Instructions for curating issues discovered from long read alignments to both haplotypes, April 17, 2023

This document is a guide to curating anomalies in HG002 diploid assemblies (currently the v0.7 verkko/rukki assembly) which suggest *missing* heterozygosity in the assembly based on DeepVariant homozygous non-reference calls on alignments of long reads to *both* assembly haplotypes (maternal and paternal). The document describes how these anomalies were detected, and the methods that we use to determine whether they are genuine errors in the assembly or false calls by DeepVariant.

This document is a work in progress, and available to edit at https://docs.google.com/document/d/1hzeNrQIkF3reoEqimIST-HFU9r1htIE2p7q8UPDEafM/edit#. Please feel free to suggest changes or additions as they occur to you.

HOW ARE THESE ANOMALIES DETECTED?

Like many assembly issues (as of April, 2023), these potentially false homozygous sites are discovered as high quality DeepVariant (DV) calls made on HG002 reads aligned to the HG002 assembly. In this case, the DV calls were made on alignments of ONT R10 simplex reads to both haplotypes. The alignments are here, with DV v1.5 calls here). This issue set includes all the homozygous non-reference (HNR) “PASS” DV v1.5 calls that had genotype quality (GQ) of 10 or greater, but is additionally filtered to limit false positives. Only calls in homozygous regions of the assembly greater than 5000 bases in size, and that have no variant call on the opposite haplotype, were retained. In other words, these issues represent homozygous sites in the assembly where a difference was missed on just one haplotype, and the ONT R10 simplex reads were long enough to stretch from flanking heterozygous spots in the assembly, where they align correctly to their respective haplotypes, into the homozygous region where they display the correct allele at the spot of the proposed assembly error. Because the long reads align correctly to their respective haplotype, the proposed correction to the assembly (the DV call) is phased correctly to its appropriate haplotype. To assure a high true positive rate for this type of potential issue, as of April 2023, our plan is to curate 15 of these variants to determine the reliability of five calls in each of three categories: DeepVariant GQ between 10 and 20, GQ between 20 and 30, and GQ between 30 and 40. The curation results will aid us in deciding which of DeepVariant’s suggested corrections to apply to the current and/or future assemblies that we attempt to polish.

ACCESSING YOUR ASSIGNED ISSUES

If you have volunteered to curate assembly issues for this April 2023 round, and we have your github username, a number of these issues will have been assigned to you for curation. You can

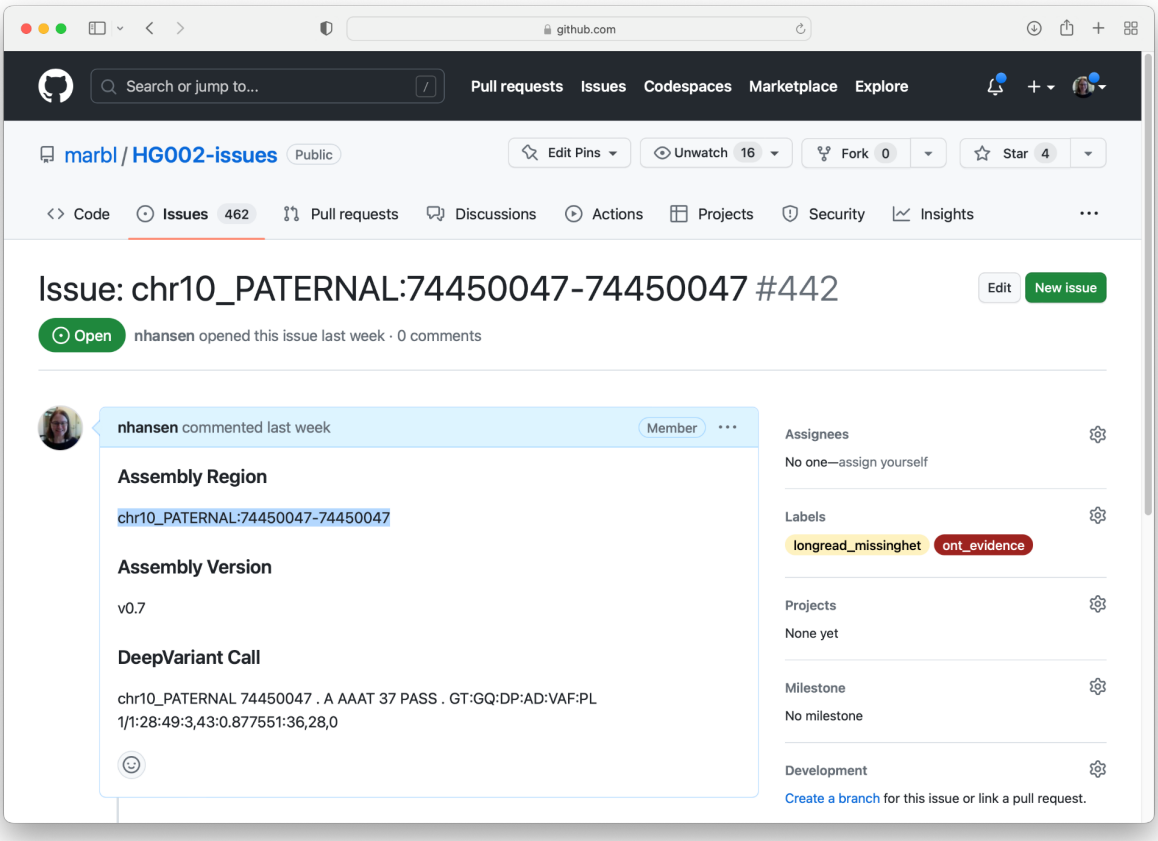
view your issues at https://github.com/marbl/HG002-issues/issues (assuming you are logged in to github) by using the “Assignee” drop down menu to select yourself. To restrict to viewing just the issues described in this document, filter the issues to include only those with the label “**longread_missinghet**”.

SESSION FILES AND VIEWING LONG READ INTERSECTED HNR CALLS IN IGV

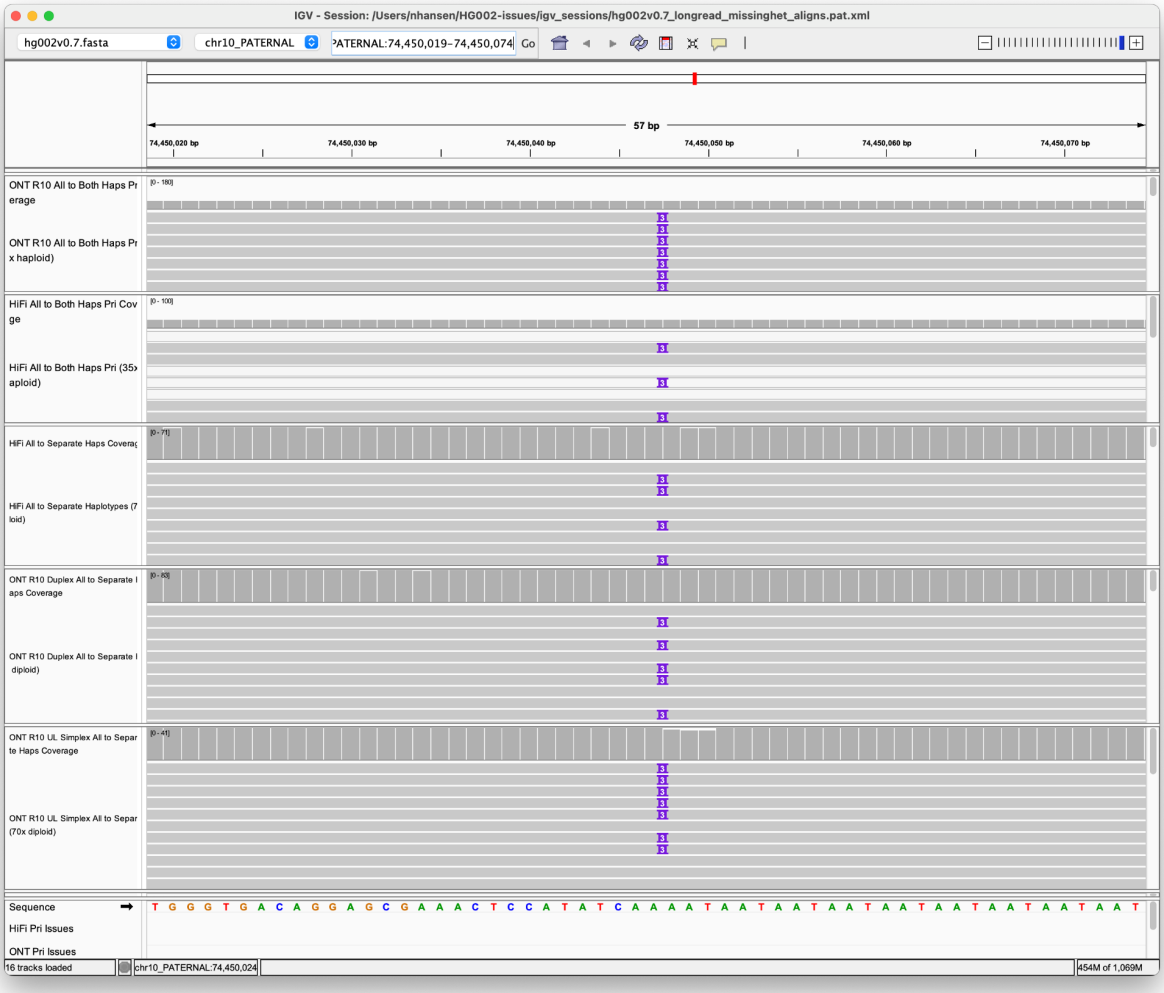
The best way to determine the underlying evidence for a DeepVariant call’s reliability is to view as much relevant data as possible for that region. One easy way to view groups of relevant data tracks in IGV is to load IGV session files. For curating this type of anomaly, we have created two IGV session files (one for issues on the maternal chromosomes, and one for issues on the paternal chromosomes) and put them on the HG002-issues github site at hg002v0.7_longread_missinghet_aligns.mat.xml and hg002v0.7_longread_missinghet_aligns.pat.xml. Once you’ve downloaded these files to your computer (if you clone the HG002-issues repository, you can update it to new versions easily), the tracks can be loaded into IGV by selecting “File…Open Session” and selecting the appropriate file’s location on your computer.

A CURATION EXAMPLE

As an example, let’s look at the DeepVariant call at chr10_PATERNAL:74,450,019-74,450,074, shown below in the GitHub HG002-issues repository. The DeepVariant call is an insertion of three bases (reference “A” becomes “AAAT”), suggesting that the assembly’s chr10_PATERNAL haplotype is missing three bases at this position.



To walk through this example, load the paternal aligned reads IGV session file (hg002v0.7_longread_missinghet_aligns.pat.xml), and copy and paste the location chr10_PATERNAL:74,450,019-74,450,074 into the position window in IGV. For our example region, our IGV window looks like this:



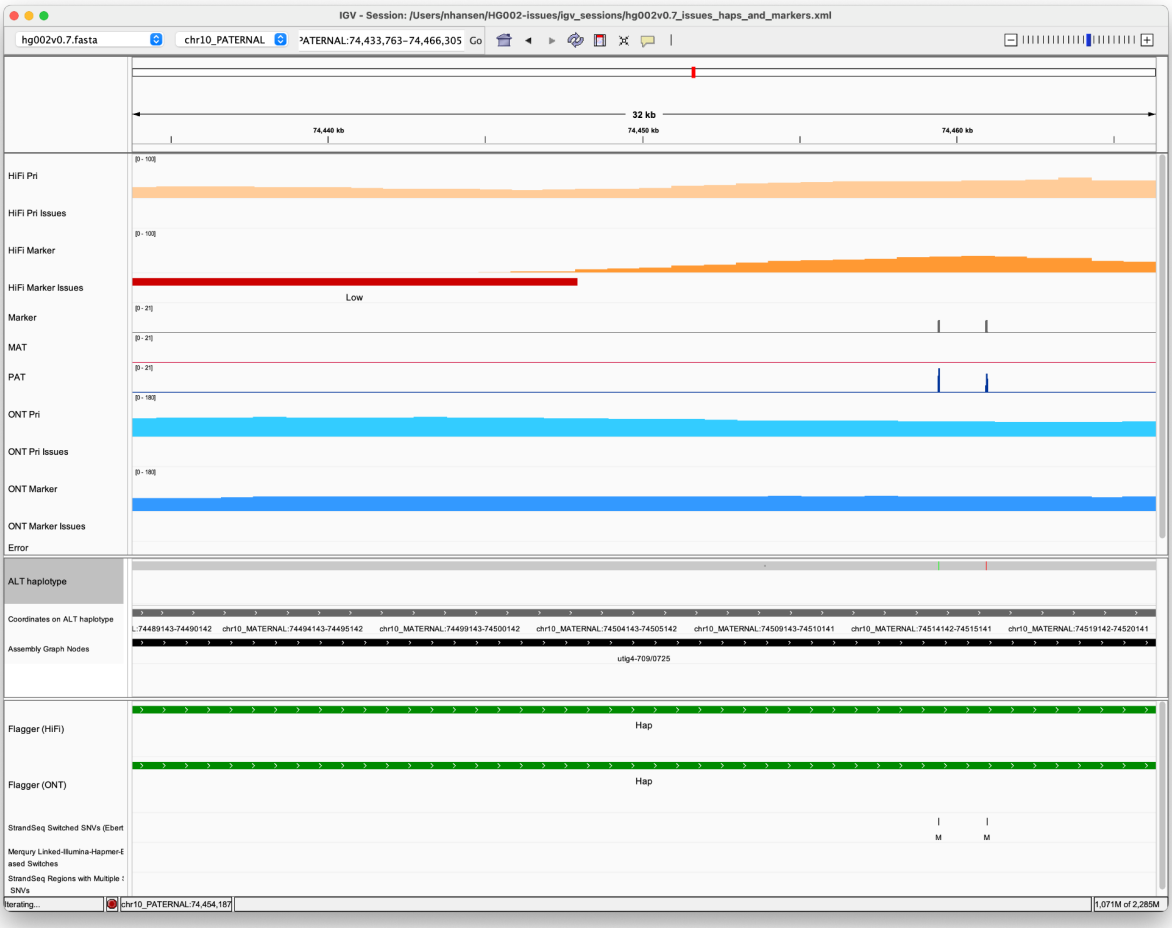
The tracks in this session file show all indels because they've all been configured in the session file to not "hide small indels". If you load tracks yourself while curating rather than using the provided IGV session file, **be sure that you also have not set each track to "hide small indels"**. Because all indels are visible, we can see the DeepVariant call, a three-base insertion of AAT in one of the many repeated AAT's, as a vertical line of insertion symbols with 3's on them just to the right of the middle of the IGV window.

The top track shows the ONT R10 ultralong simplex reads aligned to the entire assembly, and because the reads are both long and accurate, only paternal reads have aligned to this haplotype of the assembly, even though this specific spot is homozygous in the assembly for thousands of bases. Nearly all the reads in the top track show the three-base insertion with respect to the assembly.

The second track shows the HiFi reads aligned to the entire assembly, and here, many of the reads are not long enough to stretch to positions where the two assembly haplotypes differ, and are therefore colored white to show that they map equally well to both haplotypes. Those reads are equally likely to come from the maternal or the paternal haplotype, and so some of them have the three-base insertion, and some don't. The gray reads in this track, however, align better to this haplotype, so most of the gray reads have the three-base insertion.

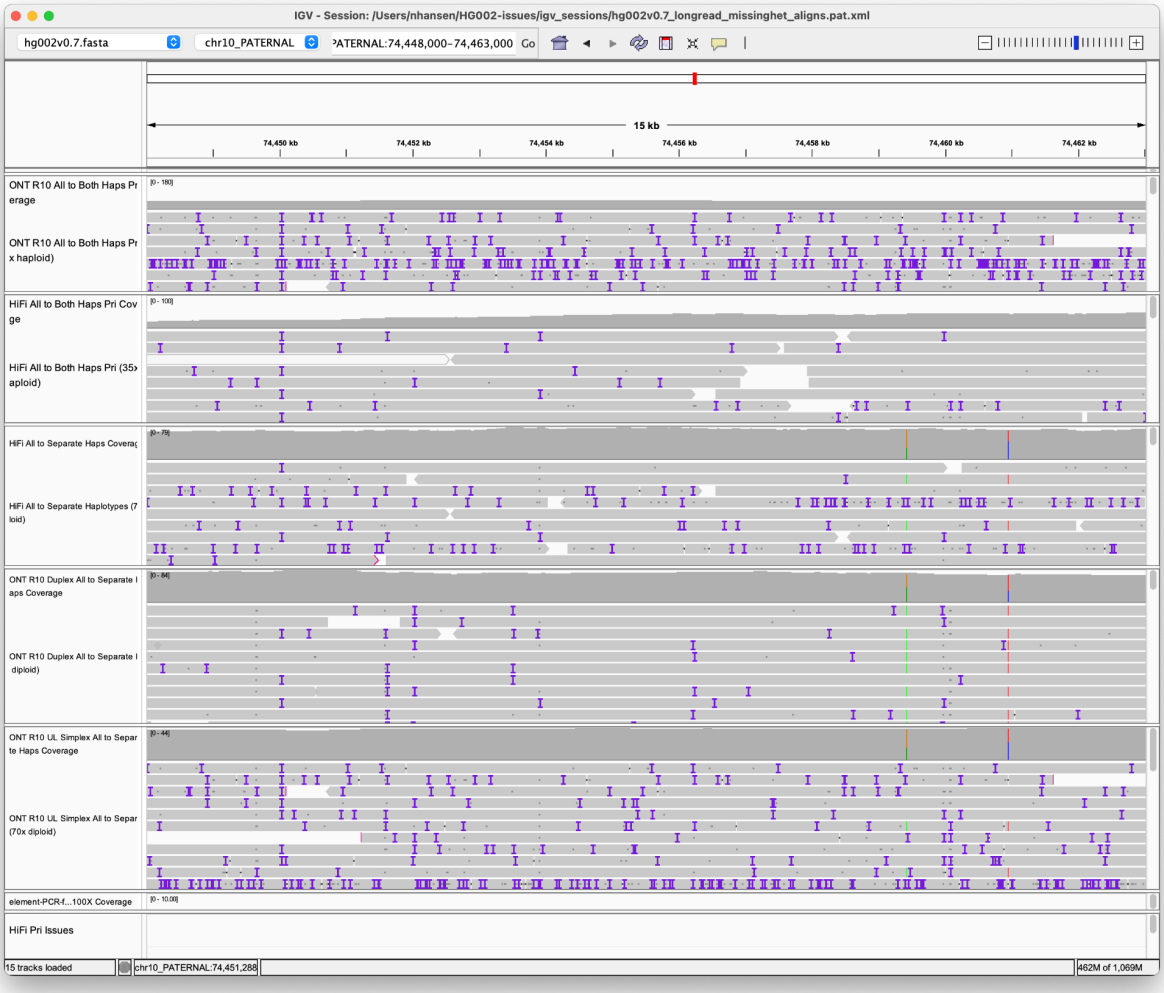
The next three tracks display alignments of all their reads to just this haplotype of the assembly. Here, both maternal and paternal reads align confidently to this haplotype, since the other haplotype was not included in the reference. So here we see reads both with and without the insertion, as expected.

Let's look at the context of this potential assembly error. If the ONT simplex reads were anchored by a flanking heterozygous locus in the assembly, we should be able to see it by zooming out. By opening a second session of IGV and loading the "hg002v0.7_issues_haps_and_markers.xml" session, we can zoom out to look for nearby differences in the two haplotypes on the "ALT haplotype":



Once we've zoomed out to 32kb, we see two single-nucleotide differences to the maternal haplotype about 10kb downstream from our potential assembly error (the assembly issue is near 74,450kb, and the two heterozygous SNVs are near 74,460kb). Since the heterozygous spots are about 10kb away from the assembly's missed heterozygous spot, it's easy to understand why (a) the ONT simplex reads easily spanned the two positions, and (b) some of the HiFi reads were mappable across them and some were not.

Now, if we return to our aligned reads session, we can center our window on both the error and the downstream SNVs by putting "chr10_PATERNAL:74,448,000-74,463,000" into the position window. In this window, we can check to see that the long reads in the top track that display the insertion in question really do stretch all the way to the heterozygous positions 10kb downstream, confirming that this is how they were able to map preferentially to the paternal haplotype. And in the tracks where reads were all mapped to just this haplotype, we can see the maternal reads as well, confirming that read with the alternate allele at the two SNV locations (a green A at the first and a red T at the second), don't have the insertion, which is only on the paternal haplotype. So in this case, all evidence seems to point to this DeepVariant call being due to a true error in the assembly.



LABELING AN ISSUE WITH YOUR DECISION

In cases like this, we can add a **priority** label to the issue on github, indicating that this is an error in the assembly that will need to be corrected. In general, if you've convinced yourself that an issue is either a true error in the assembly (i.e., the reads show a true difference from the consensus) or that the DeepVariant call was wrong and the assembly is correct, you should add a **priority** or a **false_positive** label to the issue on github, and add a few comments with screenshots of what you saw that made you make the decision you did. You will, in general, have a "co-curator" who may have already made comments or applied labels, but even if they did, it's helpful for you to add your independent assessment as well, so thank you for your efforts!

See Evaluating with IGV for further tips & tricks regarding IGV.