ETHAN MEYERS: What I'm talking about today is neural population decoding, which is very similar to what Rebecca was talking about, except for I'm talking about now more on the single neuron level and also talk a bit about some MEG at the end. But kind of to tie it to what was previously discussed, Rebecca talked a lot about, at the end, the big catastrophe. Well, you don't know if something is not there in the fMRI signal because things could be masked when you're averaging over a large region as you do when you're recording from those bold signals.

And when you're doing decoding on single neurons, that is not really an issue because you're actually going down and recording those individual neurons. And so while in general in hypothesis testing you can never really say something doesn't exist, here you can feel fairly confident that it probably doesn't, unless you-- I mean, you could do a Bayesian analysis. Anyway, all right. So kind of the very basic motivation behind what I do is, you know, I'm interested in all the questions the CBMM is interested in, how can we algorithmically solve problems and perform behaviors.

And so, you know, basically as motivation, you know, as a theoretician, we might have some great idea about how the brain works. And so what we do is we come up with an experiment and we run it. And we record a bunch of neural data. And then at the end of it, what we're left with is just a bunch of data. It's not really an answer to our question. So for example, if you recorded spikes, you might end up with something called a raster where you have trials and time. And you just end up with little indications at what times did a neuron spike.

Or if you did an MEG experiment, you might end up with a bunch of kind of waveforms that are kind of noisy. And so this is a good first step, but obviously what you need to do is take this and turn it into some sort of answer to your question. Because if you can't turn it into an answer to your question, there is no point in doing that experiment to begin with. So basically, what I'm looking for is clear answers to questions. In particular I'm interested in two things.

One is neural content. And that is what information is in a particular region of the brain, and at

what time. And the other thing I'm interested in is neural coding, or what features of the neural activity contain that information. And so the idea is, basically, if we can make recordings from a number of different brain regions and tell what content was in different parts, then we could, basically, trace the information flow through the brain and try to unravel the algorithms that enable us to perform particular tasks. And then if we can do that, we can do other things that the CBMM likes to do, such as build helpful robots that will either bring us drinks or create peace.

So the outline for the talk today is I'm going to talk about what neural population decoding is. I'm going to show you how you can use it to get at neural content, so what information is in brain regions. Then I'm going to show how you can use it to answer questions about neural coding, or how do neurons contain information. And then I'm going to show you a little bit how you can use it to analyze your own data. So very briefly, a toolbox I created that makes it easy to do these analyses.

All right, so the basic idea behind neural decoding is that what you want to do is you want to take neural activity and try to predict something about the stimulus itself or about, let's say, an animal's behavior. So it's a function that goes from neural activity to a stimulus. And decoding approaches have been used for maybe about 30 years. So Rebecca was saying MBPA goes back to 2001. Well, this goes back much further.

So in 1986, Georgopoulos did some studies with monkeys showing that he could decode where a monkey was moving its arm based on neural activity. And there was other studies in '93 by Matt Wilson and McNaughton. Matt gave a talk here, I think, as well. And what he tried to do is decode where a rat is in a maze. So again, recording from the hippocampus, trying to tell where that rat is.

And there's also been a large amount of computational work, such as work by Selinas and Larry Abbott, kind of comparing different decoding methods. But despite all of this work, it's still not widely used. So Rebecca was saying that MVPA has really taken off. Well, I'm still waiting for population decoding in neural activity to take off. And so part of me being up here today is to say you really should do this. It's really good.

And just a few other names for decoding is MVPA, multi variant pattern analysis. This is the terminology that people in the fMRI community use and what Rebecca was using. It's also called read out. So if you've heard those terms, it kind of refers to the same thing. All right, so

let me show you what decoding looks like in terms of an experiment with, let's say, a monkey. So here we'd have an experiment where we're showing the monkey different images on a screen.

And so for example, we could show it a picture of a kiwi. And then we'd be making some neural recordings from this monkey, so we'd get out a pattern of neural activity. And what we do in decoding is we feed that pattern of neural activity into a machine learning algorithm, which we call pattern classifiers. Again, you've all heard a lot about that. And so what this algorithm does, is it learns to make an association between this particular stimulus and this particular pattern of neural activity.

And so then we repeat that process with another image, get another pattern of neural activity out. Feed that into the classifier. And again, it learns that association. And so we do that for every single stimulus in our stimulus set. And for multiple repetitions of each stimulus. So you know, once this association is learned, what we do is we use the classifier or test the classifier. Here we show another image. We get another pattern of neural activity out. We feed that into the classifier.

But this time, instead of the classifier learning the association, it makes a prediction. And here it predicted the kiwi, so we'd say it's correct. And then we can repeat that with a car, get another pattern of activity out. Feed it to the classifier, get another prediction. And this time the prediction was incorrect. It predicted a face, but it was actually a car. And so what we do is we just note how often are predictions correct. And we can plot that as a function of time and kind of see the evolution of information as it flows through a brain region.

All right, so in reality, what we usually do is actually we run the full experiment. So we actually have collected all the data beforehand. And then what we do is we split it up into different splits. So here we had, you know, this experiment, let's say, was faces and cars or something. So we have different splits that have two repetitions of the activity of different neurons do two faces and two cars, and there's three different splits.

And so what we do is we take two of the splits and train the classifier, and then have the remaining split and test it. And we do that for all permutations of leaving out a different test split. So you all heard about cross-validation before? OK. One thing to note about neural populations is when you're doing decoding, you don't actually need to record all the neurons simultaneously. So I think this might be one reason why a lot of people haven't jumped on the

technique because they feel like you need to do these massive recordings.

But you can actually do something what's called pseudo populations, where you build up a virtual population that you pretend was recorded simultaneously but really wasn't. So what you do with that is you just, if on the first day you recorded one neuron, and the second day you recorded the second neuron, etc. What you can do is you can just randomly select, let's say, one trial when a kiwi was shown from the first day, another trial from the second day, et cetera. You randomly pick them. And then you can just build up this virtual population. And you can do that for a few examples of kiwis, a few examples of cars.

And then you just train and test your classifier like normal. But this kind of broadens the applicability. And then you can ask questions about what is being lost by doing this process versus if you had actually done the simultaneous recordings. And we'll discuss that a little bit more later. So I'll give you an example of one classifier, again, I'm sure you've seen much more sophisticated and interesting methods but I'll show you a very basic one that I have used a bit in the past. It's called the maximum correlation coefficient classifier. It's, again, very similar to what Rebecca was talking about.

But all you do is-- let's say this is our training set. So we have four vectors for each image, each thing we want to classify. And all we're going to do is we're going to take the average across neurons to reduce these four vectors into a single factor for each stimulus. OK, so if we did that we'd get one kind of prototype of each of the stimuli. And then to test the classifier, all we're going to do is we're going to take a test point and we're going to do the correlation between this test point and each of the kind of prototype vectors.

Whichever one has the highest correlation, we're going to say that's the prediction. Hopefully pretty simple. The reason we often use fairly simple classifiers, such as the maximum correlation coefficient classifier, is because-- or at least one motivation is because it can be translated into what information is directly available to a downstream population that is reading the information in the population you have recordings from. So you could actually view what the classifier learns as synaptic weights to a neuron. You could view the pattern of activity you're trying to classify as the pre-synaptic activity.

And then by doing this dot product multiplication, perhaps pass through some non-linearity, you can kind of output a prediction about whether there is evidence for a particular stimulus being present. All right, so let's go into talking about neural content, or what information is in a

brain region and how it needs decoding to get at that. So as motivation, I'm going to be talking about a very simple experiment. Basically, this experiment involves a monkey fixating on a point for-- well, through the duration of the trial. But first, there's a blank screen. And then after 500 milliseconds, up is going to come a stimulus.

And for this experiment, there is going to be 7 different possible stimuli that are shown here. And what we're going to try to decode is which of these stimuli was present on one particular trial. And we're going to do that as a function of time. And the data I'm going to use comes from the inferior temporal cortex. We're going to look at 132 neuron pseudo populations. This was data recorded by Ying Jang in Bob Desimone's lab. It's actually part of a more complicated experiment but I've just reduced it here to the simplest kind of bare bones nature.

So what we're going to do is we're going to basically train the classifier on one time point with the average firing rate in some bin. I think in this case it's 100 milliseconds. And then we're going to test at that time point. And then I'm going to slide over by a small amount and repeat that process. So each time we are repeating training and testing the classifier. Again, 100 milliseconds sampled every 10 seconds, or sliding every 10 seconds. And this will give us a flow of information over time.

So during the baseline period we should not be able to decode what's about to be seen, unless the monkey is psychic, in which case either there is something wrong with your experiment, most likely. Or you should go to Wall Street with your monkey. But you know, you shouldn't get anything here. And then we should see some sort of increase here if there is information.

And this is kind of what it looks like from the results. So this is zero. After here, we should see information. This is chance, or 1 over 7. And so if we try this decoding experiment, what we find is during the baseline, our monkey is not psychic. But when we put on a stimulus, we can tell what it is pretty well, like almost perfectly. Pretty simple. All right, we can also do some statistics to tell you when the decoding results are above chance doing some sort of permutation test where we shuffle the labels and try to do the decoding on shuffled labels where we should get chance decoding performance. And then we can see where is our real result relative to chance, and get p values and things like that.

It's pretty simple. How does this stack up against other methods that people commonly use? So here's our decoding result. Here's another method. Here I'm applying an ANOVA to each

neuron individually and counting the number of neurons that are deemed to be selective. And so what you see is that there's basically no neurons in the baseline period. And then we have a huge number. OK, so it looks pretty much identical.

We can compute mutual information on each neuron and then average that together over a whole bunch of neurons. Again, looks pretty simple. Or similar, I should say. Or we can compute a selectivity index. Take the best stimulus, subtract from the worst stimulus, divide by the sum. Again, looks similar. So there's two takeaway messages here. First of all, why do decoding if all the other methods work just as well? And I'll show you in a bit, they don't always.

And then the other take away message though is as a reassurance, it is giving you the same thing, right? So you know we're not completely crazy. It's a sensible thing to do in the most basic case. One other thing decoding can give you that these other methods can't is something called a confusion matrix. So a confusion matrix, Rebecca kind of talked a little bit about related concepts, basically what you have is you have the true classes here. So this is what was actually shown on each trial. And this is what your classifier predicted.

So the diagonal elements mean correct predictions. There actually was a car shown and you predicted a car. But you can look at the off diagonal elements and you can see what was commonly made as a mistake. And this can tell you, oh, these two stimuli are represented in a similar way in a brain region, where the mistakes are happening.

So another kind of methods issue is, what is the effect of using different classifiers? If the method is highly dependent on the classifier you use, then that's not a good thing because you're not telling yourself anything about the data, but you're really telling you something about the method you use to extract that data. But in general, for at least simple decoding questions, it's pretty robust to the choice of classifier you would use.

So here is the maximum correlation coefficient classifier I told you about. Here's a support vector machine. You can see like almost everything looks similar. And like when there's something not working as well, it's generally a slight downward shift. So you get the same kind of estimation of how much information is in a brain region flowing as a function of time. But maybe your absolute accuracy is just a little bit lower if you're not using the optimal method.

But really, it seems like we're assessing what is in the data and not so much about the algorithm. So that was decoding basic information in terms of content. But I think one of the

most powerful things decoding can do is it can decode what I call abstract or invariant information where you can get an assessment of whether that's present. So what does that mean?

Well, basically you can think of something like the word hello. It has many different pronunciations in different languages. But if you speak these different languages, you can kind of translate that word into some sort of meaning that it's a greeting. And you know how to respond appropriately. So that's kind of a form of abstraction. It's going from very different sound concepts into some sort of abstract representation where I know how to respond appropriately by saying hello back in that language.

Or another example of this kind of abstraction or invariance is the invariance of the pose of a head. So for example, here is a bunch of pictures of Hillary Clinton. You can see her head is at very different angles. But we can still tell it's Hillary Clinton. So we have some sort of representation of Hillary that's abstracted from the exact pose of her head, and also abstracted from the color of her pantsuit. It's very highly abstract, right?

So that's pretty powerful to know how the brain is dropping information in order to build up these representations that are useful for behavior. And I think if we were, again, going to build intelligent robotic system, we'd want to build it to have representations that have become more abstract so it can perform correctly. So let's show you the example of how we can assess abstract representations in neural data. What I'm going to look at is position invariance. So this is similar to a study that was done in 2005 by Hung and Kreiman in *Science.*

And what I'm going to do here is I'm going to train the classifier with data at an upper location. So in this experiment, the stimuli was shown at three different locations. So on any given trial, one stimulus was shown at one location. And these three locations were used, so the 7 objects were all shown at the upper location, or at the middle, at the lower. And here I'm training the classifier using just the trials when the stimuli was shown in the upper location.

And then what we can do is we can then test the classifier on those trials where the stimuli were just shown at the lower location. And we can see, if we train at the upper location, does it generalize to the lower location. And if it does, it means there is a representation that's invariant to position. Does that make sense to everyone? So let's take a look at the results for training at the upper and testing at the lower. They're down here.

So here again, I'm training at the upper location. And this is the results from testing at the

lower. Here is chance. And you can see we're well above chance in the decoding. So it's generalizing from the upper location to the lower. We can also train at the upper and test at the same upper, at the middle location. And what we find is this pattern of results. So we're getting best results when we train and test at exactly the same position.

But we can see it does generalize to other positions as well. And so we can do this full permutations of things. So here we trained at the upper, we could also train at the middle, or train at the lower location. And here if we train at the middle, we get the best decoding performance when we decode at that same middle. But again, it's generalizing to the upper and lower locations, and the same for training at lower. Get the best performance testing lower, but it again generalizes.

So if you want to just conclude this one mini study here, you know, information in IT is position invariant but not you know 100%. So we can use this technique. I'll show you a few other examples of how it can be used in slightly more powerful ways, maybe, or to answer slightly more interesting questions. So what another question we might want to ask, actually we did ask in this paper that just came out, was about the question of pose invariant identity information, so that same question about can a brain region respond to Hillary Clinton regardless of where she's looking.

And so this is data recorded by Winrich Freiwald and Doris Tsao. Winrich probably already talked about this experiment. But what they did was they had the face system here where they found these little patches through fMRI that respond more to faces than other stimuli. They went in and they recorded from these patches. And in this study that we're going to look at, they did a-- they used these stimuli that had 25 different individuals shown at eight different head orientations.

So this is Doris at eight different head orientations, but there were 24 other people who also were shown. And so what I'm going to try to do is decode between the 25 different people and see, can it generalize if I train at one orientation and test at a different one. And the three brain regions we're going to use is the most posterior region. So in this case, the eyes out here, this is like V1. This is the ventral pathway. So the most posterior region, we can combine ML and MF. We compare that to AL and to AM. I'm going to see how much position variance is there.

So again, like I said, let's start by training on the left profile and then we can test on the left profile in different trials. Or we can test on a different set of images where the individuals were

looking straight forward. So here are the results from the most posterior region, ML/MF. What we see is if we train in the left profile and test on the left profile here, we're getting results that are above chance, as indicated by the lighter blue trace.

But if we train on the left profile and test in the straight results, we're getting results that are at chance. So this patch here is not showing very much pose invariance. So let's take a look at the rest of the results. So this is ML/MF. If we look at AL, what we see is, again, there's a big advantage for training and testing at that same orientation. But now we're seeing generalization to the other orientations. You're also seeing this "U" pattern where you're actually generalizing better from one profile to the opposite profile, which was reported in some of their earlier papers.

But yeah, here you're seeing, statistically, that is above chance. Now it's not huge, but it's above what you'd expect by chance. And if we look at AM as well, we're seeing a higher degree of invariance, again, a slight advantage to the exact pose, but still pretty good. Again, this "U" a little bit but yeah, we're going to the back of the head. So what would that tell you, the fact that it's going to the back of the head, tells you it's probably representing something about hair.

What I'm going to do next, rather than just training at the left profile, I'm going to take the results of training at each of the profiles and either testing at the same or testing at a different profile. And then I'm going to plot it as a function of time. So here are the results of training and testing at the same pose. So the non-invariant case. This is ML/MF. And this AL and AM. So this is going from the back of the head anterior.

And what you see is there is a kind of an increase in this pose-specific information. Here the increase is fairly small. But there is just generally more information as you're going down. But the big increase is really in this pose invariant information. When you train at one location and test at another, that's these red traces here. And here you can see it's really accelerating a lot. It's really that these areas downstream are maybe pooling over the different poses to create opposing invariant representation.

So to carry on with this for general concept of testing invariant representations or abstract representations, let me just give you one more example of that. Here was one of my earlier studies. What I did was this study was looking at categorization. It was a study done in Earl Miller's lab. David Friedman collected the data. And what they did was they trained a monkey

to group a bunch of images together and called them cats. And then to group a number of images together and called them dogs.

It wasn't clear that the images necessarily were more similar to each other within a category versus out of the category. But through this training, the monkeys could quite well group the images together in a delayed match to sample task. And so what I wanted to know was, is there information that is kind of about the animal's category that is abstracted away from the low level of visual features. OK, so was this learning process, did they build neural representations that are more similar to each other?

So what I did here was I trained the classifier on two of the prototype images. And then I tested it on a left out prototype. And so if it's making correct predictions here, then it is generalizing to something that would only be available in the data if the monkey had-- due to the monkey's training. Modulo any low level compounds. And so here is decoding of this abstract or invariant information from the two areas. And what you see, indeed, there seems to be this kind of grouping effect, where the category is represented both in IT and PFC in this abstract way. So the same method can be used to assess learning.

So just to summarize the neural content part, decoding offers a way to clearly see what information is there and how it is flowing through a brain region as a function of time. We can assess basic information and often it yields similar results to other methods. But we can also do things like assess abstract or invariant information, which is not really possible with other methods as far as I can see how to use those other methods.

So for neural coding, my motivation is the game poker. This one study I did. Basically, when I moved to Boston I learned how to play Texas Hold'em. It's a card game where, you know-- it's a variant of poker, I'm sure most of you know, I didn't know the rules before but I learned the rules. And I could play the game pretty successfully in terms of at least applying those rules correctly, not necessarily in terms of winning money. But I knew what to do.

And prior to that, I had known other games like Go Fish, or War, or whatever. And me learning how to play poker did not disrupt my ability to play go fish. I was still bad at that as well. So somehow this information that allowed me to play this game had to be added into my brain if we believe brains cause behavior. And so in this study, we're kind of getting at that question, what changed about a brain to allow it to perform a new task?

And so to do this in an experiment with monkeys, basically, they used a paradigm that had two

different phases to it. The first phase, what they did, was they had a monkey just do a passive fixation task. So what the monkey did was, there would be a fixation dot that came up. Up would come a stimulus. There would be a delay. There would be a second stimulus. And there would be a second delay.

And then there would be a reward. And the reward was given just for the monkey maintaining fixation. The monkey did not need to pay attention to what the stimuli were at all. And on some trials the stimuli was the same. On other trials, they were different. But the monkey did not need to care about that. So monkey does this passive task. They record like over 750 neurons from the prefrontal cortex. And then what they did was they trained the monkey to deal with delayed match to sample task.

And the delayed match to sample task ran very similar. So it had a fixation. There was a first stimulus. There was a delay, a second stimulus, a second delay. So up to this point, the sequence of stimuli was exactly the same. But now after the second delay, up came a choice target, a choice image, and the monkey needed to make a saccade to the green stimulus if these two stimuli were matches. And needed to make a saccade to the blue stimulus if they were different. And so what we wanted to know was when the monkey is performing this task, it needs to remember the stimuli and whether they were matched or not, is there a change in the monkey's brain.

And so the way we're going to get at this is, not surprisingly, doing a decoding approach. And what we do is we're going to use the same thing where we train to classify at one point in time, test, and move on. And what we should find is that we're going to try to decode whether to stimuli matched or did not match. And so at the time when the second stimulus was shown, we should have some sort of information about whether it was a match or non-match if any information is present. And we can see, was that information there before when the monkey was just passively fixating, or does that information come on only after training.

So here is a schematic of the results for decoding. It's a binary task, whether a trial was a match or a non-match. So chance is 50% if you were guessing. This light gray shaded region is the time when the first stimuli came on. This second region is the time the second stimulus came on. And here is where we're kind of going to ignore, this was either the monkey was making a choice or got a juice reward. We just ignore that.

So let's make this interactive. How many people thought there was-- or think there might be

information about whether the two stimuli match or do not match prior to the monkey doing the tasks, so just in the pacification task? Two, three, four, five-- how many people think there was not? OK, I'd say it's about a 50/50 split. OK, so let's look at the passive fixation task. And what we find is that there really wasn't any information. So there's no blue bar down here.

So as far as the decoding could tell, I cannot tell whether the two stimuli match or not match in the passive fixation. What about in the active delay match to sample task, how many people think-- it would be a pretty boring talk if there wasn't.

what area?

We're talking about dorsolateral-- actually, both dorsa and ventra lateral prefrontal cortex. Yeah, indeed there was information there. In fact, we could decode nearly perfectly from that brain region. So way up here at the time when the second stimulus was shown. So clearly performing the task, or learning how to perform the task, influenced what information was present in the prefrontal cortex.

I'm pretty convinced that this information is present and real. Now the question is, and why I'm using this as an example of coding, how did this information get added into the population. We believe it's there for real and probably contributing to behavior it's a pretty big effect. All right, so here is just some single neuron results. What I've plotted here is this is a measure of how much of the variability of a neuron is predicted about whether a trial is match or non-match.

And I've plotted each dot as a neuron. I've plotted each neuron at the time where it had this maximum value of being able to predict whether a trial is match or non-match. And so this is the passive case. And so this is kind of a null distribution because we didn't see any information present about match or non-match in the passive case. When the monkey was performing the delayed match to sample task, what you see is that there's kind of a small number of neurons that become selective after the second stimulus is shown.

So it seems like a few neurons are carrying a bunch of the information. Let's see if we can quantify this just maybe a little better using decoding. So what we're going to do is we're going to take the training set and we're going to do an ANOVA to find, let's say, the eight neurons that carry the most information out of the whole population. So the 750 neurons, let's just find the eight that had the smallest p value in an ANOVA.

And so we can find those neurons. And we can keep them. And we can delete all the other

neurons. And then now we found those neurons, we'll also go to the test set and we'll delete those neurons. And now we'll try doing the whole decoding procedure on the smaller population. And by deleting the neurons on the training set, we're not really biasing our results when we start doing the classification.

So here are the results using all 750 neurons that I showed you before. And here are the results using just the eight best neurons. And what you can see is that the eight best neurons are doing almost as well as using all 750 neurons. Now I should say, there might be a different eight best at each point in time because I'm shifting that bin around. But still, at any one point in time there are eight neurons that are really, really good.

So clearly there is kind of this compact or small subset of neurons that carry the whole information of the population. Once you've done that, you might not want to know the flip of that, how many redundant neurons are there that also carry that information. So here are the results, again, showing all 750 neurons as a comparison. And what I'm going to do now is I'm going to take those eight best neurons, find them in the training set, throw them out. I'm going to also throw another 120 of the best neurons just to get rid of a lot of stuff.

So I'm going to throw out the best 128. And then we'll look at the remaining neurons and see, is there redundant information in those neurons. It's still like 600 neurons or more. And so here are the results from that. What you see is that there is also redundant information in this kind of weaker tail. It's not quite as good as the eight best or not as high decoding accuracy, but there is redundant information to it. Just to summarize this part, what we see here is that there is a few neurons that really became highly, highly selective due to this process.

So we see that there's a lot of information in this small, compact set. Here are the results from a related experiment. This was in a task where the monkey had to remember the spatial location of a stimuli rather than what an image was, like a square or circle. But anyway, small detail. Here's this big effect of this is match information, this is non-match information being decoded. So these are the decoding results that I showed you before.

Here's an analysis where an ROC analysis was done on this data. So for each neuron, they calculated how well does an individual neuron separate the match and the non-match trials. And again, pre and post training. And what you see is here, they did not see this big split that I saw with the decoding. And this was published. So the question is, why did they not see it. And the reason is because there were only a few neurons that were really highly selective. That

was enough to drive the decoding but it wasn't enough if you averaged over all the neurons to see this effect.

So essentially, there's kind of like two populations here. There's a huge population of neurons that did pick up the match information, or picked it up very weakly. And then there's a small set of neurons that are very selective. And so if you take an average of the nonselective population, it's just here. Let's say this is the pre-training population. If you take an average of post-training over all the neurons, the average would shift slightly to the right. But it might not be very detectable from the pre-training amount of information.

But if you have weights on just the highly selective neurons, you see a huge effect. So it's really important that you don't average over all your neurons but you treat the neurons as individuals, or maybe classes, because they're doing different things. So the next coding question I wanted to ask was, is information contained in what I call a dynamic population code. OK, so let me explain what that means.

If we showed a stimulus, such as a kiwi, which I like showing, we saw that there might be a unique pattern for that kiwi. And that pattern is what enables me to discriminate between all the other stimuli and do the classification. But it might turn out that there's not just one pattern for that kiwi, but there's actually a sequence of patterns. So if we plotted the patterns in time, they would actually change. So it's a sequence of patterns that represents one thing.

And this kind of thing has been shown a little bit. And actually now it's been shown a lot. But when I first did this in 2008, the kind of one study I knew of that kind of showed this was this paper by Ofer Mazor and Gilles Laurent where they did kind of the PCA analysis. And this is in like the locusts, I think, olfactory bulb. And they showed that there were these kind of trajectories in space where a particular odor was represented by maybe different neurons.

And again, I had a paper in 2008 where I examined this. And there's a review paper by King and Dehaene in 2014 about this. And there's a lot of people looking at this now. So how can we get at this kind of thing in decoding? What you can do is you can train the classifier at one point in time, and test it at a point in time like we were doing before. But you can also test at other points in time. And so what happens is if you train at a point in time that should have the information, and things are contained in a static code where there's just one pattern, then if you test at other points in time, you should do well.

Because you capture that pattern where there's good information, you should do well at other

points in time. However, if it's a changing pattern of neural activity, then when you train at one point in time, you won't do well at other points in time. Does that make sense? So here are the results-- if that will go away. Let me just orient you here.

So this is the same experiment, you know, time of the first stimulus, time of the second stimulus, chance. This black trace is what we saw before that I was always plotting in red. This is the standard decoding when I trained and tested at each point in time. This blue trace is where I train here and I tested all other points in time. So if it's the case that there's one pattern coding the information, what you're going to find is that as soon as that information becomes present, it will fill out this whole curve.

Conversely, if it's changing, what you might see is just a localized information just at one spot. So let's take a look at the movie, if that moves out of the way. OK, here is the moment of truth. Information is rising. And what you see in this second delay period is clearly we see this little peak moving along. So it's not that there's just one pattern that contains information at all points in time. But in fact, it's a sequence of patterns that each contain that information.

So here are the results just plotted in a different format. This is what we call a temporal cross training plot because I train at one point and test at a different point in time. So this is the time I'm testing the classifier. This is the time I'm training the classifier. This is the passive fixation stage, so there was no information in the population. And this is just how I often plot it. What you see is there's this big diagonal band. Here you see it's like widening a bit so it might be hitting some sort of stationary point there. But you can see that clearly there's these dynamics happening.

And we can go and we can look at individual neurons. So these are actually the three most selective neurons. They're not randomly chosen. Red is the firing rate to the non-match trials. Blue is the firing rate to the match trials. This neuron has a pretty wide window of selectivity. This other neuron here has a really small window. There's just this little blip where it's more selective or has a higher firing rate to not match compared to match. And it's these neurons that have these little kind of blips that are giving rise to that dynamics.

Here's something else we can ask about with the paradigm of asking coding questions. What we're going to do here is we're going to try a bunch of different classifiers. And here, you know, these are some questions that kind of came up. But can we tweak the classifier to understand a little bit more about population code. So here is a fairly simple example. But I

compared three different classifiers. And the question I wanted to get at was, is information coded in the total activity of a population. Or is it coded more so in the relative activity of different neurons.

So you know, in particular, in the face patches, we see that information of all neurons increases to faces. But if you think about that from a-- or maybe not information, but the firing rate increases to all faces. But if the firing rate increases to all faces, you've lost dynamic range and you can't really tell what's happening for individual faces. So what I wanted to know was, how much information is coded by this overall shift versus patterns. So what I did here was I used a Poisson Naive Bayes classifier, which takes into account both the overall magnitude and also the patterns.

I used a classifier minimum angle that took only the patterns into account. And I used a classifier called the total population activity that only took the average activity of the whole population. This classifier's pretty dumb, but in a certain sense, it's what fMRI is doing, just averaging all your neurons together. So it's a little bit of a proxy. There's paper, also, by Elias Issa and Jim DiCarlo where they show that fMRI is actually fairly-- or somewhat strongly correlated with the average activity of a whole population.

So let's see how these classifiers compare to each other to see where the information is being coded in the activity. Again, I'm going to use this study from Doris and Winrich where we're going to be looking at the pose specific phase information, just as an example. So this is decoding those 25 individuals when they're shown, trained, and tested that exact same head pose. And so what we see is we see that when we use the Poisson Naive Bayes classifier that took the pattern and also the total activity into account, and when we used the classifier that took just the pattern into account, the minimum angle, we're getting similar results.

So the overall activity was not really adding much. But if you just use the overall activity by itself, it was pretty poor. So this is, again, touching on something about what Rebecca said, when you start averaging, you can lose a lot. And so you might be blind to a lot of what's going on if you're just using voxels. There is reasons to do invasive recordings. All right, and I think this might be my last point in terms of neural coding. But this is the question of the independent neuron code.

So is there more activity if you took into account the joint activity of all neurons simultaneously, so if you had simultaneous recordings and took that into account, versus the pseudo

populations I'm doing where you are treating each neuron as if they were statistically independent. And so this is a very, very simple analysis. Here I just did the decoding in an experiment where we had simultaneous recordings and compared it to using that same data but using pseudo populations on that data, using very simple classifiers. And so here are the results.

What I found was that in this one case there was a little bit extra information in the simultaneous recordings as compared to the pseudo populations. But you know, it wouldn't really change many of your conclusions about what's happening. It's like, you know, maybe a 5% increase or something. And then this has been seen in a lot of the literature. This is the question of temporal precision or what is sometimes called temporal coding.

What happens, you know, some of the experiments I was using 100 millisecond bin, sometimes I was using 500. What happens when you change the bin size? What happens, this is pretty clear, again, from a lot of studies that I've done, when you increase the bin size, generally the decoding accuracy goes up. What you lose is temporal precision, because now you're blurring over a much bigger area. So in terms of your understanding what's going on, you have to find the right point between having a very clear result by having a larger bin versus you caring about the time information and using a smaller bin.

And I haven't seen that I need like one millisecond resolution or a very complicated classifier that's taking every single spike time into account to help me. But again, I haven't explored this as fully as I could. So it would be interesting for someone to use a method [INAUDIBLE] that people really love to claim that things are coded in patterns in time. You know, if you want to, go for it. Show me it. I've got some data available. Build a classifier that does that and we can compare it. But I haven't seen it yet.

So a summary of the neural coding. Decoding allows you to examine many questions, such as is there a compact code. So is there just a few neurons that has all the information. Is there a dynamic code. So is the pattern of activity that's coding information changing in time. Are neurons independent or is there more information coded in their joint activity. And what is the temporal precision. And this is, again, not everything, there are many other questions you could ask.

Any other questions about the neural coding? Just a few other things to mention. So you know, I was talking all about, basically, spiking data. But you can also do decoding from MEG

data. So there was a great study by Leyla where she tried to decode from MEG signals. Here's just one example from that paper where she was trying to decode which letter of the alphabet, or at least 25 of the 26 letters, was shown to a subject, a human subject in an MEG scanner. You know, see is very nice, you know, people are not psychic either.

And then at the time, slightly after the stimulus is shown, you can decode quite well. And things are above chance. And then she went on to examine position invariance in different parts of the brain, the timing of that. So you can check out that paper as well. And as Rebecca mentioned, this kind of approach has really taken off in fMRI. Here are three different toolboxes you could use if you're doing fMRI.

So I wrote a toolbox I will talk about in a minute to do neural decoding, and I recommend it for that. But if you're going to do fMRI decoding, you probably are better off using one of these toolboxes because they have certain things that are fMRI specific, such as mapping back to voxels that my toolbox doesn't have. Although you could, in principle, throw fMRI data into my toolbox as well.

And then all these studies so far I've mentioned have had kind of structure where every trial is exactly the same length, as Tyler pointed out. And if you wanted to do something where it wasn't that structured that well, such as decoding from a rat running around a maze where it wasn't always doing things in the same amount of time, there's a toolbox that came out of Emory Brown's lab that should hopefully enable you to do some of those kinds of analyses.

All right, let me just briefly talk about some limitations to decoding, just like Rebecca did with the downer at the end. So some limitations are, this is a hypothesis-based method. So we have specific questions in mind that we want to test. And then we can assess whether those questions are answered or not, to a certain degree. So that's kind of a good thing but it's also a down thing. Like if we didn't think about the right question, then we're not going to see it.

So there could be a lot happening in our neural activity that we just didn't think to ask about. And so unsupervised learning methods might get at some of that. And you could see about how much is the variable of interest you're interested in, accounting for the total variability in a population. Also, I hinted at this throughout the talk, just because information is present doesn't mean it's used. The back of the head stuff might be an example of that or not, I don't know. But you just have to interpret the results and don't know the information there. Therefore, this is the brain region doing x.

A lot of stuff can kind of sneak in. Timing information can be also really interesting. I've been exploring this summer. So if you can know the relative timing, when information is in one brain region versus another, it can tell you a lot about kind of the flow of information the computation that brain regions might be doing. So I think that's another very promising area to explore. Also, decoding kind of focuses on the computational level or algorithmic level, or really neural representations if you thought about Marr's three levels. It doesn't talk about this kind of implementational mechanistic level. So [INAUDIBLE] it's not one thing it can do.

Now if you have the flow of information going through an area and you understand that well and what's being represented, I think you might be able to back out some of these mechanisms or processes of how that can be built up. But in and of itself, decoding doesn't give you that. Also, decoding methods, computationally, can be intensive. can take up to an hour. If you do something really complicated, it can take you a week to run something very elaborate. You know, sometimes it can be quick and you can do it in a few minutes, but it's certainly a lot slower than doing something like an activity index where you're done in two seconds and then you have the wrong answer right away.

Let me just spend like five more minutes talking about this toolbox and then you can all go work on your projects and do what you want to do. So this is a toolbox I made called the neural decoding toolbox. There's a paper about it in *Frontiers in Neuroinfomatics* in 2013. And the whole point of it was to try to make it easy for people to do these analyses because [INAUDIBLE]. And so basically, here is like six lines of code that if you ran it would do one of those analyses for you. And not only is it six lines of code, but it's almost literally these exact same six lines of code.

The only thing you'd, like, replace would be your data rather than this data file. And so what you can do, the whole idea behind it is it's a kind of open science idea, you know, I want more transparency so I'm sharing my code. If you use my code, ultimately, if you could share your data, that would be great because I think I wouldn't have been able to develop any of this stuff if people hadn't shared data with me. I think we'll make a lot more progress in science if we're open and share. There you go, I'm a hippy.

And here's the website for the toolbox, www.readout.info. Just talk briefly a little bit more about the toolbox. The way it was designed is around four abstract classes. So these are kind of major pieces or objects that you can kind of swap in and out. They're like components that allow you to do different things. So for example, one of the components is a data source. This

creates the training and test set of data. You can separate that out in different ways, like there's just a standard one but you can swap it out to do that invariance or abstract analysis. Or you can do things like, I guess, change the different binning schemes within that piece of code.

So that's one component you can swap in and out. Another one are these preprocessors. What they do is they apply pre-processing to your training data, and then use those parameters that were learned on the training set to do some mechanics to the test set as well. So for example, when I was selecting the best neurons, I used a preprocessor that just eliminated-- found good neurons in the training set, just used those, and then also eliminated those neurons in the test set. And so there are different, again, components you can swap in and out with that.

An obvious component you can swap in and out, classifiers. You could throw in a classifier that takes correlations into account or doesn't. Or do whatever you want here. You know, use some highly nonlinear or somewhat nonlinear thing and see is the brain doing it that way. And there's this final piece called cross validator. It basically runs the whole cross validation loop. It pulls data from the data source, creating training and test sets. It applies the future preprocessor. It trains the classifier and reports the results.

Generally, I've only written one of these and it's pretty long and does a lot of different things, like gives you different types of results. So not just is there information loss but gives you mutual information and all these other things. But again, if you wanted to, you could expand on that and do the cross-validation in different ways. If you wanted to get started on your own data, you just have to put your data in a fairly simple format. It's a format I call raster format. It's just in a raster.

So you just have trials going this way. Time going this way. And if it was spikes, it would just be the ones and zeros that happen on the different trials. If this was MEG data, you'd have your MEG actual continuous values in there. Again, trials and time. Or fMRI or whatever. fMRI might just be one vector if you didn't have any time. And so again, this is just blown up. This was trials. This is time. You can have the little ones where a spike occurred. And then what corresponds to each trial, you need to give the labels about what happened.

So you'd have just something called raster labels. It's a structure. And you'd say, OK, on the first trial I showed a flower. Second trial I showed a face. Third trial I showed a couch. And

these could be numbers or whatever you wanted. But it's just indicating different things are happening in different trials. And you can also have multiple ones of these. So if I want to decode position, I also have upper, middle, lower. And so you can use the same data and decode different types of things from that data set.

And then there's this final information that's kind of optional. It's just raster site info. So for each site you could have just meta information. This is the recording I made on January 14 and it was recorded from IT. So you just define these three things and then the toolbox plug and play. So with some experience you should be able to do that. So that's it. I want to thank the Center for Brains, Minds, Machines for funding this work. And all my collaborators who collected the data or who worked with me to analyze it. And there is the URL for the toolbox if you want to download it.