And while this is the main goal of the project, this is very much a work in progress on both of these fronts, but hopefully what we've got will still be interesting.

A common scenario in both population and comparative genomics, especially for non-model organisms, is to have sequenced reads from a bunch of samples or species and one high-quality genome assembly. Since we've spent all this time getting this one high quality reference, its often most efficient to just map the reads from these samples to it. And with this strategy we can hopefully capture short variation like SNPs and indels, between our samples.

And there's a lot we can do with this information: we can estimate divergence or heterozygosity, we can infer substitution rates and phylogenetic relatedness, or lift over annotations from the reference genome to identify functional sites we think may have experienced positive selection.

However, these samples exist at varying levels of divergence from the reference, which we may not know exactly ahead of time,

Then, if read mapping is effective, when we map the reads and call variants to estimate divergence, we should see roughly the same estimates as their actual divergence.

So my question is: Is this what actually happens?

And to explain the reason we think we may not be accurately capturing variation with read mapping I'll show you another cartoon, which outlines the read mapping process, so we have our raw reads and our reference genome, and when we align the reads we can see where the sample differs from the reference, these are our inferred variants.

Since read mapping is sequence alignment, its effectiveness is based on sequence similarity. That means reads that contain lots of actual variants may be unmapped simply because of those variants. This is called reference bias and means that those highly diverged regions may not be reflected in the final set of variants called, so we may expect lower estimates of divergence.

Consequently if we could map those reads, we would incorporate that variation into our estimates and they would be more accurate.

There's some evidence to suggest this may be happening. In some of our projects around the lab specifically we've seen that diverged samples have lower percentages of reads mapped. So here we just

need to pay attention to the x-axis which shows the percent of reads mapped to the reference for a bunch of samples.

In this case the purple points are samples of the burrowing owl, one of which is also the reference genome used for read mapping

This one green point belongs to a different species but was still mapped to the burrowing owl genome, and you can see a shift on the x-axis. While all the burrowing owl samples have almost 100% of reads mapped, this sample has about 80%.

And this is pervasive across many re-sequencing projects.

And these unmapped reads can have affects on downstream analyses. I'm not aware of many explicit examples of this in the literature, but this one is really nice. In this paper, Armstrong et al. show that when different species of felids are mapped to a genome from their own species in blue, estimates of heterozygosity tend to go up compared to when they are all mapped to the domestic cat genome in red.

This may be a case of the variation in some reads preventing them from mapping to a diverged reference, but being able to be mapped in a closer reference and subsequently those variants being captured in these estimates of heterozygosity.

Strangely enough, the only other explicit test for reference bias I've found show's an opposite pattern. Here, Prasad et al. map the genome of the beluga whale to reference genomes from various other species, here listed on the x-axis in increasing levels of divergence from the beluga. They actuallay observe increased heterozygosity with increased divergence, unlike what Armstrong et al. saw in the cats. They attribute this to possible mis-alignments, and various other methods they tried suggest different results.

SO clearly reference bias is a problem, but we don't really know a lot about it, and we wanted to rigorously quantify it by using simulations.

So I'll briefly outline our simulation workflow. First, we needed to simulate diverged genomes from which we'd simulate reads to map.

And we did this based off of the mouse reference genome. We could've chosen any genome to do this, but I'm familiar with the mouse genome and it is chromosome level and well annotated, so this seemed good.

And we inserted differences at between 2 and 10% of sites.

Next we took each of these genomes and simulated short reads from them. And for now we've only done 30X coverage and 0.5% heterozygosity.

For both of these steps we used this read simulator called NEAT.

Besides attributing the software, the main reason I'm mentioning this so prominently is because NEAT provides as output from each run a golden BAM file and a golden VCF file containing the true locations of where reads should map and where variants were inserted. So this gives us and easy way to compare our calls against the truth.

Then we mapped the simulated reads from each diverged genome back to the original mouse genome and called variants in a pretty typical way with BWA and GATK.

And for each of these we'll assess accuracy by comparing our BAM and VCF files to the golden files generated by NEAT.

And this essentially sets us up with this situation except we know the actual level of divergence to compare to. So we'll be able to fill in these question marks and quantify how divergence effects reference bias

So, what did we find…

First I want to present just a sanity check from our simulations. Here on the x-axis I'm showing the specified divergence level for each simulation, and on the y-axis the actual percentage of sites with SNPs inserted in the golden VCF file.

The dashed grey line is the 1 to 1 line, which is where we hope our simulations fall. Each dot represents one simulated genome. And you can see the program is doing pretty well.

Though as divergence increases we are slightly under-simulating variants. We think this is probably due to the simulator double-hitting sites more often as divergence increases. But I just wanted to show this

because later on when I compare to expected levels of divergence, I'll keep labeling as what we simulated, but the underlying numbers are actually slightly lower in some cases.

Next I want to show how well the simulated reads are mapped back to the reference genome for each level of divergence. And I'll do this by showing proportional bar plots where on the x-axis we have the proportion of reads broken up into different categories based on these colors. I think most of these are self-explanatory, except for "Close map", which means that read mapped within 1 read length (which is 150bp) of the true mapping position.

And on the y-axis we have the different simulations.

I'll start with our reads simulated with 2% sequence divergence. Here we see that 80% of reads map exactly where they should, these are the ones in light blue in the bar. I think this is good, but it is still kind of striking that nearly 12% of reads are unmapped, that's this chunk in red.

And as we expected, the proportion of unmapped reads increases with divergence from the reference. But I was surprised by the extent of this increase: at 10% divergence only 38% of reads map correctly and 41% are unmapped! We also see an increase in the number of reads that map close to their expected position, but not exactly correctly.

And we can do something similar for the called variants. Here I'll show you a similar proportional bar plot, but this time the x-axis has the proportion of SNPs that were called in a couple categories. So here true positive means we simulated a variant there, and we've correctly called it as homozygous alternate compared to the reference genome. And false negative means we simulated a variant there, but did not end up calling it.

And as expected with the increasing number of unmapped reads, we see an increasing number of false negative variant calls. And these range from 7% of all variants at 2% sequence divergence to missing one third of all variants at 10% divergence.

Interestingly, the proportion of false positive variant calls, which I'm not showing here, does not increase at all. In fact, there are barely any false positive calls. This is good and I think also expected especially in such a clean simulation:

Ok so, we clearly see that reference bias is an issue. It affects how reads map and subsequent variant calls. So what can we do about it?

One possible solution that we've proposed in the past is iterative mapping. So remember, our goal is to get these unmapped reads with lots of variation to map to the reference genome.

So, with iterative mapping, after the first round of mapping, you insert the variants called back into the reference genome, generating a sort of consensus pseudo-assembly for your new sample which includes all the variation you could capture.

Then you map your reads again, this time to the consensus.

And the idea is that, since this consensus assembly already has some variation included…

That can serve as sort of an anchor point for these unmapped reads to match up to, hopefully allowing them to map and those other variants be called. And you can do this any number of times you want… generate another consensus with all the newly called variation, re-map, etc., possibly each time incorporating more variation.

And this was done by Brice Sarver and Jeff Good in 2017 on a few species of mice, and you can clearly see, especially for more diverged species, Pahari, that at least two rounds of mapping increases estimates of divergence between this species and the reference mouse genome.

And they implemented this method in software called pseudo-it, which I also worked on a lot when I did a postdoc with Jeff.

So that is a great empirical example of iterative mapping, but we also wanted to explicitly quantify its effects with simulations. Do we need to do iterative mapping in every instance to reduce bias? Or are there some cases where it may not be needed. After all, we've done a lot to speed things up with pseudo-it, but mapping and variant calling multiple times per sample can be time consuming.

So we basically re-implemented the pseudo-it pipeline for our simulations.

Instead of checking accuracy after just one round of mapping, we'll also generate a consensus pseudo-assembly from the variants we called and re-map to that. And we'll do this 2 times, so 3 total iterations of mapping, and align the genomes generated from each iteration back to the reference with minimap to assess accuracy.

So how does iterative mapping do?

Well let me just set this up for you again. So again I'll be showing proportional bar graphs of different classes of reads mapped on the x-axis. This time the y-axis represents the iteration of mapping, and each panel will represent one simulation with some level of divergence.

Let's start with the lowest level of divergence simulated, 2%. And here we see that iterative mapping has basically no effect. We map about 80% of reads correctly each time, and the proportion of unmapped reads stays consistent at about 12%.

The same trend is true for 4% divergence, roughly no change in the proportion of mapped or unmapped reads.

However, when we get to higher levels of divergence, we do start to see an effect, however maybe not what we thought it would be.

Let's focus on the 10% panel since it's the easiest to see the pattern: with one iteration of mapping we correctly map 38% of reads. After 3 iterations that goes up to 43%, a 5% improvement. That seems like a pretty good increase.

But lets notice that the proportion of unmapped reads, the red chunk, also, goes up, from 41% up to now 46%!

It seems like most of the improvement we see comes from more accurately mapping reads that were already mapped, just slightly off from their correct position. So that was unexpected, and we're still trying to figure out why.

I'll also note that, like Sarver et al, we also see almost all of our improvement after the second iteration of mapping, so more than that may be unnecessary.

Well, what about the SNPs? Well I showed you this before, which just shows the number of SNPs that actually were inserted into our simulated genomes here as the red dots.

We can look at the number of SNPs correctly called after one iteration of mapping, and this is basically just another way to show you the True Positive rate that I showed before in the bar plots. So as divergence increases, were missing more and more SNPs.

Does iterative mapping improve this?

Well, kind of. At low levels of divergence, where we're basically calling all the SNPs already, more rounds of mapping doesn't seem to really do anything. And at higher levels, again lets look at 10% divergence, we do recover a substantial number of SNPs that were missed in the first round of mapping… however, no where near the total number. So we're still far away from the red point here.

So as it is apparent here, we are making up for some of the missed calls, but definitely not all of them.

So, just to summarize, we were interested in how varying levels of divergence from the reference genome affects how variation is captured from read mapping.