PROFESSOR: Thank you. And please feel free to interrupt. I'd just as soon run this as a discussion, if you'd like. Is that permitted, do you know?

MODERATOR: Absolutely.

PROFESSOR: OK, so these are conflicts of interests for those of you who care, or you can get it in more detail here by going to this website.

And I thought I will talk about this topic of causality. You've learned quite a bit already in this course about tools for analyzing genomes from various aspects, but what you do after you analyze it is you want to test your hypotheses. And this is a very richly enabling idea, in the sense that you can go to very small cohort sizes, as we'll see-- N of one cohort sizes-- and your false positives are less of a concern if you have a high throughput way of testing them.

And so I think it's very important to know the possibilities for testing causality. And that gets us into engineering genomes-- and, in a particular, about computer-aided design. So you've talked about computer-aided analysis; now let's talk about computer-aided design of genomes, both bacterial and human.

So I just want to illustrate the idea. You might say, well, why would we want to design genomes? You can test causality, typically, by changing one base pair. Why would you want to change more than one base pair? If you have a SNP, that's great.

Well, sometimes you have multiple SNPs interacting in multigenic-- and we'll get to humans in a moment. But here's a radical example, something from the extreme edge, where you'd want to change almost every base pair in the genome-- not

make a copy of a genome but actually design, in an intelligent way-- semi-intelligent-- combinatorial as well-- a genome that has new functions, new properties.

And the four functions I submit for your consideration here is that you might want to be genetically and metabolically isolated for safety reasons or public relations reasons or both. You want to have new chemistry, new protein chemistry, new amino acids. And finally, you want to have multi-virus resistance. This is probably the most powerful of the four, where imagine that you have an organism-- whether it's industrial, agricultural, or even human-- that was resistant to all viruses, past and present-- even ones you haven't analyzed.

So how do we do this? How do we get new functionality? How do we design a genome in such a way that doesn't break? Because if you change the genome enough, you get your comeuppance. You learn you don't know as much as you think you know. You have your beautiful computer simulations from your analysis, and as soon as you test them, you start getting surprises.

So anyway, I'm going to focus on this process of designing and building and then testing. And then, so this part of the design has to have an analytic component. So we'll get back to your old friends in analytics.

So as I go down this list, maybe just show of hands of how many have been exposed to these computational tools already. So Bowtie, anybody? OK, good. See, you covered that, so I don't need to cover that.

Number two-- no? Some? SnpEff? JBrowse-- SQL, you've all heard of SQL, right? OK, good. Let's see.

So the point is each of these things is integrated into this system we call "Millstone," which is all about design and analysis. So it's this loop that goes around and around, as you'll see in just a moment-- actually, may have seen already back here. So we design it. We build it. We test it. And we analyze it.

And the analysis-- sometimes when you build it, you build a large number. You build

a combinatorial set. So this is something that's fairly unique to biological engineering-- or even to certain branches of biological engineering-- that you don't see every day in civil engineering or aeronautics. You don't build a trillion different 787s and see which one works the best. But you can in biology. And I'll give you some examples of that.

And part of the reason we could do this is just as there's next-generation sequencing, which you've heard about in this course-- and we were also involved in next-generation synthesis and next-generation inserting synthetic DNA into genomes. And you'll see all about that.

There are four different ways of doing next-generation synthesis, and it's not important for this particular class. And there are various ways of doing error correction. And these are kind of analogous to the kind of error correction that you have in electronics and computational systems, but we won't stress that analogy too much.

Here's an example, just practically, what you get when you build these oligonucleotides on chips. You might get oligos up to 300 nucleotides long. As they get longer, they tend to accumulate errors a little bit more towards the end. And so you can see that with the length, the number of errors goes up from 1 in 1,300 raw error rate to 1 in 250 raw error rate.

And then we can get rid of some of those errors with a enzymatic system called ErASE-- it doesn't really matter in this case. We can get to 1 in 6,000 without sequencing. And then with sequencing, if you're willing to clone in sequence, you can get error rates even lower. And it's important to know that fundamental limitation. You always need to think about background error in computing as well as synthesis.

You can now do combined synthesis and sequencing very closely by making cis-regulatory elements, which we did in this paper that's published-- Sri Kosuri and Dan Goodman, in particular-- where you could basically synthesize cis-regulatory elements in the genome or in a plasmid. And then you could read out the RNA

simply by RNA sequencing. The number of times you see this bar code in the RNA tells you how many times that particular construct, which could be heavily engineered-- it isn't like randomers-- you're making interesting, cis-regulatory elements. And you can make 10s of thousands of these-- millions of these constructs. We did 10s of thousands.

Then you can measure protein levels as a result of cis-regulatory. So you can have promoter elements, ribosome binding sites, and coding region mutations that you think might influence RNA and protein. And here we do proteins by having two fluorescent proteins-- a red and a green. The red is the control, and it has a very tight distribution, as you can see here.

And then the green is subject to this cis-regulatory mutations made on chips. And it has a big distribution. And you divide it up in a fluorescence-activated sorter. And you can read it out.

So here, every pixel on these two plots for RNA and protein is a separate experiment. And you can drill down and get some more information on each of these. But the basic idea is each of these was individually synthesized on the chip and individually sequenced later to determine. And the bar codes can be read out of proportion to the RNA and protein expression.

And here's an example of some surprises that come out of such studies-- and we're not just doing this for our health. So, for example, when we went into this, it was well known that codon usage effect was correlated with, and could even causally influence-- so here's an example of causality-- the expression of a protein. If you have very commonly used codons, which typically have high levels of the corresponding transfer RNA in the cell, that the observation-- and it makes sense-- is that those proteins would be expressed at higher levels.

The thing that was new was we discovered that at the end terminus, close to the cis-regulatory elements, it flips. It's the opposite. There's almost no correlation with abundant codons, and there's essentially a negative correlation here with an r squared of 0.73, right here, that shows that there's a higher expression with very

rare codons. This was published in *Science.*

And so a lot of them tend to be AT-rich, but we can separate out that component. We can separate out things like ribosome binding sites, which are AG-rich. And there's just a general trend where rare codons help expression if they're at the beginning of the gene. And you could find that out from this kind of experiment.

So now we want, if we're going to build the genome that's radically different-- let's say "radically different," here, defined as 7 to 13 codons, chains, genome-wide freed up-- liberated-- meaning that we use the synonyms in the genetic code. So there's anywhere from one to six codons for each amino acid-- three codons for stop codons. We can use that synonymous substitution table to move things around and completely free up-- get rid of every instance of a UAG and turn it into UAA. That's the first example.

And we did that genome-wide and thereby derisked it. We can now build on top of that, because we can get genomes that grow well under a variety of conditions. They're still genetically engineerable. And everywhere there's a bar there, this refers to a successful mutation in the height of the bar as refers to the efficiency of introducing those mutations.

Now we wanted to derisk another special category-- remember, I said AGA and AGG are special, in that they're the rarest coding codons. So UGA is a stop codon. AGA and AGG are arginine-encoding codons. And they're the rarest. And they also are complicated, because they tend to represent Shine-Dalgarno sites, which tend to be AG-rich regions that are involved in initiation of protein synthesis.

Anyway, so there, the number was a little large to do genome-wide, so we focused on essential genes. And so you can computationally find all the essential genes and design strategies for getting all the AGG and AGAs. And then when you synthesize those genomes, you can do them one at a time with a process called [? MAIDS, ?] which we won't go into-- too experimental.

But basically, you can essentially just go straight from oligos into the genome, and

you can do multiple ones simultaneously. And you can see which ones are hard to make and which ones are easy-- again, that's the sort of efficiency number there. You can see which ones-- if they're selected against. And some of them were actually selected against. We could not find them.

And so these are discoveries. These are examples where synonymous is not synonymous. It could mean that there's some other function, hidden, layered on top of the synonyms-- might be a ribosome binding site.

And so what we find is that we can try other, let's say other arginine codons, rather than the one we targeted. Or you sometimes can try out other codons that are not even synonymous. And eventually we found every single one of them. So there were about a dozen. They were hard at first, and then we eventually found an engineering workaround. And that illustrates a number of interesting points.

So those were all successful in essential genes. And it's our observation that if you get it to work for essential genes, getting it to work for the nonessential genes is even easier.

So then we went on, and so that's one codon at a time, two more at time. So we've derisked three codons at this point. So we went on to derisk all 13 codons-- or 13 of the 64.

And we did that in even smaller set of genes. So there are 290 essential genes in E. coli. We did 42.

And in that case, there were 400. And some examples of those-- and every one of them worked except for one. And just like the arginine codons-- that one, we tried a number of different codons, and they worked-- including non-synonymous codons. So in almost every case, you can find something that works.

And then we do biological assays that the four functions that we felt should be changed were actually changed. And here's two slides on the virus resistance. You can do, in a variety of ways, of determining how effective the virus resistance is.

Here you have about a factor of 1,000 for phage lambda, which has been mutated to be highly virulent in E. coli. This is a very pathogenic version of phage lambda.

This is T7, which is naturally quite lytic. And you can show that this is resistant to two of the three viruses that we tested. And our hypothesis is if we change more codons than just-- that was just one codon. If we change seven or so, which is what we're doing now, then it will be resistant to all viruses-- and very heavily resistant-- so resistant that the population of viruses can't mutate enough to become resistant. So all of you should be questioning that-- do I really believe that? And we can talk about that in the discussion.

So now the other big functionality is-- can we genetically, metabolically isolate these? And to do this, we took advantage of its new genetic code. Not only we've freed up a codon, we can now make that codon code for a new amino acid by another set of biochemistry.

And here's some examples. The amino acids look kind of like tyrosine or phenylalanine. Here's one that's a biphenylalanine, so it's got two benzene rings instead of one. And so it's bulkier. It's bulkier than any other amino acid, any naturally occurring one.

And we wanted to ask-- can we make those essential genes that we've been talking about-- can we make them addicted to this amino acid? And so we did by this computational protein design strategy. And the idea is we looked through every crystal structure of every essential protein in E. coli-- there's 129 or something like that, 120 crystal structures-- and systematically ask, were there any places where we could fit in a larger amino acid by carving away adjacent amino acids, such that when we then replace that larger one with a smaller one-- still keeping its surroundings mutated, so we could mutate it two, three, four, eight times-- however many amino acids nearby you need to accommodate the big amino acid-- if it no longer accommodates the small amino acids?

So you basically systematically go through every amino acid for every crystal structure and found a short list of a half dozen or so that looked promising. And so

the idea is, you put in these 2-phenol groups-- and now, when you accommodate it and shrink it down, it won't work. OK, that's the basic idea.

And in context, we wanted to have a really tough test for this. We wanted to say, not only do we want it to be addicted to this, but we don't want it to be able to escape-- either by mutation and evolution, we don't want it to escape. We don't want it to be able to escape by eating it's fellow-- its classmates-- its other E. coli.

And so the test we do is we do a-- did you have a question, anybody?

We would lyse the cells-- lyse cells of a wild-type E. coli or certain mutant strains that would produce large amounts of these. And one of the more classic ways of making an organism that's metabolically isolated so it can't survive in the wild-- it can only survive in an industrial plant or in a laboratory-- and we did this with the classic ones, which people have avoided using lysates, because it gives them bad news, which is if you grow them on lysates, you get a lot of survivors. These are the classic ones. The deletions of these two genes makes them-- they will still grow.

But this is an example of one of our designed, nonstandard amino acid strains. And we get much lower escape rates. And you'll say, even this low number here, we want to get that down to zero. And you'll see how we do that a moment. This is Mike Mee as a graduate student.

So here's a close-up of-- this is not of the active site. This just could be any place in the protein where putting in a big amino acid is going to be disruptive. So we change this leucine, innocent leucine, that's packed all around with other amino acids.

Have you guys done protein design in this class at all? Yeah? OK, so you know what I'm talking about. Rosetta, right? OK. So that's what we're using here.

We had to modify it to use nonstandard amino acids, because normally people design proteins with 20 amino acids. So we took this leucine-- we made it into this bipA. And you can see now, it's got all kinds of clashes-- three initial clashes. That's not good. So we identify those clashes and we make them smaller-- no clashes

anymore.

This is all done in the computer. This is all theoretical. Can you believe that? We'll see.

So then-- this is putting back in a small amino acid. These are some of the people that did it. So Marc and Dan are post-docs in the lab, and Ryo and Barry did the crystallography. I'm a crystallographer by training, but I'm a little out of practice.

So here is the design again, and there's the electron density. So now you can believe it, right? Because it's not just a computer model. Well, it's still a computer model, but it's based on data. And here's a comparison of the design with the X-ray structure-- not too shabby. OK.

But the question is, how well does this work in living cells? So these are cells where we've gone-- changed the whole genome so that now the stop codon, UAG, is free. It's never used, which means we can delete the release factor that normally recognizes a stop codon, which otherwise would have been lethal. And we can replace it with a transfer RNA in a tRNA synthetase that brings in this [INAUDIBLE] amino acid.

And now-- this is the one we were just looking at, the crystal structure in bold here. And it has an escape frequency which is higher-- we can crank up mutagenesis by putting it in a mutS minus background. Basically, one of the mismatched repair proteins-- we can knock it out, which increases, sort of accelerates, evolution.

And it has a noticeable escape frequency. So a more realistic scenario would be this mutS plus. And we can get escape frequencies as low as 10 to the minus 8th. These are for other mutations in that same protein. And here are mutations in another protein.

So then we said, OK, but none of these are perfect. We want something that's undetectable levels of escape. So how would we, how would you, fix this? Anybody? I'm trying to encourage you to interrupt me, so I'm interrupting you. Anybody?

You've got these things that are escaping at very low frequencies. We should be proud of that. But we want to drive it even more. Rather than 10 to the minus 8th, you want get down to 10 to minus 10th, or something like that. Suggestions?

AUDIENCE: So this is reversion of the mutations of the [INAUDIBLE].

PROFESSOR: Well, so this means that you can take the bipA, and you mutate the codon so it doesn't encode bipA anymore. It encodes something else. So it doesn't need bipA from the media. And it puts in another amino acid, and it somehow survives. So even though it's not a perfect fit, it does well enough that the enzyme is made.

AUDIENCE: So then modified multiple essential genes?

PROFESSOR: Multiple essential genes-- wow. Couldn't have said it better myself. That's what we did.

So before we could choose which two we wanted to use-- or three-- we wanted to know what the spectrum was. So we forced in all 20 standard amino acids to replace the bipA. So we said, let's mutate them intentionally-- synthetically-- and see what the spectrum is.

Now this is not going to be the natural spectrum, the sort of mutagenic spectrum-- this is our intentional-- so what we do is, we put in each of the 20. And then we do a quick selection at 20 doublings. It's a very fast evolution, not three billion years. My students didn't want to wait.

So in 20 doublings, you get a spectrum of which amino acids will substitute for bipA. In an ideal world, none of them would. But we forced them to, and these are the survivors.

And so the ones we've been talking about here, W, tryptofan, is what we'll substitute for bipA. And that kind of makes sense. It's the biggest amino acid. And that works for the [? tyrS ?], which happens to be the tRNA sythetase.

And then we picked this other one under this big red arrow for AdK-- adenosine kinase-- [INAUDIBLE] kinase-- where there's very little tryptophan that will work in

that one. But you get some escapees if you force it to take these hydrophobic aliphatics like leucine. So we made the double mutant of the-- we don't have it here-- but we've made the double mutant of the AdK and the tyrS, and it's vanishingly small.

We're probably not done. We'll keep doing this. But this is the way that you do a radical recoding and get new functions.

Any questions on that part? We're going to move onto human genome engineering. Yeah.

AUDIENCE: [INAUDIBLE] and recognize the different amino acid.

PROFESSOR: Yeah, I skipped over that because that's a little more on the biological, a little less on the computational side. So this was a work from Peter Shultz' lab and other groups. And what you do is you take a synthetase that's orthogonal, meaning it's from a completely different organism-- in this case, Methanococcus jannaschii, which is a hyperthermalphile.

You take that synthetase-- it's about as far as you can get on the evolutionary phylogenetic tree-- you bring it into E. coli. You bring in its cognate, tRNA. You change the anticodon so that it will recognize UAG, which is not what typically any tRNA normally recognizes. And that only works with certain synthetases. So only certain synthetases are blind to the anticodon-- mainly serine and leucine synthetase is in E. coli.

Anyway, so you can now evolve the active site that binds to the amino acid and the ATP. So the amino acid and ATP cause the amino acid to be [INAUDIBLE] the transfer RNA. Anyway, you can change the active site so that now it recognizes any amino acid you want to a first approximation. And you could do that through a combination of intelligent design and random mutagenesis, and there are selections for that as well.

So in general, if you're going to be doing random or semirandom mutagenesis, it's

always great to have a selection so there are selections for these things. And there now are dozens of amino acids that work fairly well in that scenario. The main thing that was limiting was not the synthetase-- I mean, you could get synthetases. It's the tRNA then had to compete with the release factor in the stop codon or had to compete with another tRNA if you use a different anticodon.

And so freeing up this codon means there's no competition. And now it works about as well as a regular amino acid. But when it has to compete, it's at a great disadvantage. Yeah.

AUDIENCE: Can you explain why changing the genetic code will cause all virus resistance?

PROFESSOR: I planted that, but thank you anyway. So there's a genetic code up there in circular form-- probably you're more used to seeing it in rectangular. But imagine that we've now derisked this UAG stop codon and these AGA and AGG codons here-- R for arginine. And we're in the process of putting all those three codons together with another four for serine and leucine.

And remember, I said serine and leucine is interesting, because you could swap out the anticodon-- the synthetase doesn't care. So that's why we picked those ones-- the three rarest ones, plus four where you can swap out the anticodon. So we could swap serine and leucine, for example. So serine and leucine also are examples of tRNAs that bind to six different codons. So moving two of them is not a big deal. So you still got four left.

So anyway, imagine that we remove them or swap them and do weird stuff with them. Every time the phage comes in, it has lots of serines and leucines that are using these, and arginines and stops. And every time it wants to put in a leucine, the ribosome puts in a serine.

Well, you can note, leucine and serine aren't that similar, and that's going to cause a mess for every single protein it makes. And there might be dozens-- maybe even hundreds for big phage-- of those codons. And so you can do the math-- that the chance of mutating one of those codons to something that will work is fairly high.

Two is squared, three to the n power, where n is the number of changes it has to make. And so if you make enough changes, population sizes have to become astronomical in order to contain one member that has changed all of its codons the right way and hasn't changed a bunch of codons that would be lethal.

AUDIENCE: So the ones that you chose, were they the rarest of the codons--

PROFESSOR: So the first three were the rarest. And part of that is because we felt we would run into the most trouble there. They may be rare for a reason. And we wanted to discover those reasons, both for biological curiosity, but also to derisk the subsequent engineering.

But the leucine and serine ones are normal. They're not that rare. But we derisked them. And remember that one where we did 13 codons on 42 essential genes? That's how we showed that, in general, it's not toxic to individual genes.

But there are examples of things where you derisk it on individual genes and you start making lots of them, and then you get so-called "synthetic lethals" where various pairs of genes conspire. But so far, most of the deleterious nature of the genomes-- where the genomes are a little bit slower growing-- it's usually due to hitchhiker mutations, not due to our design-- except in cases where it's completely not working, in which case we have to find an alternative codon. But we have to deal with all these things-- design errors, biological discovery, and hitchhikers. Yeah.

AUDIENCE: If you've already found that multiple, simultaneous mutations is unlikely, works, if they all had happened at the same time, but if you have this engineered system, if you have some way of migrating code to other-- you could end up with the spreading of your non-secret codes so that you can mutate things, one of them at a time, and accumulate.

PROFESSOR: Well, so, first of all, a phage doesn't carry along its own code. If it did, we could preempt that by making lethal genes-- that if you bring in the tRNA that has the old code, you activate the lethal gene. But I think you were talking about more a Darwinian perspective, where you have incremental changes that allow you to slog

along well enough that you can get more mutations.

The problem is, this collection of mutations-- there is no growth. Every protein is majorly messed up. And so you're not talking about, say, antibiotic resistance, where there will be kind of a gradient of antibiotics. And somewhere on the edge of the gradient, there will be just enough antibiotic to be selective, but not enough to kill it. This is something where, the instant they get into the cell, there's no gradient. They only have one code choice, and that code is something--

I think the difference between this and regular evolution is, regular evolution-- if the bacteria tried this strategy, it would be changing a little bit at a time and the phage be keeping up with it. But we took it offline, so to speak, did major code revision, and moved it back. And the phage was not watching. And the phage isn't as intelligent as hackers are.

OK, any other questions? We could stay on this topic. We don't have to go on to humans.

OK, just for fun let's go on to human genome. How many people here want to have their genome edited? All right. We'll ask in just a moment what you want to have changed.

So these are some of the tools that my colleagues and I have worked on. I've been doing this most of my career, is coming up with new tools for engineering genomes and sequencing genomes. And the one I've been talking about so far is down here at the bottom-- is Rec A and Red Beta. And the star for going forward is this Cas9 protein.

But we color-coded them here so that the recognition-- the critical thing about genome editing is finding the needle in the haystack. You want to change one base pair. You don't want to change anything else. And so something has to do that recognition.

That recognition can be Watson-Crick, so you can have DNA-DNA-- searching through the entire genome with DNA-DNA interactions, or RNA-DNA interactions, or

Watson-Crick, or protein-DNA interactions, which I'm sure you've learned about quite a bit. And so we have examples of each of these-- two examples are RNA, in blue; two examples of DNA, down in the box; and then all the rest are protein, where the protein-- the amino acid side chains are recognizing, typically, some kind of alpha helix in the major groove.

OK, so Cas9 was something that was a nice case of computational biology, in my opinion. It was found in 1987 in E. coli by Ishino and colleagues.

And it was essentially junk DNA. It was not conserved. It was repetitive, which were two of the hallmarks of junk DNA, which were very popular talk about in 1987. They were trying to shut down the Genome Project before it started, three years before it started-- before the NIH part of it started-- because they didn't want to sequence anything in the human genome that wasn't coding for proteins. I'm serious.

So anyway, this languished as junk DNA for many years. It eventually became clear to the cognoscenti bacteriologists that it might be an interesting, adaptive immunity-- kind of like antibodies-- rather than the fixed or native immunity, which were restriction enzymes. So this is kind of the adaptive version of restriction enzymes.

But it still didn't really catch on until 2013, when a couple of my post-docs and ex-post-doc and graduate students in January got it to work in humans-- so moved it from bacteria to humans-- kind of a big jump. And then it became surprisingly easy, once it made that jump, to get it to work in every organism that we and others have tried. So now 20 different organisms, at least, that this works in-- fungi, plants, and even elephants. We haven't published the elephant yet, but we have our reasons for doing that.

And the most frequently asked question-- and this, of course, should appeal to computational biologists trying to do design-- is, what about off-target? And it turns out now there are many ways of dealing with off-target-- so much so that I would be so bold-- and this is a slight speculation-- but I would say we're currently at the point where it's almost not measurable, the off-target.

And these are the different ways you can do it. So we started out, in our January 2013, with theoretical, where you would basically look for-- anybody in this room would know immediately how to do this-- would look for potential off-targets that are off by one or two nucleotides and ban those from consideration. And then you take a shorter list and do an empirical search, because this is so inexpensive. Basically, you have this guide RNA which is making a triple helix. It's binding the one strand of the DNA.

It's so easy to make those guide RNAs. It's just 20 nucleotides you have to make. You pop it into a vector where everything else is taken care of. It's so easy to do that that you can make a lot of them, and you do an empirical search. You find places that are particularly hot for the right sites and very cold for the wrong off-targets. So those are the first two methods.

Then paired nickases-- they don't make a double-strand break, which is what it does out of the box from nature. It makes a double-strand break. You have it make a single-strand nick. Then you require two of these to be coincident and near one another.

It's like the concept of PCR. You have to have two primers that are near one another. So it's a coincidence. So it's like a p squared-- if the probability is one is off by one or two or however many it takes, the chances of getting two such sites near each other is roughly p squared.

Truncated guide RNA is not something that you would necessarily guess that, if you make the guide RNA smaller, it's going to be better. But there's obviously some optimum. If you make it too long, then it can bind by any subset-- any kind of mismatched subset. If you make it too short, then from informatics standpoint, it doesn't have enough bits to recognize a place in the genome.

So it turned out that the optimal length was a little bit different from the natural length. It was about two shorter.

And finally-- and this just came out. And this is from Keith Joung and David Liu's lab,

where you get rid of the beautiful, double-strand break capacity. You can turn into a nickase, or you can make it completely nonfunctinal as a nucleus and then add nucleus domains back.

And you say, well, it seems kind of bizarre that you're doing all that work-- that you get rid of the nucleus and you add it back-- add in a different one, the FokI bacterial restriction in the nucleus. But it turns out this is the way that people have taken other DNA-binding proteins-- the zinc fingers and then the tau proteins. And so it had to be tried, and it works extremely well. And it's like the paired nickases-- you need two of these sites in order to get cleavage. And stay tuned. I'm sure there's more.

So I just want to close on this idea of causality again. I opened on it. I'll close on it. Here's an example of a double null-- myostatin double null, as the both maternal and paternal copies are missing.

There are a lot of examples of double nulls. We could talk about some later. And they're often rare. So at one point, there was only one person in the world that was characterized with this. And it's hard to do a large cohort study on this.

And they weren't really sick. The phenotype-- this little baby had heavy musculature, as if he was working out next to Arnold Schwarzenegger. But he came out this way, and he stayed that way.

But it's striking. You look at the genome and you say, wow-- a double null and a highly conserved protein. That's got to mean something. And then you can have a hypothesis of what it means based on what was known about that pathway. And it coincides with the phenotype.

And so you have a strong hypothesis, and you can test it in animals. And so here, you don't normally test it in three different animal species. But this one, there happened to be either preexisting or easy tests in cows, dogs, and mice. So that's one thing you can do to get a causality.

And the other thing is, there are cases where the animal models don't work. Either

you knew in advance they weren't going to work because they don't have that brain structure. There's nothing other than humans that have a particular kind of brain structure, so it's hard to make mutants, because you're already a mutant.

And so another option is organs on chips or organoids, because they're not really fully physiologically faithful. And this, at least, is human, but just like animal models can have artifacts, human organoids can have artifacts as well.

Here's an example of something that will be coming out in a few days that we did together with Keith Parker's lab and Bill Pu's lab. And I think this is a nice example of where you can take a hypothesis, where one base here is changed-- this G right here, is deleted-- and that's putatively what causes this cardiomyopathy that affects mitochondrial function.

And you can mutate that using the CRISPR technology I was talking about, where you use homologous remedies to go in, find that one base, change it. Or you can just make a mess near there. So one control is to not change it, and the other control is to put a little insertion, deletion in there. And of course that messes it up as well.

And so you've now constructed three isogenic strains. These are actually my cells. In the Personal Genome Project, we take volunteers like myself and establish stem cell lines. And then from the stem cell lines, we can establish, in this case, very well-ordered cardiac tissue we'll see in the next slide. And that cardiac tissue, you can test for lipid biochemistry, for other physiological parameters, for the morphology and the contractility-- so diastole and systole that you get in the cardiac muscle.

So you basically make something where you've only changed one base pair in my genome, and we've made, essentially, a version of me that's mutant. Unfortunately, I don't think I had the picture of that. I thought I did. Oh, there it is.

So here's an example-- how you get this beautiful, ribbon-like striated pattern that you expect of cardiac muscle. This is programmed from my fibroblast turned into stem cells into muscle. And then if you introduce the two mutations-- either the one

that corresponds to a patient or one that's just a mess-- you get a morphological mess. And then you can restore those by putting in the messenger RNA that will cover for the mutation.

So I'm going to open it up for questions at that point. That's causality-- I think. Questions? While we're waiting, anybody wants to volunteer what they would like to change about themselves? You can mention a specific base pair or kind of a general idea of what you'd like to change, whether you think there's any safety considerations that we should keep in mind.

AUDIENCE: The problem's delivery, right? That's the--

PROFESSOR: Delivery.

AUDIENCE: Yeah.

PROFESSOR: Yeah, fair enough. So gene therapy had a crack. People were a little overanxious, a little overambitious about over a decade ago.

And a small number of patients died from cancers, because there was random integration. Rather than this precise genome manipulation we're talking about here, there was kind of random lentiviral integration of extra copies of genes. And if you land in the wrong place, then your lentiviral or retroviral promoter will go off into oncogenes, like LMO2.

So that delivery was viral delivery, and it was random integration. We now have delivery mechanisms that are nonintegrative or integrative in a specific place or, in this case, can make precise base pair changes. So there's two levels of delivery-- one is to get it to the right tissue, and the other is to get it to the right base pair. I think both are semisolved problems.

So you can do ex vivo delivery. So you can take T cells out of a body. You can use a previous generation-- the zinc finger nucleus-- to cleave both copies of the CCR5 gene. And now people that had full-blown AIDS, you put these T cells back in their body, and then now they're AIDS resistant. Those T cells that have both copies of

the CCR5 gene missing, which is the AIDS coreceptor, are now resistant.

So that's ex vivo. That's one way to do it. Delivery to the liver is quite easy. You can do that with nonviral vectors, and a [INAUDIBLE] virus is one that's very popular. You can get it to go to almost every cell in the body, either selectively or generally. So you just want to make sure that once it goes there, it doesn't cause any damage other than the base pair you want to change.

So there are now 2,000 gene therapy trials in phase one, two, and three. It's a big change from a decade ago, where I think people had pretty much given up on gene therapy. There's now 2,000 clinical trials.

And one has emerged all the way out of phase three into full approval in Europe. Ironically, they now have genetically engineered humans in a land where they don't eat genetically modified foods. But I think they're better for it So far, it's curing diseases. Yeah.

AUDIENCE: For your noncanonical amino acids, does this open up enzymatic reactions that would be, say, impossible, do you think, with if you add a new amino acid that can [INAUDIBLE]?

PROFESSOR: So I'll just repeat the question for our viewing audience. Do nonstandard amino acids open up new enzymatic reactions? And there's already a couple of examples in the literature.

This was done prior to this wonderful strain, where there's no competition. It was done at low efficiency. But putting in one amino acid at low efficiency-- you could still get an enzyme. So even if it's, like, 10% efficiency, you produce 10 times as much enzyme, and it works.

So there were some redox-coumarin derivatives of amino acids. So coumarin-redox capabilities is not present in any of the other amino acids. And they took a protein that was very well studied-- where they had by protein design, and by random mutagenesis, and they threw the book at it-- and they could not budge the activity beyond the apparently optimal, naturally occurring activity.

They put in this amino acid, which was not randomly chosen-- it was a redox-coumarin derivative-- they put it in the active site. I think they tried out a few different things that made a small combinatorial library. But the point is, they got a tenfold improvement in the catalytic rate constants. So that's an example.

Another example, which isn't really catalytic, but it's very popular, is that you can put in polyethylene glycol-modified amino acids wherever you want rather than kind of randomly. You can put it in precisely. And this will greatly extend the serum half-life, so that normal proteins like human growth hormone, which is a approved pharmaceutical for certain uses-- not all the uses that you find on the internet, but other uses-- but it turns over very quickly in the serum.

And so if you put a polyethylene glycol in the right place on human growth hormone-- or other human protein pharmaceuticals-- they last longer. Those are two examples-- one of them definitely active site. Yeah.

AUDIENCE: This is actually a small detail from your [INAUDIBLE] study, where you looked at the structure and mutated one of the amino acids to this phenyl thing, and then you changed a bunch of other amino acids to compensate for that size. So I noticed most of the changes were to either [INAUDIBLE], but one of them was a tryptophan. So why was that?

PROFESSOR: Let's go back to that, and see if we can find that.

AUDIENCE: It was a previous slide. Yeah, this one. So it was amino acid 271.

PROFESSOR: Yeah, OK. So in each of these lines-- I didn't spend much time on this-- in each of these lines, there's one amino acid we've changed to bipA. So these three are all the same protein, and it's all the same mutation-- leucine 303 to bipA. And then all the other ones are compensating. And then here, you can see it's a different leucine and a different protein. They're all leucines-- different proteins. Now what's your question about?

AUDIENCE: My question was the compensating mutations are generally all the smaller amino

acids, right?

PROFESSOR: Oh, I see. So why phenylalanine and tryptophan?

AUDIENCE: Yeah.

PROFESSOR: Well, those are pretty close. So these are done by energy, not by eyeball. They're done all by COMP ROSETTA, where we combinatorially go through lots of side chains. So we combinatorially went through lots of proteins, lots of positions to substitute amino acid, then lots of accommodating mutations-- which is not necessarily the typical way you use this software.

Anyway, that probably is some stacking of one of the two aromatic rings onto the tryptophan. Yeah. And we tried many combinations. No doubt, we tried the phenylalanine and the tryptophan in various combinations with the other ones, and the tryptophan empirically works better.

MODERATOR: Are there any more questions?