COVID Information Commons (CIC) Research Lightning Talk

Transcript of a Presentation by Alexander Niema Moshiri (University of California San Diego), November 13, 2020

Title: RAPID: Real-time phylogenetic inference and transmission cluster analysis of COVID-19

Alexander N Moshiri CIC Profile

NSF Award #: 2028040

YouTube Recording with Slides

November 2020 CIC Webinar Information

Transcript Editor: Macy Moujabber

Transcript:

Alexander Moshiri:

Slide 1

Cool. Thank you. So hopefully people see my slides. So yeah I'm Niema Moshiri. I'm at UC San Diego. I'm in the Computer Science and Engineering Department and I'll be talking about a recent development that we've made called viral MSA.

Slide 2

Basically a standard phylogenetics workflow- if you've ever seen an evolutionary history of virus sequences, you kind of- you start with these initial sequences at the beginning that don't necessarily line up very nicely. And the first step is you do multiple sequence alignment to get them to line up. Lining them up like this tells us stuff about, you know, what is the sequence homology, what are the relationships between the sequences.

Using this alignment, you then estimate a phylogeny. And then you root the phylogeny to determine what is the common ancestor. And then there's a bunch of other downstream analyses that you might be interested in doing, you know, things like what are the clusters of outbreaks that are happening you know what demographics do we think are more at risk of getting infected by some disease? Yeah, a lot of stuff that you can do. So in general my focus is on these two steps: the multiple sequence alignment and the phylogenetic inference which are really the computational bottlenecks. And in this talk, I'll be focusing just on this single step: multiple sequence alignment.

Slide 3

So the multiple sequence alignment problem is NP-Complete, which, TLDR, that means we don't have a polynomial time solution, and as a result, many heuristics have been developed to approximate solutions. They're pretty accurate reasonably. Some of the tools that people might be familiar with are MUSCLE, ClustalOmega, and MAFFT. These are some of the common tools that implement them. However, even these heuristics generally scale quadratically with respect to the number of sequences. So in the case of SARS-CoV-2, we've had an exponential growth of sequencing happening where, I mean, this is a great thing for us. We're getting more and more sequence data, but the downside is we have to analyze those sequence data and our tools just don't scale properly. So right now- so this is actually outdated. This was from I think a week or two ago. Now we're almost at 200,000 sequences. So the tools just do not scale to enable real-time analysis.

Slide 4

So, what if instead, we knew in advance so *a priori* we knew that our sequences are going to be super similar and we already have some high confidence representative reference genome against which we can compare them to? So here, I'm showing the reference genome as a green line on the top and each of those other colored sequences below it are the sequence that I want to align with each other. Instead of aligning them amongst themselves, what I can do is just align the first sequence against the reference genome. Align the second sequence against the reference genome. Keep aligning each of the sequences independently directly against the reference genome. And then using the positions of the reference genome as anchors, I can merge the individual pairwise alignments into one multiple sequence line. So take the first column of my multiple sequence assignment see- in the first sequence this is the position that matched, then this position matched in the second, and the third, and then just collapse them together. I can do this for each position of the reference genome and then now I have a multiple sequence line. And the nice thing is that this parallelizes super nicely. Every single sequence can be aligned against the reference independently and simultaneously, and it scales linearly with the number of sequences rather than quadratically.

Slide 5

But let me take a step back real quick and think about this approach. My input is a reference genome and a bunch of sequences that are very similar to that reference genome, and my output is an alignment of every sequence against the reference genome. So if people are at all familiar with long read sequencing, this is exactly the same computational problem as mapping long reads to a reference genome. So my question was: can I leverage existing well-implemented read mapping tools to enable this type of scalable reference guided multiple sequence alignment?

Slide 6

So I developed a tool called ViralMSA which wraps around existing read mappers in order to perform reference-guided multiple sequence alignment. I wrap around a few of them, but there's one tool specifically, Minimap2, that's kind of the gold standard right now for what I'm doing. So I only recommend using my tool with that specific read mapper, but I wrap around a few of them to demonstrate that I can evolve this tool naturally as read mapping technologies evolve as well. So basically, you simply provide viral MSA reference genome and the sequences to align and then it'll handle indexing the reference genome and doing any pre-processing it needs to do, and it'll call the read mapper and then merge the results into a multiple sequence alignment.

Slide 7

And here you can see a comparison of this approach with existing best practice tools. So my tool is the blue line on the bottom and then the two other lines are other tools and you see that, in general, it's multiple orders of magnitude faster than what people are generally doing right now, and it scales very nicely. And the sequence alignments that we get are very accurate.

Slide 8

So in conclusion, this tool that I've developed enables rapid multiple sequence alignment of viral genomes. It's open source. You can get it online, and hopefully if you do any viral analyses consider using it.

Slide 9

So some acknowledgements: Heng Li developed Minimap2, which is kind of the essence of the speed and this work was supported by NSF and Google. And yeah, I'll leave questions for the chat or later in the session.