PROFESSOR: OK, welcome to RNA 2, which of course begins with a RNA 1 overview where we talked about the secondary and tertiary structure of RNA, and how one integrates dynamic programming in those algorithms. And then that is important in the way we go about measurements and it affects certain technical senses. And at an interpretation level, affects how we think about the quantitation of RNAm, which was the main topic last time.

And then today, after we have the data analyzed so that we have RNA quantitation and the random systematic errors established some idea of what the interpretation consequences are and maybe time series data, the question is, what do we do next? What we do next is basically two things. At least for today's topic.

We cluster, ask which gene expression products, whether they're RNA or protein, go up and down together. And if they go up and down together under a variety of conditions or time points in various conditions, then we want to know why. What is the mechanism by which they go up and down? And to what common goal are these gene products directed? In other words, two different whys. Why mechanistically. And why in terms of the way that they can help the entire system.

So in order to deal with clustering, we'll go into quite some detail about the options that we have for doing clustering. And you'll see they're quite a number of combinations. We'll go through distance of similarity measures, hierarchical to non-hierarchical clustering and classification. Now this is kind of the roadmap-- the overview of all the different decisions that we need to make in order to establish gene expression clustering.

Going from left to right, we've got data normalization choices, got distance metrics to choose from. Linkage methods, when we link two clusters together or two RNA types together, what methods do we use. And finally, the clustering method itself on the far right-hand side of slide three.

And then the working backwards from the clustering method, you've got two basic goals, you can think. Typically, when we think of clustering, we're mainly talking about unsupervised methods. That is to say where we're really letting the data tell us what it has to say, what gene expression products go together.

In a possibly an alternative or a [INAUDIBLE] to that would be to ask, can we use those discoveries in a sense to supervise classification? So rather than discovering what gene products go up and down together, ask those that go up and down together to use-- to allow us to classify the different conditions from which the gene expression has been ascertained. So classified as a pathological states, infectious states, cancer states and so on.

So now we're going to kind of work backwards from the unsupervised clustering methods and then move into distance metrics and linkage. So we're basically working from right to left on this chart. First, with an overview of what the goals of such quantitation classification methods should be, this has been in a previous lecture. But basically, we can start with the RNA data which we reduced to a table in the previous lecture.

You can think of it as a table of RNA expressions along the vertical axis and different conditions along the horizontal axis where we can have full change, say ratios or absolute levels. And we can do either clustering or classification. And when we do get to clustering and discovery, one of the things we can do is use motifs to get at direct causality.

These are just some buzzwords that you will find coming up in this lecture and problem set and outside. Just examples of the two types of goals of analyzing gene expression or even more general collections of quantitative data. Four examples-- four examples of unsupervised clustering, k-means clustering, self-organizing maps, single value decomposition, cluster [INAUDIBLE] analysis. You may have heard these in various contexts.

I'm lumping of all together here under this category. And we'll particularly delve into k-means as one example. We could delve into any of them, but we need to get some depth. And then just for your reference, here's some examples of the supervised learning if you were going to go into the classification.

Here's some examples of early attempts at clustering. These are particularly interesting to look at because they were early and very little prior literature. They tended to take a fresh fresher look at it than you might get in the most recent papers. Fewer assumptions and therefore, more exposition about where they feel clustering comes in from other fields and is applicable to this field.

The main dichotomy that I'm pointing out here is can cluster by gene. That is to say, by RNA or gene product or RNA protein. Or you can cluster by condition cell type or even time course. So you can think of that as-- by gene as this vertical axis, at least in the formats that most articles and this lecture will have it in. And then by condition will be your horizontal axis. Or you can do by clustering, which is clustering by both. And then down here is an example of one of many sources of free software that you can look at, both for microarray analysis and for clustering.

The general purpose of this is to divide samples into fairly homogeneous groups. Clearly, because of biological variation that can be meaningful or random, these will not be perfectly homogeneous. When we find the coregulated genes but some of the methods we talked about in previous classes, we'll want to know what the protein complexes are that are mechanistically regulating and the downstream functions of these.

Again, the major dichotomy among the unsupervised learning is whether you're doing hierarchical or non-hierarchical. We'll show an example of each. Typically, hierarchical is represented by a tree very similar to the trees that we have for sequence similarity and for pedigrees, phylogeny and so on. These are basically the terminal branches of the tree or the leaves of the tree are the individual RNA species representing a vector of different RNA quantitation.

With the non-hierarchical, you'll tend to have represented-- these are visual representations as well as underlying algorithms. They'll be represented more as a multidimensional envelope, say, a sphere or ellipse that tries to encompass a set of related gene expression values. Now we'll use diagrams like this-- mainly the two on the far left hand side of slide nine-- where you'll have a fairly tight circular or spherical clusters where it's pretty evident how they're connected. Or you can have more elongated or more inner penetrating clusters.

And how do we deal with these? The key terms that we'll try to define this is actually very similar to the ones we talked about before. We had either distance of similarity. These are in flip sides of the same coin. The greater the distance, than the less similarity. The dendrograms are the same kind of the trees that we've been seeing before.

Now the most general way of discussing distance measures is the Minkowski metric. This is actually a set of metrics. And what we're going to be talking about here are two objects which are really two-- for the purpose of discussion, two RNAs. Call them RNAx and RNAy, geneX and geneY, have key features. Meaning you have P different conditions, P time points. You will call them dimensions sometimes.

And so this means that gene expression of x under conditions one through P is it compared to the gene expression of y under those various conditions too. You can think of these as vectors with P entries in them. And so the distance is going to be some Rth root of a sum to the R power.

And we're going to go through three different examples of this. And I think you'll-- hopefully, by the time we've gone through it, you'll see the advantages of this general and the specific forms. So the three examples will have R equals 2, 1, and infinity on slide 12. These are the most common metrics. And you should see them as fairly familiar.

When R equals 2 in that formula, you now have the square root of the sum of squares. And this should remind you of your simple Cartesian plotting of the distance between two points on graph paper where you can take any diagonal, the shortest path. On the other hand, if you are navigating the streets of Manhattan, you will tend not to take diagonals through stone walls.

You'll tend to obey the blocks and you may have to go three blocks this way and four blocks that way rather than square roots. And then finally, the last one is the maximum distance you might have to go in any particular direction. So you can think that if you take the Rth root of the sum of the differences in these two measures, x and y-- measures of the two RNAs at the same condition-- that as R goes to infinity, you're going to wait up the biggest distance difference along all the different axes.

And then you'll take the R through to that. And then it'll basically be the absolute value of that difference. And so those are the three measures. But let's see some specific examples. Here we have two points. So you have-- this is the simplest possible case. Two RNAs under two different conditions.

And let's say on this arbitrary scale, the distance between y and x along the horizontal dimensions, which is say condition horizontal, is four and the condition vertical is three. That's the difference between them. And where it is absolutely relative the origin doesn't matter in any of these three metrics.

The diagonal-- the direct distance or Euclidean distance is going to be the square root of 4 squared plus 3 squared, which is going to be 5. And the Manhattan distance, you can't take that-- you can't go as the Crow flies. You have to go four blocks to the left and three blocks up. And that's seven.

And then the maximum of the two measures, if you think of these as many different measures, the biggest distance in any particular direction would be four. Now here's an example where the Manhattan distance is called the Hamming distance when all the features are binary. And why is this interesting?

I mentioned, I think, in the first lecture that many biologists and scientists in general, when they have the opportunity, will classify things as on and off even when there is some underlying quantitative nature. Transistor can be on or off for all intents and purposes. And a gene circuit or a particular gene expression can be considered off or on, 0 or 1.

And so now if you have say 17 different gene expression levels, this can be considered a 17-digit binary string. And the two genes, A and B, here can be compared. If you talk about distance rather than similarity, every time there's a conflict of 01 or 10, then you add that to the sum and you have a total of five of these cases where there's a difference. So the Hamming distance is five in this case.

So you can see that this has some intuitive appeal if you're going to be doing this Boolean system biology. Here's another one. This is a fourth measure of similarity or distance here. And we've brought it up before. The correlation coefficient. This is a way of comparing this vector of RNA expression levels x sub i with y.

So now instead of taking the difference between x and y, sub i, which is what we were doing with the Minkowski metrics, we're taking the product of those two. But if x and y are on some arbitrary scale, then we won't really have a way of comparing one experiment to another.

This is an example of normalization. We're going to use normalization a couple of different ways in this class. But they're all related in that you want to put them on a scale that's universally recognizable. Typically 0 to 1 or -1 to 1. In this case, -1 to 1.

And so what you do is in order to get it to the same center, you subtract the means from both the x and y. So now they're centroid is at 0 instead of at x bar, which is just the defined of the mean, as usual. And then to get the scale the same or on some commonly referenced scale, you divide by their product of the squares.

So the result of this, as we previously discussed correlation coefficient, is that the correlation coefficient varies between minus 1 and 1. If it is 1 on slide 16, it means that they are perfectly correlated. Which is, of course rare, but bear with us. If the gene products go up and down perfectly under all the conditions and all the time points that you look at, then they're going to get a linear correlation coefficient of 1.

If they're perfectly negatively correlated, then they will go up and down exactly out of phase or exactly when one is at this maximum, the other one will be at it's minimum. And if there's no linear correlation, then it will be a linear correlation coefficient of zero. Now there can be all kinds of complicated nonlinear relationships.

I mean, they could be very, very codependent, say, quadratic and still have a zero for their linear correlation coefficient. So exercise for the reader. Which of these is 1 minus 1 and 0? We'll start with the upper left hand one. Is that 1?

Minus 1. Good. And this one? 1 right. And zero. Great. And you will see that those have not been normalized because the correlation coefficient will do the normalization for us. In a moment, we will deal with-- we'll go back to Euclidean distances but we will do a normalization first. Now here's example of hierarchical clustering dendrograms-- just happened to be done for tumors and normal tissues.

And you can see the tumors designated by T tend to cluster together, and the normal tissues on the right hand side of slide 18 tend to cluster together. But it's not perfect. There's some interpenetration. You can see this would be a challenging classification problem.

The way that hierarchical tree was derived is you basically start by saying, each object-- gene-- and you're going to be measuring gene expression, which typically is RNA or protein. And you're going to call each individual RNA a cluster. It's a cluster of one. It's a trivial cluster.

And then as you look through each step in the hierarchical clustering, very similar to some of the greedy algorithms we use for sequence alignment, you take the two closest clusters even if they're a cluster of one and you'll merge them. And now I call that the new cluster. Now it's a cluster of two and so on and so forth. Until finally, everything is in a cluster and you've kept tract of all the who's closest to who all the way. And that produces a tree.

Now in order to generate that tree, you've got four other clustering methods. You've got the choice of the distance metric, the way of putting together the distances that you've measured. So the distance we measured can be some Minkowski or correlation coefficient. But you can put them together by either focusing on the nearest neighbor of the cluster or the furthest neighbor. That's the single link for this complete length. And we'll talk about that.

And then the other methods that we won't talk about are centroid, if you can think of the center of mass for the cluster as it emerges. And the average, which is just to say the mean of all the cross cluster pairs. If you got two clusters and you do all pairwise.

So let's do the single link versus the complete link. First, the single link in slide 21. And we're going to use exactly the same distance matrix for both of these examples. So you don't have to shift gears too much. The main thing is the only thing we're going to shift is between single and [INAUDIBLE].

And we're using Euclidean distance here, which is square root sum of square. And here you can see AB are the two closest and A and B are the furthest apart. And so the Euclidean distance for AB is 2 and AB is 6, for example. And so in the single length method, this kicks in once you start collapsing the first link.

So you make the link between A and B, that's obvious because it's the shortest distance. But how you collapse it depends on-- how you compare it to other points is what the single link method is about. So now AB is going to be treated as one unit-- one cluster. And you're going to ask, how far is AB from C?

Well, since this is a single link, you're interested in the closest distance and that's BC. And BC, from the very first leftmost matrix was three. So you fill in for the AB to C at three. And similarly the D is the closest point. From AB to D is five. It's the diagonal from B to D and so on. And that's how you've lost the top row, and it's three and five.

And now when you compare these, the next link you're going to make is going to be the smallest one in the whole table, which is three. And that happens to be the AB cluster is closest to C. And so that's going to be the next link you make. And then the rest of the game is over. It's just the ABC cluster is near D.

So you can already imagine in your mind what that tree is going to look like. A and B are going closest, and then you bring in C. And then you finally bring in D. And you might think at this point, that's the only way to do this. But the complete length version of this is exactly the same matrix.

You start up the same place. AB is still the closest one so that's the one you're going to link together first. But how you score it as you do this linking is a little different now. Because now you're concerned about completely all of the distances from the AB cluster to, say, C. Now B is close but A is far away.

And we're interested in that greater distance as well. And so the whole cluster of AB gets the distance from A to C, the longest distance, five. And so five goes in that position. And six goes as long as this is from AB to D, which again, is A to D. And so now you have a completely different-- just toggle back and forth between slide 22 and, 21 and you can see it went from three, five, four to five, six four.

So now when you make the next link-- the first link is obvious in both case, AB. The next link is now CD, because the smallest one in that two-by-two matrix is four. And that happens between C and D. And now C and D are the next link. And then now the game's over. You connect CD and AB and the link is-- so now you can see you're going to get two very different trees from the single link method on the left hand side of slide 23 is AB bringing in C and finally D.

While the complete link, you have AB and CD as two separate pairs and then they come together. Now this is the simplest possible example I could have come up with. But I think it combined with the next couple of slides will drive home the importance of the clustering method that you're using. Here the linkage method, part of it.

Again, focus in on the far left-hand side of where you have more compact spherical circular clusters or more elongated ones. We're going to take three examples here. Spherical, elongated, something in between. A single link in the middle of slide 25. And then complete link on the far right-hand side.

In a single length, Now, you can kind of see why they're called single link and complete link. This is a different way of visualizing them. Here the single length does a great job for the top and bottom clusters-- the circular and the linear forms. But when you start getting something that's somewhere in between, you get this weird single link that, at least to my eye, connects up the two clusters along the bottom here and then leaves this little cluster as the second cluster.

The complete link on the other hand, where you measure all the distances between previous clusters and the new clusters you're going to be adding goes well on the top one. And the middle one but does this weird thing with the elongated clusters where it takes a small cluster that seems, to my eye, to include things that are not that related. So the single link does well on the top and bottom and the complete link does well on the top middle.

And so you can see that depending on what you think your data are going to look like, whether they're going to look closely spaced but compact clusters that might be single length and more elongated but separated by distance, then you might want a complete link. So now where are we in this overall road map in slide 26?

We've been moving from the right where we've gone from clustering methods, supervised, unsupervised, hierarchical, non-hierarchical. We've gone through distance metrics and linkage metrics. Now let's see how it plays out with one particular non-hierarchical method. We've been focusing on hierarchical.

Now we're going to go non-hierarchical k-means and bring in issues of data normalization. In this case, gene normalization where we're trying to put genes that are wildly different in their absolute value of expression on the same scale. So one might be a very small fluctuation at a kind of medium level. Another one could be very large fluctuation from baseline up to a very high level.

And you want to account for this difference in baseline and this difference in scale. And so what you do-- and that's what all these three little normalized expression plots are. Is they represent this table, as I've mentioned, of genes along the vertical axis or gene expression levels-- genes that are going-- were we're going to measure expression levels along the vertical axis and the points or the conditions along the horizontal axis.

And so we have two representations. One is this kind of dot cluster envelope representation in the middle where you have, in this case, three dimensions. But in a case that's a little harder to visualize, multi dimensions-- 17, 15 dimensions. That's one representation where the origin is essentially the mean where you normalize it, the mean becomes zero.

And then the distance from that origin can be either positive or negative and it's the number of standard deviations from the mean. That's the way that we're going to normalize it. So each of these individual plots would be the average behavior in each of these clusters. And we'll take a look at that, the average and the deviation from the average.

But the units here in the vertical axis of these little plots will be normalized expression. Number of standard deviations within the cluster from the mean of the cluster. Now when we're going to be measuring distances between clusters where we have the same normalized expression data table-- and this is the three dimension-- this is the three dimension, in this case, or multi-dimensional representation. Where the origin is zero or the mean for each of for each of the axes and the distance from that zero mean is the number of standard deviations.

And when we'll measure it will measure this hidden distance the square root of sum of the squares over all the dimensions. And I want to emphasize that each of these clusters is not point-- if gene expression were regulated by transcription factors which bound to every site with exactly the same binding constant, then you might-- and if there were selection pressures forcing this to happen where the forcing everybody to be precisely-- everything in a cluster be precisely regulated, then these clusters would be really tight.

There'd be almost a point and there'd be no overlap between them. But in reality, there is no such selective pressures. And the transcription factors, as a result, are possibly purposefully diverse. And you get these spread clusters. And so these little blue bars on each of the points on these time series plots of normalized expression-- the three times three plots-- those little blue bars don't necessarily represent experimental error. They represent the diversity of gene expression within a cluster.

Now if you've accidentally made more-- assigned fewer clusters than is say the natural number of clusters, then you'll get more dispersion in that number than you might want. And that might be a tip off that you actually need more cluster-- you need to divide it up into more clusters and bring this down. Obviously, if you break it up in too many clusters that will have a different set of pathologies, you'll have the distance between clusters, some of the clusters will be abnormally close.

They'll be almost as if they were right touching each other. And so that's the tip off that you have too many clusters. And the number of clusters is something that you can either determine in advance or you can discover as you go. But those are the examples of criteria you might use.

Too much dispersion in those little blue air bars means that you've tried to lump too many things into one cluster. And too short a distance between adjacent clusters means that you probably divided them too finally. Now how do we begin to assess whether the clustering methods that we're using are optimal?

We've talked about all the different kind of clustering methods that you can use. One of the ways to assess whether they're are optimal-- we'll talk about many. But one is to look way outside the box to some resource that maybe the biological community has curated functions. Now they may mean this in very vague and frustrating ways, but we believe that they have done a good job.

And certainly an independent job of the experiment that's being done. The experiment that's being done is a fresh comprehensive gene expression analysis. And so if you find a cluster from a gene expression analysis that coincides with this completely independently curated database of functional categories, doesn't matter what [INAUDIBLE] means. This is some abbreviation for an Institute. Nor does it really matter what the gene names here mean.

But what you will find is that a particular set of genes, once you look it up in the database, will set off a flag that says ribosome. And you know what ribosome means. And others will be unknowns. But the point is that these will be an orderly set, a set that's perhaps enriched-- unexpectedly enriched. And you want to have some way of quantitative your surprise at finding this many of one type of function in your RNA cluster.

In a way, this is what you hope to find. It's a pleasant surprise. You want your clusters to have some coherence in their function. You also want to find some surprises, either unknowns or new combinations of functions that you didn't expect. Now this is an example of a clustering experiment. It's a popular way of representing it.

Here is the trees we've been talking about here that the closest-- the tips, the leaves here are individual genes. You can barely see them at this scale. This is a small subset of the human genes. This is a RNA expression that has been measured over time course of serum stimulation.

And considering the previous slide of different functional categorization, what you want as you hierarchically arrange these things, you've got time as the natural axis horizontally. And then you've tried to sort them so they're close together in the hierarchical tree. And you've and you've represented whether they're greatly induced or greatly suppressed during this serum stimulation.

You take one of the time-- zero time point is the reference point. And then greatly increase or decrease represented by red and green respectively. And then within each of these clusters, you have little zones where they all have the same kind of pattern of black, gray, green, and red. And so for example, the E at the bottom in the red zone here is wound healing and tissue remodeling.

And these are the genes that you might expect to be enriched in a growth stimulation paradigm, such as the one here where you're simulating with serum stimulation of fibroblasts. This is a particular example of how you might-- but you might want to quantitate this rather than just kind of showing it here. And we're going to walk through exactly how you do that quantitation in a moment.

This is just a quick snapshot of how far this clustering goes. Actually, it goes well beyond biology. But here's something that's for slide 32 out of the range of RNA expression. Here we have compounds on the vertical axis and targets, meaning proteins on the horizontal axis. And you can see all these connections between different cancer therapeutics, different cancer cell lines and potential targets.

But now back to the RNA. And we want to ask how do we assess the RNA array data collection, the clustering methods? And how do we go-- and how do we go beyond that in various directions, both as validation of the technical aspects but as showing that we're actually doing discovery and getting at mechanism. So one of the various methods we've used-- we already mentioned looking for functional categories, but another one is looking for motifs.

If we find a consistent set of motifs, this is part of the validation process as well. And these are some of the examples of algorithms. The first one that leaps to mind when mathematicians and physicists enter the field and that one that we've used a great advantage in the sequence searching part of this course was oligonucleotide frequency.

So you can use short oligonucleotides as convenient hashing keys or as ways of doing the lookup-- a very rapid lookup for sequences and in finding matches. And this is even more appropriate here for the motifs involved in transcriptional regulation because we from a variety of biological and chemical crystallographic studies that the motifs are in the range of 7 to 10 nucleotides often-- base pairs in double stranded DNA. And so you can use oligonucleotide frequencies.

However, they're limited in that they're not as rich as the weight matrices that we got when we've got a multi-sequence alignment. And when we were talking about multi-sequence alignments, we pointed out that it was hard to get the algorithms to scale beyond the pairwise. Because pairwise was n squared where n is the sequence length. And then as you go to multi-sequence alignment, it goes up exponentially with the number of sequences.

You want the number of sequences to be large though. Because the larger it is, the more you learn about the characteristics of that family of sequences. So anyway, Gibbs sampling was one of the methods that we said that we would put off to a later class. This is the later class. We'll talk about Gibbs sampling as a way of-- the idea of sampling this very large space where the large number of multi-sequences-- the multiple sequences you're comparing is that you don't want to get trapped in a local minimum.

You can have these really greedy steepest descent algorithms, but you'll get to the bottom of that pit but you won't necessarily find the global. If the sampling space is too large, even sampling won't save you because you'll sample a lot of little local [INAUDIBLE]. But, anyway, Gibbs is an example where you use randomization to find it.

Mean as example of maximization of expectations and [INAUDIBLE] and so forth are other ways of doing it. We're going to really focus in on one of these. Can't cover everything. We've talked about Gibbs sampling. And we want to put it in the context of-- and the thing that might be appealing.

Why can't we just-- if the program for transcription factor regulation is inherent in the genome, then we should just look at the genome sequence and be able to see patterns of motifs in front of genes. And then find clusters of genes that are expressed and so on. The problem with that picture-- even for one of the best case scenarios, [INAUDIBLE], which is about 12 mega bases-- as I said, these transcriptional control sites are about seven bases, let's say, of inflammation.

Here's one that will be a star for a few slides here today after now and I have break. This is GCM4. You can see it has five really full scale, two-bit conserved bases. And then the rest of the bases in this motif-- the other five bases might add up to another two bases of information or 14 bits all together.

Now 14 bits, you can think of that as 4 to the seventh power about 1 match every 16,000 bases. Now if you have a 12-megabase genome, and since it's not symmetric, you have to look at both strands. You have to think of the transcription factor scanning the DNA in both directions. Then you have 24 megabytes mega bases of sites. 24 million sites.

And at random, you expect 1 over 1,600. So you have a mean of 1,500. Now here we can bring in our old friend the Poisson distribution. And we will remember that the mean and the variance of a Poisson distribution are the same. And so the standard deviation is going to be the square root of variance, as it is for all variances and all standard deviations. And so the standard deviation is going to be about 40.

So if you expect to convince yourself that you have something interesting then you want it to be about two or three standard deviations above the mean. So your noise that you're fighting is about-- you want to get 2 and 1/2 times 40 or about 100 sites. Well, many biological phenomenon do not have 100 sites. They're not 100-- there may not be 100 GCM4 sites in the genome, for example.

And so what you need is a way of winnowing down the genomes. We're not looking through the whole genome but we're enriching in various ways. What are the various ways that we can enrich? Well, the first three we'll lumped together as ways that we can biologically cluster. Basically, that was the theme of the first few minutes of this lecture.

Ways that we can put together five genes that are-- where the gene expression products broken down together. And that would be, for example, whole genome [INAUDIBLE] data. That's the top line of slide 36. Or they could be-- and we had a little slide on this earlier of different ways that genes could show that they should go together. They could have a shared phenotype. You could do knockouts and they have similar biochemical or morphological characteristics.

And so you put them in the same functional category. That might be the source of some of the functional categories we've been talking about today. They can be conserved among different species. Species will inherit them-- will tend to inherit them as a group and others. So this is the example why genes should go together.

And then you'll reduce the sequence space to be the regulatory elements that go with those genes and not the rest of the genome. And those are the ways of selecting the genes. But then selecting the sequence itself near those genes or in those genes. You might want to eliminate protein coding regions, repetitive sequences or any other sequence is not likely to control sites.

This helps you by reducing your sequence space. That's kind of a trivial help. Actually, an important help. But in addition to that, you want-- they help you by removing traps where you're going to find motifs but that they are unlikely a priori to be relevant to transcriptional control. Which is what you're really trying to get at here to validate and to extend the discoveries you find from the unsupervised clustering.

And why do I say that? Why would protein coding regions and repetitive regions-- repetitive elements be a bias? Well, protein coding regions that for genes that cluster together, for some reason or other, likely a priori to have proteins that have similar functions. They're clustering together because they have similar functions. They might share protein domains in common.

So you will find nucleic acid motifs that are similar to one another, not because they're involved in regulation but because the geniculate code turns into protein motifs that are similar to one another. So they can accomplish a similar function. And that's why they-- and repetitive regions are definitely destined to give motifs in common because of their selfish replication properties.

The entire repetitive sequence from edge to edge will jump around the genome. And so there won't these little seven base pair motifs. They'll be a 10 kilo base motif. And that won't tell you much about transcription. Now having said that, we're in the business of sequence space reduction. Both the top three methods and this bottom method will exclude certain kinds of discoveries.

But once you find the motif by severely restricting a sequence, you can then search for that motif and pick up the examples that you might have eliminated in the first pass in a much less noisy manner. You've got this bona fide motif, now you want to find all the other examples. In a way, you're testing the specificity of the motif.

So for example, there could be RNA regulatory elements in protein coding regions. They could be some in repetitive regions. In the lecture that we gave on single nucleotide polymorphisms, I perversely chose a very interesting one that occurs in one of the most common dispersed repeats in the human genome, which is the ALU repeat. That one has regulatory significance, but we will exclude it from our search space initially so that we can get plenty of good examples in a small box.

So these are the main ways of reducing search space. And we're going to illuminate this with a particular algorithm-- a modification that gives motif sampling, which is this one where you sample the multi-sequence alignment states randomly so you don't get past the local minimum. And this is called a [INAUDIBLE] nucleic acid conserved elements. The emphasis on nucleic acid.

And what are the advantages to give a deep sampling [INAUDIBLE]? Why are we focusing in on it? Well, the [INAUDIBLE] sampling, as I said, keeps you out of local minimums. There are a number of sites per input sequence. It could be that in the genes that you've found in your cluster, some of them may have three of these motifs in front of it. Others will have one or even zero, because it could be that particular gene co-clusters is because of some other set of motifs that happen to have the same properties as the motif you're looking at it at any given moment.

So you can have zero to a large number of motifs. And that's important. This algorithm handles it. Other algorithms assume there's exactly one site per sequence. And that introduces noise. You can distribute the information content in various ways. You'll see, we can fine tune the shape of a motif in a way.

Some of these algorithms were based on proteins. Proteins have only one strand. They don't have a Watson and a Crick strand going in reverse complements one another. And so you need to make a conscious effort to adapt that algorithm so that it's-- that it, in a certain sense, recognizes the duality and the reverse complements of DNA strands. And you have to-- there's multiple distinct motifs that's different from the variable number of sites per sequence.

Once you find motif number one, it may be the dominant motif that you find again and again in a multi-sequence alignment. You have to go back and find number two. Because it could be number one isn't the only or isn't the major biologically significant motif. It could be any two or three motifs acting in concert.

So you can't just rest on your laurels when you find the first motif. And for each motif, there can be multiple examples of them per sequence. Anywhere from zero on up. So let's make this much more concrete and really drill down to a specific example.

This example-- the real example-- it's taken from the amino acid biosynthetic genes in the yeast saccharomyces. So here we've applied the two major classes to sequence reduction. The first is by biological function here. These are all amino acid biosynthetic genes, histidine, aromatic amino acids, [INAUDIBLE]. These are all on the right hand side of slide 39.

But in addition to the biological reduction of just maybe 116 genes that are involved in this process, we've also done the sequence space reduction near the gene to exclude the protein coding regions and only look at 300 to 600 bases upstream. Why 300 or 600? If the genes are really close together, you don't want to go much beyond 300 because you can enter the protein coding region of an adjacent gene.

If the genes are very far apart in this particular part of the genome, you don't want to get much more than 600 or else you'll end up in the repetitive sequences or other things that are other regulatory elements unrelated to your particular protein. Or you might end up in an RNA encoding gene. So 300 to 600 is good for this particular organism.

But you might need a different one for, say, human. You're going to have to look in introns and much further upstream, which makes it a much more difficult problem. Anyway this is the sequence reduction phase. And now let's say, well, do you see the motifs in here? I mean, those of you who are good computing should be able to do this algorithm in your head.

But here's the answer. And then we're going to-- now we're going to go through and we're going to say how we got to that answer with the Gibbs sampling alignments algorithm. The answer here is GCN4. This is the one we used to illustrate we have about seven bits of information here in this Snyder logo format. And on the lower right, it has a map score that we'll define soon enough.

Basically the higher the maps score, the better. It has to be greater than 0 to be non-random. And here's on the left hand side of slide 40 is the multisequence alignment, just like the multi sequence alignments we talked about in the last lecture-- two lectures ago. And here in red are all these arrows. They point either left to right or right to left, depending on which strand they're on so they're not exact reverse complements. Although, this does have a little bit of symmetry in it. But you can see that you have anywhere from one to two of these in front of each of these genes.

OK, so now how we get there? Let's go step by step. And some of you may find this algorithm counterintuitive at first so don't be surprised if it is. The first step is we randomly seed. We plop down, say, 10 more sequences 10 nucleotides long, arbitrarily picked that as our length and plop them down randomly on these sequences here. So we have represented seven of the 116 amino acid biosynthetic genes upstream regions here.

And we've just highlighted red arbitrarily two red, 10 [INAUDIBLE] on the top one, and then none on the second one, and then one on the third one, and so on. And then since those are given and which is the first position is given, then it's a trivial matter to line them up. Just take all the first positions and you take a sum, and that's the weight matrix.

Now you wouldn't expect since these were all randomly chosen for real sequences, you wouldn't expect this to be an astoundingly non-random weight matrix. And it's not. It has a maps score that's negative. And as I said, that's basically random. A few bases tend to stick their head up a little bit above the random noise of 0.25 if this were a random genome or whatever the base composition is.

And none of them are full scale it 2 bits. I'd say none of them are perfectly represented. So now what's the next step? That's the initial seeding and it gives you a flavor for what's going to happen next. But there's some interesting things that you can do to increase the chances of getting a good motif.

So the next thing you do is either you add another site. You add another 10 [INAUDIBLE]. So the top row of side 42-- the top sequence already has two, but you add another one. You add a third one. Sequence number, arrow four still doesn't have any. But you added a third one randomly at the top and now you've got two sequence alignments.

You really haven't been able to do anything up to this point. You've got now two multi-sequence alignments. And you ask, which one is better? Well, let's say the one on the right is a little bit better, the one you add of the sequence to is a little bit better. Now you don't just blind the program. It doesn't just blindly accept this as the better multi-sequence alignment.

It has a probability that you will accept this. And that's again, to keep you from going through a completely greedy algorithm. Every improvement is going to be probabilistic. But you will definitely very greatly tend to accept each improvement. So this was adding a sequence. That's how you might improve it.

Or you can remove one. You can add and remove another two from the top sequence here. Add one, remove one. And I asked you this if the multi sequence on the right is a little bit better. If it is, then you have a high probability of accepting those two. The add and the remove changes. These are adding or removing entire sequences. Keep going, adding and removing.

Another thing you can do is can say, well, maybe the important bases aren't all smack in a row-- 10 in a row. Maybe you want to make it a little bit longer? Maybe the motifs should be a little bit longer? Maybe some of the ones in the middle aren't important, so we'll toggle one of them off and move the columns over. So now the motifs are a little bit wider, but it still has the same number of columns.

And if that improves-- if that gives you a better map score, a greater surprise in a sense of probability that you've got-- that you would have this number of sites that are shared to this degree in this number of sequences, then you have a high probability of accepting that change. Now you're not just changing the collection of sequences you think belong to that motif family, but you're actually changing the structure of the elements that you're going to call the weight matrix. You're changing the column structure. And that's also probabilistic.

And out of all this randomness, given many cycles, you eventually get the best motif. This might be the best motif for this particular learning set. But now you want to get the second best motif. Because this isn't necessarily the biologically best motif. And this one may not act alone. It may have another one that's also enriched and it could be that their co-occurrence is even more significant than either one of them occurring singly.

So what do we do? And I think what we're going to do is we're going to take a little break. And then when we come back, your incredible curiosity will be satisfied as to how we get the second motif. So take a little break.