Today it is my honor to introduce, Dr. Atul Deshpande. Dr. Deshpande is a postdoctoral researcher in the lab of Dr. Elana Fertig in the department of oncology, at Johns Hopkins University. He has a PhD in electrical engineering from the University of Wisconsin-Madison, and his interests include the use of time series analysis and spatial statistics for modeling biological processes. He’s currently developing analysis techniques to use single cell and spacial multigenomics for the characterization of the tumor microenvironment and intracellular signaling networks. Welcome. (students applause) Well, thank you so much. And once I figure out my... Where my PowerPoint window is, we can start in earnest. Okay, yeah, thank you for the kind introduction. So, I’m Atul Deshpande, and today the title of my talk is exploring time and space for identifying gene interactions using single cell transcriptomics. So, what do time and space mean in the context of this talk? They refer to recent technological advances and the algorithms, which are the foundation...
for the projects I will be talking about. And the first advance is the ability to measure gene expression in individual cells. This in turn inspired development of algorithms that ordered these cells along the biological trajectory. Using these algorithms, we can observe changes in gene expression in a pseudo temporal reference for pseudo time, which is a measure of the progress of the biological process. The second is a more recent ability to measure gene expression within the spatial context of the tissue. But this we can analyze changes in gene expression as cellular neighborhoods change, or as the tissue type changes. So, before single cell transcriptomics, we would usually get one measurement of gene expression from a collected sample. And this is now called bulk RNA-seq in retroactively. However, as this measurement would just be an average of the population of cells, it would obscure information about the different cell types, or different cell states in the population. With single-cell RNA-seq, we can now measure gene expression in individual cells.
Depending on technology, this can range from a few hundred cells up to thousands of cells. And this allows us to observe the full heterogeneity of the cell population represented by gene expression. And using this high dimensional data we can characterize different cell types and cell states as gene expression vectors. So, one drawback of this technique is the issue of technical dropouts. Now, this is characterized by observing a lot of false zeroes, or zero inflated measurements, because we are unable to reliably measure the low iron accounts in individual cells. Now, the first project that I will discuss uses a single cell RNA-seq technology, or as it’s downstream of that. And it uses also downstream of algorithms, which order single cell data into trajectories, which represent the biology that they might be studying. For example, let’s say if you are... You have a dataset, which corresponds to stem cell differentiation, there are probably now 70 different trajectory inference methods depending on what kind of datasets you are studying,
what biology you want to study,
how big the dataset is,
or what the expected trajectory is
of the biology that you’re studying maybe.
And they attempt to order these cells based
on the expression of potentially
a few key marker genes, or how, which genes
are differentially expressed along
the biological process.
So, anytime you collect,
let’s say a single cell RNA-seq data,
you would find a mix of cells,
and that was the entire motivation
for doing this.
But that mix of cells would have
a range of cell states,
which could correspond to
from the beginning of the biological process,
to the very end of the biological process.
And what these algorithms are trying to do
is they’re trying to fit these cells
in their right place, in the biological process.
And once we do that, we can actually observe
the gene expression along this ordering.
And a lot of these methods also assign
a pseudo time to each cell,
which tells you how far along in the biology
they think, or they hypothesize that the cell is.
And so, the question that we wanted
to ask is given this pseudo temporal ordering
of the cells, which gives us
a gene expression dynamics in the pseudo temporal reference. Can we use these dynamics to infer gene regulatory networks? Or any directed networks from say, sets of genes to their targets. And the second question was whether the assigned pseudo time values help us in the network inference task. So, to make the, I guess, explanation more approachable, I will just use an example dataset. And as I explained, the concepts I've... We will just see what that means in terms of this dataset. So, this is a dataset from Semrau et al, and this is a single cell data from retinoic acid, driven differentiation. And in this mouse, embryonic stem cells differentiate into neuroectoderm and extraembryonic endoderm cells. Now the data as collected had nine samples, one before the differentiation starts and one after every six hours. So, you have data collected over 96 hours from nine samples, and each sample has 384 cells. So overall, I believe we have something like you can do the math. I guess, 2,600 cells or something like that. So, we chose to apply two trajectory inference methods to this.
So, the first one is monocle 2, which is also called Monocle DDR tree, I believe. And the second one is PAGA Tree. So, both of these methods identify a bifurcating trajectory from these cells. And so, the first one is to the left where the embryonic stem cells are actually on the right of... I'm not sure if people can see my mouse pointer, but yeah, they’re on the right of the trajectory. And then, towards the bottom left, you go into a neuroectoderm state and towards the... Right, top left, you go into an endoderm state. And on the right side, the way PAGA Tree infers trajectory is you have the embryonic stem cells on the top left. And then, it identifies a few more branches than Monocle does. But both of these identify branching trajectories. And in each case we selected the two branches, which corresponded to markers, which were, which ended up being high for neuroectoderm. So, the trajectories, the sub trajectories from each method that we’ve wanted to study was the embryonic stem cells to neuroectoderm, using these two methods. So, this as in, so we had... We have these two trajectory inference methods,
which assigned their own pseudo times,
and this is the pseudo temporal expression
dynamics for the same gene.
I did not mark which gene it was, but yeah,
so this was for the same gene.
And you can see that the dynamics
each of these trajectories gives
us is different.
First of all, the main branch,
or sub part of the trajectory that
we are considering has
a different number of cells.
And these cells may not necessarily be common
to both end.
There will be some which are common
to both of these trajectories,
but some others which are completely different.
But also, that the cell ordering itself
each method based on whatever mathematics
they use, or whatever algorithms they use,
would differ between these two methods.
So, as you see, Monocle has a higher expression
much earlier in the pseudo time,
as opposed to PAGA Tree, which has much later.
And the pseudo times here,
were not exactly 100, they’re just nominalized
to 100 just represent progress from 0%
of the biology to 100% of the biology,
or as inferred by that method.
So, now what are the challenges associated
with order single-cell data?
So, the first one is that unlike say, stock data, or say weather data, or something like that, you don’t necessarily have a uniform distribution of cells. And if you’re going to do a time series analysis, that would mean that you do not have regularly spaced time series, but you actually have irregularly spaced time series. On top of that, the pseudo time values that are assigned to the cells and ordering stem cells is uncertain. Now, finally, we recall that we had the issue of zero inflated measurements, or false zeroes in the meter because of technical dropouts. So, the question is how to overcome all these drawbacks to try and find networks from this time series data. So, the project that we had resulted in basically an algorithm called SINGE, which is single cell inference of networks from Granger ensembles. So, this was done at the Morgridge Institute for Research in Madison, Wisconsin. And these are my collaborators on this project. And let’s see, okay. So, the main concept that we build on is basically the Granger causality test.
It was introduced by Clive Granger in 1960s. And to give a very simple example of what it’s trying to say is, let’s say if you have two times series X and Y, now Granger causality tests, whether the prediction of current values of Y improves by using past values of X, in addition to past values of Y. And if that happens, then we say that X Granger causes Y. So, this is basically a lag regression between X and Y.

And if that happens, then we say that X Granger causes Y. So, this has had applications in econometrics and finance, and is also being used in computational neuroscience and biology, as noted in these examples here.

Now, the multivariate Granger causality test can be thought of as setting up and solving a vector, or regression model, where you have say, P genes, T time points and L lags. Where L lags is telling you how many, say your relationships with the past expressions you’re trying to model. And once you have that, you could think of solving this way, our model by just minimizing this objective function here. And that would give you, I guess, a few edges between the past values.
of all of the genes and your target gene.
Okay, maybe I should have explained
So, you have all the regular,
all the possible regulators of a gene,
and then you have a target gene,
and you’re trying to identify
what explains what past values
of any of these genes explains
the current values of the target gene.
And if you wanted to have
a sparse representation of this network,
or have an...
Count only a few of the edges,
you would introduce this by CT parameter,
which would ensure that the edges from say,
all of these genes to your target
are not numerous.
And you can explain the biology in a few edges.
Now, to counter the irregularity
of the time series, we use
an idea called Generalized Lasso Granger.
So, what this does is,
I’m not sure, maybe I have...
Yeah, okay, so just to recall, right?
So, you have a pseudo temporal data,
which has irregular time series,
you have missing values,
which show up as zeros here, right?
So, we want to adapt the Lasso Granger test
for irregular time series.
So, what was previously, basically coefficients from older samples in regular time series, now becomes coefficients from just timestamps in the past.

Because you might not necessarily have a sample at that point. Furthermore, we can rethink basically, the object to function as originally, if it was a dot predict between the coefficients and the values of the gene expression, we rethink that as a weighted dot predict, where basically we...

And this is the description of the weighted dot predict, where you use a Gaussian kernel to weight the inputs pseudo product based on their proximity to the timestamps that you... That correspond to these coefficients.

So, these ellipses here show kernels, where basically, if you have a timestamp corresponding to coefficient and you have no sample at that timestamp, that doesn’t necessarily mean that the input to the gene predict it is zero.
So, basically what you would do is you would just look at a bin around that timestamp, and weight input from regulators, depending on their proximity to this timestamp. If the sample is exactly at the timestamp that you expect, you would rate it highly based on discussion kernel, and the farther you move away from the timestamp, the weaker the rate of that particular sample would be. If there are say more than one cells in close proximity, it would take input from all of them. If there are no cells in the close proximity to at least take input from some cells, which are farther away, and so on. So, yeah, as in this works with irregular time series, because you don’t necessarily have to expect samples in the past at the timestamps that you wanted them to. And yeah, I think we already discussed this. So, now, as in going back to the case for... So, we had these false zeroes, right? So now, because of this kernel method, we have an inherent imputation over missing data. So, now we get what we could think of as, instead of taking all of the zeros as they are at face value,
we can treat them, or some of them as dropouts, as just missing data. And we just remove those samples now, because we can now work with irregular time series.

And because of this kernel method, we can actually work with time signature, all uniquely irregular.

We can work with... We can remove the zero valued samples and get a different, differently irregular time series for each of these genes.

And so, such an action can probably be informed by imputation techniques like magic, which help you complete, or impute zeros in the dataset. So, instead of imputing the dataset, you could just use its output to decide whether or not to remove the data from, or remove that zero from this input dataset.

So, this is just an illustration of a single generalized Lasso Granger test. So, you have the POU5F1 gene, and it’s basically, you see it’s the cells corresponding to that, or other details expression along pseudo time.

And what you also see is two trendlines predicted using a Lambda of 0.1, which is basically a sparsity constraint of 0.1. So, it would have fewer edges between the regulators and POU5F1.
And then a Lambda of 0.02, which has far more regulators. And you can see that both of these predict the trends of POU5F1 when using the past values quite well.

So, now that was just one GLG test. Now, what SINGE does, is it performs multiple such GLG tests where you sub-sample the time series different ways to get different irregular time series again. And you also use diverse hyper-parameters to effectively using these two combinations, slice the cake multiple ways and trying to look at the data. So, the type of barometers that we use are Lambda, which determines the sparsity of the network that we get, or get into metrics that we get. And we have Delta T, which gives us a time resolution of the lags between say, the past regulators and the current target timestamps and the number of likes that you have. So together, they will tell you how far behind in pseudo time should you be looking to try to predict the expression of the target. And finally, the kernel width, which tells how far, how wide the width should be around the timestamp that you are considering. Now, once we get adjacency matrices from all of these,
we get, we considered them as partial networks, and we get ranked lists from each of them. And we aggregate these rank lists using a modified border count. So, border count is something which has been used in election. It’s basically an election, I guess, result aggregating strategy, where if you have five candidates, you rank them from one to five, and then the person who has the lowest number here over all of the people that voted, they would win the vote. So, the modified border width is basically the same concept, but the only change that we did was we wanted to place more weight to a ranking, which distinguishes between say a one, the first interaction we find with the 10th interaction we find. As opposed to say, the 10,000th interaction we find with the 10,010th interaction. So, that’s why the weighting before adding these border weights is one over N squared, as opposed to say, N here. So, yeah, once we aggregate this, we get a final rank list. And so, we had to do in for trajectories, we got gene dynamics from them.
and now that results in two different networks. And there’s just showing the top 100 edges from Monocle 2 and PAGA Tree.

Now, you can obviously see that they look very different. Some of the edges I think, are common, but they can be very, very different.

So, now the question is, which of these is right, or better? So, for that we would have to first think of, okay, how do we evaluate this?

So, one way to evaluate that would be to do a precision recall evaluation. So, let’s say we have this rank list of candidate gene interactions that we just got from SINGE and a gold standard, which knows the truth. As we go down this rank list, the precision metric tells us what fraction of the prediction so far have been correct. And the recall metric tells us how many of the total interactions in the gold standard, which were correct have so far been covered.

So, the figure on the right shows a precision recall curve for two rank lists. The ideal precision recall curve would place all the edges in the gold standard at the top of the list.
So, that’s the dotted line that you see here, and the area under that precision we call curve (mumbles) blue one.

A random list in expectation would be flat. So, and it would have a precision recall curve, and the area under that curve would be 0.5.

And in this example, we can see that the precision we call curve of A, which I guess, the predictor A is better because it starts off with having more ones, or as in a high precision, and then falls as opposed to B, which rises as opposed to B, which rises from a low precision.

What it means that A gets more hits in the top of its list as opposed to B, and so on.

And so, one way to also evaluate these position we call curves is to just look at the area under the curve, which is so A here is 0.7 and B’s 0.52.

And that tells us that on an average A ranks edges better as opposed to B. Now, we would like to use near this, and the question is what could we use as a gold standard?

Now, this is real biological data that we are using, and for that, we would also need to look into the literature to find validation.
So, one good source of information is the escape database curated by the Ma’ayan lab. And this database includes the results of loss of function and gain of experiments done on genes, and also ChIP-seq experiments, which identify binding sites of transcription factors. Now, the problem being that even this database is incomplete because the gaps in biological knowledge remain and doesn’t, I guess over the time, it would be completed, filled more and more. But when we were doing this evaluation, we had to deal with what was effectively a partial gold standard, or an incomplete gold standard. So, we had these two methods and two pseudo times, which we got from that. So, what we wanted, what we did is we compared the performance of SINGE using say, Monocle 2 and the pseudo time, as well as Monocle 2 with only the ordering. And some of the least PAGA Tree fed the pseudo time and PAGA Tree with only the ordering. And so, this is how the precision recall curves...
So, we look at the average precision, which is the same thing as the area under the precision recall curve. And we also look at the average precision in the early part of the precision recall curve. And the point for that being that, in a usual workflow, you would have a combination method which would point to some important edges, and then, you would potentially tell a collaborator to try and experimentally validate that. And in that sense, you would be giving them results from the top of your list, as opposed to trying to tell how well the 10,000th edge in the list is placed in the rankings. So, with that in mind, we also look at what’s the average early precision of these curves. And for that, we basically say what happened, as to what extent is the precision maintained until 10% of the genes are regarded in the list that we have. So, the figure to the right shows a scatterplot of these, the average precision and the average early precision for these four methods, for these four options.
And what we see is that the best performing combination is using Monocle’s ordering, but not its pseudo time, and Monocle applying the pseudo time that it order, that it assigns to the cells, actually degrades the performance quite a bit. And both of the PAGA Tree options with, or without pseudo time, are in between these. So, now why would this happen? For example, and let’s take an extreme case, right? And okay, before that, there’s not necessarily something that’s wrong with Monocle, but it’s basically that for this dataset, in this instance, the pseudo time values did not necessarily make a lot of sense. So, let’s say you have perfectly ordered cells. And for the first half of the cells, you just assign a value very close to zero and the second half, to zero and the second half, you assign a value very close to one. So, even though the ordering of the cells was quite nice and reliable, just because we ended up assigning a value to the pseudo times, often times, which is completely unrealistic. We might end up losing a lot of information that we otherwise had in the dataset,
0:29:44.32 → 0:29:45.27 or in the ordering.
0:29:48.92 → 0:29:52.88 So, yeah, as an extended,
0:29:52.88 → 0:29:55.52 the ideas from this particular figure, right?
0:29:55.52 → 0:29:57.24 So, you have two methods,
0:29:57.24 → 0:29:59.03 they’re giving you two different...
0:30:00.01 → 0:30:01.323 Okay, two methods with their orderings
0:30:01.323 → 0:30:04.76 and pseudo times, so basically four cases,
0:30:04.76 → 0:30:07.59 and they all give you different rankings,
0:30:07.59 → 0:30:12.39 which have different performances
0:30:12.39 → 0:30:14.41 in terms of network evaluation.
0:30:14.41 → 0:30:16.67 And in a sense, you could say
0:30:18.711 → 0:30:21.96 that each of these PAGA Tree inference methods
0:30:21.96 → 0:30:24.87 itself with all their inefficiencies
0:30:24.87 → 0:30:27.99 and efficiencies are only partially looking
0:30:27.99 → 0:30:29.99 at the biological data.
0:30:29.99 → 0:30:33.73 So, from that perspective, each
0:30:33.73 → 0:30:37.15 of these orderings and pseudo time values
0:30:37.15 → 0:30:38.56 can be considered as sources
0:30:38.56 → 0:30:39.98 of noisy information,
0:30:39.98 → 0:30:41.58 or noisy sources of information.
0:30:42.43 → 0:30:46.357 So, instead of trying to just infer
0:30:46.357 → 0:30:52.1 one pseudo time trajectory from
0:30:52.1 → 0:30:54.81 the dataset and finding the network,
0:30:54.81 → 0:30:55.94 or say another, and finding
0:30:55.94 → 0:30:58.48 the network from that, we could think
0:30:58.48 → 0:31:01.38 of the trajectory inference method itself
0:31:01.38 → 0:31:03.14 as an additional hyper parameter
0:31:03.14 → 0:31:06.31 on top of the sparsity, and kernel bits,
0:31:06.31 → 0:31:07.52 and so on.
0:31:07.52 → 0:31:10.29 So, instead of aggregating at this point
0:31:10.29 → 0:31:11.9 after just one trajectory inference method,
we could just say that maybe we have four trajectory inference methods in the beginning. And after that, we do all of these sub sampling and application of hyper-parameters, and multiple tests. And then, we aggregate over all of these results across trajectory inference methods. So, hopefully what that would do is that would account for all the inefficiencies, or counter then inefficiencies of individual trajectory inference methods, and give us a more robust network at the end. And I have not, I guess, shown our comparisons for the other methods, which obviously isn’t in our paper. We are doing better than them. So, but you can have a look at that in the paper if you’re interested, because I just wanted to conceptually focus on these ideas a little bit more. So, I guess, one problem with trying to run four different, or five different trajectory inference methods is depending on what kind of data set you have and what kind of biology you are studying, you might not necessarily have to try only four methods. You will probably have to try multiple methods before,
which let’s say, if you know it’s a branching trajectory, you end up seeing a branching trajectory. And each of these methods would have their own input data format, up data formats, visualizations, and all of these other intricacies. So, if anyone is looking to do a lot of trajectory inference methods, I would strongly encourage you to look at that. So, these in this project, they have streamlined the use of, I think, 55 trajectory inference methods. So, you don’t necessarily need to install each one of them. You just install this project and they run each of these methods using a docker. And so, what it also helps you do is it helps you visualize all of these trajectories and evaluate them using the same, I guess, support scripts and support functions, which they also provide. And in all this, this would make your lives quite easy.
depending on what biology you want to study.
How many cells you have, what compute power you might have access to, and so on.
So, okay just some final comments on the use of, I guess, the utility of trajectory inference and pseudo times for further analysis.
And so, first of all, as in trajectories look really nice, they visually,
they give us a lot of information.
And so, based on what we saw,
we did see that there’s some,
the ordering information can help
in network inference.
The good pseudo times can help a little bit,
but if you have exceptionally bad pseudo times, it can hurt a lot as opposed to ordering.
And not every dataset is really suitable for trajectory inference.
What do I mean by that?
The dataset that I chose, I guess a lot of what is...
What particular inference methods are built around, as say,
stem cell differentiation in general, where it’s as in the biology is quite neat to begin with.
As in you start off from a single cell type, and a lot of the biology is already known.
So, you don’t have to worry, you know that it’s going to be a branching,
or bifurcating, or multi furcating trajectory. So, you know that the quality of the biology, you know what cell states to exist, to expect, and so on, and so forth. You know the markers of each of those. And so, studying something like that is much more easier using trajectory inference, or pseudo time. On the other hand, let’s say, if you had a sample from a cancer tumor in that you would find cancer cells, normal cells, a bunch of immune cells, probably 10 to 20 kinds of immune cells, and so on. So, the trajectory inference method usually tracks, or predicts places, cell states and context. Not cell types themselves. So, you wouldn’t necessarily be able to reliably run a trajectory inference method across as in using a mix of different cell types, as opposed to cell states. Now, with the stem cell differentiation, the good thing is that the cell states themselves after a point, transition into different cell types, because it’s the same cell, or same cell type which transitions through multiple cell types, through these cell states. But that’s not the case with cancer biology,
where you already start off with a mix of cell types and trajectory inference would not make sense for that mix. What people have tried is isolate, just say a T-cell type, and then try to order, or find the trajectory only for those T-cells. And there has been some success in that. So, you could run trajectory inference for a subset of the dataset, but not necessarily the entire dataset. And so, depending on what biological processes you want to study, there are trajectory inference methods, which may or may not be suitable for it. For example, a number of methods like Monocle and PAGA Tree, they try to find tree-like structures in the trajectories, so they would not be suitable for a cyclic biological process like just maintenance processes in cells. And then, there are other methods which actually try to find cell cycles, and they would not be appropriate for branching processes. And I guess, as a no single trajectory inference method, accurately represents the biology. So, it’s all basically some mathematical abstraction.
0:38:28.78 -> 0:38:31.133 of what might be happening in the cells.
0:38:35.216 -> 0:38:36.049 And yeah, as an if...
0:38:36.049 -> 0:38:37.17 If at the outset, you know
0:38:37.17 -> 0:38:41.09 what kind of trajectory to expect, then it helps
0:38:41.09 -> 0:38:42.24 in trying to
0:38:44.77 -> 0:38:45.91 at least first really,
0:38:45.91 -> 0:38:49.79 say whether the trajectory that you’re getting
0:38:49.79 -> 0:38:52.06 and the pseudo times that you get
0:38:52.06 -> 0:38:55.03 is of any worth.
0:38:55.03 -> 0:38:57.92 So, just to give you an example.
0:38:57.92 -> 0:39:00.29 So, we started off with Monocle 2
0:39:00.29 -> 0:39:02.56 as one of our examples in our paper,
0:39:02.56 -> 0:39:05.33 and then we wanted to have another method
0:39:05.33 -> 0:39:07.413 to compare the effects of different
0:39:07.413 -> 0:39:09.89 trajectory inference methods.
0:39:09.89 -> 0:39:12.85 And PAGA Tree was not necessarily the first one.
0:39:12.85 -> 0:39:14.29 We tried a number of other ones,
0:39:14.29 -> 0:39:16.08 which did not.
0:39:16.08 -> 0:39:18.38 And we knew what to expect here.
0:39:18.38 -> 0:39:21.16 We knew that there was stem cell
0:39:21.16 -> 0:39:26.16 to ectoderm trajectory and endoderm trajectory,
0:39:26.43 -> 0:39:27.98 or a branch of that.
0:39:27.98 -> 0:39:32.98 And using basically, just the first,
0:39:35.04 -> 0:39:37.56 I think we tried four methods
0:39:37.56 -> 0:39:39.4 and PAGA Tree was basically the fourth method,
0:39:39.4 -> 0:39:41.69 which gave us that kind of branching trajectory,
0:39:41.69 -> 0:39:44.87 or branching topology for the biology.
0:39:44.87 -> 0:39:49.25 And so, none of the methods you try
0:39:49.25 -> 0:39:51.803 might necessarily mean anything,
0:39:53.01 -> 0:39:55.503 unless you have some way of validating that.
0:39:56.52 -> 0:39:59.01 So, at this point, I’m gonna switch
0:39:59.01 –> 0:40:04.01 to spatial expression,
0:40:04.07 –> 0:40:05.82 or a spatial data and special analysis.
0:40:05.82 –> 0:40:08.23 So, if you have any questions
0:40:08.23 –> 0:40:12.38 about the pseudo time analysis,
0:40:12.38 –> 0:40:13.813 should we take it now, or?
0:40:19.01 –> 0:40:20.27 Does anybody have any questions
0:40:20.27 –> 0:40:22.753 on the first half of the presentation here?
0:40:26.265 –> 0:40:27.29 Oh, we can continue on,
0:40:27.29 –> 0:40:29.707 then we can come back later.
0:40:34.439 –> 0:40:35.689 Shall we go on?
0:40:40.67 –> 0:40:42.148 Sounds good.
0:40:42.148 –> 0:40:44.148 Okay.
0:40:47.73 –> 0:40:49.653 Okay, so that was all about,
0:40:51.94 –> 0:40:56.94 say how pseudo time is used in our analysis.
0:40:56.94 –> 0:41:01.617 And so, the other end of,
0:41:03.17 –> 0:41:04.47 I guess, not necessarily end,
0:41:04.47 –> 0:41:05.31 the other perspective
0:41:05.31 –> 0:41:09.6 is how is space important and how,
0:41:09.6 –> 0:41:11.05 what kind of data do we have,
0:41:12.54 –> 0:41:14.843 which give us information about space?
0:41:15.76 –> 0:41:18.45 So, the spatial context of cells
0:41:18.45 –> 0:41:21.57 is very important in many biological processes.
0:41:21.57 –> 0:41:23.69 For example, when immune cells respond
0:41:23.69 –> 0:41:26.67 to an infection, or a wound, they need
0:41:26.67 –> 0:41:28.92 to be in physical proximity of their targets.
0:41:31.01 –> 0:41:33.84 Similarly with, I guess, cancer tumor growth,
0:41:33.84 –> 0:41:37.56 and the immune response to cancer
0:41:37.56 –> 0:41:39.72 happen through intracellular signaling.
0:41:39.72 –> 0:41:42.39 Either through cytokine secretion,
0:41:42.39 –> 0:41:45.563 or through surface receptors on adjacent cells.
0:41:48.41 –> 0:41:50.17 Just knowing the relative location

28
of different cell types can also be very informative. For example, in the figure here, the information about the presence of various immune cell types nearest tumor, and the extent of immune deficient in the tumor are essential prognostic markers. And so, single cell RNA-seq, as good as it is, it associates a cell from its tissue, due to which we lose the spatial context of the cell states. But in recent years, we have been able to develop both as in spatial proteomics, which help you to image protein and densities of say, up to 30 markers at single cell resolution in the tissue. As well as spatial transcriptomics, which can measure 20,000 genes at spots in the tissue. And this was named method of the year last year in 2020, yeah, that was last year. So, this includes 5,000 barcoded spots on slide. These are added to the cells in the... Which are located in those spots. And this helps preserve the spatial context of the cells to the actual sequencing.
Now, this technology is not exactly single cell. It still provides a lot of useful spatial detail. So yeah, for explaining this project, I will use the 10x Visium sample provided by 10x genomics of a breast cancer tissue. So, the figure on the left is an H and E slide, it’s hematoxylin and eosin stain slide, which helps pathologists annotate the sample for tumor, and lesions, and so on. And the second image is that slide annotated by a pathologist, and you can see that there are different biology’s in this one slide. And for example, the lesion on top is an invasive cancer lesion, which means it can spread beyond the breast tissue, but the other lesions correspond to DCAs lesions, which are not yet classified as invasive, they could in the future be invasive. Other important annotations are those of immune cells and the stromal cells in between these lesions. For a good clinical outcome, you would hope that immune cells can infiltrate these lesions. And so the figure on the right shows the same H and E slide with overlaid Visium spots. So, each of these spots correspond
0:45:06.87 -> 0:45:08.133 to one measurement.
0:45:09.51 -> 0:45:14.4 So, this slide shows a couple of examples
0:45:14.4 -> 0:45:16.9 of spacial gene expression.
0:45:16.9 -> 0:45:18.55 So, the figure to the left
0:45:18.55 -> 0:45:21.23 is the same annotated H and E slide
0:45:21.23 -> 0:45:23.48 that will help us keep track
0:45:23.48 -> 0:45:26.87 of the biology in the slide.
0:45:26.87 -> 0:45:29.9 And so, the first figure, the middle figure,
0:45:29.9 -> 0:45:32.77 basically it shows the expression of CD8A,
0:45:32.77 -> 0:45:35.6 which is a marker of cytotoxic T-cells.
0:45:35.6 -> 0:45:37.02 Now, we see this gene expressed
0:45:37.02 -> 0:45:42.02 in the blood near the invasive and DCAs lesions,
0:45:42.48 -> 0:45:43.81 which means that the immune cells
0:45:43.81 -> 0:45:44.9 are responding to a tumor.
0:45:44.9 -> 0:45:46.55 However, we see that
0:45:46.55 -> 0:45:48.74 there’s not much infiltration of these cells
0:45:48.74 -> 0:45:49.743 within the lesions.
0:45:50.92 -> 0:45:53.99 The second marker is CD14, which is found
0:45:53.99 -> 0:45:55.74 in macrophages and dendritic cells,
0:45:56.63 -> 0:45:58.31 and its expression is much higher
0:45:58.31 -> 0:45:59.73 inside the lesions, which could point
0:45:59.73 -> 0:46:03.67 to successful infiltration of these cell types.
0:46:03.67 -> 0:46:08.24 Now, just a reminder, these the measurements
0:46:08.24 -> 0:46:09.87 that we get from 10x Visium
0:46:09.87 -> 0:46:13 are not exactly single cell, but they’re near,
0:46:14.66 -> 0:46:16.49 In a sense that each of these spots
0:46:16.49 -> 0:46:18.433 is 55 micro meters wide.
0:46:19.36 -> 0:46:21.69 And depending on what cell type
0:46:21.69 -> 0:46:22.91 you might have in that spot,
0:46:22.91 -> 0:46:26.92 it could have anywhere from one to 10 cells.
And immune cells are much smaller, so there could be up to 10 immune cells in it, but maybe only one cancer, or epithelial cell in that spot. So, as a result of gene expression of that spot is the average of the cells inside it.

Our lab has a method called CoGAPS, oesophageal CoGAPS, which is a Bayesian Markov chain Monte Carlo method for nonnegative matrix factorization. And so, as a result of the 10x Visium measurement, we now have a high dimensional matrix with 20,000 genes and around 5,000 spots. And what CoGAPS does is it helps to factorize this matrix into two low rank matrices, both of which are non-negative, which correspond to latent patterns in the data. And in the past, we have seen that these two correspond to biology’s based on the pattern markers. So, the two matrices that CoGAPS factorizes the dataset into are the amplitude matrix, which has say, 20,000 rows for 20,000 genes and N columns for the end patterns. And this helps us identify groups of co-expressed genes, which correspond to the patterns. And the pattern matrix has N rows.
and they associate the spots on the sample with patterns. So, because of the nature of the CoGAPS, factorization, and these, the columns of the matrices here, or the rows of the matrices here are not really orthogonal. They are independent, but not orthogonal. So, they could co-exist in spots, or a gene could be present in multiple processes, or a gene could be present in multiple processes, and multiple patterns, which correspond to processes. So, when we apply CoGAPS to the Visium data, the first try was basically just five patterns, and when we apply it to try and find five patterns after a factorization, we see that a number of them correspond to the pathology annotations that we see on the figure on the left. We find a pattern which corresponds to the immune cells. We find a pattern which corresponds to invasive carcinoma on the top left here. And we also find a pattern which corresponds to the DCAs lesions. And as we increase the dimensionality of CoGAPS factorization, we start seeing more and more tissue heterogeneity. For example, we now see three patterns which are associated with the mesial carcinoma, and we can see that they correspond.
to different regions in that lesion.
And this for example is completely internal,
which has no interaction with immune cells.
We have a pattern which corresponds
to immune cells, we have a pattern
which corresponds to the stromal cells.
And we also have different patterns
which highlight individual DCAs lesions.
So, one could say that potentially it’s trying,
it is finding biology’s,
which are unique to these DCAs lesions.
So, we can analyze the A matrix
to identify groups of genes associated
with each pattern, and we call these
the pattern markers.
And these help us identify pathways
that are likely expressed in these patterns,
especially in this sample,
we see a one to one association
between the pattern and the biology,
also in the biology, basically.
So, let’s see, how long do we have.
I think we’re close to...
I’ll quickly rush through these.
So, the other analysis that we can do is given,
let’s say two of these patterns,
we can try to see how these patterns interact.
So, you can see that these patterns
have a lot of spatial structure to it,
which CoGAPS was not told about.
CoGAPS, the parameters that Co-GAPS uses
have no special information, and it’s still found these spatial structures. So, and we also see that these patterns are adjacent to each other and we want to see how they interact. So, what we do is we find, basically we estimate the kernel density of each of these patterns, which is a function of both the pattern intensity at a spot, as well as the spatial clustering of hyper intensities.

And we compare that against another distribution obtained by the density estimation after randomizing the locations of these pattern densities. So, the intensities which are beyond distal distribution are the ones that we... Are the spots which correspond to these outliers are the ones that we count as hotspots of pattern activity. Similarly, we can find the hotspots of immune response.

And when we combine both of them, we find regions where cancer is active, regions where immune cells are active, and regions where both of them are active. And this is the interaction region. And in this region, we are trying to find genes which correspond to this interaction between cancer and immune, and which are not necessarily markers of...
And regular markers of cancer and immune.
So, genes which are specifically related
to the non-linear interactions
between these patterns.
And to that end, basically we hypothesize
that since CoGAPS is already
an approximation of the dataset
with a linear combination of the patterns,
the residuals of CoGAPS,
of the CoGAPS estimate from the dataset
could point us to the non-linear interactions
between the patterns.
And we are only looking at the region
where both of the patterns are active
and comparing the residuals of CoGAPS
in that region to the residuals
in only the cancer region,
and only the immune region.
And now, this can be done for each of these,
I guess, pattern combinations,
and we can find what corresponds
to pattern interaction
between these pairs of patterns.
So, for future work, as part
of the data collection in clinical trials,
we’re already collecting both spacial
and single cell transcriptomics
and proteomics from patients.
So, we are trying to integrate all
of this into one big dataset,
which would represent the tumor microenvironment,
which would help us characterize the patient sample as a whole. And we would also like to infer intracellular signaling networks the same way as we were trying to do it using time, but now using space where intracellular signaling is a function of the distance between the cells and the types of neighboring cells for a target cell.

The learnings from these projects would go into a spatial temporal model of tumor growth and response to therapy, which can be used into building a digital patient or digital clone, where we can try to test what therapies might work on what patients.

So, these are the people who have been, and of course, 10x Genomics who were kind enough to give us the sample for studying, as well as my collaborators on this project. Thank you so much.

And I can take questions now, sorry for the overshooting time.

Do we have any questions to look at? People on Zoom? Yeah, question (mumbles). Going back to the time series slides. Mm-hmm.
Can you talk about how you know if you have good, or bad pseudo times? And is there a way to fix bad pseudo times? So, yeah, as in what I’ve not shared on here, we also, we knew for example, that we were studying... We wanted to study a trajectory which goes from stem cells to neuroectoderm, and we had markers. And I think, some (mumbles) themselves. They have identified markers of stem cells neuroectoderm  and endoderm cells. So, if we’re looking at the trajectories of the markers along the pseudo time to see if those make sense. For example, a marker which is supposed to be high in stem cells would, should be tapering down to zero along pseudo time, and a marker, which is supposed to be high in neuroectoderm should be increasing with pseudo time. So, we had, I think six oral markers to each of stem cells, neuroectoderm and endoderm cells. And we were trying to confirm the combination that neuroectoderm markers increase with pseudo time, but the other two decrease, or the endoderm shouldn’t decrease necessarily, but it shouldn’t have
a monotonic increase like the neuroectoderm one.
And it should not be present in the initial.
Does that...
So, that was one way to do it, basically.
Thank you.
Any other questions?
So, with the combination of many cells,
and the spatial stuff, is there any hope
of getting a temporal signal out of any of that,
or is that (indistinct)?
In spatial did you mean?
Yeah.
So, I think,
the issue would be, I guess,
not in clinical, I suppose.
In a sense that, okay, are you thinking
about pseudo temporal, or just clinical?
Yeah.
Pseudo temporal, I think there might
be some possibility,
and I’ve been thinking of
as in, we would still have to isolate,
I guess, cell types, for example.
So, one of the problems with that
is that as I mentioned,
the spots are not exactly single cell, right?
So, especially, let’s say if you’re trying
to do a pseudo temporal ordering
of CD8 T-cells,
they are more,
much likely than not, co-localized
0:58:44.46 –> 0:58:48.83 with other cell types, which would also,
0:58:48.83 –> 0:58:51.25 I guess, corrupt the expression
0:58:51.25 –> 0:58:53.03 that you are seeing.
0:58:53.03 –> 0:58:57.18 So, that would make it slightly different.
0:58:57.18 –> 0:59:01.3 We could think of ordering the spot
0:59:01.3 –> 0:59:02.944 as a whole, basically.
0:59:02.944 –> 0:59:03.777 And my…
0:59:04.91 –> 0:59:07.12 I belong to a school of thought that basically,
0:59:07.12 –> 0:59:08.446 if you have a…
0:59:08.446 –> 0:59:09.6 And then, so what people try to do
0:59:09.6 –> 0:59:13 with say, this kind of data,
0:59:13 –> 0:59:14.56 this spacial Visium data,
0:59:14.56 –> 0:59:16.83 where you have say, up to 10 cells,
0:59:16.83 –> 0:59:21.83 they try to resolve this into cell types.
0:59:22.73 –> 0:59:25.01 So, they would compare that to, there is I think,
0:59:25.01 –> 0:59:27.703 one paper called RTCD, or RCTD.
0:59:29.94 –> 0:59:32.223 RCTD robust cell type decomposition.
0:59:33.47 –> 0:59:35.51 So, what they do is basically,
0:59:35.51 –> 0:59:37.87 they take the spatial data,
0:59:37.87 –> 0:59:41.06 they have a reference single cell data,
0:59:41.06 –> 0:59:46.06 and they try to assign each spot,
0:59:46.99 –> 0:59:51.03 or a resolve each spot into a mixture
0:59:51.03 –> 0:59:53.93 of the cell types that might exist
0:59:53.93 –> 0:59:55.543 in the single cell data.
0:59:56.75 –> 1:00:01.01 And that could help you to say,
1:00:01.01 –> 1:00:04.38 identify what the mixture in general is.
1:00:04.38 –> 1:00:08.96 But my as in my thought is that we could
1:00:08.96 –> 1:00:12.97 just think of each spot as some representation
1:00:15.2 –> 1:00:16.82 of the biology in that neighborhood.
1:00:16.82 –> 1:00:19.71 So, each spot could just represent
1:00:19.71 –> 1:00:22.36 a neighborhood, as opposed to trying to find
what the individual cells are.
And that would basically abstract out
the representation and the biology to that
of the spots.
And we’ll have to think about how to do that,
but I think there could be some ordering to that,
but we’ll need to see what makes sense.
And then, for a lot of cells, cell states,
they are quite well-characterized.
For example, if you say that a T-cell
is activated, or a T-cell as naive,
or exhausted, you know what markers to expect.
But what would you be able to say
for spots instead?
The other thing to think of is,
especially with say, the proteomics as well,
where you can get actual single cell
and distributions, and neighborhood characterization.
You could think of it as can you,
so the same thing that...
The same ideas that were used
for pseudo temporal ordering of cells,
can they be used for pseudo temporal
ordering of neighborhoods?
For example, if you have a cell neighborhood,
which as they’re presented as whatever,
the central cell, and it’s five neighbors.
Now, depending on, are they all tumor?
Then maybe they have...
They’re basically deep in the cancer,
which has never been visited by an immune cell,
1:01:58.14 –> 1:01:59.38 is that a mix of tumor
1:01:59.38 –> 1:02:01.57 and activated immune cells?
1:02:01.57 –> 1:02:03.93 So, that is basically an active tumor
1:02:03.93 –> 1:02:06.15 immune interaction that’s happening.
1:02:06.15 –> 1:02:10.22 Is that exhausted T-cells and tumor,
1:02:10.22 –> 1:02:11.1 where basically the tumor
1:02:11.1 –> 1:02:15.69 has fought back and tried to suppress the...
1:02:15.69 –> 1:02:16.93 Or it’s basically sent signals
1:02:16.93 –> 1:02:21.13 to suppress the immune response, and so on.
1:02:21.13 –> 1:02:22.433 So, perhaps there could be
1:02:22.433 –> 1:02:24.73 a trajectory of neighborhoods,
1:02:24.73 –> 1:02:28.81 where you could say that depending on all
1:02:28.81 –> 1:02:31.13 the possible combinations that you expect
1:02:31.13 –> 1:02:33.453 in cellular neighborhoods,
1:02:35.33 –> 1:02:39.96 this current neighborhood is this far along
1:02:39.96 –> 1:02:42.68 that process, or that branch of a process.
1:02:44.393 –> 1:02:46.621 That was a long and winding answer.
1:02:46.621 –> 1:02:47.74 (chuckles) I don’t know if
1:02:48.68 –> 1:02:51.69 that necessarily answered it. Thank you.
1:02:51.69 –> 1:02:54.49 Thank you, any last questions?
1:02:54.49 –> 1:02:55.77 I wanna be mindful of time.
1:02:55.77 –> 1:02:58.333 Any questions that come to you, or?
1:03:06.287 –> 1:03:09.277 All right, well if not, thank you again.
1:03:09.277 –> 1:03:11.168 (students applaud) We really appreciate that.
1:03:11.168 –> 1:03:14.752 Thank you a lot.
1:03:14.752 –> 1:03:15.977 You have a wonderful (indistinct).
1:03:15.977 –> 1:03:16.81 Mm-hmm.
1:03:20.394 –> 1:03:24.394 (lecturer mumbles indistinctly)
1:03:26.984 –> 1:03:31.067 (students chatter indistinctly)