we discovered, but there was no need-- they didn't have high throughput data. So there was no need to use this. It was done by a statistician, and it's very hard to understand. I mean, I can barely understand it myself.

So in some extent, it was reinventing the wheel. But we do it a little differently. And then we were able to apply this. So it turns out that we were actually able to get some really good results.

For example, if I use the results of that algorithm to divide the patients into different groups, then it's more significant than using some other method of doing things. So there are methods like this that could be-- I think that will be available. And unfortunately, it does require some mathematical training, but for people here that are interested in this and can invest the time and energy, I mean, I think knowing that the better algorithms will give you a lot of insights into what's going on underneath your data.

AUDIENCE: [INAUDIBLE]

PETER PARK: Well, that's the problem right?

AUDIENCE: You know, because just so many things will always be a problem. Even if they randomly [INAUDIBLE] had nothing to do with the survival, they can always find [INAUDIBLE].

PETER PARK: But not as well, right? Yeah, but it's actually not that likely, if you actually do the calculation. But--

AUDIENCE: [INAUDIBLE]

PETER PARK: Yeah, you could find some combinations that will-- and it's true for anything, right? But even for the binary case, if you have 10 in each group, what's the probability that you're going to find a random predictor that will be lined-- if it's continuous data-- that will be lined up exactly as in the order of the patients? It's not that likely.

So if it's not binary, right? It's much less likely.

AUDIENCE: So you'd think the binary [INAUDIBLE].

PETER PARK: Yeah, yeah, exactly. Yeah, so that's the problem, right? Even for these Kaplan-Meier curves that people show, they need to adjust for multiple testing in some way, right? So you can certainly find genes that will give you a significant p-value, even though the data are random.

So some people do permutation tests to adjust for the p-value. So conclusions-- I don't need to go with that here.

So let's see-- just as a final comment. To summarize briefly, I would say that some of the recent papers have done it pretty well. I think, analysis. I think that involves not just doing the typical cluster, and then finding the p-value for the Kaplan-Meier curves, but actually getting a good score of some sort, and then incorporating that into the multivariate model.

And then, I mean, that part is actually not trivial. But if you're familiar with this statistical package, something like this, it is actually not hard. So you should be able to, I think, if you understand these papers really well, there are a lot of little steps that you have to worry about, you should probably be able to do everything on your own, I think. So without too much difficulty.

So finally, the software that I use all the time is R-- are you guys familiar with the software? What statistics software, if any, do you guys know? MATLAB? MATLAB, a lot of people know. So R is-- so if you go to a statistics department, they'll use one of two softwares. One is at SAS, people use that. And the other one is S+.

So SAS has been around forever. It's like Fortran-- it's like it's been around forever. There's a lot of people that have written good software for it. You know that it works. But it's very clumsy. A lot of things that you want to do, you can't do.

They have a better Windows-driven package now for SAS. But even just a few years ago, you will draw plot like using asterisks on a text kind of page, so draw little characters on your screen as a plot. But I think the younger generation that are more computer savvy are more likely to use S+ unless they have to use SAS in their statistics courses.

So S+ is more like MATLAB, so it's fairly powerful, and it gives you a lot of freedom. And R-- so there's so the reason it's called S+ is there is a statistical language called S-- I don't know why it's called S. Why C? Programming language called C.

And then some-- and this was developed at Bell Labs. And some company took the code, and then made that as a product, and they called it S+. And then now there's a bunch of people, a lot of whom worked on-- a lot of whom are familiar with S that developed R.

So actually the guy that wrote one of the two co-authors of the original R is at Dana-Farber, and he's been doing a lot of work in microarray analysis, as well. But so the software is free, and it's like Linux. People contribute-- it's well tested. And very powerful.

And that's what I use all the time. There are some issues like memory management, and certain data types. But I think things are improving. And so I'm quite happy with it. And everyone that I've recommended to it are pretty happy with it.

It has a feel of MATLAB, like you don't declare variables. You just kind of use them. Matrix manipulations are very similar. Everything is done in vector form. I showed this to one guy in one of the labs at the children's hospital-- he's a fellow. And then he was trying to do some microarray data analysis in Excel, like doing some sort of permutation.

And I showed him how to do this, and he said, this one line, I mean, it took me three hours to do this in Excel, this is one line in R. He's very happy.

AUDIENCE: [INAUDIBLE]

PETER PARK: MATLAB has more statistics, but I think R has more sophisticated statistics. So MATLAB is more engineering oriented. MATLAB has better graphics. It's probably better for solving for algorithms-- like if you want to solve a large matrix, it's probably better in MATLAB. Just because a lot more people are working with MATLAB, and it's a big company, they spent a lot of money on it.

But for statistics, which means, I think, R is more sophisticated. You have a lot more options.

Yeah, so I mean, I used to use MATLAB all the time. But now, I'm happy with R. But if you know one, it's easy to pick up the other. I mean, you always get confused as to how do I comment something out, or that sort of thing. But they're basically the same flavor-- the same reason that MATLAB is popular.

Let's see, steep learning curve, but worthwhile in the long run. And there is something called Bioconductor-- I don't if anyone has heard of this.

So this was an effort that Robert Gentleman, who's at the Dana-Farber Institute, who was one of the founders of R, started. So I was of more of a part of it-- more part of this project before, and he suggested at the very beginning for the title, or for the name of the project, MAD Men-- like microarray data management. But in the end, it was decided that NIH would not want to fund something called MAD Men

So he came up with Bioconductor, which is a much better name. So if you go to bioconductor.org, there are packages that you can download. So what's happening now is-- I think this package has caught on, so that if someone writes or comes up with a good algorithm, they'll write a routine and deposit it somewhere in this website so that you can download it and run it. And that's been done for a lot of good algorithms out there.

So if you want to normalize your cDNA data using some fancy method, it's there. Download it. It's pretty easy. And there's a Windows version of this that's pretty easy. And R comes in Linux and Windows, but it gets compiled for-- gets ported to Windows frequently, and I've had no problems with it.

I think someone was trying to port it to Apple, but I don't know how successful it is. I wouldn't recommend it. But I'm pretty happy with it. This is what I use all the time. This is what a lot of people use who do microarray analysis.

OK, so I see that I have ended three minutes early.

AUDIENCE: [INAUDIBLE]

PETER PARK: Yeah, so the problem-- yeah, there is. The problem is that I think R is just a bad name, because you can't really search for R.

You can go to a Bioconductor R, and there's a link back to R, or you can do-- you can if you do a statistical package R. R used to be-- I mean, it used to be hard to run these things. But Robert and his friends have really made this easy, so that you basically click a button and it'll install on your Windows. So all right.