nature portfolio

Peer Review File

Genotyping Short Tandem Repeats Across Copy Number Alterations, Aneuploidies, and Polyploid Organisms

Corresponding Author: Professor Maria Anisimova

This file contains all reviewer reports in order by version, followed by all author rebuttals in order by version.

Version 0:

Reviewer comments:

Reviewer #1

(Remarks to the Author)
Key results:
ConSTRain is described as an STR variant caller designed to genotype STRs in the presence of copy number variants (CNVs) and polyploid genomes. The authors demonstrate that ConSTRain has high accuracy when tested on a euploid human genome and is competitive with existing STR variant callers. Additionally, they showcase its performance in analyzing data from polyploid species and in detecting structural variant-driven repeat instability in tumoroid cells, yielding promising results.

Originality and Significance:
ConSTRain is an original tool that addresses a gap in current STR variant calling methods by accounting for polyploidy and CNVs. The study's findings are relevant to multiple fields, including genomics, structural variant analysis, and cancer research. Its applications could benefit clinical research on repeat instability as well as agricultural studies, particularly for polyploid crop species. Given the increasing recognition of STRs' role in human disease and gene regulation, this manuscript presents novel applications of STR variant calling that should be of immediate interest to a broad audience in genetics, bioinformatics, and related disciplines.

The manuscript is robust, with clear methods and well-supported results. The GitHub page is well-organized, and the documentation is clear and helpful. However, there are a few aspects that could be further clarified in the manuscript:

Line 248: Could you elaborate on how segmental duplications affect the results and the potential issues that could arise if they are not filtered out?

Lines 257–268: Please include memory usage metrics for these analyses.

Line 366: The authors acknowledge potential caveats regarding assumption violations, correctly noting that this issue is not exclusive to ConSTRain. However, it would be helpful to suggest ways users can mitigate this issue, either in the methodology or discussion. Adding a couple of sentences on best practices for datasets with skewed distributions could strengthen the manuscript—perhaps introducing checks for PCR duplicates to reduce artificially inflated alleles?

Lines 377–380: Looking forward to seeing this implemented in the future!

(Optional) Lines 381–386: Consider providing CNA catalogs for some model organisms (e.g., human, maize, mouse) on the ConSTRain GitHub page.

(Optional suggestion/potential future implementations): - Addition of a merging VCF functionality to produce multi-sample VCF files; - Support for imperfect repeats.

(Optional): Providing subsetted or "mock" BAM files in the resources subdirectory could help users test ConSTRain with example data. This could include a CNA BAM file for demonstration purposes.

Reviewer #2

(Remarks to the Author)
This papers describes a new tool - ConSTRain - for genotyping tandem repeats using spanning reads. It is written in rust, and is faster than existing tools (HipSTR, GangSTR, and therefore also ExpansionHunter, etc.). It is also easy to install, and has good documentation on GitHub. Importantly, ConSTRain is the first tool to add support for non-diploid genomes and regions with copy number alterations and so enables much better characterization of TRs in non-diploid organisms. Benchmarking results in the paper show that ConSTRain has comparable accuracy to GangSTR and HipSTR.

Below are some minor comments and requests for clarification. My only substantial request is to stratify accuracy by allele length, and not just by sequencing depth or motif size.

Minor points:

- It would be helpful if the "Vocabulary" section defined "Allele length distribution". I'm used to hearing this phrase refer to the distribution of allele lengths in a population (ie. y-axis = # of haplotypes or individuals). Later in the paper, it becomes clear that "Allele length distribution" refers to the # of reads supporting each allele.

- Similarly, it would be helpful to define "depth of coverage" at an STR locus, as well as "spanning read". My understanding is "depth of coverage" refers to the number of *spanning* reads (as suggested on line #88) and spanning reads are those that have alignments starting >= 1bp to the left of the reference repeat interval and ending >= 1bp to the right of the interval (?).

- Assuming my understanding above is correct, I would expect depth of coverage to be significantly affected by the number of repeats in the reference and by the allele size - eg. alleles of length 100bp would have significantly fewer spanning reads than 10bp alleles. I would then expect this to be problematic for assumption 3 (ie. when alleles at a locus have different sizes), and cause the --min/max-norm-depth filters to bias results against allele sizes that differ substantially from the reference allele size. It would be helpful if the paper could explicitly discuss this.

- Figure 1 caption:
- typo: "An STR locus are loaded" => "An STR locus is loaded"
- Also, "(2) Reads overlapping the STR region are extracted from the alignment file, ". In the text, line #88 says only spanning reads are extracted - which is meaningfully different from "overlapping reads".

- line #82 wording: "Finally, each STR has a copy number, which indicates how many instances of the STR locus exist in the genome" - was confusing on the first reading because the phrase ".. in the genome" implies that an STR locus can exist in multiple locations (rather than on multiple copies of a chromosome). It made me wonder if this was talking about STR loci within segmental duplications.

- line #153 clarification: "... ConSTRain allows for the filtering of STR loci based on their normalised depth of coverage" When a locus is filtered out, does this mean it will be excluded from the output entirely, or just that it's genotype will be cleared (while leaving the allele depth distribution), or neither?

- "ConSTRain's accuracy is competitive with existing STR variant callers"
In addition to accuracy, this section compares tool runtimes. It would be helpful to also mention ConSTRain's memory requirements and how they vary with thread count (in my test, it used less than 4Gb in single-threaded mode on a catalog with ~150k loci, and < 3Gb per thread when using multiple threads).

Major points (all related to benchmarking):

- The reported benchmarks for filtered and unfiltered calls don't currently provide information about the allele sizes involved. I expect that the overwhelming majority of the true genotypes are homozygous reference (or at most differ by +/-1 or +/-2 repeats from the reference allele size), so the current benchmark is mostly measuring each tool's ability to accurately call homozygous reference genotypes. Although it's natural for a truth set to have this distribution of allele sizes, I think it's still important to also report accuracy by allele size, for example by sharing a histogram per allele (rather than per genotype) where the x-axis = true allele size minus the reference allele size, and y-axis = number or fraction of alleles called correctly. It would be especially helpful to stratify by allele size when comparing results before/after filtering as I suspect the --min/max-norm-depth thresholds preferentially filter out loci with significant expansions/contractions relative to the reference, thus enriching for easier-to-call loci.

- line #287: "For loci with genotype AAB they usually reported a genotype that was homozygous for the more abundant of the two alleles, missing the other allele length. In a subset of these loci (7.53% for GangSTR, 6.49% for HipSTR) a heterozygous AB genotype was reported."
This is surprising - particularly for simulated data! If GangSTR and HipSTR genotypes are wrong for > 90% of AAB loci, which if I understand correctly are just loci where ~66% of reads support allele A while 33% support allele B, it suggests that both GangSTR and HipSTR are extremely sensitive to allele read balance - which seems implausible to me. On average there should be 15 reads spanning the B allele - more than enough for these tools to detect it. Am I misunderstanding something?

- I would additionally request an explicit comparison of accuracy by allele size - taking a representative set of loci and generating simulated data for a wide enough range of allele sizes that it shows where each tool stops being able to accurately estimate the allele size (similar to the plots in the GangSTR paper - Mousavi 2019 Figure 3B-E). This type of analysis could also be used to address my previous point by plotting results for AAB genotypes while setting either A or B to the reference allele size. I'd be surprised if your results differed significantly from the high GangSTR accuracy indicated by Mousavi 2019 Figure 3B-E.

- In the paper, accuracy is defined as an exact match between the biallelic call and the true genotype (line #194) - which is a good starting point - but obfuscates the degree to which the evaluated tools deviate from the true allele size when they don't get it exactly right. In my tests, while GangSTR errors tend to be underestimates of the larger expansions, ConSTRain errors tend to report homozygous reference genotypes. Arguably, this is an important limitation worth discussing.

Discussion section:

- line #371 "However, it does highlight that an incorrect inference will be made if the observed read distribution strongly deviates from the underlying genotype. This is an issue for variant calling in general, and not specific to our method." To me this points to opportunities for adjusting ConSTRain's assumptions and model to better match complexities in the data rather than a limitation of variant calling in general.

- line #375 wording: "in-phase indel mutations" is a confusing phrase - maybe something like "expansions or contractions by integer multiples of the repeat unit length"? Also, was it not possible to simply trim the true allele sizes to integer multiples of the repeat unit in order to avoid this issue(?) Finally, here (and/or on github and/or in tool warning messages), it's worth specifying whether ConSTRain expects the input catalog repeat interval sizes to be an integer multiple of the repeat unit length, and also whether ConSTRain expects the start coordinates in the catalog to be 0-based (since, even though the BED format is officially 0-based, HipSTR and GangSTR used 1-based coordinates in their catalog BED files, so there can be confusion).

- When discussing limitations, I think it's important to reiterate that the tool only handles perfect repeats, and clarify what errors are likely to occur when it encounters alleles with interrupted sequences - does it just ignore reads with the interrupted repeats? Also, given that the genotyping approach is based on spanning reads, it's worth underscoring that ConSTRain is currently unable to detect expansions whose size approaches or exceeds the read length. Additionally, since the HG002 benchmarking was done with 250bp reads (rather than the 150bp reads used in most existing WGS datasets), it's worth mentioning that, at least for the small number of large expansions present in the HG002 truth set, these benchmarking results are likely better than what can be expected with shorter read lengths.

- For completeness, even if you don't directly benchmark against it, I think it's worth mentioning ExpansionHunter (since it's currently a widely-used tool) and it's strengths/weaknesses relative to ConSTRain.

Version 1:

Reviewer comments:

Reviewer #1

(Remarks to the Author)
I have reviewed the authors' revised manuscript and their point-by-point response to the reviewers' comments. I appreciate the modifications made in response to both my suggestions and those of Reviewer #2. The authors have addressed all concerns appropriately. I have no further comments, and I believe the manuscript is now suitable for publication. Thank you for this work.

Reviewer #2

(Remarks to the Author)
Thank you to the authors for addressing my comments. I recommend the manuscript for publication.

My only remaining suggestions, if the authors agree with them, would be to further expand on the revision in response to my comment 2.9. There, the added text includes:

"In this simulation, the longest allele to generate at least ten reads was 66 base pairs. The longest allele to generate any reads was 81 base pairs in length. This suggests that the detection limit for ConSTRain with a sequencing depth of 30X and a read length of 150bp is somewhere within this range of allele lengths. The influence of this effect was also apparent when we plotted the accuracy of allele-level length calls as a function of allele length (Supplementary Figure 4B&C)."

I think it would be helpful to add another sentence or two on how the effect is apparent in Figures 4B & C, since at first glance, the accuracy doesn't change much across allele lengths - with the blue dots' trend gradually descending, but then moving back up (especially in 4C). Also, it would be more informative to use absolute counts and a symmetric-log scale on the 2nd y-axis in 4B & C (ie. instead of 'proportion of STR loci'). This would allow readers to better see the tail of the

distribution and better understand the difference between the histograms in B & C - particularly in terms of how many alleles are being discarded by the filter.

Point-by-point response to reviewers (first round of revisions)
Communications Biology submission COMMSBIO-25-1875-T

The original reviewer comments are shown in black text, our responses are in blue text.

Reviewer #1 (Remarks to the Author):

Key results:
ConSTRain is described as an STR variant caller designed to genotype STRs in the presence of copy number variants (CNVs) and polyploid genomes. The authors demonstrate that ConSTRain has high accuracy when tested on a euploid human genome and is competitive with existing STR variant callers. Additionally, they showcase its performance in analyzing data from polyploid species and in detecting structural variant-driven repeat instability in tumoroid cells, yielding promising results.

Originality and Significance:
ConSTRain is an original tool that addresses a gap in current STR variant calling methods by accounting for polyploidy and CNVs. The study's findings are relevant to multiple fields, including genomics, structural variant analysis, and cancer research. Its applications could benefit clinical research on repeat instability as well as agricultural studies, particularly for polyploid crop species. Given the increasing recognition of STRs' role in human disease and gene regulation, this manuscript presents novel applications of STR variant calling that should be of immediate interest to a broad audience in genetics, bioinformatics, and related disciplines.

The manuscript is robust, with clear methods and well-supported results. The GitHub page is well-organized, and the documentation is clear and helpful. However, there are a few aspects that could be further clarified in the manuscript:

1.1: Line 248: Could you elaborate on how segmental duplications affect the results and the potential issues that could arise if they are not filtered out?
Thank you for the suggestion. We now elaborate more on our motivation for filtering out STR loci located in segmental duplications. Specifically, we've added the following text (lines 269-271):
"These loci are problematic because they are located in blocks of DNA that have highly similar homologues elsewhere in the genome. These homologues may be located far away from each other, potentially on different chromosomes. This means that it is impossible to determine the genomic origin of short sequencing reads mapping to segmental duplications."

1.2: Lines 257–268: Please include memory usage metrics for these analyses.
Thank you for the suggestion. We now include memory usage metrics in this table for each tool and added the following sentence to the table caption: “Memory usage statistics are based on the 'Maximum resident set size' reported when running each tool under the GNU time command with the --verbose flag.”

1.2: Line 366: The authors acknowledge potential caveats regarding assumption violations, correctly noting that this issue is not exclusive to ConSTRain. However, it would be helpful to suggest ways users can mitigate this issue, either in the methodology or discussion. Adding a couple of sentences on best practices for datasets with skewed distributions could strengthen the manuscript—perhaps introducing checks for PCR duplicates to reduce artificially inflated alleles?
Thank you for your suggestion. We now explicitly suggest to mark duplicate alignments in BAM files before running ConSTRain, so that they can properly be filtered out (lines 176-178):
“One potential source of an increase in the number of reads mapping to a genomic region are PCR duplicates. To address this, ConSTRain ignores all alignments for which the 'PCR or optical duplicate' flag is set. Thus, it is advisable to mark duplicate alignments explicitly before running ConSTRain, for example using samtools markdup.”

1.3: Lines 377–380: Looking forward to seeing this implemented in the future!
Thank you for your interest!

1.4: (Optional) Lines 381–386: Consider providing CNA catalogs for some model organisms (e.g., human, maize, mouse) on the ConSTRain GitHub page.
Thank you for the suggestion. Since CNA’s are unique to each individual, we felt that it was beyond the scope of the ConSTRain GitHub page to include such catalogues directly. However, large sequencing efforts (e.g., 1000Genomes, The Cancer Genome Atlas, GTEx etc.) typically include CNA annotations for each individual in the cohort. We now link to some of these catalogues in the GitHub documentation to provide a starting point for researchers interested in studying the interplay between STR variability and CNAs.

1.5: (Optional suggestion/potential future implementations): - Addition of a merging VCF functionality to produce multi-sample VCF files; - Support for imperfect repeats.
Thank you for the suggestions. The reason we did not implement merging VCF files ourselves is that there are already standard bioinformatics tools that have very robust implementations to perform this task (see, for example, bcftools merge).

We agree that implementing support for imperfect repeats would be a valuable addition to ConSTRain. However, it would entail a substantial rework and extension of the tool compared to its current state.

1.6: (Optional): Providing subsetted or "mock" BAM files in the resources subdirectory could help users test ConSTRain with example data. This could include a CNA BAM file for demonstration purposes.
Thank you for the excellent suggestion. We now include some mock files in the ConSTRain/tests/data folder of the GitHub repository, and added examples of how to run ConSTRain with these files in the GitHub documentation.

Reviewer #2 (Remarks to the Author):

This papers describes a new tool - ConSTRain - for genotyping tandem repeats using spanning reads. It is written in rust, and is faster than existing tools (HipSTR, GangSTR, and therefore also ExpansionHunter, etc.). It is also easy to install, and has good documentation on GitHub. Importantly, ConSTRain is the first tool to add support for non-diploid genomes and regions with copy number alterations and so enables much better characterization of TRs in non-diploid organisms. Benchmarking results in the paper show that ConSTRain has comparable accuracy to GangSTR and HipSTR.

Below are some minor comments and requests for clarification. My only substantial request is to stratify accuracy by allele length, and not just by sequencing depth or motif size.

Minor points:

2.1: - It would be helpful if the "Vocabulary" section defined "Allele length distribution". I'm used to hearing this phrase refer to the distribution of allele lengths in a population (ie. y-axis = # of haplotypes or individuals). Later in the paper, it becomes clear that "Allele length distribution" refers to the # of reads supporting each allele.
Thank you for your suggestion. We now extended the 'Vocabulary' section to also explicitly define 'allele length distribution', 'depth of coverage', and 'spanning read'. We hope this improves the clarity of the manuscript. The new text is on lines 81 to 85 of the updated manuscript:
"During genotyping, ConSTRain extracts all reads that span an STR locus. *Spanning reads* are defined as those reads for which the alignment starts at least 5 bp before the STR locus, and extends at least 5 bp beyond the STR locus. The STR allele lengths observed in all spanning reads for a locus gives the *allele length distribution* for that locus. The total number of spanning reads in the allele length distribution is called the *depth of coverage*."

2.2: - Similarly, it would be helpful to define "depth of coverage" at an STR locus, as well as "spanning read". My understanding is "depth of coverage" refers to the number

of *spanning* reads (as suggested on line #88) and spanning reads are those that have alignments starting >= 1bp to the left of the reference repeat interval and ending >= 1bp to the right of the interval (?).
Please see our response to comment 2.1, where we also address this point.

2.3: - Assuming my understanding above is correct, I would expect depth of coverage to be significantly affected by the number of repeats in the reference and by the allele size - eg. alleles of length 100bp would have significantly fewer spanning reads than 10bp alleles. I would then expect this to be problematic for assumption 3 (ie. when alleles at a locus have different sizes), and cause the --min/max-norm-depth filters to bias results against allele sizes that differ substantially from the reference allele size. It would be helpful if the paper could explicitly discuss this.
We agree that this is an important issue to discuss. Please see our response to comment 2.9 below for how we updated the manuscript to address this.

2.4: - Figure 1 caption:
- typo: "An STR locus are loaded" => "An STR locus is loaded"
Thank you for catching this, we have corrected the typo.
2.5: - Also, "(2) Reads overlapping the STR region are extracted from the alignment file, ". In the text, line #88 says only spanning reads are extracted - which is meaningfully different from "overlapping reads".
Thank you for highlighting this inconsistency. We have updated the Figure 1 caption: “Reads that completely span the STR region are extracted from the alignment file…”

2.6: - line #82 wording: "Finally, each STR has a copy number, which indicates how many instances of the STR locus exist in the genome" - was confusing on the first reading because the phrase ".. in the genome" implies that an STR locus can exist in multiple locations (rather than on multiple copies of a chromosome). It made me wonder if this was talking about STR loci within segmental duplications.
We have rephrased this sentence and added an example to clarify (lines 86&87): “Finally, each STR has a copy number, which indicates how many homologues of the STR locus exist in a sample. For example, for loci located on the autosomes in human samples the copy number is two (in the absence of CNAs).”

2.7: - line #153 clarification: "... ConSTRain allows for the filtering of STR loci based on their normalised depth of coverage" When a locus is filtered out, does this mean it will be excluded from the output entirely, or just that it's genotype will be cleared (while leaving the allele depth distribution), or neither?
Thank you for your comment, this is indeed an important point to specify. We added the following text to the manuscript to specify that filtered loci are still included in the output VCF (lines 159-162): “Loci that are filtered out are still reported in the VCF output,

including FORMAT fields describing the depth of coverage, allele length distribution, and more. The only difference is that for these loci no genotype will be reported. This means that a filtered locus will still be considered in future ConSTRain runs starting from this VCF file (see below)."

2.8: - "ConSTRain's accuracy is competitive with existing STR variant callers"
In addition to accuracy, this section compares tool runtimes. It would be helpful to also mention ConSTRain's memory requirements and how they vary with thread count (in my test, it used less than 4Gb in single-threaded mode on a catalog with ~150k loci, and < 3Gb per thread when using multiple threads).
Thank you for the suggestion. We now include memory usage metrics in this table for each tool and added the following sentence to the table caption: "Memory usage statistics are based on the 'Maximum resident set size' reported when running each tool under the GNU time command with the --verbose flag."

Major points (all related to benchmarking):

2.9: - The reported benchmarks for filtered and unfiltered calls don't currently provide information about the allele sizes involved. I expect that the overwhelming majority of the true genotypes are homozygous reference (or at most differ by +/-1 or +/-2 repeats from the reference allele size), so the current benchmark is mostly measuring each tool's ability to accurately call homozygous reference genotypes. Although it's natural for a truth set to have this distribution of allele sizes, I think it's still important to also report accuracy by allele size, for example by sharing a histogram per allele (rather than per genotype) where the x-axis = true allele size minus the reference allele size, and y-axis = number or fraction of alleles called correctly. It would be especially helpful to stratify by allele size when comparing results before/after filtering as I suspect the --min/max-norm-depth thresholds preferentially filter out loci with significant expansions/contractions relative to the reference, thus enriching for easier-to-call loci.
Thank you for your comment. To show how the effectiveness of ConSTRain's genotyping is affected by STR locus lengths, we implemented two new analyses: In the first, we generated a range of different allele lengths for an STR locus, simulated sequencing reads for each allele, and mapped them back to the human reference genome. We find that ConSTRain recovers fewer reads for longer allele lengths in alignments. Secondly, we now show how ConSTRain's genotyping accuracy is associated with STR allele lengths in the HG002 benchmark. Graphs for these new analyses are included as Supplementary Figure 4. We also added text to the Methods and Results section to discuss these new analyses:

Methods (lines 208-215):
"We computationally generated a range of allele lengths of a trinucleotide STR located

at chr3:63912684-63912714 in the *ATXN7* gene. We generated alleles between 5 and 40 repeat units (15bp to 120bp), spanning the range of allele lengths observed in healthy individuals. We included enough genomic context upstream and downstream of the repeat locus to make each sequence 20000bp in length. For each generated sequence, we simulated 2x150 paired-end reads to a depth of 30X. Since we were not interested in modelling sequencing errors for this analysis, reads were simulated using wgsim with the command-line arguments -e 0 -r 0 -R 0 -X 0 -S 42 -1 150 -2 150. We then mapped these reads to the GRCh38 reference sequence with minimap2 and genotyped the trinucleotide STR with ConSTRain using default command-line parameters."

Results (lines 280-287):
"The (normalised) depth of an STR locus is expected to be affected by the STR allele length: longer alleles are less likely to be spanned by sequencing reads, and will thus be less well represented in the allele length distribution ConSTRain extracts. To explicitly show this effect, we generated a range of allele lengths for a trinuleotide repeat locus in the ATXN7 gene (see Methods). As expected, Supplementary Figure 4A shows that ConSTRain finds progressively fewer spanning reads for longer allele lengths. In this simulation, the longest allele to generate at least ten reads was 66 basepairs. The longest allele to generate any reads was 81 basepairs in length. This suggests that the detection limit for ConSTRain with a sequencing depth of 30X and a read length of 150bp is somewhere within this range of allele lengths. The influence of this effect was also apparent when we plotted the accuracy of allele-level length calls as a function of allele length (Supplementary Figure 4B&C)."

2.10: - line #287: "For loci with genotype AAB they usually reported a genotype that was homozygous for the more abundant of the two alleles, missing the other allele length. In a subset of these loci (7.53% for GangSTR, 6.49% for HipSTR) a heterozygous AB genotype was reported."
This is surprising - particularly for simulated data! If GangSTR and HipSTR genotypes are wrong for > 90% of AAB loci, which if I understand correctly are just loci where ~66% of reads support allele A while 33% support allele B, it suggests that both GangSTR and HipSTR are extremely sensitive to allele read balance - which seems implausible to me. On average there should be 15 reads spanning the B allele - more than enough for these tools to detect it. Am I misunderstanding something?
Thank you very much for drawing our attention to this. These sentences contain an error in reporting on our part, so your surprise is warranted. The statement in the manuscript is 'the wrong way around' from our analyses: in the case of an AAB genotype, GangSTR and HipSTR reported a heterozygous AB genotype in **92.47%** **93.51%** of the time, respectively.

We have corrected this statement. The two sentences now read (lines 319-321):
"For loci with genotype AAB they usually reported a heterozygous AB genotype. In a

subset of these loci (7.53% for GangSTR, 6.49% for HipSTR) a genotype that was homozygous for the more abundant of the two alleles was reported, missing the other allele length."

2.11: - I would additionally request an explicit comparison of accuracy by allele size - taking a representative set of loci and generating simulated data for a wide enough range of allele sizes that it shows where each tool stops being able to accurately estimate the allele size (similar to the plots in the GangSTR paper - Mousavi 2019 Figure 3B-E). This type of analysis could also be used to address my previous point by plotting results for AAB genotypes while setting either A or B to the reference allele size. I'd be surprised if your results differed significantly from the high GangSTR accuracy indicated by Mousavi 2019 Figure 3B-E.
Please see our response to comment 2.9, where we address this point as well.

2.12: - In the paper, accuracy is defined as an exact match between the biallelic call and the true genotype (line #194) - which is a good starting point - but obfuscates the degree to which the evaluated tools deviate from the true allele size when they don't get it exactly right. In my tests, while GangSTR errors tend to be underestimates of the larger expansions, ConSTRain errors tend to report homozygous reference genotypes. Arguably, this is an important limitation worth discussing.
Thank you for your comment, we now explicitly reiterate in our Discussion section that ConSTRain is limited by the read length (lines 420-426): "Further, since ConSTRain is limited by the sequencing read length it will never be able to genotype alleles that are longer than the read length. This is an issue when genotyping large repeat expansions, such as those observed in some human diseases. In such cases, ConSTRain will either fail to find a genotype (for homozygous expansions) or report a homozygous genotype for the shorter allele (in cases where only one allele is expanded). However, we observed in our HG002 benchmark that over 95% of repeat loci were shorter than 30bp in length, which means that a standard 30X sequencing experiment with 150bp reads will be sufficient to resolve the vast majority of STRs in the human samples."

Discussion section:

2.13: - line #371 "However, it does highlight that an incorrect inference will be made if the observed read distribution strongly deviates from the underlying genotype. This is an issue for variant calling in general, and not specific to our method." To me this points to opportunities for adjusting ConSTRain's assumptions and model to better match complexities in the data rather than a limitation of variant calling in general.
Thank you for your comment, we agree that the previous formulation disregarded the chance of improvement opportunities. We have rewritten this sentence to reflect this

(lines 403-408):
“This is an issue for variant calling in general, and addressing it would require a more sophisticated genotyping approach than the one currently implemented in ConSTRain. Such an approach could, for example, incorporate a genotyping model that corrects for the fact that longer STR alleles are expected to be spanned by fewer sequencing reads and are therefore underrepresented in the allele length distributions compared to shorter alleles. We did not implement such a genotyping approach here, since our primary focus was to create a method that extends the realm of STR genotyping to different ploidies beyond standard human physiology. It is, however, an important point for future improvements to ConSTRain.”

2.14: - line #375 wording: "in-phase indel mutations" is a confusing phrase - maybe something like "expansions or contractions by integer multiples of the repeat unit length"? Also, was it not possible to simply trim the true allele sizes to integer multiples of the repeat unit in order to avoid this issue(?) Finally, here (and/or on github and/or in tool warning messages), it's worth specifying whether ConSTRain expects the input catalog repeat interval sizes to be an integer multiple of the repeat unit length, and also whether ConSTRain expects the start coordinates in the catalog to be 0-based (since, even though the BED format is officially 0-based, HipSTR and GangSTR used 1-based coordinates in their catalog BED files, so there can be confusion).
Thank you for your comment, we have rephrased this sentence to improve clarity (line 411): “…ConSTRain only considers mutations where integer multiples of the repeat unit are inserted or deleted…”

We have also updated the GitHub documentation to say that BED coordinates must be 0-based, open-ended.

2.15: - When discussing limitations, I think it's important to reiterate that the tool only handles perfect repeats, and clarify what errors are likely to occur when it encounters alleles with interrupted sequences - does it just ignore reads with the interrupted repeats? Also, given that the genotyping approach is based on spanning reads, it's worth underscoring that ConSTRain is currently unable to detect expansions whose size approaches or exceeds the read length. Additionally, since the HG002 benchmarking was done with 250bp reads (rather than the 150bp reads used in most existing WGS datasets), it's worth mentioning that, at least for the small number of large expansions present in the HG002 truth set, these benchmarking results are likely better than what can be expected with shorter read lengths.
Thank you for your suggestion. We added the following sentences to the Discussion section (lines 417-420): “In general, ConSTRain currently only considers perfect repeat loci. If a sample contains mutations that interrupt the repeat locus or alter the length of the locus by a number of basepairs that is not equal to the repeat period, the genotype

reported by ConSTRain will not match the actual genotype in the sample. This is because ConSTRain will discard sequencing reads that do not conform to its expectations, which will in turn lead to incorrect genotype inferences."

2.16: - For completeness, even if you don't directly benchmark against it, I think it's worth mentioning ExpansionHunter (since it's currently a widely-used tool) and it's strengths/weaknesses relative to ConSTRain.
Thank you for your suggestion. We added a paragraph to the Discussion where we discuss ExpansionHunter (lines 433-439): "Finally, it is worth mentioning that STR variant callers are an area of active development, where many different tools are available. In our benchmarks we only compared ConSTRain directly to GangSTR and HipSTR, since these are two methods that are very similar in how they analyse alignments and how they represent STR loci. Another notable method is ExpansionHunter. This is a method that allows for a more complex specification of STR reference allele sequences compared to ConSTRain, and can also resolve expansions beyond the sequencing read length in euploid human samples. There are many other methods available, each with different strengths and weaknesses, but a comprehensive review of these methods is beyond the scope of this manuscript."

Point-by-point response to reviewers (second round of revisions)
Communications Biology submission COMMSBIO-25-1875-T

The original reviewer comments are shown in black text, our responses are in blue text.

Reviewer #1 (Remarks to the Author):
I have reviewed the authors' revised manuscript and their point-by-point response to the reviewers' comments. I appreciate the modifications made in response to both my suggestions and those of Reviewer #2. The authors have addressed all concerns appropriately. I have no further comments, and I believe the manuscript is now suitable for publication. Thank you for this work.

Thank you for your insightful feedback throughout the review process, and for supporting our manuscript for publication.

Reviewer #2 (Remarks to the Author):
Thank you to the authors for addressing my comments. I recommend the manuscript for publication.

Thank you very much for recommending our manuscript for publication. We appreciate the feedback you have provided throughout this review process, which has served to improve the quality and robustness of the final manuscript.

2.1: My only remaining suggestions, if the authors agree with them, would be to further expand on the revision in response to my comment 2.9. There, the added text includes:

"In this simulation, the longest allele to generate at least ten reads was 66 base pairs. The longest allele to generate any reads was 81 base pairs in length. This suggests that the detection limit for ConSTRain with a sequencing depth of 30X and a read length of 150bp is somewhere within this range of allele lengths. The influence of this effect was also apparent when we plotted the accuracy of allele-level length calls as a function of allele length (Supplementary Figure 4B&C)."

I think it would be helpful to add another sentence or two on how the effect is apparent in Figures 4B & C, since at first glance, the accuracy doesn't change much across allele lengths - with the blue dots' trend gradually descending, but then moving back up (especially in 4C). Also, it would be more informative to use absolute counts and a symmetric-log scale on the 2nd y-axis in 4B & C (ie. instead of 'proportion of STR loci'). This would allow readers to better see the tail of the distribution and better understand the difference between the histograms in B & C - particularly in terms of how many alleles are being discarded by the filter.

Thank you for this final suggestion. We agree that the panels we added to Supplementary Figure 4 did not clearly convey our message. We have now replaced these panels with new graphs. Supplementary Figure 4B now shows a histogram of the number of alleles observed as a function of allele length, both before and after filtering. This provides a direct comparison and shows how many alleles are removed for each allele length. Supplementary Figure 4C now only shows the accuracy per allele length (the blue dots in the previous version of the graph), both before and after filtering. We also added trendlines for visual clarity.

Additionally, we expanded our discussion of this Supplementary Figure in the Results section:

“This suggests that the detection limit for ConSTRain with a sequencing depth of 30X and a read length of 150bp is somewhere within this range of allele lengths. We also observed this effect when investigating the number and accuracy of allele-level length calls in the HG002 sequencing data. The longest allele length that was observed at least 10 times in the HG002 sequencing data was 120bp or 100bp before and after read depth filtering, respectively. (Supplementary Figure 4B). Accounting for the longer read length in the HG002 data, this is roughly in line with what we observed in the simulated *ATXN7* reads. Overall, we found a decrease in accuracy with increasing allele lengths (Supplementary Figure 4C). There were some pronounced dips in accuracy for alleles between 40 and 60 bp in length. This was due to mono- and dinucleotide repeats, for which the longest loci in our dataset are in this range of allele lengths. These STRs become harder to genotype accurately as they increase in length, decreasing the overall genotyping accuracy. For higher allele length ranges, where mono- and dinucleotide STRs are no longer present, the overall accuracy rises again.”