broadinstitute / gatk Public

Code Issues 1.3k Pull requests 140 Discussions Actions Projects

Releases Tags Find a release

Apr 14

lbergelson 4.6.2.0 76edc75 Compare

4.6.2.0

Latest

Download release: gatk-4.6.2.0.zip
Docker image: https://hub.docker.com/r/broadinstitute/gatk/

Highlights of the VERSION release:

- **Funcotator Data Location Moved**
  We've moved the location that `FuncotatorDataSourceDownloader` pulls data from because it turned out to be rather expensive to host it there. If you use this in a pipeline we would appreciate it if you updated to the new version. (#9131)
  - Old:
    - gs://broad-public-datasets/funcotator/
    - https://console.cloud.google.com/storage/browser/broad-public-datasets/funcotator
  - New:
    - gs://gcp-public-data--broad-references/funcotator/
    - https://console.cloud.google.com/storage/browser/gcp-public-data--broad-references/funcotator
- **New SV Tools** There are several new tools to work with SV Data from GATK-SV `SVStratify` and `GroupedSVCluster` (#8990)
- **CallableLoci** was ported from GATK3 since it is useful in some situations. (#9031)

- **New BQSR argument** `--allow-missing-read-group` to work around a rare but annoying issue where BQSR fails if a Read Group is completely filtered from the training data but present at application time. (#9020)

Full list of changes:

- **New Tools**
  - Add SVStratify and GroupedSVCluster tools #8990
  - Port of `CallableLoci` from GATK3 #903
- **Flow Mode Calling**
  - Tiny performance improvement #9077
- **Mutect2+**
  - Many small changes to Mutect2 pipelines to support Permutect #9094, #9136, #9138
- **Funcotator**
  - Updated references to the funcotator datasets bucket to point to the new google bucket by **@KevinCLydon** in #9131
- **SV Calling**
  - Prioritize het calls when merging clustered SVs #9058
- **Notable Enhancements**
  - BQSR: avoid throwing an error when read group is missing in the recal table, and some refactoring. by **@takutosato** in #9020
- **Bug Fixes**
  - VariantRecalibrator R script fix so new versions of R work. #9046
  - Addressed an edge case in ScoreVariantAnnotations that can occur when one variant type is not present in the input VCF #9112
  - Fix an annoying warning by excluding logback-classic #9128
  - Close a FeatureReader after use #9078
- **Miscellaneous Changes**

- Option to retain source IDs on VariantContext merge #9032

- **Documentation**
  - Update Python compatibility information in README.md #9047
- **Dependencies**
  Many dependencies updated including bug fixes and security patches
  - Update Htsjdk 4.1.3-> 4.2.0 in
  - Update Picard 3.3.0 -> 3.4.0 #9143
  - Update logback-core from 1.4.14 to 1.5.13 #9079
  - Update GenomicsDB #9059
  - Update Netty #9120
  - Exclude bad version of bouncycastle library #9129
  - Bump org.apache.commons:commons-vfs2 from 2.9.0 to 2.10.0 #9130
  - Update parquet to 1.15.1 #9144
- **Developer Infrastructure**
  - Update upload_artifact in github actions #9061
  - Update gradle sonatype plugin #9133

Full Changelog: `4.6.1.0...4.6.2.0`

Contributors

takutosato and KevinCLydon

Assets 3

0 Join discussion

Oct 30, 2024

lbergelson 4.6.1.0 `02c87bf` Compare

4.6.1.0

Download release: gatk-4.6.1.0.zip
Docker image: https://hub.docker.com/r/broadinstitute/gatk/

Highlights of the 4.6.1.0 release:

- Modernize the aging Conda environment with up to date python dependencies. All the python tools have been updated appropriately. This will enable easier integration of new machine learning tools.
  - ***If you use python tools outside of the docker, you must rebuild your conda environment for this release***
- `CNNScoreVariants` has been replaced by `NVScoreVariants`, a rewritten and modernized version. The python code for this tool was written by members of NVIDIA Genomics Research.
  - Thank you Babak Zamirai, Ankit Sethia, Mehrzad Samadi, George Vacek and the whole NVIDIA genomics team!
  - This GATK blog post has more of the story from when we first made the tool available for testing.
- New `Funcotator` argument `--prefer-mane-transcripts` which improves transcript selection and lays groundwork for upcoming improvements.
- New argument `--variant-output-filtering` which lets you restrict output variants based on the input intervals. This replaces and imrpoves on `--only-output-calls-starting-in-interval` and works with `SelectVariants` and other VariantWalkers. This is useful to prevent duplicating variants when splitting an input VCF into multiple shards.

Full list of changes:

- **CNNScoreVariants -> NVScoreVariants** (#8004, #9010, #9009)
  - CNNScore variants has been replaced by NVScoreVariants, scripts that use it should be updated to use NVScoreVariants instead.
  - The training tools (CNNVariantTrain, CNNVariantWriteTensors)have been removed. If you need to retrain the model for your data type you should continue to use GATK 4.6.0.0. New training tools are in development to work alongside NVScoreVariants and will be added in subsequent releases.

- **New Tools**
  - New tool `GtfToBed` to convert Gencode GTF files to BED files (#7159, #8942)
  - New tool for internal use `VcfComparator` (#8933, #8973)
- **Joint Calling GVS**
  - Adds QD and AS_QD emission from VariantAnnotator on GVS input (#8978)
- **GenomicsDB**
  - Switch to logging a warning instead of an exception for intervals in query that were not part of GenomicsDBImport (#8987)
- **Funcotator**
  - Added a '--prefer-mane-transcripts' mode that enforces MANE_Select tagged Gencode transcripts where possible)(#9012)
- **SV Calling**
  - Handle CTX_PP/QQ and CTX_PQ/QP CPX_TYPE values inSVConcordance (#8885)
  - Complex SV intervals support by **@mwalker174** (#8521)
  - Require both overlap and breakend proximity for depth-only SV clustering (#8962)
- **Flow Based Calling**
  - Modified HaplotypeBasedVariantRecaller to support non-flow reads (#8896)
  - FlowFeatureMapper: X_FILTERED_COUNT semantics adjusted and documented more accurately (#8894)
  - Changes to flow arguments in haplotype caller from Picard (see Picard release notes
- **Miscellaneous Features**
  - Added a check for whether files can be created and executed within the configured tmp-dir (#8951)
- **Documentation**
  - Clarify in the README which git lfs files are required to build GATK (#8914)
  - Add docs about citing GATK (#8947)
  - Update Mutect2.java Documentation (#8999)
  - Add more detailed conda setup instructions to the GATK README (#9001)
  - Adding small warning messages to not to feed any GVCF files to these tools (#9008)

- **Refactoring**
  - Swapped mito mode in Mutect to use the mode argument utils (#8986)
- **Tests**
  - Adding a test to capture an expected edge case in Reblocking (#8928)
  - Update the large CRAM files to v3.0 (#8832)
  - Update CRAM detector output files (#8971)
  - Add dependency submission workflow so we can monitor vulnerabilities (#9002)
- **Dependencies**
  Updating dependencies to make use of modern frameworks with fewer vulnerabilities was a focus of this release.
  - Updated Python and PyMC, removed TensorFlow, and added PyTorch in conda environment. (#8561)
  - Rebuild gatk-base docker image (3.3.1) in order to pull in recent patches (#9005)
  - Updates to java build and dependencies (#8998, #9006, #9016)
    - Update to the Gradle 8.10.2
    - Improvements to `build.gradle` to use of features like consuming publishes Bills of Materials (BOMs)
    - Update many direct and transitive java dependencies to fix security vulnerabilities.
    - Update Htsjdk 4.1.1 to 4.1.3
    - Update Picard 3.2.0 to 3.3.0
    - Update hdf5-java-bindings to version 1.2.0-hdf5_2.11.0 (#8908)

Contributors

mwalker174

▶ **Assets** 3

0 Join discussion

Jun 29, 2024

droazen 4.6.0.0 64348bc Compare

4.6.0.0

Download release: gatk-4.6.0.0.zip
Docker image: https://hub.docker.com/r/broadinstitute/gatk/

Highlights of the 4.6.0.0 release:

- We've fixed a serious CRAM writing bug that affects GATK versions 4.3 through 4.5 and Picard versions 2.27.3 through 3.1.1. This bug can, in limited cases, lead to reads with an incorrect base sequence being written. See this comment to GATK issue 8768 and the full release notes below for more details on what conditions trigger the bug.
    - ***To help users detect whether their CRAM files are affected, we've released a CRAM scanning tool called `CRAMIssue8768Detector` that can detect whether a particular CRAM file is affected by this bug. If you suspect that some of your CRAM files may have been affected, please run this tool on them for confirmation!***
- By overwhelming popular demand, we've switched back to using the standard `./.` representation for no-calls in `GenotypeGVCFs` and `GenomicsDB` instead of `0/0` with `DP=0`. This reverts the change described in our article GenotypeGVCFs and the death of the dot.
    - We intend to publish a new article shortly to replace that older article with further details on this change. When we do so, we'll link to it from here.
- The `Mutect2` germline resource can now have split multiallelic format
- Added an `--inverted-read-filter` argument to allow for selecting reads that fail read filters from the command line easily
- We've fixed a number of issues with HTTP support, mainly affecting the loading of side inputs such as indices over HTTP
- Reduced the number of layers in the GATK docker image to help users running into docker quota issues

Full list of changes:

- **Important CRAM writing bug fix and detection tool**
  - We've updated to `HTSJDK` 4.1.1 and `Picard` 3.2.0 (#8900), which fix a serious bug in the CRAM writing code first reported in GATK issue 8768
  - This issue affects GATK versions 4.3.0.0 through 4.5.0.0, and is fixed in GATK 4.6.0.0.
  - This issue also affects Picard versions 2.27.3 through 3.1.1, and is fixed in Picard 3.2.0.
  - The bug is triggered when writing a CRAM file using one of the affected GATK/Picard versions, and both of the following conditions are met:
    - At least one read is mapped to the very first base of a reference contig
    - The file contains more than one CRAM container (10,000 reads) with reads mapped to that same reference contig
  - When both of these conditions are met, the resulting CRAM file may have corrupt containers associated with that contig containing reads with an incorrect sequence.
  - Since many common references such as hg38 have N's at the very beginning of the autosomes and X/Y, many pipelines will not be affected by this bug. However, users of a telomere-to-telomere reference, users doing mitochondrial calling, and users with reads aligned to the alt sequences will want to scan their CRAM files for possible corruption.
  - The other mitigating circumstance is that when a CRAM is affected, the signal will be overwhelmingly obvious, with the mismatch rate typically jumping from sub-1% to 80-90% for the affected regions, making it likely to be caught by standard QC processes.
  - ***We've released a CRAM scanning tool called `CRAMIssue8768Detector` (#8819) that can detect whether a particular CRAM file is affected by this bug. If you suspect that some of your CRAM files may have been affected, please run this tool on them for confirmation!***

- **Joint Calling**
  - We've switched back to using the standard `./.` representation for no-calls in `GenotypeGVCFs` and `GenomicsDB` instead of `0/0` with `DP=0` (#8715) (#8741) (#8759)
    - This reverts the change described in our article GenotypeGVCFs and the death of the dot
  - Fix for `GenotypeGVCFs` with mixed ploidy sites (#8862)

- Fix for `GnarlyGenotyper` when PLs are null (#8878)
- Fixed bug in `ReblockGVCF` when removing annotations (#8870)
- Enable `ReblockGVCF` to subset AS annotations that aren't "raw" (pipe-delimited) (#8771)
- Remove header lines in `ReblockGVCF` when we remove FORMAT annotations (#8895)
- `ReblockGVCF` : Add malaria spanning deletion exception regression test with fix (#8802)
- Restore some `GnarlyGenotyper` tests (#8893)

- **HaplotypeCaller**
  - Fix to long deletions that overhang into the assembly window causing exceptions in `HaplotypeCaller` (#8731)

- **Mutect2**
  - The `Mutect2` germline resource can now have split multiallelic format (#8837)
  - Make the `Mutect2` haplotype and clustered events filters smarter about germline events (#8717)
  - Added the DragSTR model to the Mutect2 WDL (#8716)
  - Improvements to `Mutect2` 's `Permutect` training data mode (#8663)
  - Bigger `Permutect` tensors and `Permutect` test datasets can be annotated with truth VCF (#8836)
  - `Mutect2` WDL and GetSampleName can handle multiple sample names in BAM headers (#8859)
  - `Permutect` dataset engine outputs contig and read group indices, not names (#8860)
  - Normal artifact LOD is now defined without the extra minus sign (#8668)

- **CNV Calling**
  - Fixed the GT header in `PostprocessGermlineCNVCalls` 's `--output-genotyped-intervals` output (#8621)

- **SV Calling**
  - Reduced `SVConcordance` memory footprint (#8623)
  - Rewrote complex SV functional annotation in `SVAnnotate` (#8516)
  - We now handle the `CTX_INV` subtype in `SVAnnotate` (#8693)

- **Flow-based Calling**
  - SNVQ recalibration tool added for flow-based reads (#8697)
  - Bug fix in flow-based allele filtering (#8775)
  - Fixed a bug in flow-based `AlleleFiltering` that ignored more than a single sample (#8841)
  - Fixed an edge case in flow-based variant annotation (#8810)
- **Notable Enhancements**
  - Added an `--inverted-read-filter` argument to allow for selecting reads that fail read filters from the command line easily (#8724)
  - Inverted `SoftClippedReadFilter` to conform to the standard filtering logic (#8888)
  - Reduced the number of docker layers in the GATK image from 44 to 16 (#8808)
  - `VariantFiltration` : added a `--mask-description` argument to write custom mask filter description in VCF header (#8831)
  - `GatherVcfsCloud` is no longer beta (#8680)
- **Miscellaneous Changes**
  - `GetPileupSummaries` now uses the standard `MappingQualityReadFilter` instead of a custom `--min-mapping-quality` argument (#8781)
  - `Funcotator` : suppress a log message about b37 contigs when not doing b37/hg19 conversion (#8758)
  - Output the new image name at the end of a successful cloud docker build (#8627)
  - Exclude the test folder from code coverage calculations (#8744)
  - Removed deprecated genomes in the cloud docker image that was causing CNN WDL test failures (#8891)
  - Re-commit large test files as lfs stubs (#8769)
  - Standardize test results directory between normal/docker tests (#8718)
  - Improve failure message in `VariantContextTestUtils` (#8725)
  - Update the `setup_cloud` github action (#8651)
  - Parameterize the logging frequency for ProgressLogger in `GatherVcfsCloud` (#8662)
- **Documentation**
  - Updated the README to include list of popular software included in docker image (#8745)

- **Dependencies**
  - Updated `HTSJDK` to 4.1.1, which fixes the CRAM writing bug described above (#8900)
  - Updated `Picard` to 3.2.0, which fixes the CRAM writing bug described above (#8900)
  - Updated `GenomicsDB` to 1.5.3, which supports M1 Macs and switches no-call representation back to `./.` (#8710) (#8759)
  - Updated `http-nio` to 1.1.1, which fixes several URL-handling bugs with HTTP support (#8889)
  - Updated several miscellaneous dependencies to fix security vulnerabilities (#8898)

▶ **Assets** 3

👍 6 🎉 2 7 people reacted

8 Join discussion

Dec 13, 2023

droazen 4.5.0.0 8317d8b Compare

4.5.0.0

Download release: gatk-4.5.0.0.zip
Docker image: https://hub.docker.com/r/broadinstitute/gatk/

Highlights of the 4.5.0.0 release:

- `HaplotypeCaller` now supports custom ploidy regions that can be specified via a new `--ploidy-regions` argument, overriding the global `-ploidy` setting
- The default `SmithWaterman` implementation for `HaplotypeCaller` and `Mutect2` is now the hardware-accelerated version, resulting in a significant speedup
- `Funcotator` has a new datasource release that brings in the latest version of `Gencode` and several other key data sources

- We've updated our dependencies and our docker environment to greatly cut down on known security vulnerabilities
- We've greatly improved support for `http` / `https` inputs in GATK-native tools (though most Picard tools bundled with GATK do not yet support it)
- We've ported some additional DRAGEN features to `HaplotypeCaller` that bring us closer to functional equivalence with DRAGEN v3.7.8
- `GenomicsDBImport` now has support for Azure storage `az://` URIs
- `GnarlyGenotyper` now has haploid support
- Lots of important bug fixes, including a fix for a bug in the Intel GKL that could cause output files to intermittently fail to be compressed properly

Full list of changes:

- **HaplotypeCaller**
  - HaplotypeCaller now supports custom ploidy regions (#8609)
    - Added a new argument to `HaplotypeCaller` called `--ploidy-regions` which allows the user to input a `.bed` or `.interval_list` with the "name" column equal to a positive integer for the ploidy to use when calling variants in that region
    - The main use case is for calling haploid variants outside the PAR for XY individuals as required by the VCF spec, but this provides a much more flexible interface for other similar niche applications, like genotyping individuals with other known aneuploidies
    - The global `-ploidy` flag will still provide the background default (or the built-in ploidy of 2 for humans), but the user-supplied values will supersede these in overlapping regions
  - Changed the `SmithWaterman` implementation to default to `FASTEST_AVAILABLE` (#8485)
  - Fixed a bug in pileup calling mode relating to the number of haplotypes (#8489)
  - Huge simplication of genotyping likelihoods calculations -- no change in output (#6351)
  - Be explicit about when variants are biallelic (#8332)
  - Fixed debug log severity for read threading assembler messages (#8419)
  - Fixed issue with visibility of the `--dont-use-softclipped-bases` argument (#8271)

- **Mutect2**
  - Added a `--base-qual-correction-factor` to allow a scale factor to be provided to modify the base qualities reported by the sequencer and used in the `Mutect2` substitution error model (#8447)
    - Set to zero to turn off the error model changes introduced in GATK 4.1.9.0
  - Fixed a bug in `FilterMutectCalls` for GVCFs (#8458)
    - When using GVCFs with `Mutect2` (for example with the Mitochondria mode), in the filtering step ADs for symbolic alleles are set to 0 so it doesn't contribute to overall AD. There was an off-by-one error that removed the alt allele AD rather than the `<NON_REF>` allele AD. This led to NaNs and errors when a site had no ref reads (for example a GT of `[ref,alt,<NON_REF>]` and AD of `[0,300,0]` would accidentally be changed to an AD of `[0,0,0]` if the alt index was removed instead of the `<NON_REF>` index).
- **DRAGEN-GATK**
  - Added implementations of the "columnwise detection" and "PDHMM" (partially-determined HMM) features from DRAGEN to bring us much closer to functional equivalence with DRAGEN v3.7.8 (#8083)
  - Development work to prepare the way for the final missing DRAGEN 3.7.8 feature, "joint detection":
    - Graph method for PDHMM event groups that unifies finding/merging and overlap/mutual exclusion (#8366)
    - Rewrote haplotype construction methods in `PartiallyDeterminedHaplotypeComputationEngine` (#8367)
    - More refactoring in `PartiallyDeterminedHaplotypeComputationEngine` and preparing for joint detection (#8492)
    - Innocuous housekeeping changes in the partially-determined haplotypes code (#8361)
    - Clarify cryptic bitwise operations in the partially-determined haplotype `EventGroup` subclass (#8400)
- **Joint Calling**
  - Added haploid support to `GnarlyGenotyper` (#7750)
  - Fix to allow `GenotypeGVCFs` to properly handle events not in minimal representation (#8567)
  - `ReblockGVCF` : added a `--keep-site-filters` argument to keep site-level filters (#8304) (#8308)

- `ReblockGVCF` : added a `--add-site-filters-to-genotype` argument to move site-level filters to genotype-level filters (#8484)
- `ReblockGVCF` : added a `--format-annotations-to-remove` argument to specify format-level annotations to remove from all genotypes in final GVCF (#8411)
- `ReblockGVCF` : added a check to make sure the input VCF is a GVCF rather than a single sample VCF (#8411)
- Improved an error message in `GnarlyGenotyper` (#8270)
- Added a `mergeWithRemapping()` method in `ReferenceConfidenceVariantContextMerger` to perform allele remapping prior to genotyping (#8318)
- GVS (Genomic Variant Store) development:
  - Incorporated changes from the GVS branch to existing files (#8256)
  - Incorporated build changes from the GVS branch (#8249)
  - Merged non-GVS bits required by the GVS branch [VS-971] (#8362)

- **GenomicsDB**
  - Allow `GenomicsDBImport` to accept Azure `az://` URIs as input (#8438)
  - Updated to a newer `GenomicsDB` release with Java 17 support, improved error messages/logging, and generally improved performance (#8358)

- **Funcotator**
  - New data source release V1.8 (#8512)
    - Updated `Gencode` to version 43, and also updated `COSMIC` , `Clinvar` , and several other datasources to their latest versions
    - The data sources are now split by reference into separate hg19 and hg38 bundles to cut down on size
  - Fixed support for newer `Gencode` GTF versions by making the `GencodeGTFField` parsing more permissive (#8351)
  - Fixed `Funcotator` VCF output renderer to correctly preserve B37 contig names on output for B37 aligned files (#8539)
  - Fix bug in VCF comparison code that causes `Funcotator` to crash with certain datasources (#8445)
  - Connected the splice site window size to CLI parameters (#8463)
  - Allow `LocatableXsvFuncotationFactory` to read gzipped files (#8363)

- **CNV Calling**

- Matched gCNV pipeline arguments to those that were shown to have good performance in running large exome cohorts (#8234)
- Added resource usage section to the `GermlineCNVCaller` java doc (#8064)

- **SV Calling**
  - Added support for breakend replacement alleles in `SVCluster` (#8408)
    - Implements allele collapsing for "breakend replacement" BND alleles, as described in section 5.4 of the VCFv4.2 spec
  - Size similarity linkage and bug fixes for SV matching tools (#8257)
    - Added size similarity criterion to the `SVConcordance` and `SVCluster` tools. This is particularly useful for accurately matching smaller SVs that have a high degree of breakpoint uncertainty, in which case reciprocal overlap does not work well. PESR/mixed variant types must have size similarity, reciprocal overlap, and breakend window criteria met. Depth-only variants may have either size similarity + reciprocal overlap OR breakend window criteria met (or both).
  - Updated SV split-read strand validation and clustering (#8378)
    - Adds some flexibility to the allowed split-read strand annotations on SV records:
      - Allow INS -+ strands
      - Allow INV null strands
      - When clustering, only require that strands match for INV/BND records
  - Sample set and annotation improvements for `SVConcordance` (#8211)

- **Mitochondrial pipeline**
  - Added a variable for the user to specify the java heap size in Picard in the MT pipeline (#8406)
  - Exposed runtime attributes as arguments in the MT pipeline (#8413) (#8417)

- **Flow-based Calling**
  - New/updated flow-based read tools (#8579)
    - Added a new `GroundTruthScorer` tool to score reads against a reference/ground truth
    - Updated `FlowFeatureMapper`
  - Created an `AddFlowBaseQuality` tool that writes reads from flow-based SAM/BAM/CRAM files that pass criteria to a new file while adding a base-quality attribute (BQ) (#8235)
  - Added an experimental tool `FlowPairHMMAlignReadsToHaplotypes` that aligns flow-based reads to set of haplotypes / templates (#8305)

- Fixed an issue with reads that contain the tp tag sometimes being incorrectly identified as flow-based (#8337)
- Minor changes and fixes to flow-based annotations (#8442)
- Removed a line in `FlowBasedAnnotation` that contained a bug and thus was meaningless (#8421)
- Additional annotation in FeatureMap (#8347)
- Removed unnecessary flow-based argument and option (#8342)
- `GroundTruthScorer` doc update (#8597)
- Removed unnecessary and buggy validation check (#8580)

- **Notable Enhancements**
  - Major security fixes in our dependencies and docker environment
    - Updated the GATK base docker image to Ubuntu 22.04 for security fixes and newer versions of genomics packages like `samtools` and `bcftools` (#8610)
    - Updated GATK dependencies to address known security vulnerabilities, and added a vulnerability scanner to `build.gradle` (#8607)
  - Greatly improved HTTP support (#8611)
    - Updated the `http-nio` library and made tweaks to HTSJDK to make it available in more places. The new version of `http-nio` should provide much more reliable access to http(s) file paths. This is supported by all methods accessing Paths, and includes SAM/BAM/CRAM and VCF/Feature file...

Read more

▶ **Assets** 3

🎉 12 12 people reacted

Mar 16, 2023

droazen 4.4.0.0 2dbc025 Compare

4.4.0.0

Download release: gatk-4.4.0.0.zip
Docker image: https://hub.docker.com/r/broadinstitute/gatk/

Highlights of the 4.4.0.0 release:

- We've moved to Java 17, the latest long-term support (LTS) Java release, for building and running GATK! Previously we required Java 8, which is now end-of-life.
    - Newer non-LTS Java releases such as Java 18 or Java 19 may work as well, but since they are untested by us we only officially support running with Java 17.
- Significant enhancements to `SelectVariants`, including arguments to enable `GVCF` filtering support and to work with genotype fields more easily.
- A new tool `SVConcordance`, that calculates SV genotype concordance between an "evaluation" VCF and a "truth" VCF
- Bug fixes and enhancements to the support for the Ultima Genomics flow-based sequencing platform introduced in GATK 4.3.0.0

Full list of changes:

- **Flow-based Variant Calling**
    - `FlowFeatureMapper`: added surrounding-median-quality-size feature (#8222)
    - Removed hardcoded limit on max homopolymer call (#8088)
    - Fixed bug in dynamic read disqualification (#8171)
    - Fixed a bug in the parsing of the T0 tag (#8185)
    - Updated flow-based calling `Mutect2` parameters to make them consistent with the `HaplotypeCaller` parameters (#8186)
- **SelectVariants**
    - Enabled GVCF type filtering support in `SelectVariants` (#7193)
        - Added an optional argument `--ignore-non-ref-in-types` to support correct handling of VariantContexts that contain a NON_REF allele. This is necessary because every variant in a GVCF file would otherwise be assigned the type MIXED, which makes it impossible to filter for e.g. SNPs.
        - Note that this only enables correct handling of GVCF input. The filtered output files are VCF (not GVCF) files, since reference blocks are not extended when a variant is filtered out.
    - `SelectVariants`: added new arguments for controlling genotype JEXL filtering (#8092)

    - `-select-genotype` : with this new genotype-specific JEXL argument, we support easily filtering by genotype fields with expressions like 'GQ > 0', where the behavior in the multi-sample case is 'GQ > 0' in at least one sample. It's still possible to manually access genotype fields using the old `-select` argument and expressions such as `vc.getGenotype('NA12878').getGQ() > 0` .
    - `--apply-jexl-filters-first` : This flag is provided to allow the user to do JEXL filtering before subsetting the format fields, in particular the case where the filtering is done on INFO fields only, which may improve speed when working with a large cohort VCF that contains genotypes for thousands of samples.

- **SV Calling**
  - Added a new tool `SVConcordance` , that calculates SV genotype concordance between an "evaluation" VCF and a "truth" VCF (#7977)
  - Recognize MEI DELs with ALT format DEL:ME in `SVAnnotate` (#8125)
  - Don't sort rejected reads output from `AnalyzeSaturationMutagenesis` (#8053)

- **Notable Enhancements**
  - `GenotypeGVCFs` : added an `--keep-specific-combined-raw-annotation` argument to keep specified raw annotations (#7996)
  - `VariantAnnotator` now warns instead of fails when the variant contains too many alleles (#8075)
  - Read filters now output total reads processed in addition to the number of reads filtered (#7947)
  - Added `GenomicsDB` arguments to the `CreateSomaticPanelOfNormals` tool (#6746)
  - Added a `DeprecatedFeature` annotation and a process for officially marking GATK tools as deprecated (#8100)
  - Prevent tool `close()` methods from hiding underlying errors (#7764)

- **Bug Fixes**
  - Fixed issue causing `VariantRecalibrator` to sometimes fail if user provided duplicate -an options (#8227)
  - `ReblockGVCF` : remove A,R, and G length attributes when `ReblockGVCF` subsets an allele (#8209)
    - Previously if an input gVCF had allele length, reference length, or genotype length annotations in the FORMAT field, `ReblockGVCF` would not remove all of them at sites where an allele was dropped. This makes the output gVCF invalid since the annotation length no longer matches the length described in the header at those sites. Now we fix up F1R2, F2R1, and AF annotations and

remove any other annotations that are not already handled that are defined as A, R, or G length in the header.

- Fixed a `gCNV` bug that breaks the inference when only 2 intervals are provided (#8180)
- Fixed NPE from unintialized logger in `GenotypingEngine` (#8159)
- Fixed asynchronous Python exception propagation in `StreamingPythonExecutor` / `CNNScoreVariants` (#7402)
- Fixed issue in `ShiftFasta` where the interval list output was never written (#8070)
- Bugfix for the type of some output files in the somatic CNV WDL (#6735) (#8130)
- `MergeAnnotatedRegions` now requires a reference as asserted in its documentation (#8067)

- **Miscellaneous Changes**
  - Deprecated an untested `VariantRecalibrator` argument and an old `ReblockGVCF` argument that produced invalid GVCFs (#8140)
  - Removed old `GnarlyGenotyper` code with a diploid assumption to prepare for adding haploid support to `GnarlyGenotyper` (#8140)
  - `ReblockGVCF` : add error message for when tree-score-threshold is set but the TREE_SCORE annotation is not present (#8218)
  - `TransferReadTags` : allow empty unaligned bams as input (#8198)
  - Refactored `JointVcfFiltering` WDL and expanded tests. (#8074)
  - Updated the carrot github action workflow to the most recent version, which supports using `#carrot_pr` to trigger branch vs master comparison runs (#8084)
  - Replaced uses of `File.createTempFile()` with `IOUtils.createTempFile()` to ensure that temp files are deleted on shutdown (#6780)
  - Don't require python just to instantiate the `CNNScoreVariants` tool classes. (#8128)
  - Made several `Funcotator` methods and fields protected so it is easier to extend the tool (#8124) (#8166)
  - Test for presence of ack result message and simplify `ProcessControllerAckResult` API (#7816)
  - Fixed the path reported by the gatkbot when there are test failures (#8069)
  - Fixed incorrect boolean value in `DirichletAlleleDepthAndFractionIntegrationTest` (#7963)
  - Removed two ancient and unused `HaplotypeCaller` test files that are no longer needed (#7634)
  - Added scattered gCNV case WDL to dockstore file (#8217)

- **Documentation**
  - Updated instructions for installing Java in the README (#8089)
  - Added documentation on `OMP_NUM_THREADS` and `MKL_NUM_THREADS` to `GermlineCNVCaller` and `DetermineGermlineContigPloidy` (#8223)
  - Improvements to `PileupDetectionArgumentCollection` documentation (#8050)
  - Fixed typo in documentation for `VariantAnnotator` (#8145)
- **Dependencies**
  - Moved to `Java 17` , the latest LTS Java release, for building/running GATK (#8035)
  - Updated `Gradle` to 7.5.1 (#8098)
  - Updated the GATK base docker image to 3.0.0 (#8228)
  - Updated `HTSJDK` to 3.0.5 (#8035)
  - Updated `Picard` to 3.0.0 (#8035)
  - Updated `Barclay` to 5.0.0 (#8035)
  - Updated `GenomicsDB` to 1.4.4 (#7978)
  - Updated `Spark` to 3.3.1 (#8035)
  - Updated `Hadoop` to 3.3.1. (#8102)
  - Require `commons-text` 1.10.0 to fix a security vulnerability (#8071)

▶ **Assets** 3

🎉 1 1 person reacted

2 Join discussion

Oct 12, 2022

droazen 4.3.0.0 8dbb78f Compare

4.3.0.0

Download release: gatk-4.3.0.0.zip
Docker image: https://hub.docker.com/r/broadinstitute/gatk/

Highlights of the 4.3.0.0 release:

- Support for the Ultima Genomics flow-based sequencing platform
- A next-generation suite of tools for variant filtration based on site-level annotation, intended to eventually supersede the older `VariantRecalibrator` workflow
- `CompareReferences` and `CheckReferenceCompatibility` : new tools for comparing and checking compatibility with genomic references
- Support in `HaplotypeCaller` / `Mutect2` for supplementing the variants discovered in local assembly with variants discovered via a pileup-based approach

Full list of changes:

- **Support for the Ultima Genomics flow-based sequencing platform** (#7876)
    - Added a new `--flow-mode` argument to `HaplotypeCaller` which better supports flow-based calling
        - Added a new Haplotype Filtering step after assembly which removes suspicious haplotypes from the genotyper
        - Added two new likelihoods models, `FlowBasedHMM` and the `FlowBasedAlignmentLkelihoodEngine`
    - Added a new `--flow-mode` argument to `Mutect2` which better supports flow-based calling
    - Added support for uncertain read end-positions in `MarkDuplicatesSpark`
    - Added a new tool `FlowFeatureMapper` for quick heuristic calling of bams for diagnostics
    - Added a new tool `GroundTruthReadsBuilder` to generate ground truth files for Basecalling
    - Added a new diagnostic tool `HaplotypeBasedVariantRecaller` for recalling VCF files using the `HaplotypeCallerEngine`
    - Added a new tool breaking up CRAM files by their blocks, `SplitCram`
    - Added a new read interface called `FlowBasedRead` that manages the new features for FlowBased data
    - Added a number of flow-specific read filters
    - Added a number of flow-specific variant annotations
    - Added support for read annotation-clipping as part of clipreads and GATKRead

- Added a new `PartialReadsWalker` that supports terminating before traversal is finished

- **Next-generation suite of tools for variant filtration based on site-level annotations** (#7954) (#8049)
  - This tool suite is intended to eventually supersede the older `VariantRecalibrator` workflow
  - The new tools include:
    - `ExtractVariantAnnotations` : extracts site-level variant annotations, labels, and other metadata from a VCF file to HDF5 files
    - `TrainVariantAnnotationsModel` : trains a model for scoring variant calls based on site-level annotations
    - `ScoreVariantAnnotations` : scores variant calls in a VCF file based on site-level annotations using a previously trained model

- **New Reference Comparison Tools**
  - `CompareReferences` : a new tool for analyzing the differences between references at both the dictionary and the base level (#7930) (#7987) (#7973)
    - In its default mode, this tool uses the reference dictionaries to generate an MD5-keyed table comparing the specified references, and does an analysis to summarize the differences between the references provided.
    - Comparisons are made against a "primary" reference, specified with the `-R` argument. Subsequent references to be compared may be specified using the ``--references-to-compare` argument.
    - A supplementary table keyed by sequence name can be displayed using the `--display-sequences-by-name argument` ; to display only sequence names for which the references are not consistent, run with the `--display-only-differing-sequences` argument as well.
    - MD5s can be recalculated from the actual sequence when missing from the dictionary
    - When run with `--base-comparison FULL_ALIGNMENT` , the tool performs full-sequence alignment on the differing reference sequences to produce a VCF with SNPs and Indels. However, this mode ignores IUPAC / N bases.
    - Running with `--base-comparison FIND_SNPS_ONLY` finds single-base differences between differing reference sequences of the same length. This mode can handle IUPAC / N bases correctly, but not indels.
    - To perform the full-sequence alignment, GATK now packages a distribution of `MUMmer` for x86_64 Mac and Linux, which can be invoked from within the GATK

using the new `MummerExecutor` class.

- `CheckReferenceCompatibility` : a new tool to check a BAM/CRAM/VCF for compatibility against a set of references (#7959) (#7973)
  - This tool generates a table analyzing the compatibility of a BAM/CRAM/VCF input file against provided references.
  - The tool works to compare BAM/CRAMs (specified using the -I argument) as well as VCFs (specified using the -V argument) against provided reference(s), specified using the `--references-to-compare` argument.
  - When MD5s are present, the tool decides compatibility based on all sequence information (MD5, name, length); when MD5s are missing, the tool makes compatibility calls based only on sequence name and length.

- **HaplotypeCaller/Mutect2**
  - Added an optional "Pileup Detection" step to `Mutect2` and `HaplotypeCaller` before assembly that supplements the variants from local assembly with variants that show up in the pileups (#7432)
  - Fixed a `Mutect2` `IndexOutOfBoundException` with germline resource (#7979)
  - `Mutect3` dataset enhancements: optional truth VCF for labels, seq error likelihood annotation (#7975)
  - Added `Mutect3` dataset generation to the `Mutect2` WDL (#7992)
  - `GetPileupSummaries` now streams its output rather than storing it in memory (#7664)
  - Fixed a rare edge case in the `AdaptiveChainPruner` where the `JavaPriorityQueue` is undefined for tied elements (#7851)

- **SV Calling**
  - `CondenseDepthEvidence` : a new tool that combines adjacent intervals in DepthEvidence files (#7926)
  - `LocusDepthtoBAF` : a new tool that merges locus-sorted LocusDepth evidence files, calculates the bi-allelic frequency (baf) for each sample and site, and writes these values as a BafEvidence output file (#7776)
  - `PrintReadCounts` : a new tool that prints (and optionally subsets) an read depth (DepthEvidence) file or a counts file as one or more (for multi-sample DepthEvidence files) counts files for CNV determination (#8015)
  - `CollectSVEvidence` : fixed a bug where trailing SNP sites and depth intervals without read coverage were being omitted from the output (#8045)
  - `CollectSVEvidence` : added read depth generation and raw-counts output (#8015)

- Improved `PrintSVEvidence` performance by tweaking the `MultiFeatureWalker` traversal (#7869)
- Fixes related to `BafEvidence` (biallelic-frequency of a sample at some locus) (#7861)
- Fixed a bug where the end coordinate was being incorrectly compared when sorting discordant read pair evidence (#7835)
- Sort output from `SVClusterEngine` (#7779)
- Remove abandoned SV filtering project and unneeded build dependency (#7950)

- **CNV Calling**
  - Fix a no-call genotype ploidy bug in `JointGermlineCNVSegmentation` (#7779)
  - Added numerical-stability tests and updated test data for all `ModelSegments` single-sample and multiple-sample modes (#7652)
  - Added a gCNV integration test to detect numerical differences in the outputs (#7889)

- **GenomicsDB**
  - `GenomicsDBImport` : added the ability to specify explicit index locations via the sample name map file (#7967)
    - Each line in the sample name map file may now optionally contain a third column with the path/URI to the index. This is useful when the index is not in the same location as the corresponding GVCF.

- **Bug Fixes**
  - Fixed an issue where we weren't properly merging AD values when combining GVCFs and no PLs were present (#7836)
  - Fixed a bug in `ReblockGVCF` that could cause the first position on a contig to be dropped (#8028)
  - Fixed an allele-ordering issue in the allele-specific annotation code (#7585)
  - `VariantRecalibrator` : type change int -> long to prevent tranche novel variant count overflow (#7864)
  - Fixed an issue with tabix index generation (#7858)
  - Fixed a bug in `SiteDepthCodec` (#7910)

- **Miscellaneous Changes**
  - `VariantsToTable` now includes all fields when none are specified (#7911)

- `SelectVariants` now warns the user about poor performance when the sample names in the VCF header are unsorted (#7887)
- `VariantRecalibrator` now has a `--dont-run-rscript` argument to disable execution of its R script but still output the actual R script file (#7900)
- Added some generic read tag/expression filters for use on numeric tags (#7746)
- Replaced Travis CI with Github Actions for our continuous testing (#7754)
- Switched over to Github Actions for building our nightly docker image (#7775)
- Created a new `build_docker_remote.sh` script for building the docker image remotely with Google Cloud Build (#7951)
- Added an argument mode manager for group arguments and a demonstration of how it might be used in `HaplotypeCaller --dragen-mode` (#7745)
- Added unit tests for the `Utils.concat()` methods (#7918)
- Added a test to validate WDLs in the scripts directory. (#7826)
- Added a `use_allele_specific_annotation` arg and fixed task with empty input in the `JointVcfFiltering` WDL (#8027)
- Fixed an issue in the GATK stats script in which the first day's downloads on a new release were set to 0 (#7794)
- Fixed a typo in the Dockerfile that broke git lfs pull (#7806)
- Removed unused code in the `utils.solver` package (#7922)
- Corrected the time for GATK nightly build cron jobs (#7784)
- Disabled the red "X" from failing `CodeCov` builds and de...

Read more

▸ **Assets** 3

👍 10 10 people reacted

Apr 13, 2022

droazen 4.2.6.1 33bda5e Compare

4.2.6.1

Download release: gatk-4.2.6.1.zip
Docker image: https://hub.docker.com/r/broadinstitute/gatk/

Highlights of the 4.2.6.1 release:

This release contains a single bug fix for `GenotypeGVCFs` to fix an erroneous `IllegalStateException` ("No likelihood sum exceeded zero -- method was called for variant data with no variant information.") in the edge case where unnormalized PLs are present at monomorphic sites.

▶ **Assets** 3

👍 7 7 people reacted

Apr 8, 2022

droazen 4.2.6.0 3b0bc03 Compare

4.2.6.0

Download release: gatk-4.2.6.0.zip
Docker image: https://hub.docker.com/r/broadinstitute/gatk/

Highlights of the 4.2.6.0 release:

- Important bug fixes for the joint calling tools (GenotypeGVCFs / GenomicsDB)
  - GATK 4.2.5.0 contained two joint genotyping bugs that are now fixed in GATK 4.2.6.0:
    - `GenotypeGVCFs` can throw NullPointerExceptions in some cases with many alternate alleles.
    - The expectation-maximization component of the QUAL calculation was disabled, leading to false positive, low quality alleles at some multi-allelic sites.
  - **If you are running these tools in 4.2.5.0 we strongly recommend updating to 4.2.6.0**
- Fixed a "Bucket is a requester pays bucket but no user project provided" error that occurred when accessing requester pays buckets in Google Cloud Storage even when the `--gcs-project-for-requester-pays` argument was specified

- **If you continue to encounter problems accessing requester pays Google Cloud Storage buckets in 4.2.6.0, please let us know by filing a Github issue!**

- Two new tools for the Structural Variation calling pipeline: `SVAnnotate` and `PrintSVEvidence`
- Some fixes to genotype-given-alleles mode in `HaplotypeCaller` and `Mutect2`

Full list of changes:

- **Joint Calling (GenotypeGVCFs / GenomicsDB)**
  - GATK 4.2.5.0 contained two joint genotyping bugs which are now fixed in 4.2.6.0:
    - `GenotypeGVCFs` can throw NullPointerExceptions in some cases with many alternate alleles.
      - Fixed in:
        - Fix for `NullPointerException` when GenomicsDB has more ALT alleles than specified maximum and many GQ0 hom-ref genotypes allow variants to pass the QUAL filter (#7738)
    - The expectation-maximization component of the QUAL calculation was disabled, leading to false positive, low quality alleles at some multi-allelic sites.
      - Fixed in:
        - Fix multi-allelic QUAL calculation and restore some missing ALT annotation data in `ReblockGVCFs` (#7670)
  - Mention acceptable compressed VCF file extensions in `GenomicsDBImport` error message (#7692)

- **SV Calling**
  - Added a new tool `SVAnnotate` (#7431)
    - `SVAnnotate` adds functional annotations for SVs called by `GATK-SV` (#7431)
  - Added a new tool `PrintSVEvidence` (#7695)
    - `PrintSVEvidence` is a tool that can merge any number of files containing one of five types of evidence of structural variation. It's also capable of subsetting regions or samples. It's used to merge evidence from a cohort in the `GATK-SV` pipeline.
  - Added start/end coordinate validation to `SVCallRecord` (#7714)

- **HaplotypeCaller / Mutect2**
  - Fixed an edge case in `HaplotypeCaller` where filtered alleles in the vicinity of forced-calling alleles could result in empty calls (#7740)

        - This affects users who run genotype given alleles mode in non-GVCF mode
    - Fixed a bug in `HaplotypeCaller` and `Mutect2` where force-calling alleles were lost upon trimming by placing allele injection after trimming (#7679)
    - Added a debug ``--pair-hmm-results-file` argument that dumps the the exact inputs/outputs of the PairHMM to a file (#7660)
    - Some changes to `Mutect2` to support the future `Mutect3` (#7663)
        - Added training data for the Mutect3 normal artifact filter
        - Output tensors for Mutect3 as plain text rather than VCF

- **RNA Tools**
    - `TransferReadTags` : a new tool that transfers a read tag from an unaligned bam to the matching aligned bam (#7739).
        - This tool allows us to retrieve read tags that get lost when converting a SAM file to fastqs, then back to SAM (which is necessary if e.g. running fastp to clip adapter bases before alignment).
    - `PostProcessReadsForRSEM` : a new tool that re-orders and filters reads before running RSEM, which has stringent requirements on the input SAM (https://github.com/deweylab/RSEM) (#7752).

- **Funcotator**
    - Added custom `VariantClassification` severity ordering. (#7673)
        - Users can now customize the severity ratings of the various `VariantClassifications` using the new `--custom-variant-classification-order` argument
    - Added logging statements to the b37 conversion process explaining why the automatic b37 conversion does or does not take place on their VCFs (#7760)

- **VariantRecalibrator**
    - Added regularization to covariance in GMM maximization step to fix convergence issues in `VariantRecalibrator` (#7709)
        - This makes the tool more robust in cases where annotations are highly correlated

- **Bug Fixes**
    - Fixed a "Bucket is a requester pays bucket but no user project provided" error that occurred when accessing requester pays buckets in Google Cloud Storage even when `--gcs-project-for-requester-pays` was specified (#7700) (#7730)

- Fix for the `PossibleDeNovo` annotation to work without Genotype Likelihoods (#7662)
  - `PossibleDeNovo` checks each trio's genotype (including parent hom ref genotypes) for likelihoods even though it doesn't actually use the PLs. The PLs can get dropped if GVCFs are reblocked which means this annotation no longer works as expected. This changes the check to look for GQs instead of PLs as the GQs are used as part of the annotation.
- Fixed a bug with the `--mate-too-distant-length` in `MateDistantReadFilter` not being configurable (#7701)

- **GATK Engine**
  - Added a new `MultiFeatureWalker` traversal to the GATK engine (#7695)
  - Removed an ancient, unused option to track unique reads in a `LocusIteratorByState` (#6410)

- **Miscellaneous Changes**
  - Added back the `jcenter` repository resolver to our gradle build, fixing a "Could not find biz.k11i:xgboost-predictor:0.3.0" error when building GATK from source (#7665)
  - We now properly update the `latest` tag in the `broadinstitute/gatk-nightly` Dockerhub repo (#7703)
  - The docker build now only does a `git lfs pull` on `src/main/resources/large` (#7727)
  - Install git lfs with --force in the `Dockerfile` (#7682)
  - Fix WDL generation for `MultiVariantWalkers` by adding a companion index to the `MultiVariantWalker` input variant arg (#7689)
  - Added google apps script to automatically update GATK release stats. (#7637)
  - Updated the GATK stats script to be more universally usable (#7759)
  - Added `JointCallExomeCNVs` to `.dockstore.yml` and included a note in the WDL (#7719)

- **Documentation**
  - Corrected the docs for the `--heterozygosity` argument in the `GenotypeCalculationArgumentCollection` (#7661)

- **Dependencies**
  - Updated `Picard` to `2.27.1` (#7766)
  - Updated `google-cloud-nio` to `0.123.25` (#7730)

▶ **Assets** 3

2 people reacted

Feb 4, 2022

droazen 4.2.5.0 da7cd83 Compare

4.2.5.0

Download release: gatk-4.2.5.0.zip
Docker image: https://hub.docker.com/r/broadinstitute/gatk/

Highlights of the 4.2.5.0 release:

- Fixed a `GenotypeGVCFs` `IllegalStateException` error reported by multiple users in #7639
- Added a new tool `SVCluster` that clusters structural variants based on coordinates, event type, and supporting algorithms.

Full list of changes:

- **Joint Calling (GenotypeGVCFs / GenomicsDB)**
  - Fixed an `IllegalStateException` in `GenotypeGVCFs` arising from GenomicsDB output with too many alts and no likelihoods, and also added a `--genomicsdb-max-alternate-alleles` argument that is separate from the `--max-alternate-alleles` argument used by `GenotypeGVCFs` (#7655)
    - This fixes the `GenotypeGVCFs` error reported in #7639
    - The new `--genomicsdb-max-alternate-alleles` argument is required to be at least one greater than the `--max-alternate-alleles` argument, to account for the NON_REF allele.
  - `ReblockGVCF` : fixed an edge case where hom-ref "variant" records with no data had wrong-sized PLs and didn't merge with adjacent blocks (#7644)
- **SV Calling**

- Added a new tool `SVCluster` that clusters structural variants based on coordinates, event type, and supporting algorithms. (#7541)
  - Primary use cases include:
    - Clustering SVs produced by multiple callers, based on interval overlap, breakpoint proximity, and sample overlap.
    - Merging multiple SV VCFs with disjoint sets of samples and/or variants.
    - Defragmentation of copy number variants produced with depth-based callers.

- **Mutect2**
  - The palindrome ITR artifact transformer now skips reads whose contigs are not in sequence dictionary (#6968)
    - This fixes a NullPointerException error in `Mutect2` reported in #6851

- **GATK Engine**
  - Added a new read filter, `ExcessiveEndClippedReadFilter` (#7638)
    - This filter will keep reads that have fewer than the specified number of clipped bases on either end.
    - Designed with long reads in mind, and as a result has a default value of 1000.

▶ **Assets** 3

Jan 4, 2022

lbergelson 4.2.4.1 5b87367 Compare

4.2.4.1 the log4j strikes back

Download release: gatk-4.2.4.1.zip
Docker image: https://hub.docker.com/r/broadinstitute/gatk/

Highlights of the 4.2.4.1 release:

- Fix more newly discovered log4j2 vulnerabilities. Now that people are paying attention they are finding all sorts of things.

Full list of changes:

- **Build System**
  - Upgrade our build from Gradle 5.6 to the newest 7.3.2 (#7609)
  - This fixes some gradle bugs which were blocking development
- **GenomicsDB**
  - Update to genomicsdb 1.4.3 (#7613) which fixes #7598
  - Fix bug which caused --max_alternate_alleles to be ignored when using GenomicsDB (#7576)
- **Miscellaneous Changes**
  - Update .dockstore.yml (#7595)
  - Fix developer doc in AS_RMSMappingQuality (#7607)
- **Dependencies**
  - Update log4j to 2.17.1 (#7624)(#7615)
  - Upgrade to Barclay 4.0.2. (#7602)
  - Update to genomicsdb 1.4.3 (#7613)

▶ **Assets** 3

EXHIBIT 1069.32

broadinstitute / gatk Public

Code Issues 1.3k Pull requests 140 Discussions Actions Projects

Releases Tags Find a release

Dec 15, 2021

lbergelson 4.2.4.0 2d3d4aa Compare

4.2.4.0 the log4shell edition

Download release: gatk-4.2.4.0.zip
Docker image: https://hub.docker.com/r/broadinstitute/gatk/

Highlights of the 4.2.4.0 release:

- Fix a major security bug due to log4j vulnerability. (CVE-2021-44228)
- Improvement to calculation of ExcessHet in joint genotyping. (GenotypeGVCFs, GnarlyGenotyper, ExcessHet).

Full list of changes:

- **Funcotator**
  - Aligned the Funcotator checkIfAlreadyAnnotated test with the Funcotator engine code. (#7555)
- **GenotypeGVCFs / ExcessHet**
  - Removed undocumented mid-p correction to p-values in exact test of Hardy-Weinberg equilibrium and updated corresponding tests. We now report the same value as ExcHet in bcftools. Note that previous values of 3.0103 (corresponding to mid-p values of 0.5) will now be 0.0000. (#7394)
  - Updated expected ExcessHet values in integration test resources and added an update toggle to GnarlyGenotyperIntegrationTest.

- Updated ExcessHet documentation.

- **Miscellaneous Changes**
  - Delete an unused .gitattributes file which was unintentionally stored in git-lfs and caused an error message to appear sometimes when checking out the repository. (#7594)
  - Remove trailing tab in VariantsToTable output header (#7559)

- **Documentation**
  - Updated AUTHORS file to remove a contributor's name at their request. (#7580)
  - Remove outdated javadoc line in AssemblyBasedCallerUtils (#7554)

- **Dependencies**
  - Updated log4j to version 2.13.1 -> 2.16.0 to patch CVE-2021-44228 (#7605)

▶ **Assets** 3

❤️ 4 👀 1 4 people reacted

Nov 2, 2021

droazen 4.2.3.0 f31f019 Compare

4.2.3.0

Download release: gatk-4.2.3.0.zip
Docker image: https://hub.docker.com/r/broadinstitute/gatk/

Highlights of the 4.2.3.0 release:

- Notable bug fixes for `Mutect2` and `Funcotator`
- Support in `CombineGVCFs` and `GenotypeGVCFs` for "reblocked" GVCFs as produced by the `ReblockGVCF` tool. Reblocked GVCFs have a significantly reduced storage footprint.
- More control over the Smith-Waterman parameters in `HaplotypeCaller` and `Mutect2`

- A new Fragment Allele Depth (`FAD`) variant annotation similar to the `AD` annotation except that allele support is considered per read pair, not per individual read
- GenomicsDB bug fixes and enhancements

Full list of changes:

- **HaplotypeCaller/Mutect2**
  - Fixed a bug where `Mutect2` failed to filter germline variants with alternate representations (#7103)
    - This caused variants with alternative representations in gnomAD to not be recognized as being the same as called variants in some cases. This resulted in variants that were called and not filtered, but they should have been filtered by "germline".
  - Exposed Smith-Waterman parameters as tool arguments in `HaplotypeCaller` , `Mutect2` , and `FilterAlignmentArtifacts` . (#6885)
    - Enables use of alternative parameters for different event representation (e.g. three consecutive SNPs instead of two small indels)
  - Can now specify the Smith-Waterman implementation in `FilterAlignmentArtifacts` (#7105)
  - Added a `--debug-assembly-variants-out` diagnostic option to output a side VCF with variants detected by assembly for `HaplotypeCaller` and `Mutect2` (#7384)
  - `Mutect2` : the `--genotype-germline-sites` argument is no longer marked as experimental (#7533)
- **GenotypeGVCFs / CombineGVCFs**
  - Updated `CombineGVCFs` and `GenotypeGVCFs` to handle "reblocked" GVCFs with diploid data that are potentially missing hom-ref genotype PLs (#7223)
  - Homozygous reference genotypes with no PLs and zero depth are now output as no-calls by `GenotypeGVCFs` (#7471)
  - Bug fixes for `GenotypeGVCFs` / `GnarlyGenotyper` when allele-specific annotations have empty values due to lack of informative reads or no depth (#7491) (#7186)
- **GenomicsDB**
  - Added a new `--call-genotypes` GenomicsDB argument, enabling output of called genotypes (i.e. not ./.) when tools like `CombineGVCFs` and `SelectVariants` read from a GenomicsDB workspace (#7223)

- Added a `--bypass-feature-reader` argument to `GenomicsDBImport` to allow the C-based htslib VCF reader implementation to be used instead of the Java implementation (#7393)
  - Using this option will reduce memory usage and potentially speed up the import process
- Updated to GenomicsDB 1.4.2 (#7520)
  - This release fixes a commonly-encountered bookkeeping issue with GenomicsDB array fragments. Should fix errors of the type: "Error: Cannot read from buffer; Error: cannot load book-keeping" as reported in #7012
  - Full release notes are here: https://github.com/GenomicsDB/GenomicsDB/releases/tag/v1.4.2

- **Funcotator**
  - Fixed a `StringIndexOutOfBoundsException` in the protein change prediction code that could be triggered by certain indels. The fix avoids the crash by adding additional bounds checking. (#7513)
  - Allow `FilterFuncotations` to process multi-transcript genes (#7506)

- **CNV Calling**
  - CNV WDLs now handle BAM/CRAM index paths explicitly, as for cases where the index is not in the same path as its file (#7518)
  - gCNV in the CASE mode now fills in all hidden DenoisingModelConfig and CopyNumberCallingConfig arguments from the input model configuration (#7464)
  - Exposed number of samples used for estimating denoised copy ratios in gCNV via a new `--num-samples-copy-ratio-approx` argument (#7450)

- **SV Calling**
  - `JointGermlineCNVSegmentation` : bug fixes and refactoring (#7243)
    - A number of bugs, particularly with max-clique clustering, have been fixed, as well as a parameter swap bug in `JointGermlineCNVSegmentation`
    - Reworks classes used by `JointGermlineCNVSegmentation` for SV clustering and defragmentation. The design of `SVClusterEngine` has been overhauled to enable the implementation of `CNVDefragmenter` and `BinnedCNVDefragmenter` subclasses. Logic for producing representative records from a collection of clustered SVs has been separated into an `SVCollapser` class, which provides enhanced functionality for handling genotypes for SVs more generally.

- **Notable Enhancements**

    - Added a new Fragment Allele Depth (`FAD`) variant annotation (#7511)
        - This annotation is identical to the `AD` annotation except that allele support is considered per read pair, not per individual read

- **Miscellaneous Changes**
    - `SplitIntervals` : added new tool arguments to control output file naming (#7488)
    - Fixed an issue that caused the Travis CI test suite reports to fail to be uploaded (#7525)
    - Updated Travis CI authentication information (#7521)

- **Documentation**
    - Updated `StrandBiasBySample` documentation (#7283)
    - Updated `MarkDuplicatesSpark` documentation (#7191) (#7535)
    - Added a comment to ``.travis.yml` about the checkout depth (#7421)

- **Dependencies**
    - Updated to `GenomicsDB` 1.4.2 (#7520)
    - Updated `sqlite-jdbc` library to a newer version to support M1 Macs (#7519)

▶ **Assets** 3

Aug 18, 2021

droazen 4.2.2.0 cafc6ce Compare

4.2.2.0

Download release: gatk-4.2.2.0.zip
Docker image: https://hub.docker.com/r/broadinstitute/gatk/

Highlights of the 4.2.2.0 release:

- The `ReblockGVCF` tool is now out of beta with several important improvements. This tool can be used to postprocess `HaplotypeCaller` GVCFs to decrease filesize.

- `FilterMutectCalls` now has a `--microbial-mode` argument that sets filters to defaults appropriate for microbial calling
- Important bug fixes to `CalibrateDragstrModel` and `Funcotator`

Full list of changes:

- **New Tools**
  - `ShiftFasta` : create a fasta with the bases shifted by an offset (#6694)
- **ReblockGVCF**
  - `ReblockGVCF` is now out of beta (#7419)
  - Improved `ReblockGVCF` output to eliminate overlapping reference blocks and reference gaps following trimmed deletions (#7122)
  - Fixed bugs associated with input no-call genotypes and fixed an off-by-one error at contig starts (#7404)
  - Fixed an error on ref blocks with missing DPs (if `--floor-blocks` arg is not provided); fixed rare cases where spanning deletion (*) allele is incorrectly modified (#7400)
- **Mutect2**
  - `FilterMutectCalls` : added a `--microbial-mode` argument that sets filters to defaults appropriate for microbial calling (#6694)
- **ValidateVariants**
  - Added an optional argument to check for GVCF reference blocks overlapping variants or other reference blocks (#7405)
- **DRAGEN-GATK**
  - Fixed a thread safety issue in `CalibrateDragstrModel` that could cause intermittent `ArrayIndexOutOfBoundsException` (#7417)
  - Added documentation for `ComposeSTRTableFile` (#7409)
- **Funcotator**
  - Fixed an issue where the `Match_Norm_Seq_Allele1` and `Match_Norm_Seq_Allele2` fields were not being populated in MAF output (#7422)
- **Mitochondrial pipeline**

- Removed calls to `FilterNuMTs` and `FilterLowHetSites`, which are no longer being used (#7325)

- **CNV Calling**
  - Fixed a bug resulting from prefix strings of less than 3 characters when creating temporary files in `GermlineCNVCaller` and improved documentation of corresponding utility methods. (#7411)

- **Documentation**
  - Fixed an argument name typo in the `CombineGVCFs` docs (#7413)
  - Fixed the wording of a comment in `MultiVariantDataSource` (#7388)

▶ **Assets** 3

Jul 30, 2021

droazen 4.2.1.0 9951f77 Compare

4.2.1.0

Download release: gatk-4.2.1.0.zip
Docker image: https://hub.docker.com/r/broadinstitute/gatk/

Highlights of the 4.2.1.0 release:

- Several important fixes to HaplotypeCaller and the new DRAGEN-GATK code introduced in GATK 4.2.0.0

- Started laying the groundwork in `Mutect2` for `Mutect3`, which will be more machine learning focused

- `LocalAssembler`: a new tool that performs local assembly of small regions to discover structural variants (#6989)

- Support for multi-sample segmentation in `ModelSegments`

- Major speed improvements and several important fixes to `Funcotator`
- A new version of the Intel Genomics Kernel Library (GKL), with many important fixes and improvements
- A new version of GenomicsDB, with improved cloud support
- A GATK-wide option to shard VCFs on output, which is often useful for pipelining
- GATK support for block compressed interval (`.bci`) files, which is useful when working with extremely large interval lists

Full list of changes:

- **New Tools**
  - `LocalAssembler` : a new tool that performs local assembly of small regions to discover structural variants (#6989)
- **HaplotypeCaller**
  - Fixed a rare edge case in DRAGEN mode that could result in negative GQs when `USE_POSTERIOR_PROBABILITIES` is set (#7120)
  - Fixed a rare edge case (mainly affecting DRAGEN mode) that could cause the PL arrays to be deleted when genotyping in `HaplotypeCaller` (#7148)
  - Fixed a bug in the `AlleleLikelihoods` that could result in new evidence X being assigned arbitrary likelihoods left over from previous evidence (#7154)
  - Fixed a "Padded span must contain active span" error caused by invalid feature file intervals that weren't being checked for validity against the sequence dictionary (#7295)
  - Do not add the artificial haplotype read group to the bamout file when `--bam-writer-type NO_HAPLOTYPES` is specified (#7141)
  - Suppressed excessive log output related to `JumboAnnotation` warnings in `HaplotypeCaller` (#7358)
- **DRAGEN-GATK**
  - `CalibrateDragstrModel` : fixed a sporadic out-of-memory error (#7212)
  - `CalibrateDragstrModel` : fixed an "IllegalArgumentException: Start cannot exceed end" error (#7212)
- **Mutect2**

- Added a training data mode (`--training-data-mode`) to `Mutect2` to prepare for `Mutect3` (#7109)
  - Training data mode collects data on variant- and artifact-supporting read sets for fitting a deep learning filtering model
- Better error bars for samples with small contamination in `CalculateContamination` (#7003)

- **Funcotator**
  - Greatly improved `Funcotator` performance by optimizing the VCF sanitization code (#7370)
    - In our tests, this change appears to speed up the tool by roughly 2x
  - Updated the Gencode GTF Codec to be more permissive with transcript and gene types (#7166)
    - Now the Gencode GTF Codec no longer restricts `transcriptType` and `geneType` to a limited set of values. These fields are now each stored as a String. This allows for arbitrary values in these fields and will help to future-proof (and species-proof) the GTF parser.
    - Fixes "IndexFeatureFile Error to Run Funcotator with Mouse Ensembl GTF" (#7054)
  - Now can decode codons containing IUPAC bases into amino acids. (#7188)
  - Updated the tool to allow for protein changes with N / IUPAC bases. (#6778)
    - Added the ability to have IUPAC bases in either the ref/alt alleles OR in the reference when calculating the amino acid sequence. In this case, the code will no longer throw a user exception, but will log a warning and will produce ? amino acids in the case that they cannot be decoded from the amino acid table. Currently this will happen any time an N or IUPAC base is in the region to be coded into amino acids.
    - Added AminoAcid.UNDECODABLE as a placeholder for any unknown / undecodable amino acid (such as in the case of an ambiguous IUPAC base).
  - `Funcotator` now checks whether the input has already been annotated, and by default throws an error in that case.
    - We also added a `--reannotate-vcf` override argument to explicitly allow reannotation (#7349)

- **CNV Calling**
  - Enabled multi-sample segmentation in `ModelSegments` (#6499)
  - Removed mapping error rate from estimate of denoised copy ratios output by gCNV, and updated sklearn. (#7261)

- Moved gCNV sample QA check into the Postprocessing task in the WDL (#7150)

- **SV Calling**
  - Added `LocalAssembler`, a new tool that performs local assembly of small regions to discover structural variants (#6989)

- **The Genomics Kernel Library (GKL)**
  - Updated to GKL version 0.8.8, and remove the FPGA PairHMM as an option (#7203)
    - This is a significant update to the GKL that comes with many fixes and improvements:
      - Update ISAL and OTC Zlib libraries to latest version (Q1 2021)
      - Fixed 3 reproducible issues and retested out of 4 more in GKL
      - Updated build for Centos 7 and Current Mac.
      - Ran valgrind on limited C unit tests (passed)
      - Major improvements to input validation
      - Major updates to Error handling and propagation.
      - Added Negative space unit testing coverage
      - Regular Static Code Scanning
      - Good overall quality of life improvement for the software

- **GenomicsDB**
  - Moved to GenomicsDB 1.4.1, and add a toggle between the GCS Connector and native GCS support (#7224)
    - This release allows for the direct use of the native GCS C++ client instead of the GCS Cloud Connector via HDFS. The GCS Cloud Connector can still be used with GenomicsDB via the ``--genomicsdb-use-gcs-hdfs-connector option`
    - Using the native client with GCS allows for GenomicsDB to use the standard paradigms to help with authentication, retries with exponential backoff, configuring credentials, etc., and also helps with performance issues with GCS. See #7070.
  - Allow specifying S3 and Azure blob storage uri's to GenomicsDB in addition to GCS and HDFS (#7271)
  - Fixes related to the GenomicsDB upgrade (#7257)
    - Fixed an issue where the combine operation for certain fields needs to take care to not remap missing fields to NON_REF
    - Fixes "Regression in GenomicsDBImport progress meter" #7222

        - Adds tests for "GenomicsDBImport Creating Workspace Where REF is Inappropriately N?" #7089
    - Improved the error message in `GenomicsDBImport` when failing to open a `FeatureReader` (#7375)

- **Mitochondrial pipeline**
    - Added median coverage metric to the mitochondrial pipeline (#7253)

- **Notable Enhancements**
    - Added a GATK-wide option (`--max-variants-per-shard`) to shard VCFs on output (#6959)
        - Sharded output is often extremely useful for pipelining
    - Added GATK support for block compressed interval (`.bci`) files (#7142)
    - Added an `AlleleDepthPseudoCounts` (DD) genotype annotation. (#7303)
        - Similar to AD, the new annotation (DD) captures the depth of each allele's supporting evidence or reads, however it does so by following a variational Bayes approach looking into the likelihoods rather than applying a fixed threshold. This turns out to be more robust in some instances.
        - To get the new non-standard annotation in `HaplotypeCaller` you need to add `-A AllelePseudoDepth`
    - We now track the source of variants in `MultiVariantWalkers` , which is important for some tools such as `VariantEval` (#7219)

- **Bug Fixes**
    - Fixed key ordering bugs in the implementations of `Histogram.median()` and `CompressedDataList.iterator()` (#7131)
        - These bugs could result in incorrect RankSumTest annotations in some cases
    - Fixed the `DepthPerSampleHC` and `StrandBiasBySample` annotations to not spam the logs with "Annotation will not be calculated" warnings (#7357)
    - `VariantEval` : fixed contig stratification to defer to user-defined intervals (#7238)

- **Miscellaneous Changes**
    - The `ProgressMeter` can now be completely disabled for all tools / traversals by overriding `GATKTool.disableProgressMeter()` (#7354)
    - We now authenticate with Dockerhub in our Travis builds, to help avoid tests failing due to quota issues (#7204) (#7256)
    - Migrated `VariantEval` to be a `MultiVariantWalkerGroupedOnStart` (#6973)

- `VariantEval` : added an argument to specify the `PedigreeValidationType` (#7240)
- Converted `InfoFieldAnnotation` / `GenotypeAnnotation` into interfaces. (#7041)
- Allow `MultiVariantWalkerGroupedOnStart` subclasses to view/set `ignoreIntervalsOutsideStart` (#7301)
- `PedigreeAnnotation` : consolidate code, provide getters, and allow `PedigreeValidationType` to be set (#7277)
- `ASEReadCounter` : added a warning for variants lacking GT fields (#7326)
- Added filters to `dockstore.yml` so that only the master branch and the releases get synced to Dockstore (#7217)
- Fixed a compatibility issue between Java 11 and `log4j2` (#7339)
- We now update the gcloud package signing key at the start of every docker build (#7180)
- Updated our Artifactory key (#7208)
- Disabled some Spark dataproc tests because of dependency issues. (#7170)
- Removed some embedded licenses from scripts (#7340)

- **Documentation**
  - Variant annotation documentation: removed broken links to related annotations from the tool docs (#7307)
  - Updated the link to an article on Jexl expressions (#7317)
  - Fixed several broken links in docs for the CNV tools (#7309)
  - Fixed broken links in the docs for `Funco...

Read more

▶ **Assets** 3

Feb 19, 2021

droazen 4.2.0.0 1f7fade Compare

4.2.0.0

Download release: gatk-4.2.0.0.zip
Docker image: https://hub.docker.com/r/broadinstitute/gatk/

Highlights of the 4.2.0.0 release:

- We've worked closely with Illumina to port a number of significant innovations for germline short variant calling from their DRAGEN pipeline to GATK. These improvements will form the basis of the upcoming open-source implementation of the DRAGEN pipeline which we're calling DRAGEN-GATK
- A number of other fixes and improvements to `HaplotypeCaller` to improve the phasing of variant calls and to fix edge cases with indels and spanning deletions
- A new pipeline for gCNV exome joint calling

Full list of changes:

- **DRAGEN-GATK** (#6634) (#7063)
  - With this release we've worked closely with Illumina to make improvements to the GATK `HaplotypeCaller` to allow it to output germline short variant calls that are functionally equivalent to the calls made by their DRAGEN 3.4.12 pipeline. See our blog post on DRAGEN-GATK for more details on these improvements. A full `DRAGEN-GATK` pipeline that leverages these new features will be released in the near future as a WDL workflow script in the WARP repo on GitHub as well as a featured workspace in Terra.
  - Below is a summary of the improvements we've ported from DRAGEN in this release. We recommend that most users wait until the complete `DRAGEN-GATK` pipeline is released as a WDL workflow before evaluating these features, though advanced users comfortable with building their own pipelines are welcome to try them out now:
    - **DragSTR**: a port of DRAGEN's model for STRs (Short Tandem Repeats) that adjusts HMM indel priors based on empirical reference contexts for better indel calling.
      - Using DragSTR involves running two new tools prior to the `HaplotypeCaller`:
        - `ComposeSTRTableFile`: scans a reference for STR sites and outputs a table file with a subsample of the available STR sites across the genome.
        - `CalibrateDragstrModel`: given the STR table for a reference produced by `ComposeSTRTableFile` and the reads for a specific

sample, generates a model for potential sequencing errors for STR sites of various sizes for that sample.
    - After running these tools, you then run `HaplotypeCaller` with the **`--dragstr-params-path`** argument to pass it the DragSTR model generated by `CalibrateDragstrModel`.
  - **BQD (Base Quality Dropout)** and **FRD (Foreign Read Detection)**: two new genotyper error models ported from DRAGEN
    - The `Base Quality Dropout (BQD)` model penalizes variants with low average base quality scores and high average sequencing cycle counts among genotyped reads and reads that were otherwise excluded from the genotyper to model read-context dependent sequencing errors.
    - The `Foreign Read Detection (FRD)` model uses an adjusted mapping quality score as well as read strandedness information to penalize reads that are likely to have originated from somewhere else on the genome or from contamination.
    - To activate the BQD and FRD models, run `HaplotypeCaller` with the **`--dragen-mode`** argument.
  - Added a new variant QUAL score model that reports the variant QUAL score as the posterior of the reference genotype based on the sample-dependent DRAGEN STR and flat SNP priors.

- **HaplotypeCaller**
  - We now add physical phasing information (PGT/PID/PS attributes) to genotypes with spanning deletion alleles (#6937)
  - Fixed two phasing bugs (#7019)
    - Fixed "HaplotypeCaller emitting incorrect phasing when genotyping hom-het-het" (#6463)
    - Fixed "Phased variants do not have the same phase set identifier" (#6845)
  - Fixed quality score calculation for sites with spanning deletions (#6859)
    - This fixes a bug in the AlleleFrequencyCalculator that was causing quality to be overestimated for sites with * alleles representing spanning deletions.
  - Added the ability for indels to be recovered from dangling heads in the assembly graph, and a new `--num-matching-bases-in-dangling-end-to-recover` argument for filtering dangling ends (#6113) (#7086)
  - Improved handling of indels/spanning deletions in the cigar base quality adjustment code. (#6886)
    - This aims to better handle the edge cases that come up when mates have mismatching numbers of bases at the start or end of the reads relative to each-

other.
  - Fixed a bug where overlapping reads in subsequent assembly regions could have invalid base qualities (#6943)
  - Convert non-ACGT IUPAC bases to N in HaplotypeCaller prior to assembly to prevent a crash (#6868)
  - Renamed the `--mapping-quality-threshold` argument to `--mapping-quality-threshold-for-genotyping` , and updated its documentation to be less confusing (#7036)
  - Added an option for `HaplotypeCaller` and `Mutect2` to produce a bamout without artificial haplotypes (#6991)
  - Updated the `--debug-graph-transformations` argument to emit the assembly graph both before and after chain pruning (#7049)

- **Mutect2**
  - Fixed the `--dont-use-soft-clipped-bases` argument in `Mutect2` to actually work as intended (#6823)
    - Due to a bug, this option did nothing because a copy of the original reads was modified. By deleting the unnecessary mapping quality filtering (this is totally redundant with the M2 read filter), we finalize (and thereby discard soft clips if requested) an assembly region made from the original reads, not a copy.
  - Fixed a bug in the `Mutect2` engine active region code that could affect the ability to call tumor alts when the normal has a different alt at the same site (#6908)
  - Removed an obsolete cram to bam conversion step in the `Mutect2` WDL (#6970)
  - Updated the `Mutect2` whitepaper in `docs/mutect/mutect.pdf` to accurately reflect current filter names, and updated the section on `FilterAlignmentArtifacts` (#6967)

- **CNV Calling**
  - A new pipeline for gCNV exome joint calling (#6554)
    - Added a new tool (`JointGermlineCNVSegmentation`) and associated workflow (`scripts/cnv_wdl/germline/joint_call_exome_cnvs.wdl`) to combine gCNV segments and calls across samples
    - `JointGermlineCNVSegmentation` segments and genotypes CNV calls from the germline CNV pipeline jointly across multiple samples.
    - The workflow in `scripts/cnv_wdl/germline/joint_call_exome_cnvs.wdl` produces a joint, multi-sample genotyped VCF.
    - For whole genomes, we recommend CNVs as part of a full SV callset with https://github.com/broadinstitute/gatk-sv (soon to be added to Terra)

- `GermlineCNVCaller` now restarts inference once with a new random seed when inference diverges. Also added a new entry point to PythonScriptExecutor that returnes ProcessOutput. (#6866)
  - This is intended to alleviate transient issues with GermlineCNVCaller inference in which the ELBO converges to a NaN value, by calling the python gCNV code with an updated random seed input.
- `CreateReadCountPanelOfNormals` : fixed a bug in the logic for filtering zero-coverage samples and intervals (#6624)
- `FilterIntervals` : fixed a bug in the tool logic when filtering on annotations and -XL is used to exclude intervals (#7046)

- **SV Calling**
  - `PrintSVEvidence` : a new tool that prints any of the Structural Variation evidence file types: read count (RD), discordant pair (PE), split-read (SR), or B-allele frequency (BAF) (#7026)
    - This tool is used frequently in the GATK-SV pipeline for retrieving subsets of evidence records from a bucket over specific intervals. Evidence file formats comply with the current specifications in the existing GATK-SV pipeline.

- **GenomicsDB**
  - Introduced a new feature for `GenomicsDBImport` that allows merging multiple contigs into fewer GenomicsDB partitions (#6681)
    - Controlled via the new `--merge-contigs-into-num-partitions` argument to `GenomicsDBImport`
    - This should produce a huge performance boost in cases where users have a very large number of contigs. Prior to this change, GenomicsDB would create a separate folder/partition for each contig, which slowed down import to a crawl when there were many contigs.

- **Funcotator**
  - Added sorting by strand order for transcript subcomponents (#7065)
    - This fixes an issue where the coding sequence, protein prediction, and other annotations could be incorrect for the hg19 version of Gencode, due to the individual elements of each transcript appearing in numerical order, rather than the order in which they appear in the transcript at transcription time.
  - Updated the Funcotator tutorial link in the tool documentation. (#6920) (#6925)

- **Mitochondrial pipeline**

- Simplified the max_reads_per_alignment_start argument in mitochondria_m2_wdl/AlignAndCall.wdl (#6904)
- Remove the unused "autosomal_coverage" parameter from the Filter task in mitochondria_m2_wdl/AlignAndCall.wdl (#6888)

- **Notable Enhancements**
  - Add a `-O` option to save the output to a file in the following tools: `FlagStat`, `CountBases`, `CountReads`, `CountVariants`, and `CountBasesInReference` (#7072)
  - `DepthOfCoverage`: added a new gene_statistics output file (#7025)
  - `ReblockGVCF`: allow reblocking with no P...

Read more

▶ **Assets** 3

Oct 9, 2020

droazen 4.1.9.0 9d5727d Compare

4.1.9.0

Download release: gatk-4.1.9.0.zip
Docker image: https://hub.docker.com/r/broadinstitute/gatk/

Highlights of the 4.1.9.0 release:

- A major update to `Funcotator`, bringing in the latest Gencode release, fixing compatibility issues with dbSNP, and more!
- Two new tools, `GeneExpressionEvaluation` and `ReferenceBlockConcordance`
- Significant performance improvements to `DepthOfCoverage` and `SelectVariants`
- Some important bug fixes:
  - Fixed a bug in `HaplotypeCaller` and `Mutect2` where we were losing insertion events that immediately followed a deletion

- A fix for the "CreateSomaticPanelOfNormals output PoN has much less variants in 4.1.8.0 than before" issue reported in #6744
- A fix for a frequently-encountered `NullPointerException` in the `AS_StrandBiasTest` annotation when running `CombineGVCFs` reported in #6766

Full list of changes:

- **New Tools**
  - `GeneExpressionEvaluation` : a tool for evaluating gene expression from RNA-seq reads aligned to whole genome (#6602)
    - This tool counts fragments to evaluate gene expression from RNA-seq reads aligned to the genome. Features to evaluate expression over are defined in an input annotation file in gff3 fomat. Output is a tsv listing sense and antisense expression for all stranded grouping features, and expression (labeled as sense) for all unstranded grouping features.
  - `ReferenceBlockConcordance` : a new tool to evaluate concordance of reference blocks in GVCF files (#6802)
    - This tool compares the reference blocks of two GVCF files against each other and produces three histograms:
      - *Truth block histogram*: Indicates the number of occurrences of reference blocks with a given confidence score and length in the truth GVCF
      - *Eval block histogram*: Indicates the number of occurrences of reference blocks with a given confidence score and length in the eval GVCF
      - *Confidence concordance histogram*: Reflects the confidence scores of bases in reference blocks in the truth and eval VCF, respectively. An entry of 10 at bin "80,90" means that there are 10 bases which simultaneously have a reference confidence of 80 in the truth GVCF and a reference confidence of 90 in the eval GVCF.
- **HaplotypeCaller/Mutect2**
  - Fixed a bug in `HaplotypeCaller` and `Mutect2` where we were losing insertion events that immediately followed a deletion (#6696)
  - Added a workaround for an issue with multiallelics in the `CreateSomaticPanelOfNormals` pipeline (#6871)
    - This fixes the "CreateSomaticPanelOfNormals output PoN has much less variants in 4.1.8.0 than before" issue reported in #6744

- Made improvements to the `Mutect2` active region detection code that resulted in recovering some low-AF calls that we were missing (#6821)
- Made the `HaplotypeCaller` / `Mutect2` adaptive pruner smarter in complex graphs, resulting in modest improvements to indel sensitivity when using the adaptive pruning option (#6520)
- Fixed a bug in variation event detection code that could sometimes lead to mistreating indel assembly windows as SNP assembly windows (#6661)
- Fixed a bug in `FragmentUtils` where insertion quals were used instead of deletion quals when adjusting base qualities for two overlapping reads from the same fragment (#6815)
- Fixed a concurrent modification exception error for local runs of `HaplotypeCallerSpark` (#6741)
- Marked the `--linked-de-bruijn-graph` argument as Advanced rather than Hidden (#6737)
- Made a small tweak to `Mutect2` 's callable sites count (#6791)
- Added a "requester pays" option to `Mutect2` WDL tasks that access bams for use with Google Cloud "requester pays" buckets (#6879)

- **Funcotator**
  - A major set of updates to `Funcotator` (#6660)
    - Updated to the latest Gencode release
    - Fixed the contig naming compatibility issue with dbSNP reported in #6564 ("hg38 dbSNP has incorrect contig names")
    - Now both hg19 and hg38 have the contig names translated to "chr__"
    - Added 'lncRNA' to GeneTranscriptType.
    - Added "TAGENE" gene tag.
    - Added the MANE_SELECT tag to FeatureTag.
    - Added the STOP_CODON_READTHROUGH tag to FeatureTag.
    - Updated the GTF versions that are parseable.
    - Fixed a parsing error with new versions of gencode and the remap positions (for liftover files).
    - Added test for indexing new lifted over gencode GTF.
    - Added Gencode_34 entries to MAF output map.
    - Pointed data source downloader at new data sources URL.
    - Minor updates to workflows to point at new data sources.
    - Updated retrieval scripts for dbSNP and Gencode.
    - Added required field to gencode config file generation.

        - Now gencode retrieval script enforces double hash comments at top of gencode GTF files.
        - Fixed an erroneous trailing tab in MAF file output reported in #6693
    - Added a maximum version number for data sources in `Funcotator` (#6807)
    - Added a "requester pays" option to the `Funcotator` WDL for use with Google Cloud "requester pays" buckets (#6874)
    - `FuncotateSegments` : fixed an issue with the default value of --alias-to-key-mapping being set to an immutable value (#6700)

- **GenomicsDB**
    - Updated to GenomicsDB Version 1.3.2, which brings better propagation of errors messages from the GenomicsDB library (#6852)
        - Using the GATK option GATK_STACKTRACE_ON_USER_EXCEPTION will now also output a limited C/C++ stacktrace

- **CNV Tools**
    - Fixed a bug in the `KernelSegmenter` : the minimal data to calculate the segmentation cost should be `2 * windowSize` , rather than `windowSize` (#6835)
    - Germline CNV WDL improvements for WGS (#6607)
        - Modified gCNV WDLs to improve Cromwell performance when running on a large number of intervals, as in WGS
        - Added optional disabled_read_filters input to CollectCounts
        - Enabled GCS streaming for CollectCounts and CollectAllelicCounts
    - Added a "requester pays" option to the germline and somatic CNV WDLs for use with Google Cloud "requester pays" buckets (#6870)

- **Mitochondrial Pipeline**
    - Fix to correctly handle spaces in sample names in the Mitochondria WDL (#6773)
    - Exposed a `max_reads_per_alignment_start` argument in the Mitochondria WDL (#6739)
    - Updated the `HaploChecker` Dockerfile to reflect the correct haplocheck CLI (#6867)

- **Notable Enhancements**
    - Significantly improved the performance of `DepthOfCoverage` by removing slow string formatting calls (#6740)

        - In a test run with default arguments locally the runtime for a WGS full chr15 drops from ~8.9 minutes to ~4.7 minutes after this patch
    - Significantly improved the performance of `SelectVariants` with large numbers of samples by changing an operation to scale linearly instead of quadratically with the number of samples (#6729)
        - On one example with several thousand samples there was a speed up from ~5 minutes to 0.1 minutes
    - WDL generation: made several improvements to automatic WDL generation, annotated additional tools for WDL generation, and added a section to the README with instructions on generating WDLs for GATK tools (#6800)
    - Added a suite of utility methods for working with Google BigQuery: `BigQueryUtils` (#6759) (#6861)
    - The GATK docker image can now be built with a simple `docker build .` command (no extra arguments needed) (#6764) (#6842) (#6782)
    - Added a Dockstore yml file with workflow descriptions for the WDLs in the GATK repo, to facilitate automatic publication to Dockstore (#6770)

- **Bug Fixes**
    - Fixed a `NullPointerException` in the `AS_StrandBiasTest` annotation reported in #6766 (#6847)
    - Fixed a bug with soft clips in `LeftAlignIndels` (#6792)
    - `VariantRecalibrator` : uniquify annotations to fix the error reported in #2221 (#6723)
    - Fixed an issue where `ContextCovariate` in `BaseRecalibrator` mistakenly assumed that all non-ACGT bases in the read are N (#6625)
    - Fixed a crash in `CountBasesSpark` when using the `-L` option (#6767)

- **Miscellaneous Changes**
    - Significant refactoring of the SV discovery classes (#6652)
    - `FilterVariantTranches` : report more info when the ref alleles don't match (#6723)
    - We now report the target url in exceptions thrown by `HtsgetReader` (#6799)
    - Added more information to error messages in `AssemblyRegion` for contigs not in the reference dictionary (#6781)
    - Improved an error message in `GATKRead.setMatePosition()` (#6779)
    - Updated the Barclay WDL template for compatibility with the Debian distribution (#6841)

- Temporarily disabled `HtsgetReader` tests to work around issues caused by a server-side upgrade. (#6804)
- Re-enabled an `IndexFeatureFile` test for uncompressed BCF. (#6716)

- **Documentation**
  - Marked `LearnReadOrientationModel` as a `DocumentedFeature` (#6726)
  - Added a gentle warning about loss of True Positives with the default `FilterIntervals` params (#6751)
  - Updated the README to mention that the conda environment is not officially supported on macOS at this time. (#6788)
  - Fixed a typo in the example command for `SplitIntervals` (#6869)
  - Fixed a typo in the `--tmp-dir...

Read more

▶ **Assets** 3

Jul 20, 2020

droazen 4.1.8.1 297f24e Compare

4.1.8.1

Download release: gatk-4.1.8.1.zip
Docker image: https://hub.docker.com/r/broadinstitute/gatk/

Highlights of the 4.1.8.1 release:

- This is a minor point release intended primarily to push out a needed enhancement to the `Mutect2` pipeline.
- This release also introduces a new framework for the auto-generation of WDLs for GATK/Picard tools. Over the next several GATK releases, we intend to hook GATK/Picard tools up to the new WDL generator, with the ultimate goal of having WDLs automatically published for all tools with each release.

Full list of changes:

- **Mutect2**
  - We now allow for the passing of additional arguments to `GetPileupSummaries` from the `Mutect2` WDL (#6713)
- **GATK Engine**
  - Added a new framework for the auto-generation of WDLs for GATK/Picard tools (#6504)
    - Over the next several GATK releases, we intend to hook GATK/Picard tools up to the new WDL generator, with the ultimate goal of having WDLs automatically published for all tools with each release
- **Bug Fixes**
  - Fixed an error (reported in #6664) when trying to read `.vcf` / `.tbi` files located in a path that contains spaces in the name (#6702)
- **Miscellaneous Changes**
  - Removed a few GATK classes that are redundant with Picard classes. (#6678)
- **Documentation**
  - Added instructions for running Spark tools in LOCAL mode to the README (#6682)
  - Removed documentation reference to a GATK 3.x annotation that no longer exists (#6679)
- **Dependencies**
  - Updated HTSJDK to `2.23.0` (#6702)

▶ **Assets** 4

Jun 26, 2020

droazen 4.1.8.0 bc0994c Compare

4.1.8.0

Download release: gatk-4.1.8.0.zip
Docker image: https://hub.docker.com/r/broadinstitute/gatk/

Highlights of the 4.1.8.0 release:

- A major new release of `GenomicsDB` (1.3.0), with enhanced support for shared filesystems such as NFS and Lustre, support for MNVs, and better compression leading to a roughly 50% reduction in workspace size in our tests. This also includes a fix for an error in `GenotypeGVCFs` that several users were encountering when reading from GenomicsDB.
- A major overhaul of the `PathSeq` microbial detection pipeline containing many improvements
- Initial/prototype support for reading from HTSGET services in GATK
  - *Over the next several releases, we intend for HTSGET support to propagate to more tools in the GATK*
- Fixes for a couple of frequently-reported errors in `HaplotypeCaller` and `Mutect2` (#6586 and #6516)
- Significant updates to our Python/R library dependencies and Docker image

Full list of changes:

- **New Tools**
  - `HtsgetReader` : an experimental tool to localize files from an HTSGET service (#6611)
    - *Over the next several releases, we intend for HTSGET support to propagate to more tools in the GATK*
  - `ReadAnonymizer` : a tool to anonymize reads with information from the reference (#6653)
    - This tool is useful in the case where you want to use data for analysis, but cannot publish the data without anonymizing the sequence information.
- **HaplotypeCaller/Mutect2**

- Fixed an "evidence provided is not in sample" error in `HaplotypeCaller` when performing contamination downsampling (#6593)
  - This fixes the issue reported in #6586
- Fixed a "String index out of range" error in the `TandemRepeat` annotation with `HaplotypeCaller` and `Mutect2` (#6583)
  - This addresses an edge case reported in #6516 where an alt haplotype starts with an indel, and hence the variant start is one base before the assembly region due to padding a leading matching base
- Better documentation for `FilterAlignmentArtifacts` (#6638)
- Updated the `CreateSomaticPanelOfNormals` documentation (#6584)
- Improved the tests for `NuMTFilterTool` (#6569)

- **PathSeq**
  - Major overhaul of the PathSeq WDLs (#6536)
    - This new PathSeq WDL redesigns the workflow for improved performance in the cloud.
    - Downsampling can be applied to BAMs with high microbial content (ie >10M reads) that normally cause performance issues.
    - Removed microbial fasta input, as only the sequence dictionary is needed.
    - Broke pipeline down to into smaller tasks. This helps reduce costs by a) provisioning fewer resources at the filter and score phases of the pipeline and b) reducing job wall time to minimize the likelihood of VM preemption.
    - Filter-only option, which can be used to cheaply estimate the number of microbial reads in the sample.
    - Metrics are now parsed so they can be fed as output to the Terra data model.
    - CRAM-to-BAM capability
    - Updated WDL readme
    - Deleted unneeded WDL json configuration, as the configuration can be provided in Terra
  - Added an `--ignore-alignment-contigs` argument to `PathSeq` filtering that lets users specify any contigs that should be ignored. (#6537)
    - *This is useful for BAMs aligned to hg38, which contains the Epstein-Barr virus (chrEBV)*

- **GenomicsDB**
  - Upgraded to `GenomicsDB` version 1.3.0 (#6654)
    - Added a new argument `--genomicsdb-shared-posixfs-optimizations` to help with shared POSIX filesystems like NFS and Lustre. This turns on disable file

locking and for GenomicsDB import it minimizes writes to disks. The performance on some of the gatk datasets for the import of about 10 samples went from 23.72m to 6.34m on NFS which was comparable to importing to a local filesystem. Hopefully this helps with Issue #6487 and #6627. Also, fixes Issue #6519.
    - This version of GenomicsDB also uses pre-compression filters for offset and compression files for new workspaces and genomicsdb arrays. The total sizes for a GenomicsDB workspace using the same dataset as above and the 10 samples went from 313MB to 170MB with no change in import and query times. Smaller GenomicsDB arrays also help with performance on distributed and cloud file systems.
    - This version has added support to handle MNVs similar to deletions as described in Issue #6500.
    - There is added support in `GenomicsDBImport` to have multiple contigs in the same GenomicsDB partition/array. This will hopefully help import times in cases where users have many thousands of contigs. Changes are still needed from the GATK side to make use of this support.
    - Logging has been improved somewhat with the native C/C++ code using spdlog and fmt and the Java layer using apache log4j and log4j.properties provided by the application. Also, info messages like No valid combination operation found for INFO field AA - the field will NOT be part of INFO fields in the generated VCF records will only be output once for the operation.
  - Made `VCFCodec` the default for query streams from `GenomicsDB` (#6675)
    - This fixes the frequently-reported `NullPointerException` in `GenotypeGVCFs` when reading from GenomicsDB (see #6667)
    - Added a `--genomicsdb-use-bcf-codec` argument to opt back in to using the BCFCodec, which is faster but prone to the above error on certain datasets

- **CNV Tools**
  - `DetermineGermlineContigPloidy` can now process interval lists with a single contig (#6613)
  - `FilterIntervals` now filters out any singleton intervals (#6559)
  - Fixed an inaccurate error message in `SVDDenoisingUtils` (#6608)

- **Docker/Conda Overhaul** (#5026)
  - Our docker image is now built off of Ubuntu 18.04 instead of 16.04
    - This brings in newer versions of several important packages such as `samtools`

- Updated many of the Python libraries installed via our conda environment and included in our Docker image to newer versions, resolving several outstanding issues in the process
- R dependencies are now installed via conda in our Docker build instead of the now-removed `install_R_packages.R` script
  - Due to this change, we recommend that tools that use R packages (e.g., to create plots) should now be run using the GATK docker image or the conda environment.
- *NOTE:* significant updates and changes to the Ubuntu version, native packages, and R/python packages may result in corresponding numerical changes in results.

- **Mitochondrial Pipeline**
  - Minor updates to the mitochondrial pipeline WDLs (#6597)

- **Notable Enhancements**
  - `RevertSamSpark` now supports CRAMs (#6641)
  - Fixed a `VariantAnnotator` performance issue that could cause the tool to run very slowly on certain inputs (#6672)
  - More flexible matching of dbSNP variants during variant annotation (#6626)
    - Add all dbsnp id's which match a particular variant to the variant's id, instead of just the first one found in the dbsnp vcf.
    - Be less brittle to variant normalization issues, and match differing variant representations of the same underlying variant. This is implemented by splitting and trimming multiallelics before checking for a match, which I suspect are the predominant cause of these types of matching failures.
  - Added a `--min-num-bases-for-segment-funcotation` argument to `FuncotateSegments` (#6577)
    - This will allow for segments of length less than 150 bases to be annotated if given at run time (defaults to 150 bases to preserve the previous behavior).
  - `SplitIntervals` can now handle more than 10,000 shards (#6587)

- **Bug Fixes**
  - Fixed interval summary files being empty in `DepthOfCoverage` (#6609)
  - Fixed a crash in the BQSR R script with newer versions of R (#6677)
  - Fix crash when reporting error when trying to build GATK with a JRE (#6676)
  - Fixed an issue where `ReadsSourceSpark.getHeader()` wasn't propagating the reference at all when a CRAM file input resides on GCS, so it always resulted in a "no reference was provided" error, even when a reference was provided. (#6517)

- Fixed an issue where `ReadsSourceSpark.checkCramReference()` always tried to create a Hadoop Path object for the reference no matter what file system it lives on, which fails when using a reference on GCS. (#6517)
- Fixed an issue where the tab completion integration tests weren't emitting any output (#6647)

- **Miscellaneous Changes**
  - Created a new `ReadsDataSource` interface (#6633)
  - Migrated read arguments and downstream code to `GATKPath` (#6561)
  - Renamed `GATKPathSpecifier` to `GATKPath`. (#6632)
  - Add a read/write roundtrip Spark integration test for a CRAM and reference on HDFS. (#6618)
  - Deleted redundant methods in `SVCigarUtils`, and rewrote and moved the rest to `CigarUtils` (#6481)
  - Re-enabled tests for HTSGET now that the reference server is back to a stable version (#6668)
  - Disabled `SortSamSparkIntegrationTest.testSortBAMsSharded()` (#6635)
  - Fixed a typo in a `SortSamSpark` log message. (#6636)
  - Removed incorrect logger from `DepthOfCoverage`. (#6622)

- **Documentation**
  - Fixed annotation equation rendering in the tool docs. (#6606)
  - Adding a note as to how to filter on MappingQuality in `DepthOfCoverage` (#6619)
  - Clarified the doc...

Read more

▶ **Assets** 3

Apr 23, 2020

droazen 4.1.7.0 4ec2a4b Compare

4.1.7.0

Download release: gatk-4.1.7.0.zip
Docker image: https://hub.docker.com/r/broadinstitute/gatk/

Highlights of the 4.1.7.0 release:

- Added allele-specific filtering to the mitochondrial pipeline.
  - *Allele-specific filtering is important for mitochondrial calling because there are many more multi-allelic sites than in the germline autosome.*
- A fix for the frequently-encountered "Smith-Waterman alignment failure" error in `HaplotypeCaller` and `Mutect2`
- Initial support for http(s) paths for BAM inputs, including signed urls
- A new tool, `DownsampleByDuplicateSet`, to randomly sample a fraction of duplicate sets from an input bam sorted by UMI

Full list of changes:

- **New Tools**
  - `DownsampleByDuplicateSet`: a new tool to randomly sample a fraction of an input bam sorted by UMI. (#6512)
    - Given a bam grouped by unique molecular identifier (UMI), this tool drops a specified fraction of duplicate sets and returns a new bam.
    - A duplicate set refers to a group of reads whose fragments start and end at the same genomic coordinate *and* share the same UMI.
    - The input bam must first be sorted by UMI using FGBio GroupReadsByUmi.
    - Use this tool to create, for instance, an insilico mixture of duplex-sequenced samples to simulate tumor subclones.
- **HaplotypeCaller/Mutect2**
  - Fixed a regression in `HaplotypeCaller` and `Mutect2` where alt haplotypes with a deletion at the end of the padded region caused exceptions (#6544)
    - This bug produced error messages like the following: "*Smith-Waterman alignment failure. Cigar = 275M with reference length 275 but expecting reference length of 303*"
  - Fixed an `ArrayIndexOutOfBoundsException` in `GenotypeUtils.computeDiploidGenotypeCounts()` caused by mistakenly assuming ploidy two for no-calls (#6563)

- Added more control over scattering in the `Mutect2` PON WDL to allow arbitrarily fine scattering, reducing the memory required for downstream runs of `GenomicsDBImport` (#6527)
- Invert `--correct-overlapping-quality` argument in `HaplotypeCaller` to `--do-not-correct-overlapping-quality` (#6528)

- **Mitochondrial Pipeline**
  - Added allele-specific filtering to the mitochondrial pipeline (#6399)
    - Allele-specific filtering is important for mitochondria because there are many more multi-allelic sites than in the germline autosome and therefore, downstream tools have access to more of the good allele data.
    - These Mutect2 filters used in the MT pipeline are now allele-specific: `weak_evidence`, `base_qual`, `map_qual`, `duplicate`, `strand_bias`, `strand_artifact`, `position`, `contamination`, and `low_allele_frac`.
    - They are added to the `AS_FilterStatus` annotation in the INFO field.
    - The `numt_chimera` and `numt_novel` filters have been replaced by the `possible_numt` filter.
    - Two new filtering tools have been added: `NuMTFilterTool` for the `possible_numt` filter and `MTLowHeteroplasmyFilterTool` for the `mt_many_low_hets` filter, both of which are allele-specific.
    - The `--split-multi-allelics` option of the `LeftAlignAndTrimVariants` tool now splits the annotations in the FORMAT and INFO fields that are of type A and R (allele-specific, and allele-specific with reference).
    - The `VariantFiltration` tool now has an `--apply-allele-specific-filters` option that will apply masks at the allele level. Before this addition, sites that should not be masked, but had deletions that spanned a masked site would have been masked. Now, if this option is specified, only the alleles spanning the masked site will be masked.

- **GATK Engine**
  - Added initial support for http(s) paths for BAM inputs, including signed urls (#6526)

- **Miscellaneous Changes**
  - Exposed maximum copy ratio and point size for CNV plotting tools (#6482)
  - Decreased an epsilon value in `VariantRecalibrator` so that our production exome joint genotyping tests pass (#6534)
  - Migrated reference arguments and downstream code to `GATKPathSpecifier` (#6524)

- Removed obsolete `isCompatibleWithSparkBroadcast()` method. (#6523)

- **Documentation**
  - Cleaned up the handling of some missing values in auto-generated GATK tool documentation (#6565)
    - Now docs won't include null, "", or [] in the default value list.
  - Added a README for the CNN variant scoring workflow, and added an input JSON for `Mutect2` workflow files located in GCS buckets (#6542)
  - Fixed a typo in a ploidy prior example in the docs for `DetermineGermlineContigPloidy` (#6531)

▶ **Assets** 3

Mar 25, 2020

droazen 4.1.6.0 bdb2e15 Compare

4.1.6.0

Download release: gatk-4.1.6.0.zip
Docker image: https://hub.docker.com/r/broadinstitute/gatk/

Highlights of the 4.1.6.0 release:

- `Funcotator` now supports ENSEMBL GTF files (and non-human species)
- A beta port of the GATK3 tool `DepthOfCoverage`, a tool to assess sequence coverage by a wide array of metrics, partitioned by sample, read group, library, or gene (#5913)
- Several important bug fixes and enhancements to `HaplotypeCaller` and `Mutect2`, *including*:
  - A fix for an often-reported issue where `HaplotypeCaller` could produce reads starting with deletions during the realignment step and error out.
  - A fix for another often-reported issue where `Mutect2` could emit MNPs despite `--max-mnp-distance` being 0, causing downstream errors in `GenomicsDB` about MNPs

not being supported.

Full list of changes:

- **New Tools**
  - A beta port of the GATK3 tool `DepthOfCoverage`, a tool to assess sequence coverage by a wide array of metrics, partitioned by sample, read group, library, or gene (#5913)
    - This port fixes several bugs and changes some behavior present in the GATK3 version:
      - Fixed a longstanding bug in GATK3 DepthOfCoverage where using multiple partition types results in column header and body lines having mismatching ordering causing incorrect output.
      - The old version used to merge adjacent and overlapping intervals when generating interval summary files. This is no longer the case as in GATK4 adjacent and overlapping intervals are tabulated as separate lines in the output (This also applies to gene lists which would previously have been merged as well).
      - Changed the behavior of gene list coverage to no longer count introns when generating interval summaries for gene lists.
      - Added support for RefSeqGeneList files as optional gene list input.
- **HaplotypeCaller**
  - Fixed a bug where single-base intervals led to no calls (#6507)
    - This fixes the issue reported in #6495 "HaplotypeCaller doesn't detect alternate alleles with 1 bp intervals"
  - Clean leading deletions from reads realigned to best haplotypes (#6498)
    - This fixes the issue reported in #6490 "HaplotypeCaller might be producing bogus reads with deletions at their alignment start during realignment to best haplotype step"
  - Fixed an edge case when haplotypes have leading insertion after trimming (#6518)
- **Mutect2**
  - `Mutect2` can now filter MNVs with orientation bias (#6486)
  - Added an experimental pileup-based read error corrector, which in our evaluations reduces false positives and improves speed at no cost to sensitivity (#6470)
  - Switched CigarBuilder's order for adjacent indels to be deletion first (#6510)

        - Fixes #6473 "Mutect2 (GATK 4.1.5.0) emitting MNPs despite max-mnp-distance 0"
        - This also resolves downstream errors in `GenomicsDB` about not supporting MNPs
    - Fixed several bugs involving `getReadCoordinateForReferenceCoordinate()` (#6485)
        - Fixes #6342 "Mutect2 occasionally writes nonsense / invalid values for MPOS info tag"
        - Fixes #6314 "GATK4.1.3.0 Mutect2 enable-all-annotations option error"
        - Fixes #6294 "ReadPosRankSumTest with leading insertions"
        - Fixes #5492 "ReadPosRankSumTest doesn't work for two deletions with one base in between"

- **Funcotator**
    - `Funcotator` now supports ENSEMBL GTF files (and non-human species) (#6477) (#6492)
        - Users can now create datasources for any species for which ENSEMBL has an annotated GTF file and the corresponding coding region FASTA file
        - When creating new data sources, the user must still use `gencode` as the parent folder for the GTF data source subfolders. For example, for E. coli MG1655:
            - DATASOURCES
                - gencode
                - ASM584v2
                - Escherichia_coli_str_k_12_substr_mg1655.ASM584v2.44.gtf
                - Escherichia_coli_str_k_12_substr_mg1655.ASM584v2.cds.all.fa
                - ...
                - gencode.config
        - For more information on creating data sources see the Funcotator tutorial on the GATK Forums.
        - An example datasource for E. coli MG1655 can be found in the large test files for Funcotator
        - For ENSEMBL datasources for vertebrates: ftp://ftp.ensembl.org/pub/
        - For ENSEMBL datasources for other species: ftp://ftp.ensemblgenomes.org/pub/

- **CNV Calling**
    - Upgrade CNV WDLs to 1.0 spec (#6506)

- Fixed an off-by-one segmentation argument in `ModelSegments`. (#6497)

- **Miscellaneous Changes**
  - Simplified cigar and clipping code; added tests and fixed a few bugs including #6130 (#6403)
  - Refactored and enhanced ArgumentsBuilder (#6474)
  - Allow all GATKSparkTools to set the SBI index granularity (#6458)
  - Delete NioBam and related classes (#6479)
  - Clean up old interval code (#6465)
  - Remove duplicate copy of the NIO prefetching code (#6464)
  - Fix ignored test in GATKReadAdaptersUnitTest (#6471)
  - Fix alternate spellings of De Bruijn in the codebase (#6472)

- **Documentation**
  - Fix a broken set of javadoc references in FeatureDataSource (#6478)

▶ **Assets** 3

EXHIBIT 1069.66

broadinstitute / gatk Public

Code Issues 1.3k Pull requests 140 Discussions Actions Projects

Releases Tags Find a release

Feb 28, 2020

droazen 4.1.5.0 7f9b849 Compare

4.1.5.0

Download release: gatk-4.1.5.0.zip
Docker image: https://hub.docker.com/r/broadinstitute/gatk/

Highlights of the 4.1.5.0 release:

- A new, improved version of the `--linked-de-bruijn-graph` mode for `HaplotypeCaller` and `Mutect2` that has better sensitivity compared to the previous linked DeBruijn graph implementation (#6394)
- A new version of `GenomicsDB` that fixes many frequently-reported issues
- `LeftAlignIndels` now works for multiple indels
- `VariantAnnotator` and `Concordance` are now out of beta
- A significant number of bug fixes to major tools like `GenotypeGVCFs` and `SelectVariants`

Full list of changes:

- **HaplotypeCaller**
  - New, improved version of the `--linked-de-bruijn-graph` mode for `HaplotypeCaller` and `Mutect2` that has better sensitivity compared to the previous linked DeBruijn graph implementation (#6394)

        - Running `HaplotypeCaller` in this mode will reduce the number of erroneous haplotypes discovered which can improve genotyping, phasing, and runtime.
        - Changed the haplotype recovery step to check that it covers all paths through the graph even if there are poorly supported paths in the JunctionTrees. Added the argument `--disable-artificial-haplotype-recovery` to disable this behavior.
        - Added the ability to expand graph kmer size after haplotype recovery in the event that there was a failure due to overcomplicated assembly graphs.
        - Added code to squeeze extra sensitivity out of the junction trees by tolerating SNP errors when threading the junction trees themselves
    - Realigning to best haplotype handles indels better (#6461)
    - Fixed issue #5434 on inconsistent selection of reads for the PL, AD, and DP calculations. (#6055)
    - Fixed bug where SNP and indel pseudocounts were swapped in the `AlleleFrequencyCalculator` (#6401)
    - The qual used in `HaplotypeCaller`'s `isActive()` method now matches that of `GenotypeGVCFs`. That is, they both now use the new qual. (#6343)
    - Skip non-nucleotide alleles in force-calling mode, fixing bug (#6405)
    - Fixed the hidden/experimental `--error-correct-reads` argument to actually correct the bases and qualities (#6366)
    - Removed the deprecated and obsolete `--use-new-qual-calculator` argument (#6398)
    - Refactored code related to windows and padding for assembly and genotyping, with slight changes to HMM padding for indels (#6358)

- **Mutect2**
    - Improved `SomaticClusteringModel` (#6337)
    - Sped up Mutect2 reference confidence model with fast likelihoods model (#6457)
    - Modified Fragment creation for Mutect2 to not fail for supplementary reads (#6327)
    - Uniqify PG IDs in `FilterAlignmentArtifacts` (#6304)
    - Fixed error in RealignmentEngine due to converting from exclusive to inclusive interval ends (#6404)
    - Added an error message for no callable sites in Mutect2 (#6445)
    - Changed filter reporting in Mutect2 (#6288)
    - Fixed force-calling mode in M2 mito WDL (#6359)
    - Pass the reference to the realignment filter in the Mutect2 WDL (#6360)
    - Deleted the old orientation bias filter (#6408)

- Made callable sites a Long to avoid integer overflow (#6303)

- **GenomicsDB**
  - Move to `GenomicsDB` 1.2.0 (#6305)
    - Fixes an issue with `GenomicsDBImport` erroring out due to duplicate fields in the Info, Format, and/or Filter fields. (#6158)
    - Fixes an issue with `GenomicsDBImport` not completing for mixed ploidy samples (#6275)
    - This version uses a 64-bit htslib to workaround overflow issues when computed annotation sizes exceed the 32-bit integer space

- **Joint Calling**
  - `GenotypeGVCFs` : improved checking for upstream deletions in the `GenotypingEngine` (#6429)
    - Fixes rare cases where `GenotypeGVCFs` could emit a variant with a spanned allele (*), and a genotype that references the spanned allele, but fail to emit the upstream spanning variant.
  - `GenotypeGVCFs` : Don't call the NON_REF allele in genotypes or ADs (#6437)
  - Parse combined `AS_QUALapprox` values from older reblocked GVCFs properly (#6442)
  - Added a force output sites argument to `GenotypeGVCFs` (#6263)
  - Remove extraneous alleles in GenotypeGVCFs force-output mode (#6406)

- **CNV Calling**
  - Copy temporary files early in gcnvkernel to avoid inadvertent temporary directory cleanup. (#6297)
  - Enabled streaming of counts.tsv/counts.tsv.gz files in gCNV CLIs. (#6266)
  - Fixed shard index in PostprocessGermlineCNVCalls log message. (#6313)
  - gCNV vcf cleanup (#6352)
  - Index output VCFs for GCNV postprocessing (#6330)

- **Notable Enhancements**
  - `VariantAnnotator` is now out of beta (#6402)
  - `Concordance` is out of beta (#6397)
  - `LeftAlignIndels` now works for multiple indels (#6427)
  - `FilterVariantTranches` can now handle cases where there are only SNPs or only indels, and not both (#6411)

- Added new read filters for `NotProperlyPaired` and for `MateDistant` (#6295)
- Made the `.git` directory optional during build (#6450)

- **Bug Fixes**
  - Handle zero-weight Gaussians correctly in `VariantRecalibrator` (#6425)
  - Fixed the `--invalidate-previous-filters` argument in `VariantFiltration` to work as intended (ie., roll back all variants to unfiltered status) (#6412)
  - Fixed a bug where `SelectVariants` takes forever on many-allelic somatic samples (#6446)
  - Make sure `SelectVariants` outputs variants in correct order (assuming input vcf is correctly sorted) (#6444)
  - Fixed a NPE crash in `VariantEval` when run with no intervals/reference (#6283)
  - Fixed a NPE crash in `FastaReferenceMaker` (#6435)
  - Fixed an out-of-bounds error in `CountNs` annotation (#6355)
  - Fixed a bug in hardClipCigar function that caused incorrect cigar calculation (#6280)
  - `AnalyzeSaturationMutagenesis` : fixed bug in codon calling for in-frame inserts (#6332)

- **Miscellaneous Changes**
  - Collect split read and paired end evidence files for GATK-SV pipeline (#6356)
  - Add "PASS" filter line for `ApplyVQSR` and `FilterMutectCalls` (#6436)
  - Added engine functionality for accessing the user defined intervals without merging them (#5887)
  - Trim intervals loaded from interval files. (#6375)
  - Propagate read group filters in `ReadGroupBlackListReadFilter` . (#6300)
  - Modified ANDed read filter output message for readability (#6315)
  - Clearly label the number of reads processed in the `BaseRecalibrator` log output (#6447)
  - Clearly label the `CountReads` tool output (#6449)
  - Improved the error messages for missing contigs in the reference (#6469)
  - Avoid a copy and reverse operation in `CigarUtils.isGood()` (#6439)
  - Fixed `GenotypeAlleleCount` 's toString() method (#6376)
  - Minor Funcotator WDL updates. (#6326)
  - Added a `getPairOrientation()` method to `GATKRead` (#6420)
  - Merged `GATKProtectedVariantContextUtils` methods into other classes (#6409)

- Deleted a lot of unused VCF constants (#6361)
- Deleted some unused genotyping code (#6354)
- Fixed incoherent unit test cases in allele subsetting utils (#6448)
- Add Python script executor error message for SIGKILL exit code 137. (#6414)
- Pip install pinned numpy. (#6413)
- Do not install R on travis, and only run the R tests on the Docker. (#6454)
- Fixes for `IndexFeatureFile` error reporting. (#6367)
- Temporarily remove dead Berkeley mirror to unblock builds. (#6422)
- Disable CNNVariantPipelineTest.testTrainingReadModel until failures are resolved. (#6331)
- Delete unused JsonSerializer (#6415)
- Delete empty file SparkToggleCommandLineProgram.java. (#6311)

- **Documentation**
  - Clarify the definition of the `NON_REF` allele (#6431)
  - Clarify behavior of `SplitIntervals` for lists of adjacent intervals (#6423)
  - Update docs to reflect the fact that `TandemRepeat` works with `HaplotypeCaller` (#5943)
  - Update LeftAlignIndels documentation (#6177)
  - Update hyperlink to new GATK forum page in the README (#6381)
  - Add minValue/minRecommended value to ApplyBQSRArgumentCollection (#6438)
  - Small README fixes (#6451)
  - Fix some GATK doc issues (#6318)
  - Update copyright date in LICENSE.TXT (#6383)

- **Dependencies**
  - Updated `HTSJDK` to 2.21.2 (#6462)
  - Updated `Picard` to 2.21.9 (#6462)
  - Updated `Disq` to 0.3.5 (#6323)
  - Updated `GenomicsDB` to 1.2.0 (#6305)
  - Updated `TestNG` to 7.0.0 (#5787)

▶ **Assets** 3

Nov 27, 2019

lbergelson 4.1.4.1 c61cb50 Compare

4.1.4.1

Download release: gatk-4.1.4.1.zip
Docker image: https://hub.docker.com/r/broadinstitute/gatk/

Highlights of the 4.1.4.1 release:

- New **experimental** `HaplotypeCaller` assembly mode which improves phasing, reduces false positives, improves calling at complex sites, and has 15-20% speedup vs the current assembler. It is enabled with option `--linked-de-bruijn-graph`. This mode is still experimental and not recommended for production use yet.
- `IndexFeatureFile` improvements:
  - now cloud enabled
  - changed controversial `F` argument to `I` instead.
- Bug fixes and improvements in `GenomicsDB`, `Mutect2`, variant annotation, and more!

Full list of changes:

- **New Tools**
  - `PrintReadsHeader`: a new tool to print a BAM/SAM/CRAM header to a file (#6153)
- **HaplotypeCaller**
  - Experimental prototype of JunctionTree based haplotype finding. (#6034) #5925
  - Fix a genotyping bug were reference/alt likelihoods were capped differently. (#6196)
- **Mutect2**
  - `Mutect2` now warns but does not fail when three or more reads have the same name. (#6240)
  - Fixed the random seed at the beginning of `FilterMutectCalls` (#6208)
  - `GetSampleName` and `GetPileupSummaries` in the M2 pipeline are no longer beta. (#6215)
  - Increase number of iterations in `CalculateContamination` to 30. (#6282)

- Handled an edge case with high scatter count in M2 WDL. (#6216)
- Use ArgumentsBuilder in M2 tests. (#6219)

- **Joint Calling**
  - Allele-specific VQSR convergence fix. (#6262)
  - Fix to Allele Fraction annotation bug in multisample vcfs. (#6251)
  - Fix RAW_MQ header inconsistencies after reblocking. (#6276)
  - Mark SNP/indel mode argument in `GatherTranches` as required so tranches are named properly. (#6273)

- **CNV Calling**
  - Fixed model parameter assignment typo in gCNV ploidy model (#6285)
  - Added docker option to the gcnv QC tasks. (#6185)
  - Added epsilons to overdispersion in gCNV models to avoid NaNs. (#6245) #4824 #6226 #6227
  - Assert that ELBO did not become NaN during each step of inference of gCNV. (#6186)
  - Added ability to override `THEANO_FLAGS` environment variable in gCNV tools. (#6244) #6235
  - Removed erroneous short argument names in R scripts for CNV plotting. (#6197)

- **GenomicsDB**
  - Allow GATK to configure annotation processing instead of hardcoding values in GenomicsDB GDB-39
  - High ploidy sites with many genotypes no longer causes an overflow error. GDB-54
  - Add missing libcurl in the native GenomicsDB library. #6122 GDB-66
  - No longer crashes when vcfbufferstream from htslib appears to be invalid. GDB-67
  - Propagated native GenomicsDB exceptions as java IOExceptions. GDB-68
  - Fix issue when using vid protobuf interface and there is more than 1 config. GDB-70
  - Cleanup GenomicsDB vid combine protobuf mapping overrides. #6190

- **Miscellaneous Changes**
  - Cloud-enable `IndexFeatureFile` and change input arg name from -F to -I. (#6246) #6161
  - WDL to run `ReadsPipelineSpark` on a multicore machine (#6213)
  - Replace `TwoPassReadWalker` with more general `MultiplePassReadWalker`. (#6154)

- Abolish unfilled likelihoods and revamp `VariantAnnotator`. (#6172)
- Improve exception message in `ValidateVariants`. (#6076)
- Fix Syntax Warning when running GATK with python 3.8 (#6231)

- **Developer / Testing**
  - Report errors logs in github comment (#6247) 6234
  - Add .java-version to gitignore to support jenv users. (#6232)
  - Restart test JVM after every 100 test classes do reduce out of memory failures. (#6093)
  - Running the cloud tests on java 11 on travis. (#6210)

- **Documentation**
  - Clarify definition of PGT in VCF header (#6221)
  - docs for paired reads in Mutect2 somatic genotyping (#6264)
  - Fix some typos in the allele subsetting code. (#6265)

- **Dependencies**
  - Update picard to 2.21.2 (#6253)
  - Update disq to 0.3.4 (#6252)
  - update htsjdk to 2.21.0 (#6250)
  - Update to Genomicsdb 1.1.2.2 (#6206) (#6188)

▶ **Assets** 3

Oct 8, 2019

droazen 4.1.4.0 0a50314 Compare

4.1.4.0

Download release: gatk-4.1.4.0.zip
Docker image: https://hub.docker.com/r/broadinstitute/gatk/

Highlights of the 4.1.4.0 release:

- Major improvements and fixes to `Mutect2`, including more intelligent handling of paired reads during genotyping and better filtering.
- Important bug fixes to `HaplotypeCaller`, the joint calling pipeline, and `Funcotator`
- Beta support for building/testing on Java 11 (#6119) (#6145)
  - *We encourage you to try this out and give us feedback!*

Full list of changes:

- **New Tools**
  - `AlleleFrequencyQC`: a QC tool that uses `VariantEval` to bin variants in 1000 Genomes by allele frequency. For each bin, we compare the expected allele frequency from 1000 Genomes with the observed allele frequency in the input VCF. This was designed with arrays in mind, as a way to discover potential bugs in our pipeline. #6039)
- **Mutect2**
  - `Mutect2` genotyping now forces paired reads to support the same haplotype (#5831)
  - New `FilterAlignmentArtifacts` now realigns a locally-assembled unitig of all variant read pairs (#6143)
  - Fixed a `Mutect2` bug that overfiltered by one variant (#6101)
  - Fixed a small gene panel edge case for `CalculateContamination` (#6137)
  - Fixed a small gene panel edge case in orientation bias filter (#6141)
  - Unified the NIO and non-NIO M2 WDLs (call-caching will now work on Terra) (#6108)
  - Updated `Mutect2` pon WDL to WDL 1.0 (#6187)
  - Removed `Oncotator` from the M2 WDL (`Funcotator` is still there) (#6144)
  - Fixed an issue in the M2 WDL that could cause the Funcotate task to be ignored by tools such as dxWDL (#6077)
  - Some miscellaneous code refactoring/improvements (#6184) (#6136) (#6107) (#6159)
- **HaplotypeCaller**

- `HaplotypeCaller` now force-calls like `Mutect2` : the `-genotyping-mode GENOTYPE_GIVEN_ALLELES` argument is gone (now you only need to specify `--alleles force-calls.vcf`) and alleles are now force-called *in addition* to any other alleles (#6090)
- Renamed `--output-mode EMIT_ALL_SITES` to `--output-mode EMIT_ALL_ACTIVE_SITES` , and clarified the documentation for the argument (#6181)
- Fixed a rare bug in the genotyping engine where it could emit untrimmed alleles for SNP sites (#6044)
- Fixed some sources of non-determinism in the `HaplotypeCaller` that in rare cases could cause the output to vary slightly given the same inputs (#6195) (#6104)
- Deleted the old exact AF calculation model (#6099)

- **Joint Calling**
  - Fixed a regression in GATK 4.1.3.0 that caused us to not emit the `AS_QD` annotation when running a joint calling pipeline with `CombineGVCFs` (`GenomicsDB` was unaffected) (#6168)
  - Fixed allele-specific annotation array length issues when alleles are subset in tools such as `GenotypeGVCFs` (#6079)
  - Changed `AS_RankSum` outputs to "." for missing values rather than "nul" (#6079)

- **Funcotator**
  - Fixed a bug that caused `Funcotator` to outputs fields in wrong order in some cases when writing a VCF (#6178)
    - Specifically, `Funcotator` would output functation fields in the wrong order when there was more than 1 site in a VCF data source with the exact same position and alleles and it matched one of the variants being annotated

- **Mitochondrial pipeline**
  - Renamed the output vcf with the name of the sample and supplied a default value for `autosomal_median_coverage` (meaning you'll now get the `NuMT` filter even if you don't provide the actual autosomal coverage) (#6160)

- **Miscellaneous Changes**
  - Beta support for building/testing on Java 11 (#6119) (#6145)
  - `UpdateVCFSequenceDictionary` now supports replacing an invalid sequence dictionary in a VCF (#6140)
  - `CountFalsePositives` now requires an intervals file (#6120)

- `AnalyzeSaturationMutagenesis` : use supplementary alignments to identify large deletions (#6092)
- `AnalyzeSaturationMutagenesis` : an insert at the start codon is not in the ORF (#6121)
- Added a check for null sequence dictionaries in the dictionary validation code (#6147)
- Update SV Spark pipeline example shell scripts saving results to GCS (#6114)
- Update public key for installing R in docker (#6116)
- Log exceptions during deletion on JVM exit instead of throwing (#6125)
- Don't fail the build if we're in a git worktree folder (#6169)
- Free a bit of memory fir the test suite by disabling mysql and postgress on travis (#6085)
- Delete bogus index files for queryname sorted CRAMs. (#6149)
- Cleanup GenomicsDB debugging test output (#6089)

- **Documentation**
  - Fixed mitochondria mode documentation in `FilterMutectCalls` (#6174)
- **Dependencies**
  - Updated `HTSJDK` to 2.20.3 (#6126)
  - Updated `Picard` to 2.21.1 (#6205)
  - Updated `google-cloud-nio` to 0.107.0 (#6042)
  - Updated `Gradle` to 5.6 (#6106)

▶ **Assets** 3

Aug 9, 2019

droazen 4.1.3.0 afc0e44 Compare

4.1.3.0

Download release: gatk-4.1.3.0.zip
Docker image: https://hub.docker.com/r/broadinstitute/gatk/

Highlights of the 4.1.3.0 release:

- `GnarlyGenotyper` , a new beta joint genotyping tool which, along with `ReblockGVCF` , forms part of a forthcoming more scalable version of our joint genotyping pipeline that we call the "GATK Biggest Practices" pipeline
- `FuncotateSegments` , a new beta companion tool to `Funcotator` that performs functional annotation on a segment file (`.seg`) rather than a VCF
- `GenomicsDBImport` now has the ability to incrementally update an existing GenomicsDB workspace
- Several important bug fixes to `HaplotypeCaller` and `Mutect2`

Compatibility notes:

- `GermlineCNVCaller` models built in cohort mode with previous releases are no longer compatible. Users should rebuild these models with this release before running `GermlineCNVCaller` in case mode. See the **CNV Tools** section below for more details.

Full list of changes:

- **New Tools**
  - **GnarlyGenotyper** (beta tool) (#4947) (#6075)
    - The `GnarlyGenotyper` is designed to perform joint genotyping on cohorts of at least tens of thousands of samples called with `HaplotypeCaller` and post-processed with `ReblockGVCF` to produce a multi-sample callset in a super highly scalable manner.
    - Caveats:
      - `GnarlyGenotyper` is intended to be used with GVCFs for which low quality variants have already been removed, derived from post-processing `HaplotypeCaller` GVCFs with `ReblockGVCF` . See the "Biggest Practices" usage example in the `ReblockGVCF` docs for details.
      - `GnarlyGenotyper` does not subset alternate alleles and can return some highly multi-allelic sites. PLs will not be output for sites with more than 6 alts to save space.
      - `GnarlyGenotyper` assumes all diploid genotypes
    - Annotations:
      - To generate all the annotations necessary for VQSR, input variants to the `GnarlyGenotyper` must include the `QUALapprox` and `VarDP` annotations

along with the latest `RAW_MQandDP` annotation.
        - If allele-specific annotations are present, they will be used appropriately and a new `AS_AltDP` annotation giving the total depth across samples for each alternate allele will be added.
    - A GATK "Biggest Practices" pipeline including the `GnarlyGenotyper` is forthcoming pending some fixes improving on the above caveats.
  - **FuncotateSegments** (beta tool) (#5941)
    - A companion tool to `Funcotator` that performs functional annotation on a segment file (`.seg`) rather than a VCF
    - The Somatic CNV pipeline can optionally run this tool for functional annotation
- **HaplotypeCaller/Mutect2**
  - Fixed a regression in `HaplotypeCaller` / `Mutect2` that caused some variants to be lost at sites with high complexity (#5952)
  - Fixed a GGA (GENOTYPE_GIVEN_ALLELES) mode bug in `HaplotypeCaller` / `Mutect2` where added alleles' cigars could have soft clips (#6047)
    - This bug would manifest as a "Cigar cannot be null" error
  - Fixed a bug where cached indel informativeness values could be incorrectly applied to the wrong sites in `HaplotypeCaller` / `Mutect2` (#5911)
  - Fixed an edge case in `HaplotypeCaller` / `Mutect2` where dangling end merging creates cycles (#5960)
  - Added hidden arguments to the assembly engine to track found haplotype counts and kmers used (#6049)
  - Fixed a bug in `CalculateContamination` when contamination is indistinguishable from zero (#5971)
  - Fixed a bug where normal p value argument in `FilterMutectCalls` was declared static (#5982)
- **CNV Tools**
  - Added `FuncotateSegments` as an option to the Somatic CNV WDL (#5967)
  - Added QC metrics to the Germline CNV workflow (#6017)
  - Enabled GC-bias correction by default in CNV workflows (#5966)
  - Added denoised coverage file concatenation output to gCNV postprocessor (#5823) *Note:* The addition of this feature breaks compatibility with gCNV cohort-mode models built with previous releases.

- Changed cr.igv.seg output of ModelSegments to give log2 Segment_Mean. (#5976)
- Fixed CNV plotting script to allow spaces in input filenames. (#5983)

- **GenomicsDBImport**
  - Added support for making incremental updates to existing workspaces (#5970)
    - This can be done using the new `--genomicsdb-update-workspace-path` argument
  - Fixed a crash in `GenomicsDBImport` on queries at positions inside deletions (#5899)
  - Treat AS_QUALapprox and AS_VarDP strings as array of int vectors (#5933)

- **Mitochondrial Calling Pipeline**
  - Added NIO support and updated to WDL 1.0 (#6074)

- **Spark Tools**
  - Removed the beta label from many simple Spark tools (#5991)
  - Bug fix for reading references from GCS on Spark (#6070)
  - Eliminated an unnecessary sort step in `HaplotypeCallerSpark` (#5909)
  - Fixed `BaseRecalibratorSpark` failure on a cluster due to system classloader issue (#5979)
  - Added a WDL for `ReadsPipelineSpark` (#5904)
  - Added a command-line argument to toggle using NIO on reading for Spark (#6010)
  - Added advanced arguments to `MarkDuplicatesSpark` to allow non-queryname sorted inputs when specifying multiple input bams and to treat unsorted inputs as queryGroup-sorted (#5974)
  - Clarified the behavior of `MarkDuplicatesSpark` when given multiple input bams, and improved the sorting behavior if given a mix of queryname-sorted and query-grouped bams (#5901)
  - Changed `spark.yarn.executor.memoryOverhead` to `spark.executor.memoryOverhead` as promoted by Spark 2.3 (#6032)
  - Handle newly-added arguments in `ApplyBQSRUniqueArgumentCollection` (#5949)

- **Miscellaneous Changes**
  - Added a new `BaseQualityHistogram` variant annotation to generate base quality histograms (#5986)
  - Added a new `SoftClippedReadFilter` that can filter out reads where the ratio of soft-clipped bases to total bases exceeds some given value (#5995)

- Fixed a serious bug in `ValidateVariants` where the tool would silently do no validation in the default case when a DBSNP file was not provided (#5984)
- Fixed a "Record covers a position previously traversed" error in `ValidateVariants` for GVCFS with multiple contigs (#6028)
- The `RMSMappingQuality` annotation now requires the `--allow-old-rms-mapping-quality-annotation-data` argument to run with GVCFs created by older versions of the GATK (#6060)
- Added a simple TSV/CSV/XSV writer with cloud write support as an alternative to TableWriter (#5930)
- `Funcotator` : added Funcotator stand-alone WDL to supported area (#5999)
- Extracted the `GenotypeGVCFs` engine into publicly accessible class/function (#6004)
- Refactored `VariantEval` methods to allow subclasses to override (#5998)
- `AnalyzeSaturationMutagenesis` : arbitrarily choose 1 read for disjoint pairs, dump rejected reads, and various other improvements (#5926) (#6043)
- Normalized some AssemblyRegion args in `HaplotypeCallerSpark` (#5977)
- Don't redundantly delete temporary directories in `RSCriptExecutor` (#5894)
- Treat all source files as UTF-8 for java, javadoc (#5946)
- Updated an out-of-date argument name in an error message for the `CycleCovariate`
- Changed an error about "duplicate feature inputs" to be a UserException (#5951)
- Got rid of `ExpandingArrayList` in favor of `ArrayList` (#6069)
- Disabled Codecov for now on travis due to spurious errors (#6052)
- Lowered the Xms value in the test JVM (#6087)
- Updated the travis installed R version to 3.2.5, matching our base docker image (#6073)
- Fixed an erroneous warning about GCS test configuration (#5987)
- Added a code of conduct (#6036)

- **Documentation**
  - `FilterVariantTranches` documentation fix and improvement (#5837)
  - Updated `FilterMutectCalls` usage examples (#5890)
  - Added `--max-mnp-distance 0` to usage example in `CreateSomaticPanelOfNormals` docs (#5972)
  - Updated the `MarkDuplicatesSpark` documentation to no longer contain a misleading usage example (#5938)

- Added a clarification to the README to warn users to set their Gradle JVM properly in Intellij after setup (#6066)
- Added links to download Java 8 to the README (#6025)
- Remove non-ascii chars from javadoc (#5936)

- **Dependencies**
  - Updated HTSJDK to 2.20.1 (#6083)
  - Updated Picard to 2.20.5 (#6083)
  - Updated Disq to 0.3.3 (#6083)
  - Updated Spark to 2.4.3 (#5990)
  - Updated Gradle to 5.4.1 (#6007)
  - Updated GenomicsDB to 1.1.0.1 (#5970)

▶ **Assets** 3

Apr 23, 2019

droazen 4.1.2.0 fb16ae1 Compare

4.1.2.0

Download release: gatk-4.1.2.0.zip
Docker image: https://hub.docker.com/r/broadinstitute/gatk/

Highlights of the 4.1.2.0 release:

- Two new tools, `MethylationTypeCaller` and `AnalyzeSaturationMutagenesis` (see below for descriptions)
- Significant improvements to `GENOTYPE_GIVEN_ALLELES` mode in `Mutect2` and `HaplotypeCaller`
- Fixed a serious bug in `Funcotator` that could cause END positions to be wrong for some deletions in MAF output
- Significant updates to the mitochondrial calling pipeline

Full list of changes:

- **New Tools**
  - **MethylationTypeCaller** (#5762)
    - Identifies methylated bases from bisulfite sequencing data. Given a bisulfite sequenced, methylation-aware aligned BAM and a reference, it outputs methylation-site coverage to a specified output vcf file.
  - **AnalyzeSaturationMutagenesis** (#5803)(#5883)
    - Processes reads from a saturation mutagenesis experiment, an experiment that systematically perturbs a mini-gene to ascertain which amino-acid variations are tolerable at each codon of the open reading frame. Its main job is to discover variations from wild-type sequence among the reads, and to summarize the variations observed.
- **Mutect2**
  - Made significant improvements to `GENOTYPE_GIVEN_ALLELES` mode in `Mutect2` and `HaplotypeCaller` (#5874). These improvements are described in more detail in #5857
  - `CalculateContamination` now works much better for very small gene panels (#5873)
  - We now correctly handle inputs with 100% contamination in `Mutect2` filtering (#5853)
  - `Mutect2` now uses natural logarithms internally (#5858). This does not change any outputs.
  - Minor update to the `Mutect2` PON WDL (#5859)
- **Funcotator**
  - Fixed a serious bug that could cause END positions to be wrong for some deletions in MAF output (#5876)
  - The tool now throws a user error for an AD field with only 1 value in MAF mode (#5860)
  - Added a new filter to `FilterFuncotations`. For two autosomal recessive genes, MUTYH and ATP7B, homozygous variants and compound heterozygous variants will be tagged and added to the output vcf. (#5843)
- **Mitochondrial Calling Pipeline**

- Updated the pipeline for the new `Mutect2` filtering scheme and pulled filtering after the liftover and recombining of the VCF. (#5847)
- Made the subsetting of the WGS bam fast by using `PrintReads` over just chrM instead of traversing the whole bam for NuMT mates. (#5847)
- Moved polymorphic NuMTs based on autosomal coverage to a filter (it was an annotation before) (#5847)
- Added an option to hard filter by VAF (#5847)
- Bug fix for large input files to the mitochondrial pipeline (we now include the size of the input BAM/CRAM when calculating disk size, when necessary) (#5861)

- **Structural Variation Calling Pipeline**
  - Bug fix to `QNameFinder` to handle reads with negative unclipped starts (#5864)

- **Miscellaneous Changes**
  - Added a `--min-fragment-length` argument to the `FragmentLengthReadFilter` (#5886)
  - Added a `--spark-verbosity` argument to control verbosity of Spark-generated logs (#5825)
  - Added a new `WalkerBase` abstract class to be used for all built-in walkers (#4964)
  - Exposed transient attributes in the `GATKRead` API (#5664)
  - Convert more code to use `GATKPathSpecifier` (#5870) (#5832). This also fixes an `InvalidPathException` on Windows machines.
  - Fixes to the test suite related to the recent introduction of a codec for Picard interval lists (#5879)
  - Eliminated an error message during the Docker build in Travis logs by creating a directory before copying to it. (#5878)

- **Documentation**
  - Updated the `Mutect2` WDL README with `Funcotator` information (#5892)
  - Updated a usage example for `CreateHadoopBamSplittingIndex` (#5898)

▶ **Assets** 3

Mar 28, 2019

droazen 4.1.1.0 ea3032d Compare

4.1.1.0

Highlights of the 4.1.1.0 release:

- A substantial (~33%) speedup to the `HaplotypeCaller` in GVCF mode (`-ERC GVCF`)
- Major updates to `Mutect2` , including completely overhauled filtering and smarter handling of overlapping read pairs.
- A tensorflow update for `CNNScoreVariants` that speeds up the tool by roughly ~2X when using the 2D model.
- Important updates to the mitochondrial calling pipeline, and improved memory usage in the CNV pipeline.
- Important bug fixes to `Funcotator` , `VariantEval` , `GenomicsDBImport` , and other tools, as well as to the `--pedigree` argument for annotations.

Docker image: https://hub.docker.com/r/broadinstitute/gatk/

Full list of changes:

- **HaplotypeCaller**
  - Greatly improved the performance of the ReferenceConfidenceModel using dynamic programming and caching (#5607)
    - This speeds up whole-genome GVCF mode calling (`-ERC GVCF`) by ~33% in our tests!
  - Optimized some additional performance hotspots in the ReferenceConfidenceModel (#5616) (#5469) (#5652)
  - Can now write VCF outputs to Google Cloud Storage (GCS) (#5378)
  - Don't output variants with no ALT allele if the * (spanning deletion) allele gets dropped (#5844)
  - Added a `--force-active` argument that marks all regions as active. Useful for debugging/diagnostics. (#5635)
  - `HaplotypeCallerSpark` : made performance improvements to allow the tool to run on WGS in strict mode (#5721)
  - Fixed rare infinite recursion bug in `KBestHaplotypeFinder` (also affects `Mutect2`) (#5786)

- **Mutect2**
  - Overhaul of `FilterMutectCalls`, which now applies a single threshold to an overall error probability (#5688)
    - `FilterMutectCalls` automatically determines the optimal threshold.
    - The new somatic clustering model learns tumors' allele fraction spectra and overall SNV and indel mutation rates in order to improve filtering.
    - Includes a rewrite of `Mutect2` documentation -- better organization and now includes command line examples in addition to math.
  - `Mutect2` now modifies base and indel qualities of overlapping paired reads to account for PCR error rather than discarding reads (#5794)
    - This especially improves indel sensitivity.
  - Optimized `Mutect2` read orientation filtering by collecting F1R2 counts from within Mutect2 itself, greatly reducing wall-clock and CPU time (#5840)
  - New `Mutect2` panel of normals workflow using `GenomicsDB` for scalability (#5675)
    - Panel of normals removes germline variants in order to contain only technical artifacts, and contains information about artifact prevalence
  - Rewrote `Mutect2` active region likelihood as special case of full somatic likelihoods model, which reduces runtime by 5% (#5814)
  - `Funcotator` updates in `Mutect2` WDL (#5742) (#5735)
  - Prune assemby graph before checking for cycles (#5562)
  - Refactor `Mutect2` inheritance so that it doesn't have inactive arguments (#5758)
  - Added CRAM support to the `Mutect2` WDL (#5668)
  - Split MNPs in `Mutect2` PON WDL, fixing a potential bug (#5706)
  - Handle negative infinity log likelihoods from PairHMM in `Mutect2` (#5736)
  - Fixed overfiltering in `Mutect2` in GGA alleles mode with no reads (#5743)
  - Correct some `Mutect2` VCF header lines (#5792)
  - Handle unmarked duplicates with mate MQ = 0 in `Mutect2` (#5734)
  - Output sample names in `Mutect2` PON header (#5733)
  - Avoid error due to finite precision error in `Mutect2` PON creation (#5797)
  - Update `Mutect2` javadoc to reflect v4.1 changes. (#5769)
  - Renamed the `OxoGReadCounts` annotation to `OrientationBiasReadCounts` (#5840)
- **CNNScoreVariants**
  - We now use the latest Intel-optimized tensorflow (#5725)
    - This speeds up the 2D CNN by roughly 2X in our tests!
  - `FilterVariantTranches` is out of beta (#5628)

- Fixed `CNNScoreVariants` hanging when the conda environment is not set up (#5819)
  - We now make sure that the GATK tool Python package is present before executing streaming Python commands.
- Extensive updates to the CNN WDLs (#5251)

- **Mitochondrial Calling Pipeline**
  - Added an option to recover all dangling branches, on by default for MT calling (#5693)
    - Fixes a large number of missed calls
  - Use adaptive pruning in the mitochondria pipeline (#5669)
  - Changed defaults in mitochondria mode in response to `Mutect2` filtering overhaul (#5827)
  - Allowed the MT pipeline to work on bams with a mix of single and paired-end reads (#5818)
  - Added a hard filter to M2 for polymorphic NuMTs and low VAF sites (#5842)
  - Updated the `haplochecker` version to `0.1.2` to fix a bug with flipping the major and minor hg headers in its output (#5760)
  - Added the rest of the mitochondria joint-calling pipeline (#5673)
    - Merging and genotyping "somatic" GVCFs from `Mutect2`
  - Added a read filter for unmapped reads and their mates (#5826)
  - Refactored the MT WDL to make validations easier (#5708)
  - Updated a variable name in MT WDL to match gatk-workflows version (#5694)

- **GenotypeGVCFs**
  - Added an option to merge intervals for better `GenotypeGVCFs` performance on `GenomicsDB` exome input (#5741)
  - Trim per-allele FORMAT annotations and optionally retain raw AS annotations (#5833)
    - `GenotypeGVCFs` now uses the header info to determine if FORMAT lists need to be subset when alleles are dropped
    - Fixes "F1R2 and F2R2 annotations not updated by GenotypeGvcfs" (#5704)

- **Funcotator**
  - Non-locatable data sources can create funcotations again (#5774)
    - Fixes a bug where `Funcotator` was not adding funcotations from non-locatable data sources

- Fixed handling of symbollic alleles when determining best transcript for `GencodeFuncotation` creation. (#5834)
- `FilterFuncotations` : support for multi-allelic variants (#5588)
- `FilterFuncotations` : support for gnomAD for allele frequency in `ClinVarFilter` and `LofFilter` , with a new argument telling it which dataset of gnomAD or ExAC to use (#5691)
- Added `#` as a character to be sanitized by `VCFOutputRenderer` (#5817)
- Added in Markdown files for Funcotator forum posts (#5630)
- Updated `Funcotator` documentation with a FAQ section to respond to user comments (#5755)

- **CNV Tools**
  - Improved memory usage in gCNV (#5781)
  - Improved memory requirements of `CollectReadCounts` (#5715)
  - Added some fixes for minor CNV issues (#5699)
  - Added io_commons.read_csv to address issues with formatting of sample names in gCNV (#5811)
  - Added gCNV PROBPROG 2018 extended abstract, archived notes on CNV methods, and deleted some legacy documentation (#5732)

- **Miscellaneous Changes**
  - `SelectVariants` can now write VCF outputs to Google Cloud Storage (GCS) (#5378)
  - `VariantEval` bug fix: don't require the output file to already exist (#5681)
  - Fixed the `--pedigree` argument in the `PossibleDeNovo` annotation (#5663)
  - `GenomicsDBImport` : fixed a core dump when querying overlapping deletions (#5799)
  - `GatherPileupSummaries` : a new tool that combines the output of `GetPileupSummaries` from disjoint scatter jobs (#5599)
  - `VariantsToTable` : add splitting for allele-specific annotations and ADs (#5697)
  - `CalculateGenotypePosteriors` : fix reported bug where no-call genotypes with no reads get genotype posterior probabilities and calls (#5667)
  - Added a new argument to Spark tools enabling the user to control whether to sort the reads on output (#4874)
  - `ReadsPipelineSpark` : fixed an "Interval not within the bounds of a contig" error (#5645)
  - `Concordance` : fixed the tool to allow for no variation alleles in the truth data. (#5718)

- `ReblockGVCF` : fix sites with zero AD to actually use SITE-level DP value as intended in (#5835)
- Change `UpdateVCFSequenceDictionary` to use the specified dictionary uniformly (#5093)
- Fixed gatk-nightly Docker builds (https://hub.docker.com/r/broadinstitute/gatk-nightly/) (#5759)
- Print the Picard/HTSJDK versions in addition to the GATK version when running with `--version` (#5757)
- `IndexFeatureFile` : fixed a crash on VCFs with 0 records (#5795)
- `PrintBGZFBlockInformation` : removed the file extension check so that we can accept bams (#5801)
- Added a new read filter: `IntervalOverlapReadFilter` (#5656)
- Add NIO Path support to `TableReader` and `TableWriter` (#5785)
- Replaced `IntervalsSkipList` with `OverlapDetector` (#4154)
- Removed some unused arguments in VCF merging code (#5745)
- Kebab-case some arguments in `LocusWalker` and `LocusWalkerSpark` (#5770)
- Removed an unnecessary IllegalArgumentException in `PairHMM` (#5705)
- Removed accidental uses of log4j v1 (#5682)
- Improvements to Spark evaluation scripts (#5815)
- Extract tests from `PrintReadsIntegrationTest` to share with the Spark version. (#5689)

- **Documentation**
  - Improved the documentation for the `StrandOddsRatio` annotation (#5703)
  - Fixed the descriptions of some `HaplotypeCaller` arguments (#5658)
  - Update `VariantRecalibrator` example code to reflect new tagged argument syntax (#5710)
  - Corrected javadoc for the `InbreedingCoeff` annotation (#5768)
  - `CalculateGenotypePosteriors` : minor updates to javadoc and logger type (#5601)
  - Added and Updated javadoc for `SortSamSpark` and `MarkDuplicatesSpark` (#5672)
  - Added a link to a "GitHub basics for researchers" article at top of the GATK README (#5643)
  - Updated the main GATK README to remove outdated references to the Intel conda environment (#5753)
  - Trimmed overly-long tool...

Read more

▶ **Assets** 3

Jan 29, 2019

droazen 4.1.0.0 335fac0 Compare

4.1.0.0

It's been a year since the GATK `4.0.0.0` release in January 2018, and we decided that it was time to package up the past year's worth of GATK improvements into a new major release, which we're calling version `4.1.0.0` !

To commemorate this milestone, we'll be publishing a series of in-depth technical articles and blog posts covering the major new features in version `4.1.0.0` on the official GATK blog.

Below we've compiled the highlights of the new features added between versions `4.0.0.0` and `4.1.0.0` . If you're interested in seeing *only* the changes between the last release (`4.0.12.0`) and this release (`4.1.0.0`), click here instead.

Official docker image is here: https://hub.docker.com/r/broadinstitute/gatk/

Major changes between versions 4.0.0.0 and 4.1.0.0 (January 2018 to January 2019):

- **Next-Gen VQSR Replacement For Single-Sample**
  - New suite of tools `CNNScoreVariants` , `CNNVariantTrain` , `CNNVariantWriteTensors` , and `FilterVariantTranches`
  - `CNNScoreVariants` is now out of beta and ready for production use
  - Performs variant training and scoring using a convolutional neural network.
  - Single-sample only
  - Produces better results than the legacy `VariantRecalibrator` (VQSR) and comparable or better results to third-party tools like `DeepVariant`

    - Sophisticated 2D model that uses the reads

- **Major HaplotypeCaller Improvements**
    - Now genotypes and outputs spanning deletions
    - Now outputs VCF spec-compliant phased variants
    - Can emit MNPs via a new `--max-mnp-distance` argument
    - Important fix to the reference confidence calculation upstream of indels
    - New `HaplotypeCaller` priors for variants sites and homRef blocks
        - Added new `--population-callset` argument allowing an external panel of variants to be specified to inform the frequency distribution underlying the genotype priors
        - Added new `--num-reference-samples-if-no-call` argument to control whether to infer (and with what effective strength) that only reference alleles were observed at sites not seen in any panel

- **Major Mutect2 Improvements**
    - `Mutect2` is now out of beta
    - Support for multi-sample calling
    - Lots of support for high-depth calling such as cfDNA, UMIs, mitochondria, including a new active region likelihood, probabilistic assembly graph pruning that adjusts to the local depth, a new mitochondria mode, and new filters for blood biopsy and mitochondria
    - Now outputs VCF spec-compliant phased variants
    - Can emit MNPs via a new `--max-mnp-distance` argument
    - Added a genotype given alleles (GGA) mode
    - New STR indel error model that improves sensitivity and precision in STR (short-tandem repeat) contexts
    - Many new/improved filters to reduce false positives (eg., `FilterAlignmentArtifacts`)
    - Mutect2 now automatically recognizes and removes end repair artifacts in regions with inverted tandem repeats. This is extremely important for some FFPE samples.
    - New probabilistic orientation bias tool
    - Got rid of many questionable indels showing up in bamout of Mutect2 and the HaplotypeCaller
    - Big improvements to CalculateContamination, especially when tumor has lots of CNVs
    - NIO support in Mutect2 WDL

- Significant speed improvements
- Improved allele fraction estimation
- Initial GVCF output support

- **Mitochondrial Calling**
  - Added `--mitochondria-mode` to `Mutect2` and `FilterMutectCalls`. This increases sensitivity and only applies filters that are optimized for mitochondria.

- **New allele frequency / qual score model**
  - Is now the default in `HaplotypeCaller` and `GenotypeGVCFs`
  - Optimized for greater speed, should resolve many `GenotypeGVCFs` memory issues
  - Rare numerical finite precision issues in the allele-specific qual have been resolved

- **Major Improvements to the CNV (Copy Number Variation) tools**
  - The CNV tools are now out of beta.
    - This includes the tools: `AnnotateIntervals`, `CallCopyRatioSegments`, `CollectAllelicCounts`, `CollectReadCounts`, `CreateReadCountPanelOfNormals`, `DenoiseReadCounts`, `DetermineGermlineContigPloidy`, `FilterIntervals`, `GermlineCNVCaller`, `ModelSegments`, `PostprocessGermlineCNVCalls`, `PreprocessIntervals`, `PlotDenoisedCopyRatios`, and `PlotModeledSegments`
  - Completed the `GermlineCNVCaller` (gCNV) pipeline and made various performance/runtime improvements to both the methods and WDLs.
  - Major changes include the addition of new tools (`PostprocessGermlineCNVCalls`, `FilterIntervals`, and `CollectReadCounts`, which replaces `CollectFragmentCounts`), as well as improvements to existing tools (notably, `AnnotateIntervals`).
  - Improved support for various formats, namely VCF output in the gCNV pipeline, IGV-compatible .seg output in the `ModelSegments` somatic CNV pipeline, and CRAM support for all CNV WDLs.
  - Developed tools and WDLs for tagging and filtering of germline events in the `ModelSegments` somatic CNV pipeline.

- **Funcotator Official Release**
  - Funcotator is now out of beta
  - Huge number of bug fixes and accuracy improvements. Output for several fields is now more correct than other well-known functional annotation tools.

- Some new features include:
  - MAF output support
  - NIO support for datasources
  - gnomAD support
  - dbsnp support
  - Support for Mitochondrial amino acid sequence/protein change strings
  - 5'/3' flank support
  - Major performance improvements due to added caching
  - Added ALL mode for transcript selection (`--transcript-selection-mode ALL`) which will output full annotation fields for all transcripts
- Created a new `FuncotatorDataSourceDownloader` tool to download data sources
- Added an experimental `FilterFuncotations` tool

- **MarkDuplicatesSpark is now a Validated, Scalable Replacement for MarkDuplicates**
  - MarkDuplicatesSpark is now out of beta
  - Rewritten version of the tool matches Picard `MarkDuplicates` output and has greatly improved performance and scalability
  - Supports multiple BAM inputs
  - Indexes BAM outputs on-the-fly in parallel on a cluster

- **Additional Tools Ported from GATK3**
  - Ported `VariantAnnotator`
  - Ported `VariantEval`
  - Ported `FastaAlternateReferenceMaker` and `FastaReferenceMaker`
  - Ported `LeftAlignAndTrimVariants`
  - Restored `GenotypeGVCFs` `--include-non-variant-sites` argument

- **Major Improvements to the SV (Structural Variation) Tools**
  - Improvements to collection and calling of events based on discordant read pair evidence.
  - A new scaffolding algorithm greatly improves the contiguity of local assemblies, increasing sensitivity.
  - Regions of excessive sequencing depth are excluded from evidence collection and assembly, improving runtime performance.
  - A major overhaul of our algorithm for calling events based on local assemblies improves accuracy and allows for the accurate reporting of small complex SVs.

- A machine learning (xgBoost) based classifier for SV evidence improves runtime and increases accuracy by determining which regions should be fed into the local assembly workflow.

- **Spark Improvements**
  - New Disq Spark library allows faster and more accurate loading of formats like BAM and VCF
  - `HaplotypeCallerSpark` now has a "strict mode" that closely matches the regular `HaplotypeCaller`
  - Created `RevertSamSpark`, a parallelized Spark version of Picard's `RevertSam` tool
  - Migrated most Spark tools that take a reference and/or VCF to use Spark's intrinsic file copying mechanism instead of broadcast to distribute the reference and VCFs to worker nodes -- a big performance win!

- **GenomicsDB Improvements**
  - Allele-specific annotation support
  - Multi-interval support (with some performance caveats)
  - Support for sites-only queries
  - Support for returning the GT field in queries
  - New protobuf-based API to allow configuration without editing JSON files
  - Added in machinery to allow per-annotation combine operations to be specified
  - Allow for hdfs and gcs URI's to be passed to GenomicsDB
  - Migrated from `com.intel.genomicsdb` to `org.genomicsdb`

- **"Goodies" Worth Mentioning**
  - Added fasta.gz support to the `-R/--reference` argument in walker tools
  - `SelectVariants` can now drop specific annotation fields from the output vcf
  - `CalculateGenotypePosteriors` now supports indels
  - New tool `ReblockGVCF` to merge reference blocks in single-sample GVCFs for smaller filesizes
  - Improved MQ calculation accuracy, especially at sites with many uninformative reads; concomitant with new annotation tag and format
  - The `-L` argument now supports GCS (Google Cloud Storage) for interval list files / bed / vcf files in walker tools
  - Added support for "Requester Pays" GCS (Google Cloud Storage) buckets via new `--gcs-project-for-requester-pays` argument
  - Added GCS (Google Cloud Storage) output (-O) support to more tools

- Improved Python integration (eliminated timeouts and reliance on prompt synchronization) means fewer glitches during runs of ML-based tools
- A significantly (~33%) smaller GATK docker image
- Changed argument tagging syntax from "--arg tag:value" to "--arg:tag value"
  - Affects command-line interface for `VariantRecalibrator`, `VariantEval`, `VariantFiltration`, and `VariantAnnotator`

Changes between versions 4.0.12.0 and 4.1.0.0 *only*:

- Many tools are now out of beta and ready for production use!
  - `CNNScor...

Read more

▶ **Assets** 3

Dec 17, 2018

droazen 4.0.12.0 02682d5 Compare

4.0.12.0

Highlights of this release include support for outputting phased variants in `HaplotypeCaller` / `Mutect2`, restoring the `--include-non-variant-sites` argument to `GenotypeGVCFs`, a port of the GATK3 tool `VariantEval`, a new library (Disq, https://github.com/disq-bio/disq) for working with BAM/CRAM/VCF/etc. formats on Spark, and GCS (Google Cloud Storage) support in `Funcotator`.

As usual, a docker image for this release can be downloaded from https://hub.docker.com/r/broadinstitute/gatk/

Full list of changes in this release:

- `HaplotypeCaller` / `Mutect2`
  - Output VCF spec-compliant phased variants in HaplotypeCaller and Mutect2

  - Added an experimental adaptive pruning option for local assembly (#5473)
  - Improved implementation of allele-specific new qual (#5460)
  - Use cigar complexity to break ties in uninformative reads' best haplotypes (#5359)
  - Improved handling of regions that are too short after trimming in HaplotypeCaller and in Mutect2 (Closes issue #5079)
  - Optimization in `CigarUtils` to shortcut to M-only CIGAR when provably optimal (#5466)
  - Changed SUPPORTED_ALLELES_TAG from SA to XA (#5418)

- `HaplotypeCaller`
  - Fixed bug in GGA mode caused by split multallic sites with genotypes (#5365)
  - The debug command line argument is now passed correctly in HaplotypeCaller (fixed issue #4943) (#5455)

- `Mutect2`
  - Big improvements to CalculateContamination's model for determining hom alt sites (#5413)
  - Reduce false negatives from mapping quality filter on long indels in Mutect2 (#5497)
  - Added a mismatch ratio option in realignment filter (#5501)
  - Made Mutect2 read position filter default much less stringent (#5487)
  - Fixed M2 bug for germline resources with AF=. (#5442)
  - Fix read position annotation bug in M2 filter (#5495)
  - Cleaner Mutect2 VCF fields (#5510)
  - Moved PerAlleleAnnotations to the INFO field (#5518)
  - Removed unnecessary inheritance of M2 filtering arguments collection (#5498)

- `GenotypeGVCFs`
  - Restored the --include-non-variant-sites argument from GATK3 to GenotypeGVCFs (#5219)

- Ported the GATK3 tool `VariantEval` to GATK4 (#5043)

- Replaced the Hadoop-BAM library with the newly-developed Disq library (https://github.com/disq-bio/disq) for efficiently working with BAM/CRAM/VCF/etc. formats on Spark (#5138)

  - Improves Spark performance across-the-board, and fixes many edge-case bugs in Hadoop-BAM

- `Funcotator`
  - Added GCS support to Funcotator data sources, so that data sources can now be accessed directly from GCS buckets (#5425)
  - Added support for annotating 5'/3' flanks (#5403)
  - Funcotator now creates default annotations for difficult variants. (#5374)
  - Funcotator now can create annotations for symbollic alleles and masked alleles (#5406)
  - Funcotator now can match between hg19 and b37 data sources. (#5491)
  - Added in regression tests and fixes for correctness of many annotations (#5302)
  - Now DE_NOVO_START_IN_FRAME and DE_NOVO_START_OUT_FRAME are correct. (#5357)
  - Added cDNA Strings for Intronic Variants (#5321)
  - VCF data sources create an ID field for the ID of the variant used for the annotation (#5327)
  - Funcotator now computes MT protein changes. (#5361)
  - Funcotator now correctly populates transcript position. (#5380)
  - Added a script that can create data sources from BED files. (#5438)
  - Updated testing Gencode data sources to fully exercise test data set (#5423)
  - Moved validation test data out of large files area. (#5381)
  - Updated top-level class documentation for Funcotator. (#4655)
  - Added scripts to liftover gnomAD. Also bugfixes for Funcotator NIO. (#5514)

- `HaplotypeCallerSpark`
  - Added a "strict mode" that allows `HaplotypeCallerSpark` to closely match the output of the regular `HaplotypeCaller` (#5416)
  - Now extends AssemblyRegionWalkerSpark (#5386)

- `MarkDuplicatesSpark` : Added a few of the remaining unimplemented useful features from Picard (#5377)

- `CNV workflows`
  - Changed `FilterIntervals` to operate on the intersection of intervals in all inputs. (#5408)
  - Fixed RAM usage parameter error in combine_tracks.wdl (#5358)

  - Various other improvements to combine_tracks.wdl (#5384)
  - Fixed gCNV WDL broken by Cromwell update on FireCloud. (#5407)
  - Replaced bash script in gCNV ScatterIntervals task with updated version of IntervalListTools. (#5414)

- `CNNScoreVariants`
  - Check for and require hardware AVX support (#5291)
- Changed `SelectVariants` so that it can handle multiple rsIDs separated by ';' in a VCF file (#5464)
- Miscellaneous Changes
  - Added `setIsUnplaced()` to the `GATKRead` API to distinguish reads with no mapping information (#5320)
  - Fixed an integer overflow bug in the `RMSMappingQuality` annotation (#5435)
  - Fixed floating-point bug in MannWhitneyU on some JVMs. (#5371)
  - Standardized the output argument for `LeftAlignIndels` (#5474)
  - `SplitIntervals` now produces an `.interval_list` file (#5392)
  - Fixed a bug with GATK_GCS_STAGING in the GATK launcher script #1338 (#5452)
  - Added ExampleReadWalkerWithVariantsSpark.java and tests (#5289)
  - Add description getter and javadoc in GATKReportTable (#5443)
  - Fixed message in GATKAnnotationPluginDescription (#5444)
  - Replaced some uses of PrintWriter (#5461)
  - Refactor GVCFWriter to allow push/pull iteration. (#5311)
  - Add scripts/dataproc-cluster-ui to release bundle. (#5401)
  - Marked `VariantAnnotator` as a `@DocumentedFeature` (#5480)
  - Removed obsolete intel conda environment references. (#5482)
  - Deleted the CountSet class (#5467)
  - Test framework: disabled gcloud login on travis for non-cloud non-wdl tests (#5335)
  - Updated Spark scripts to reflect changes from #5386 and #5127. (#5415)
  - Fixed jexl logging and updated VariantFiltration doc. (#5422)
  - Fixed some dead links in the README (#5405)
- Dependencies
  - Updated htsjdk to 2.18.1 (#5486)
  - Updated Picard to 2.18.16. (#5412)

- Updated Intel-GKL dependency to 8.6 (#5463)

▶ **Assets** 3

Oct 23, 2018

lbergelson 4.0.11.0 5d95d60 Compare

4.0.11.0

A release which includes major improvements to Mitochondrial calling in Mutect2 as well as bug fixes and improvements:

As always a docker is available here: https://hub.docker.com/r/broadinstitute/gatk/

Mutect2 and HaplotypeCaller changes:

- Added `--mitochondria-mode` to `Mutect2` and `FilterMutectCalls`. This increases sensitivity and only applies filters that are optimized for mitochondria. A best practices WDL for calling mitochondrial variants on WGS data will be available in the future. (#5193)
- Strand based annotations will use both reads in an overlapping read pair (#5286)
- Realignment filter annotates the VCF with passing and failing read counts (#5328)
- New filters and annotation to support blood biopsy that count and filter based on N's at variant sites (#5317)
- Fixed bug for M2 GGA alleles with zero coverage (#5303)
- Fixed error in genotype given alleles mode when input alleles have genotypes (#5341) #5336
- Add new annotations to bamout to make understanding calls easier (#5215)
- Fixed a typo.

CNV Pipeline:

- Added FilterIntervals to perform annotation-based and count-based filtering in the gCNV pipeline. (#5307) closes #2992 #4558

Spark:

- Removed WellformedReadFilter from CountReadsSpark (#5329)
- Support fasta.gz in GATKSparkTool (#5290) closes #5258

Other:

- CNN variant update models validate scores cleanup training (#5175)
- combine_tracks.wdl supports GISTIC2 conversion (and bugfix) (#5287) closes #5284 #5283
- handle normal reads in validation sample in BasicSomaticValidator (#5322)

GenomicsDB:

- Allow for hdfs and gcs URI's to be passed to GenomicsDB (#5197)

SelectVariants:

- Enable SelectVariants to drop specific annotation fields from output vcf. (#5254) closes #5235

SplitNCigarReads:

- Added defensive check to OverhangFixingManager splices for non-reference spanning reads (#5298) closes #5293
- Fixed SplitNCigarReads ArrayIndexOutOfBounds error for reads with long deletions (#5285) closes #5230

Testing:

- Added a toggle to update the expected outputs in HaplotypeCallerIntegrationTest (#5324)
- Added a new servicekey.json for travis (#5308) closes #5305
- Added full-sized B37 and HG38 references to our large test data (#5309) closes #5111
- Added in new data sources for funcotator testing. (#5296)

▶ **Assets** 3

Oct 9, 2018

droazen 4.0.10.1 4dd7ba8 Compare

4.0.10.1

This is a small release that improves the calculation of the `MQ` (mapping quality) annotation, which provides an estimate of the overall mapping quality of reads supporting a variant call. It also introduces a number of experimental improvements to the CNV workflows, as well as a bug fix to `LocusWalkerSpark`.

As usual, a docker image for this release can be downloaded from https://hub.docker.com/r/broadinstitute/gatk/

Full list of changes in this release:

- Improve MQ calculation accuracy (#4969)
  - Change raw MQ to a tuple of (sumSquaredMQs, totalDepth) for better accuracy where there are lots of uninformative reads or called single-sample variants with homRef genotypes.
  - Note that incorporating this change into a pipeline will require a concomitant update to this version for GenomicsDBImport and GenotypeGVCFs.
- Updated `SimpleGermlineTagger` and somatic CNV experimental post-processing workflow with several experimental changes that improve precision results, and expand possible evaluations, of GATK CNV (#5252)
  - New script `combine_tracks.wdl` for post-processing somatic CNV calls. This wdl will perform two operations:
    - Increases precision by removing:
      - germline segments. As a result, the WDL requires the matched normal segments.
      - Areas of common germline activity or error from other cancer studies.
    - Converts the tumor model seg file to the same format as AllelicCapSeg, which can be read by ABSOLUTE. This is currently done inline in the WDL.
      - This is not a trivial conversion, since each segment must be called whether it is balanced or not (MAF =? 0.5). The current algorithm relies on hard filtering and may need updating pending evaluation.
      - For more information about AllelicCapSeg and ABSOLUTE, see:

        - Carter et al. *Absolute quantification of somatic DNA alterations in human cancer*, Nat Biotechnol. 2012 May; 30(5): 413–421
        - https://software.broadinstitute.org/cancer/cga/absolute
        - Brastianos, P.K., Carter S.L., et al. *Genomic Characterization of Brain Metastases Reveals Branched Evolution and Potential Therapeutic Targets* (2015) Cancer Discovery PMID:26410082
    - Changes to GATK tools to support the above:
        - `SimpleGermlineTagger` now uses reciprocal overlap to in addition to breakpoint matching when determining a possible germline event. This greatly improved results in areas near centromeres.
        - Added tool `MergeAnnotatedRegionsByAnnotation`. This simple tool will merge genomic regions (specified in a tsv) when given annotations (columns) contain exact values in neighboring segments and the segments are within a specified maximum genomic distance.
    - New scripts `multi_combine_tracks.wdl` and `aggregate_combine_tracks.wdl` which run `combine_tracks.wdl` on multiple pairs and combine the results into one seg file for easy consumption by IGV.
- `LocusWalkerSpark`: fix issue where intervals with no reads were being dropped (#5222)
    - This fixes the bug reported in #3823
- Added `SparkTestUtils.roundTripThroughJavaSerialization()` method for better serialization testing on Spark (#5257)
- Build system: set the same compiler flags for all gradle JavaCompile tasks (#5256)

▶ **Assets** 3

EXHIBIT 1069.102

Oct 3, 2018

droazen 4.0.10.0 7368c62 Compare

4.0.10.0

Highlights of this release include a new tool `ReblockGVCF`, a bug fix for a crash in `Mutect2`, and a more efficient distribution mechanism for the reference and VCFs in Spark tools.

As usual, a docker image for this release can be downloaded from https://hub.docker.com/r/broadinstitute/gatk/

Full list of changes in this release:

- Added a new experimental tool `ReblockGVCF` (#4940)
  - A tool to merge reference blocks in single-sample GVCFs for smaller filesizes
- `Mutect2`:
  - Fixed a bug in the `PalindromeArtifactClipReadTransformer` (#5241)
    - This filter would crash with an out-of-bounds error for fragment lengths and/or mate start positions that went off the end of a contig.
  - Changed the way the log10AlleleFractions are calculated in `SomaticLikelihoodsEngine`: now we use the mean of the posterior of the allele fractions. (#5231)
  - Reword comments in Mutect2 WDL to not refer to the old orientation bias filter as deprecated. (#5196)
  - Cited CGA in Mutect docs (#5228)

- `HaplotypeCaller` : Allow MNP calling in GVCF mode with stern warnings about not trying joint-genotyping from the resulting GVCFs. (#5182)
  - `HaplotypeCaller` will now allow you to output MNPs in GVCF mode with a warning, however since joint genotyping of MNPs is unsupported, `CombineGVCFs` and `GenomicsDBImport` will now refuse to process GVCFs containing MNPs.
- `GATK Spark tools` :
  - Migrated most Spark tools that take a reference and/or VCF to use Spark's intrinsic file copying mechanism instead of broadcast to distribute the reference and VCFs to worker nodes (#5127) (#5221)
    - This improves the performance of Spark tools that take a reference and/or VCF as side inputs, as the new distribution mechanism doesn't load the entire contents of the files into memory like broadcast did.
    - As a side effect of this change, support for 2bit references has been removed from tools that were migrated to the new distribution mechanism (in particular, `BaseRecalibratorSpark` and `HaplotypeCallerSpark`).
    - The CNV Spark tools have not yet been migrated, and still support 2bit references for now.
  - Bug fix: ensure that intervals with no reads are not dropped by the `SparkSharder` (#5248)
- `Funcotator` :
  - Added command line exclusion lists, so that users can prune fields from the output. (#5226)
  - Added Funcotator excluded fields option explicitly to the M2 WDLs. (#5242)
- Fix a multithreaded race condition in `GenotypeLikelihoodCalculators` by synchronizing updates of shared genotype likelihood tables. (#5071)
  - This bug affected `HaplotypeCallerSpark` , but not the regular `HaplotypeCaller`
- `GenomicsDB` : added in machinery to allow per-annotation combine operations to be specified (#4993)
- `GATK Engine` : Hooked up `CountingVariantFilter` to `VariantWalkers` (#4954)
- `StreamingPythonScriptExecutor` : added a new message to the `StreamingProcessController` ack FIFO protocol to allow additional message detail to be passed as part of a negative ack. (#5170)

    - This improves exception message propagation for fatal errors when running Python tools.

- `gCNV WDLs` :
    - Tar calls from all samples. (#5225)
        - This fixes an issue where the gCNV WGS cohort germline WDL was outputting vcf files with names that do not correspond to the actual samples inside the files.
    - Added multi-sample functionality to gCNV case mode WDL, and added a wrapper for gCNV case mode WDL to help optimize cloud computation cost. Also optimized how data is sent to postprocessing task in gCNV WDLs. (#5176)
- `gCNV kernel` : Enforced ViterbiSegmentationEngine to analyze single samples only (#5176)
- Added a `dataproc-cluster-ui` script to easily open the Spark UI on dataproc clusters (#5188)
- Fixed pom issues that prevented publishing to maven central (#5224)
- Added `tabix` to the docker base image (#5247)

▶ **Assets** 3

Sep 20, 2018

droazen 4.0.9.0 6e352bb Compare

4.0.9.0

Highlighting this release are some important fixes and improvements to the `HaplotypeCaller` , in particular support for genotyping spanning deletions and a fix to the reference confidence calculation around indels. This release also brings support for "Requester Pays" GCS (Google Cloud Storage) buckets, `fasta.gz` support to the `-R` / `--reference` argument, a port of `LeftAlignAndTrimVariants` from GATK3, a new tool `FuncotatorDataSourceDownloader` to download `Funcotator` datasources, and bug fixes to `Mutect2` , `VariantRecalibrator` , and `SelectVariants` .

As usual, a docker image for this release can be downloaded from https://hub.docker.com/r/broadinstitute/gatk/

- `HaplotypeCaller`
  - Fixed the reference confidence calculation upstream of indels (#5172)
    - Improve hom-ref GQs near indels in GVCFs. Also consider bases on either side of indels informative if local assembly has been performed.
    - The previous behavior generated some PL=0,0,0 no-calls because the CIGAR of reads containing indels wasn't taken into account when determining which reads were informative for the indel reference confidence model. The local realignment wasn't being used inside the active region previously either, which has been fixed. A related change considers bases on either side of indels informative if local assembly has been performed (but not during active region detection). Both result in far fewer 0,0,0 calls. Unfortunately there are still some 0,0,X homRef calls related to #5171.
  - Make HaplotypeCaller genotype and output spanning deletions (#4963)
    - Modifies HaplotypeCaller so that it can output and genotype spanning deletion alleles represented by the * allele.
    - Fixes #2960
    - Previously, the output of HaplotypeCaller would not include spanning deletion alleles when run in single sample VCF mode or in genotype given alleles mode, even when that genotype would be more appropriate. In the joint calling workflow GenotypeGVCFs adds genotypes for spanning deletions, although the input likelihoods will not be broken out to specifically account for spanning deletion alleles.
  - Simplify HaplotypeBAMWriter code. #944 (#5122)
- `Mutect2`
  - Mutect2 now emits DP values in the FORMAT field (#5185)
  - Add `--get-af-from-ad` option to recalculate the allele fraction based on AD instead of the Bayesian estimate (#5118)
    - Recommended for mitochondrial applications
  - Fixed a `StringIndexOutOfBoundsException` crash in the ReferenceBases annotation when a variant is within 10 base pairs of the end of a chromosome (#5151)
  - Restore base quality filter code that got removed unintentionally in #4895. (#5123)
  - Remove extra space in the `MutectVersion` header line (previously was `Mutect Version`) (#5184)

- Added support for "Requester Pays" GCS (Google Cloud Storage) buckets via new `--gcs-project-for-requester-pays` argument (#5140)
- Added fasta.gz support to the `-R` / `--reference` argument in walker tools (#5120)
- Added GCS/NIO support to the `--tmp-dir` argument (#4469)
- Upgraded `google-cloud-java` to the official 0.62.0 release, and move off of our custom fork of the library. This release includes the retry for transient `502` errors that we added to our fork in GATK 4.0.8.0 (#5194) (#5135)
- Ported the `LeftAlignAndTrimVariants` tool from GATK3 (#5144)
- `VariantRecalibrator` : the serialized model now sets annotation order (#3655)
    - This addresses a problem where serialized GMMs for VQSR assumed that the annotation order would be the same between the commands that generated them and the commands that used them. VQSR no longer depends on the commandline order of the annotations.
- `SelectVariants` : Drop sites with the * allele as the only ALT when running with `--exclude-non-variants` (#5129)
- `Funcotator` :
    - Created a new `FuncotatorDataSourceDownloader` tool to download data sources. (#5150)
    - Add an experimental `FilterFuncotations` tool (#4991)
    - Updated COSMIC to annotate protein change strings with their counts. (#5181)
    - Fix INDEL start/stop position and alleles for VCF gencode output. (#5131)
    - Get datasource version from a manifest file instead of the README (#5149)
    - Extract a new `FuncotatorEngine` to make it easier to write additional tools in the future that leverage Funcotator's annotation engine (#5134)
    - Handle character encoding error cases. (#5124)
- `CNNScoreVariants` :
    - Add WDLs and JSONs to run `CNNScoreVariants` in a single-sample workflow (#4774)
    - Added `--python-profile` argument to enable Python profiling. (#4953)
- `CNV tools` :

  - Produce an IGV-compatible seg file alongside the copy ratio calls in `CallCopyRatioSegments` (#5115)
  - Added optional mappability and segmental-duplication annotation to `AnnotateIntervals`. (#5162)
  - Improvements and refactoring of the `Nucleotide` class (#4846)

- `SV tools`:
  - Bug fix to read name mangling in `ExtractOriginalAlignmentRecordsByNameSpark` (#5107)
  - Added an `InsertSizeDistribution` class to represent expected insert-size distribution (normal and log-normal distributed) parameterized by insert size mean and stddev (#4827)
  - Added documentation clarification and additional validation to `SVInterval` (#5157)
  - Test and utils clean up (#5116)

- `MarkDuplicatesSpark`:
  - Switched `MarkDuplicatesSpark` tile-parsing code to use shorts in order to match Picard (#5165)
  - Added better error messages around missing read groups in `MarkDuplicatesSpark` (#5177)

- Clone read base qualities rather than reference them directly in the read clipper code to prevent unsafe array operations (#4926)

- Fix three bugs in the `AlignmentUtils` class (#3494)
  - The treatment of D-over-D in function applyCigarToCigar() was backward.
  - In function `createReadAlignedToRef()` the read start position passed to the `leftAlignIndel()` call was incorrect if the haplotype has an indel relative to reference.
  - When the `leftAlignIndel()` call drops any leading D operator in the result cigar, the read start position needs to be adjusted accordingly.

- Test infrastructure improvements:
  - Split out `gatk-testUtils` as a separate artifact in our build system(#5112)
  - Skip push builds if there is a pull request (cuts down on total number of travis builds by about half) (#5156)
  - We now share the test settings between the main build and the docker tests (#5155)

- Documented use of `--temp-dir` with `GenomicsDBImport`. (#5047)
- Deleted obsolete experimental tool `MarkDuplicatesGATK` in favor of `MarkDuplicatesSpark` (#5166)
- Deleted obsolete experimental tool `BaseRecalibratorSparkSharded` (#5192)
- Upgraded htsjdk to version 2.16.1 (#5168)
- Upgraded Picard to version 2.18.13. (#5173)

▶ **Assets** 3

Aug 16, 2018

droazen 4.0.8.1 bc98370 Compare

4.0.8.1

This is a small bug fix release to fix an issue with unpaired reads in `Mutect2`, as well as small fixes and improvements to `Funcotator`, `FilterVariantTranches`, and `MarkDuplicatesSpark`.

As usual, a docker image for this release can be downloaded from https://hub.docker.com/r/broadinstitute/gatk/

Full list of changes in this release:

- `Mutect2`: Fixed a "Cannot get mate information for an unpaired read" error that could occur with certain datasets containing unpaired reads that pass all the M2 read filters and show evidence of a SNV (#5121)
- `Funcotator`:
  - Fixes to the splice site logic. (#5106)
    - Funcotator now ignores leading indel bases when checking if variants are within the splice site boundaries (eg. if a leading base in an indel, which is preserved between the reference and alternate alleles, is within the splice site boundary

but the bases that have been changed are NOT, then the variant is now correctly labeled as NOT a splice site).
    - Populate the DB SNP validation status field properly (#5046)
        - Funcotator will now populate the MAF DB SNP Validation status field with proper values (e.g. "by1000genomes") instead of boolean value (e.g. "TRUE")
        - Funcotator now handles multiple records in a VCF funcotation factory that have the same pos, ref, and alt combination, even if equivalent and not exact matches.

- `FilterVariantTranches` :
    - Add an `--invalidate-previous-filters` argument to remove old filters left over from previous runs (off by default) (#5042)
    - Add `--snp-tranche` and `--indel-tranche` arguments to replace the previous `--tranche` argument (#5042)

- Updated `MarkDuplicatesSpark` scoring and comparison code to reflect changes in Picard (#5023)
    - Updated the scoring code to no longer take into account the unclipped start position of mismatching reads. Also changed the score to be a double packed short value in order to better reflect Picard scoring code.

- Other Changes:
    - Added new `IOUtils.isHDF5File()` utility method (#5082)
    - Add jitpack support for building GATK snapshots (#5056)
    - Fixed broken link in Travis to docker test failure reports (#5108)

▶ **Assets** 3

Aug 14, 2018

droazen 4.0.8.0 1e98c6d Compare

4.0.8.0

This release features some significant changes to `Mutect2` that improve both performance and correctness, as well as a bug fix to `GenomicsDBImport` for large interval lists.

As usual, a docker image for this release can be downloaded from https://hub.docker.com/r/broadinstitute/gatk/

Full list of changes in this release:

- `Mutect2`
  - Handle overlapping mates in M2 active region detection, causing fewer false active regions (#5078)
    - Makes Mutect2 ~25% faster in many cases with no loss of accuracy!
  - Filter M2 calls that are near other filtered calls on the same haplotype (#5092)
    - A very effective new filter that significantly reduces false positives
  - New Orientation Bias Filter (#4895)
    - New, improved orientation bias model, without which the M2 pipeline is not viable for NovaSeq data.
  - Changed the default AF slightly for M2 tumor-only mode (just a small tweak) (#5067)
  - Optimize some Mutect-related tools (#5073)
    - Everything that inherits from `AbstractConcordanceWalker` (this includes the `Concordance` tool and `MergeMutect2CallsWithMC3`) is now much faster on the cloud
  - Fixed edge case for M2 palindrome transformer (#5080)
    - Fixed an edge case involving reads assigned huge fragment lengths
  - Allowing counts for supporting alt reads in the validation normal. (#5062)
    - Added useful information suggesting possible normal artifacts in somatic validation tool.
  - M2 wdl doesn't emit unfiltered vcf, which is redundant (#5076)
- `GenomicsDBImport`
  - Fix for issue where we could run out of file handles when working with large interval lists (#5105)
  - Display warning when using large interval lists with `GenomicsDBImport` (#5102)
- Updated `MarkDuplicatesSpark` tie-breaking rules to reflect changes in picard (#5011)
- Added the ability for `CompareDuplicatesSpark` to output mismatching reads (#4894)
- Updated our `google-cloud-java` fork to 0.20.5-alpha-GCS-RETRY-FIX (#5099)

- We now retry on 502 and UnknownHostException errors when using NIO

- `SV Tools :`

  - Various improvements (#4996)
    - output a single VCF for new interpretation tool
    - bring MAX_ALIGN_LENGTH and MAPPING_QUALITIES annotations from CPX variants to re-interpreted simple variants
    - add new CLI argument and filter assembly based variants based on annotation MAPPING_QUALITIES, MAX_ALIGN_LENGTH
    - filter out variants of size < 50
  - Bug fix for the extreme edge case where after alignments de-overlapping, an alignment block is only 1 base long (#4962)
  - Turn back on checking variant info fields against header in SV vcf writing (turned off temporarily long time ago but slipped attention after implementation stablized) (#5084)

▶ **Assets** 3

Jul 30, 2018

cmnbroad 4.0.7.0 b6a630a Compare

4.0.7.0

Some important fixes in this release include a new version of GenomicsDB with a fix for the stack overflow seen when using large interval lists, and an updated Docker image with a fix for the missing R/ggplot2 dependencies.

As usual, a docker image for this release can be downloaded from https://hub.docker.com/r/broadinstitute/gatk/.

Docker

- Restore missing R/ggplot2 dependencies on the Docker image. [#5040 (https://github.com//pull/5040)

GenomicsDB

- Fix GenomicsDBImport stack overflow when using large number of intervals #4997

Mutect2

- Don't use very short stubs of clipped reads for genotyping #5057
- Add maxRetries to runtime in M2 WDLs #5049
- Fix an edge case bug in PalindromeArtifactReadTransformer #5038
- Make orientation bias filtering default to true #5019
- Added option for ValidateBasicSomaticShortMutations to output a vcf #4999
- Add Mutect2 PalindromeArtifactReadTransformer to hard clip inverted tandem repeats insertion artifacts #4998
- Making MAF become the output of Funcotator in M2 WDL and multiple transcript fix. #4941

CNV Tools

- Exposed ability to blacklist intervals in CNV WDLs. #5027
- Added output of IGV-compatible .seg files to ModelSegments. #5048

Structural Variants

- Add BreakpointEvidence filter based on classifier #4769
- Address more edge cases in assembly alignments #5044
- Refactor AssemblyContigAlignmentsConfigPicker #4971
- Fix an edge case in assembly contig alignment picker where no good mappings to canonical mappings exist #5005
- Trim down ref bases for CPX variants #4970

Funcotator

- VCF Funcotation Factory will recognize equivalent alleles (even when not exact) #4977

Other

- Include docs for new variant quality score model #5008
- Engine changes related to migration of GATK3 VariantEval to GATK4 #4495
- Fix position annotations to use position in original, not clipped, read #4956
- Add cmd line to VCF generated by GATKSparkTool #4981

▶ **Assets** 3

Jul 6, 2018

droazen 4.0.6.0 d6a7fcd Compare

4.0.6.0

Highlights of this release include:

- A new version of `GenomicsDB` that brings many long-requested features such as support for multiple intervals in `GenomicsDBImport`
- A significantly (~33%) smaller GATK docker image
- An important bug fix for the `-new-qual` option in `GenotypeGVCFs` / `HaplotypeCaller` / `Mutect2`

As usual, a docker image for this release can be downloaded from https://hub.docker.com/r/broadinstitute/gatk/

Full list of changes in this release:

- GenomicsDB: new version with many long-awaited features and bug fixes (#4645)
    - Multi-interval support in `GenomicsDBImport` (#3269)
        - Now you can specify multiple `-L` intervals when importing variants into GenomicsDB using `GenomicsDBImport`, instead of having to specify one interval per invocation.
    - New protobuf-based API to allow configuration without editing JSON files
    - Support for sites-only queries
    - Support for returning the genotype (GT) field in queries
    - Fixed bug where records with spanning deletion alleles could cause reads from GenomicsDB to fail (#4716)
- Reduced the size of the GATK docker image by approximately 33%, from ~5.3 GB to ~3.5 GB (#4955)

- Fixed a regression in the `-new-qual` option for `GenotypeGVCFs` / `HaplotypeCaller` / `Mutect2` that was introduced in GATK `4.0.5.0` (#4980)
    - There was a precision issue in the `AlleleFrequencyCalculator` when running with `-new-qual` that could cause a crash at certain sites (specifically, sites with spanning deletions and highly unlikely alt alleles).
- `HaplotypeCaller` : don't count qual = 0 sites as polymorphic for GVCF mode (#4967)
- `ValidateBasicSomaticShortMutations` : added a new optional argument to produce summary table output (#4982)
- `ExtractOriginalAlignmentRecordsByNameSpark` : added a new optional argument to invert the logic in the read-name filtering (#4944)
- Separated out the "variant calling" integration tests from the rest of the integration tests to speed up overall test suite runtime in travis (#4984)

▶ **Assets** 3

Jun 29, 2018

cmnbroad 4.0.5.2 ee81df1 Compare

4.0.5.2

Highlights of this release include major Funcotator performance improvements on hg19/b37 inputs, a newly rewritten Java version of FilterVariantTranches, HaplotypeCaller bamout improvements, and improved Python integration by eliminate timeouts.

As usual, a docker image for this release can be downloaded from https://hub.docker.com/r/broadinstitute/gatk/.

Funcotator Improvements

- Improve handling of hg19/B37 references (#4586).
    - Fixed performance bug involving excessive cache misses when querying datasources, resulting in major

performance improvements when running on HG19/B37 data (performance increased by approx. 30x with v1.4.20180615 of
the standard Funcotator data sources) (#4586).
  - Automatically detect when B37 data run against hg19 data source and convert contig names to be hg19 compliant.
  - Assumes all data sources for the hg19 reference are compliant with hg19 contig names. User-created data
sources will have to honor this.
  - Perform additional validation on input data to ensure a given reference FASTA has a sequence
dictionary that is a superset of the given VCF. This is a more stringent check than is automatically
performed by the GATK. Can be disabled with the `--disable-sequence-dictionary-validation` flag.
  - Released new version of datasources to go with this release (1.4.20180615), necessary because the data
sources needed to be made consistent with hg19 (before they were a mix of hg19 and b37 contig names).
  - Updated the minimum required data source version to be the latest release.
  - Updated the `getDbSNP.sh` and `createSqliteCosmicDb.sh` data source scripts to preprocess those data sources
to have hg19-compliant contigs names.
  - Removed the `--allow-hg19-gencode-b37-contig-matching` flag.
  - Removed the `--allow-hg19-gencode-b37-contig-matching-override` flag.
- User defined transcripts were being used as a filter rather than a priority order. The filtering step has been eliminated. Fixes #4918 (#4931)
- Added custom MAF fields to MafOutputRenderer (#4917)
- LocatableXsv data sources now produce at most 1 funcotation per allele pair. (#4936)
- LocatableXsv data sources now provide the correct number of funcotations (#4915)
- Preserve VCF fields in MAF output (#4872)
- Fixing error when spanning deletions overlap coding regions (#4881)

HaplotypeCaller/Mutect2

- Improvements to FilterMutectCalls. Eliminates about 3% of all false positives in DREAM while reducing sensitivity by about 0.1%
- Fix many questionable -bamout alignments where, because of a bad choice of Smith-Waterman parameters,

deletions were preferred over single-base substitutions.(#4858)
Result is many fewer spurious indels in the -bamout output.

- Introduced new SmithWaterman parameters affecting realignment of the reads to their best haplotype. This
  also changes some annotations that depend on the alignment, such as `BaseQualityRankSum` and `ReadPositionRankSum`.
  The changes are slight and make things more correct.
- Modify the behavior of (BaseGraph) getNextReferenceVertex for non-ref paths (#4889)

FilterVariantTranches

- Rewrite VCF Tranche filtering in java, with tests (#4800)

Engine

- StreamingPythonExecutor no longer uses timeouts or relies on prompt synchronization. (#4757)
- Allow concordance tools (AbstractConcordanceWalker) to use NIO for truth call set (#4905)
- Add pre- and post- apply variant transformer to VariantWalkerBase

MarkDuplicatesSpark

- Fixed a missing special case in MarkDuplicates ReadsKey code to better match current picard results (#4899)
- Reworked the keys for MarkDuplicatesSpark to be sufficient for grouping on their own. (4878)
- Improve error message for MarkDuplicates duplicates readnames issues (#4879)

Structural Variants

- Add tests for AssemblyContigWithFineTunedAlignments (#4961)
- Fix no index output for assembly bam file (#4945)
- Overhaul tests on assembly-based non-complex breakpoint and type inference code (#4835)
- Simple fix to remove trailing slash in GCS_SAVE_PATH to avoid double slashes in GCS_RESULTS_DIR (#4873)

Misc:

- Upgrading picard 2.18.2 -> 2.18.7 (#4949)
- Update htsjdk 2.15.1 -> 2.16.0 (#4914)
- Added support to PrintReadsSpark for non-coordinate sorted bams (#4853)
- Adding --sort-order option to SortSamSpark (#4545)
- Increased boot disk size on GATK tasks in M2 wdl to accomodate 4.0.5.0 docker (#4877)

▶ **Assets** 3

Jun 11, 2018

droazen 4.0.5.1 24b4f7b Compare

4.0.5.1

This is primarily a bug fix release to fix a crash in the help system (#4875). The issue was that tools that use annotations (which includes `Mutect2`, `HaplotypeCaller`, `GenotypeGVCFs`, `CombineGVCFs`, and `VariantAnnotator`) would crash when trying to print their help text. This could be triggered by running with an explicit `--help`, or by typing an invalid tool command line.

This release also brings in some improvements to `Funcotator`, including a new mode to output annotations for all transcripts.

As usual, a docker image for this release can be downloaded from https://hub.docker.com/r/broadinstitute/gatk/

Full list of changes in this release:

- Fix crash when displaying help text for tools that use annotations (#4876)
- `Funcotator` improvements (#4838) (#4870)
  - Added `ALL` mode for transcript selection (`--transcript-selection-mode ALL`) which will output full annotation fields for all transcripts
  - IGR annotation are no longer reported if there are any transcripts that would result in a non-IGR annotation for a given variant
  - VCF Datasources now have to match both the alt and ref alleles to be added as annotations to a variant

  - Added the `--allow-hg19-gencode-b37-contig-matching-override` flag to allow for even more permissive matching contig names between B37 and HG19 references (primarily designed to be used in development)
  - Updated the experimental Funcotator WDL to work properly in cromwell
  - Refactored internals of `Funcotator` to use `FuncotationMap` objects to store annotations
  - Additional tests to ensure VCF and MAF protein change strings are equivalent
  - Other minor internal bugfixes for testing
- Fix to the Oncotator command line in the `Mutect2` WDL (#4862)
- Removed unsupported `Mutect2` WDLs (these now live on Firecloud) (#4836)

▶ **Assets** 3

Jun 7, 2018

droazen 4.0.5.0 f4225b8 Compare

4.0.5.0

Highlights of this release include the ability to emit MNPs in `Mutect2` and `HaplotypeCaller` via a new `--max-mnp-distance` argument, much better active region detection for low allele fractions in `Mutect2`, new priors for variants sites and homRef blocks in `HaplotypeCaller`, a new tool `FilterAlignmentArtifacts` to filter false positive alignment artifacts in the `Mutect2` pipeline, performance improvements to `CNNScoreVariants` and `Funcotator`, and a new `--sites-only-vcf-output` GATK engine argument to suppress genotypes when writing VCFs.

As usual, a docker image for this release can be downloaded from https://hub.docker.com/r/broadinstitute/gatk/

Full list of changes in this release:

- `Mutect2`
  - Made `Mutect2` active region determination much better for low allele fractions (#4832)

        - In particular, this makes `Mutect2` vastly better for mitochondrial and cfDNA calling
    - `Mutect2` can now emit MNPs according to adjustable distance threshold specified via `--max-mnp-distance` (#4650)
    - Tweaked `Mutect2` read position filter to handle non-biological (eg FFPE) insertions better (#4851)
    - Fixed `Mutect2` bug where triallelic normal artifacts were sometimes hidden from filtering engine (#4809)
    - `Mutect2` STR filter now also looks at insertions (#4845)
        - This lowers the indel false positive rate dramatically.
    - `Mutect2 WDL`:
        - now outputs MAF segmentation (#4837)
        - now runs `FilterAlignmentArtifacts` (#4848)
        - now uses lenient validation in `SortSam` (#4844)

- Added new tool `FilterAlignmentArtifacts` (#4698)
    - Filters false positive alignment artifacts (that is, apparent variants due to reads being mapped to the wrong genomic locus) from a VCF callset by checking variant-supporting reads and their mates.
    - By considering the realignment of the read and its mate, it saves a lot of variants, especially in low-complexity regions, from being filtered as mapping errors.

- `HaplotypeCaller`
    - `HaplotypeCaller` can now emit MNPs according to adjustable distance threshold specified via `--max-mnp-distance` (#4650)
    - New `HaplotypeCaller` priors for variants sites and homRef blocks (#4793)
        - Added new `--population-callset` argument allowing an external panel of variants to be specified to inform the frequency distribution underlying the genotype priors
        - Added new `--num-reference-samples-if-no-call` argument to control whether to infer (and with what effective strength) that only reference alleles were observed at sites not seen in any panel
        - As a side effect of this change, `CalculateGenotypePosteriors` now supports indels.
    - GCS/NIO output support for the `-bamout` argument (#4721)

- `-new-qual` in `HaplotypeCaller` / `Mutect2` / `GenotypeGVCFs` no longer counts spanning deletions as support for variant qual (#4801)

- `CNNScoreVariants`
  - Performance improvements to the prep of the input tensors in the 2D model (#4735)
  - Bug fix to prevent a crash on the ends of the mitochondrial contig (#4751)
- `GATK Engine`
  - Added a new traversal type `TwoPassVariantWalker` that does two passes over its input variants (#4744)
  - Enable the `-L` argument to read feature files (such as `.bed` or `.vcf` files) from non-local Paths, including GCS buckets (#4854)
  - Added `--sites-only-vcf-output` argument to the GATK engine to suppress genotype fields when writing VCFs (#4764)
  - Tools that use annotations now use the barclay annotation plugin (#4674)
  - Added new `ReadQueryNameComparator` (#4731)
  - Automatically schedule temporary resource files for delete on exit (#4616)
- `Spark tools`
  - Added support for `g.vcf.gz` files in Spark. #4274 (#4463)
  - Spark tools can now write SAM files #4295. (#4471)
  - Added a `--output-shard-tmp-dir` argument to specify the parts directory for un-sharded BAM writing (#4666)
- `MarkDuplicatesSpark`
  - Fixed `MarkDuplicatesSpark` so it handles supplementary reads with unmapped mates properly (#4785)
  - Added a distinction between PCR orientation and Optical Duplicates orientation in `MarkDuplicatesSpark` (#4752)
  - Fixed serialization crash in `MarkDuplicatesSpark` (#4778)
  - Fixed queryname partitioning bug where asking for queryname sort would result in reads with the same name being split between partitions (#4765)
  - Changed `MarkDuplicatesSpark` to sort non-queryname sorted bams before processing to ensure marking is consistent across shards (#4732)
  - Renamed some `MarkDuplicatesSpark` arguments to follow the "kabob-style" convention (#4715)
  - `MarkDuplicatesSpark` now uses the Picard `OpticalDuplicatesFinder` directly (#4750)
  - `MarkDuplicatesSpark` now uses Picard metrics code directly (#4779)

- `BwaSpark` : disable sequence dictionary validation when aligning reads #4131 (#4308)
- `Funcotator`
  - Major performance improvements due to added caching and other optimizations (#4740)
  - Various fixes (#4783) (#4817) (#4770)
    - Sanitize special characters when outputting VCF so that VCF validation passes
    - Ordering specified in the header did not match the variants and hg19/b37 - VCF datasources were being inconsistently processed, inducing a lot of missed annotations.
    - Added Funcotator tests for Clinvar and Gencode v28 in hg38, and mixed chr/no-chr GENCODE.
    - Eased restrictions so that Gencode v28 would be recognized as a valid gtf. Future versions of Gencode will not fail just based on the version number and warning will be emitted instead.
    - Refining handling of transcripts with missing sequence info.
    - Refactored UTR VariantClassification handling.
    - Added warning statement when a transcript in the UTR has no sequence info (now is the same behavior as in protein coding regions).
    - Added tests to prevent regression on data source date comparison bug.
    - Fixed DNA Repair Genes getter script.
    - Fixed an issue in COSMIC to make it robust to bad COSMIC data.
    - Gencode no longer crashes when given an indel that starts just before an exon.
    - Fixed the SimpleKeyXsvFuncotationFactory to allow any characters to work as delimiters (including characters used in regular expressions, such as pipes).
    - Modified several methods to allow for negative start positions in preparation for allowing indels that start outside exons.
    - Fixed an issue in 5' UTR processing that would cause variant alleles with length > 1 to throw an exception (fixes issue #4712).
    - Fixed a bug in the version detection for Funcotator data sources that would prevent newer data source versions from being detected as compatible (date comparison error).
  - Gencode data sources now have names preserved from config files. (#4823)
- `GCNV` kernel tunings (#4720)
  - Fixed a minor issue in sampling error estimation that could lead to NaN (as a result of division by zero)

  - Introduced separate internal and external admixing rates
  - Introduced two-stage inference for cohort denoising and calling
  - Capped phred-scaled qualities to maximum values permitted by machine precision in order to avoid NaNs and overflows.
  - Took a first step toward tracking and logging parameters during inference, starting with the ELBO history.

- Validation of sequence dictionaries from multiple BAMs now throws warning instead of exception in CNV workflows. (#4758)

- `SV tools`

  - Tweak BWA to allow "gappier" alignments in local assemblies (#4708)
  - Added a new experimental tool named `CpxVariantReInterprepterSpark` to extract barebone-annotated simple variants from an GATK-SV discovery pipeline produced VCF containing complex variants (#4602)
  - Fix "UnhandledCaseSeen" error in `StructuralVariationDiscoveryPipelineSpark` (#4677)

- Added new `SingleSequenceReferenceAligner` class to align against an on-the-fly single contig reference using Bwa-Mem (#4780)

- Updates to the conda environment for Python-based tools (#4749)

  - Fix #4741, where newer versions of conda appear to treat relative references in the environment yml as being relative to the yml file instead of relative to the cwd (based on observation).
  - Add a second conda yml file (`gatkcondaenv.intel.yml`) for environments that use Intel hardware acceleration and the Intel Tensorflow package (based on #4735).
  - Added a gradle task (`condaEnvironmentDefinition`) to generate the conda yml files from a single template to ensure that all the environment definitions remain in sync. This task also generates the Python package archive.
  - Added a gradle task (`localDevCondaEnv`) to create or update a local (non-Intel) conda environment. This is a shortcut for use during development when you're iteratively changing/testing Python code and want to update the conda env.

- Added a new WEX test bam to `src/test/resources/large` , with a companion target interval list (#4756)

- Add slightly modified version of GATK3 github issue template (#4796)

- Updated htsjdk to 2.15.1 (#4830)

▶ **Assets** 3

Apr 26, 2018

droazen 4.0.4.0 7641f53 Compare

4.0.4.0

Highlights of this release include major performance improvements to `MarkDuplicatesSpark`, better sensitivity and precision in STR (short tandem repeat) contexts for `Mutect2`, support for a "genotype given alleles" mode in `Mutect2`, dbSNP support for `Funcotator`, and several important bug fixes to `CombineGVCFs`.

As usual, a docker image for this release can be downloaded from https://hub.docker.com/r/broadinstitute/gatk/

Full list of changes in this release:

- `MarkDuplicatesSpark`
  - New, optimized version of the tool with greatly improved performance and scalability (#4656)
  - Note that this tool is still marked as beta, and has a number of known issues. The current version is suitable for evaluation/profiling purposes only.
- `Mutect2` improvements
  - Added a GGA (genotype given alleles) mode activated via the `--genotyping-mode GENOTYPE_GIVEN_ALLELES` and `--alleles` arguments (#4601)
  - Better sensitivity and precision in STR (short-tandem repeat) contexts (#4690)
  - New, supported Mutect2 NIO-enabled WDL that works in Firecloud (#4710)
  - Better default AF for M2 tumor-normal mode (#4690)
  - Restored explicit PASS (as opposed to empty) filter in Mutect2 (#4644)
  - Fixed Mutect2 failure for germline resource without AF (#4607)
  - Fixed a bug in the Mutect2 WDL bamout where scatters with overlapping assembly regions failed (#4613)
  - Fixed extra filtering args being deactivated in Mutect2 WDL due to typo

- `CombineGVCFs` : several important bug fixes
  - ReferenceConfidenceVariantContextMerger fixes for spanning deletions, and use the correct types for the median calculation. (#4680)
  - Handle trailing reference blocks correctly (#4615)
  - Fix and test for calculating intermediate band interval start locations. (#4681)
- `Funcotator`
  - Added dbSNP support via a new VcfFuncotationFactory. (#4593)
  - Fixed the refContext annotation. (#4605)
  - Fixed calculation of GC content to be correct. (#4608)
  - Fixes for HG38 exception and better logging. (#4563)
  - *Note:* only datasource releases `1.2.20180329` and later will work with this version of Funcotator
- `HaplotypeCaller` : Fixed a bug that caused the `--comp` and `--input-prior` arguments to not be settable by the user (#4703)
- `CNNScoreVariants` : Better numerical consistency between python and java, and transpose bug fix (#4652)
- `CNV Tools`
  - A new framework to support automated evaluation of GATK CNV (#4276)
  - Enabled zero eigensamples to be specified for `CreateReadCountPanelOfNormals` (#4502)
  - Exposed maximum chunk size in CNV panel of normals. (#4528)
  - Changed CNV PoN to filter on equality to interval median percentile. (#4503)
- `SV Tools`
  - Breakpoint location and type inference unit (#4562)
  - Scaffold local assemblies (#4589)
  - Use the latest version of fermilite jni (#4622)
  - Update sv scripts to only copy a single bam file and index, and respect project parameter (#4646)
  - Various bug fixes (#4670) (#4623)
- Added GCS (Google Cloud Storage) output support to the following tools: `ApplyBQSR` , `SplitNCigarReads` , `ClipReads` , `LeftAlignIndels` , `RevertBaseQualityScores` , and `UnmarkDuplicates` (#4695) (#4424)

- Mark the `--disable-tool-default-read-filters` argument as advanced, and add a warning to its documentation string (#4671)
  - *Many tools do not function correctly without their default read filters turned on, so this argument is intended only for advanced users who know what they're doing!*
- `ParallelCopyGCSDirectoryIntoHDFSSpark` : allow the tool to take a filename glob to subset files to copy (#4624)
- Picard: updated to version 2.18.2 (#4676)

▶ **Assets** 3

EXHIBIT 1069.126

Mar 27, 2018

droazen 4.0.3.0 6979f37 Compare

4.0.3.0

This release brings a major update to our experimental neural-network-based `VariantRecalibrator` replacement, initial `MAF` support in `Funcotator`, as well as some updates to `Mutect2` and the `CNV` tools.

As usual, a docker image for this release can be downloaded from https://hub.docker.com/r/broadinstitute/gatk/

Summary of changes in this release:

- A major update to our experimental neural-network-based suite of variant scoring tools, which will eventually replace the `VariantRecalibrator` (#4245)
  - The `NeuralNetInferenceTool` has been renamed to `CNNScoreVariants`
  - Baseline models are now included in the distribution.
  - Added additional tools to write tensors and to train your own models given a VCF of validated calls, an unfiltered VCF and a confident region: `CNNVariantTrain`, `CNNVariantWriteTensors` and `FilterVariantTranches`
  - Read-level 2D models are now supported via the tensor-type read_tensor argument. 2D models at present are significantly slower than the 1D models.
- `Funcotator`:
  - Added prototype support for outputting `MAF` files (and many bug fixes) (#4472)
- `Mutect2:`

- `CalculateContamination` emits its segmentation and `Mutect2` germline model uses it (#4509)
- Option to emit (but still filter) all germline sites in `Mutect2` (#4522)
- Made number of samples to put variant site in `Mutect2` PON adjustable (#4566)
- Added `Oncotator` filtering enabled in `Mutect2` WDL. (#4423)

- `CNV` tools:
  - Replaced `CollectFragmentCounts` with `CollectReadCounts`. (#4564)
  - Allowed use of zero eigensamples in `DenoiseReadCounts`. (#4411)
  - Changed filtering of normal hets on overlap with copy-ratio intervals in `ModelSegments` to be consistent with filtering of case hets. (#4510)
  - Updated PostprocessGermlineCNVCalls (segments VCF writing, WDL scripts, unit tests, integration tests) (#4396)

- Miscellaneous changes:
  - `Concordance`: added option to analyze contributions of different filters (#4520)
  - Exposed the `-pairHMM` / `--pair-hmm-implementation` argument in `HaplotypeCaller`, which was previously hidden (#4494)
  - Set the default `samjdk.compression_level` to 2 (was previously 1) (#4547)
  - Upgraded to Spark 2.2.0 (#4314)
  - Changed Spark sharding of queryname-sorted bams to better handle secondary and supplementary reads (#4473)
  - Added logging output to the bam writing step for spark tools (#4501)
  - `git-lfs` is now required to compile the GATK
  - Added a registry for deprecated/unported tools. (#4505)
  - Updated the Hadoop GCS connector from 1.6.1 to 1.6.3. (#4590)
  - Added a large runtime resource directory to `git-lfs`, and exposed it to the Docker build. (#4530)
  - We now include full tool documentation in the GATK binary distribution zip (#4377)
  - Made our maven artifacts much smaller by preventing gradle uploadArchives from including distZip and distTar (#4569)
  - Added chr20 and chr21 alt contigs to the `GRCh38` reference snippet used for testing (#4548)

▶ **Assets** 3

Mar 2, 2018

droazen 4.0.2.1 8a78790 Compare

4.0.2.1

This is a small bug fix release containing fixes for the following issues:

- `HaplotypeCaller` : fix the `-contamination` / `-contamination-file` arguments, which were not working properly, and add tests (#4455)
- Fixes/improvements to the GATK configuration file mechanism (#4445)
  - If a Java system property is specified explicitly on the user's command line, allow it to override the corresponding value in the GATK config file
  - Bundle an example GATK configuration file with the GATK binary distribution. This config file can be edited and passed to the GATK via the `--gatk-config-file` argument.
  - There are still some configuration-related TODOs/known issues: in particular, the `gatk` front-end script currently bakes in some system properties internally, which will always override the corresponding values in the config file. We plan to patch the `gatk` script to no longer set these system properties internally, and delegate to the config file instead.
- `Mutect2` : minor bug fixes and improvements (#4466)
  - Fix "FilterMutectCalls trips on non-int value in MFRL tag" (#4363)
  - Fix ordering of allele trimming vs. variant annotation (#4402)
  - Fix "CalculateContamination gives >100% results" (#3889)
  - Disable the `MateOnSameContigOrNoMappedMateReadFilter` by default (#3514)
  - Make mapping quality threshold in `GetPileupSummaries` modifiable (#4011)
- `SV Tools:` Add a scan for intervals of high depth, and exclude reads from those regions from SV evidence (#4438)
- In the GATK docker image, run the GATK using the fully-packaged binary distribution jars, rather than the unpackaged jars (#4476). This fixes a number of minor issues reported by users of the docker image.

▶ **Assets** 3

Feb 26, 2018

lbergelson 4.0.2.0 2bb2f50 Compare

4.0.2.0

This is a small release that includes a new Beta tool, a port of `VariantAnnotator` from Gatk3, as well as some bug fixes and other improvements. `Mutect2` is no longer beta.

- `Mutect2` and `FilterMutectCalls` are now no longer beta! (#4384)
- new tool `VariantAnnotator` (#3803):
  - ported tool from GATK3
  - first beta release
- `Spark Improvements` :
  - fix a major performance regression that harmed performance of spark tools (#4428)
  - `SortReadFileSpark` renamed -> `SortSamSpark` (#4442)
  - minor improvements to Kryo registration (#4451)
- new `CNV` Tumor only WDL (#4414)
- Viterbi segmentation and segment quality calculation for gcnvkernel (#4335)
- Other Bug Fixes and Improvements:
  - update to latest GKL, improves performance of GZIP level 2 compression (#4379)
  - `CalculateGenotypePosteriors` fixed bug that caused duplicates in the output VCF as well as several other issues (#4352, #4431)
  - Display a more prominent warning message for Beta and Experimental tools. (#4429)
  - non-zero Picard tool exit codes now cause a non-zero exit from gatk (#4437)
  - removed support for deprecated Google Reference API (#4266)
  - Improve evidence info dumps and SV pipeline management (#4385)
  - oncotator docker uses default docker if not specified (#4394)
  - Added check for non-finite copy ratios in ModelSegments pipeline. (#4292)

- make FASTQ reader remove phred bias from quals (#4415)

▶ **Assets** 3

Feb 9, 2018

droazen 4.0.1.2 dbea0da Compare

4.0.1.2

This is a small bug fix release to fix issues in the WDLs for `Mutect2` and the `CNV` tools. It also includes a newer version of the `GKL` (Genomics Kernel Library) with some compression-related performance improvements.

As usual, a docker image for this release can be downloaded from https://hub.docker.com/r/broadinstitute/gatk/

- `Mutect2` WDL:
  - Handle sample names with spaces correctly (#4360)
  - Pass VCF indices correctly (#4381)
- `CNV` somatic pair workflow and somatic panel workflow WDLs:
  - Fixed `mem_gb_for_model_segments` parameter and exposed additional memory parameters (#4364)
- Update to `GKL` version 0.8.3 with compression-related performance improvements (#4311)

▶ **Assets** 3

Feb 1, 2018

droazen 4.0.1.1 390274b Compare

4.0.1.1

This is a small bug fix release that fixes the following:

- Fix sorting bug in `GatherTranches`. Gathered tranches should now be closer to target truth sensitivity in the lower range (~90%).
- `Mutect2` WDL: fix memory requests to request MB instead of GB.
- CNV somatic pair workflow WDL: added missing `Oncotator` optional arguments
- Prevent printing a stack trace when the user specifies the name of a tool that doesn't exist. Instead print suggestions for similar tool names.

▶ **Assets** 3

Jan 30, 2018

droazen 4.0.1.0 d6e7635 Compare

4.0.1.0

Highlights of this release include a preview version of a future neural-network-based VQSR replacement, the ability to generate a VCF from the `GermlineCNVCaller` output, allele-specific annotation support in `GenomicsDBImport`, as well as a number of important post-4.0 bug fixes. See below for the full list of changes.

As usual, a docker image for this release can be downloaded from https://hub.docker.com/r/broadinstitute/gatk/

Changes in this release:

- New experimental tool `NeuralNetInference` (#4097)
  - An eventual VQSR replacement.
  - Performs variant score inference with a 1D Convolutional Neural Network with a pre-trained model. This is faster but not as high quality the 2D model which is coming along with training and tranche-style filtering in the next GATK release (#4245).

    - Tool name subject to change!
- `GenomicsDBImport`:
    - Add support for allele-specific annotations (#4261) (#3707)
    - Allow sample names with whitespace in the sample name map file (#3982)
    - Fix segfault crash on long path names (#4160)
    - Allow multiple import commands to be run in the same workspace directory (#4106)
    - Fix segfault crash during import when flag fields not declared in the VCF header (#3736)
    - Improve warning message when PLs are dropped for records with too many alleles (#3745)
- CNV tools:
    - Added `PostprocessGermlineCNVCalls` tool for generating VCFs from GermlineCNVCaller output (#4254)
    - Exposed bounds for determining copy-neutral region in `CallCopyRatioSegments` (#4263)
    - Added support for CRAM inputs to CNV WDLs (#4257)
    - Miscellaneous bug fixes, documentation updates, and WDL cleanup.
- `HaplotypeCaller`
    - Fix the `--min-base-quality-score` / `-mbq` argument, which previously had no effect (#4128). This fix also affects `Mutect2`.
    - Fix a "contig must be non-null and not equal to *, and start must be >= 1" error by patching an edge case in the ReadClipper code: when reverting soft-clipped bases of a read at the start of a contig, don't explode if you end up with an empty read (#4203)
- `Mutect2`:
    - Smarter contamination model (#4195)
    - Removed the `--dbsnp` and `--comp` arguments. The best practice now is to pass in `gnomAD` as the `germline-resource`.
    - Removed a number of other arguments that were `HaplotypeCaller`-specific and not appropriate for `Mutect2`, such as `--emit-ref-confidence`.
    - Mutect2 WDL: CRAM support (#4297)
    - Mutect2 WDL: Compressed vcf output and Funcotator options (#4271)
    - Miscellaneous WDL cleanup
- `HaplotypeCallerSpark`:
    - Fixes to the tool that make its output much closer to that of the non-Spark `HaplotypeCaller` (#4278). Note that this tool (unlike the non-Spark `HaplotypeCaller`) is still in beta, and should not be used for any real work. There

are still major performance issues with the tool that in practice prevent running on certain kinds of large data and in certain modes.
    - Disallow writing a `.vcf.gz` when in GVCF mode, as this combination currently doesn't work (#4277)
- `BwaSpark`:
    - set more reasonable default set of read filters (#4286)
- `PathSeq`:
    - Add WDL for running the `PathSeq` pipeline with a README and example JSON input. (#4143)
- Fix piping between Picard tools run via the GATK by changing logging output to stderr (#4167)
- Disallow unindexed block-compressed tribble files as input to walkers (#4240) (#4224). This works around a bug in HTSJDK that could cause such files to appear truncated. Until the HTSJDK bug is fixed, block-compressed `.vcf.gz` files (and similar files) will need to be accompanied by an index, which can be generated using the `IndexFeatureFile` tool.
- Restore `.list` as an allowed extension for files containing multiple values for command-line arguments (#4270). The previous extension `.args` is also still allowed. This feature allows users to provide a file ending in `.list` or `.args` containing all of the values for an argument that accepts multiple values (for example: a list of BAM files), instead of typing all the values individually on the command line.
- Fix conda environment creation to work better with the release distribution. (#4233)
- `IndexFeatureFile`: more informative error message when trying to index a malformed file (#4187)
- Suggest using BED files as a way to resolve ambiguous interval queries. (#4183)
- Set Spark parameter userClassPathFirst = false #3933 (#3946)
- Update to HTSJDK 2.14.1 (#4210)

▶ **Assets** 3

Jan 9, 2018

droazen 4.0.0.0 1418c4a Compare

4.0.0.0

4.0.0.0 general release

▶ **Assets** 3

Oct 16, 2017

droazen 4.beta.6 fd8749d Compare

4.beta.6

Pre-release

This release brings a critical bug fix to the `GenomicsDBImport` tool related to sample ordering, plus a new tool `FixCallSetSampleOrdering` to repair vcfs generated using the pre-`4.beta.6` version of the tool. See the description of the bug in #3682 to determine whether you are affected. Do *not* run `FixCallSetSampleOrdering` unless you are sure that you are affected by the bug in #3682.

Other highlights include upgrading to the latest version of the Picard tools, and adding engine support for reading Gencode GTF files.

A docker image for this release can be found in the `broadinstitute/gatk` repository on dockerhub. Within the image, cd into `/gatk` then run `gatk-launch` commands as usual.

Note: Due to our current dependency on a snapshot of `google-cloud-java`, this release cannot be published to maven central.

Full list of changes for this release:

- Fixed sample name reordering bug in GenomicsDBImport (#3667)
- New tool FixCallSetSampleOrdering to repair vcfs affected by #3682 (#3675)
- Integrate latest Picard tools via Picard jar. (#3620)
- Adding in codec to read from Gencode GTF files. Fixes #3277 (#3410)
- Upgrade to HTSJDK version 2.12.0 (#3634)
- Upgrade to GKL version 0.7 (#3615)

- Upgrade to GenomicsDB version 0.7.0 (#3575)
- Upgrade Mockito from 1.10.19 -> 2.10.0. (#3581)
- Add GVCF support to VariantsSparkSink (#3450)
- Fix writing variants to GCS buckets (#3485)
- Support unmapped reads in Spark. (#3369)
- Correct gVCF header lines (#3472)
- Dump more evidence info for SV pipeline debugging (#3691)
- Add omitFromCommandLine=true for example tools (#3696)
- Change gatkDoc and gatkTabComplete build tasks to include Picard. (#3683)
- Adding data.table R package. (#3693)
- Added a missing newline in ParamUtils method. (#3685)
- Fix minor HTML issues in ReadFilter documentation (#3654)
- Add CRAM integration tests for HaplotypeCaller. (#3681)
- Fix SamAssertionUtils SortSam call. (#3665)
- Add ExtremeReadsTest (#3070)
- removing required FASTA reference input that was needed before (for its dict) for sorting variants in output VCF, now using header in input SAM/BAM (#3673)
- re-enable snappy use in htsjdk (#3635)
- fix 3612 (#3613)
- pass read metadata to all code that needs to translate contig ids using read metadata (#3671)
- quick fix for broken read (mapped to no ref bases) (#3662)
- Fix log4j logging by removing extra copy from the classpath.#2622 (#3652)
- add suggestion to regularly update gcloud to README (#3663)
- Automatically distribute the BWA-MEM index image file to executors for BwaSpark (#3643)
- Have PSFilter strip mate number from read names (#3640)
- Added the tool PreprocessIntervals that bins the intervals given by the user to be used for coverage collection. (#3597)
- Cpx SV PR serisers, part-4 (#3464)
- fixed bug in which F1R2 and F2R1 annotation kept discarded alleles (#3636)
- imprecise deletion calling (#3628)
- Significant improvements to CalculateContamination (#3638)
- Adds supplementary alignment info into fastq output, also additional... (#3630)
- Adding tool to annotate with pair orientation info (#3614)

- add elapsed time to assembly info in intervals file (#3629)
- Created a VariantAnnotationArgumentCollection to reduce code duplication and added a StandardM2Annotation group (#3621)
- Docs for turning assembled haplotypes into variant alleles (#3577)
- Simplify spark_eval scripts and improve documentation. (#3580)
- Renames StructuralVariantContext to SVContext. (#3617)
- Added KernelSegmenter. (#3590)
- Fix bug in for allele order independant comparison (#3616)
- Docs for local assembly (#3363)
- Added a method to VariantContextUtils which supports allele alt allele order independant comparison of variant contexts. (#3598)
- Fixed incorrect logger in CollectAllelicCounts and RecalibrationReport. (#3606)
- updating to newer htsjdk snapshot (#3588)
- clear diffuse high frequency kmers (#3604)
- update SmithWatermanAligner in preparation for native optimized aligner (#3600)
- added spark tool for extracting original SAM records based on a file containning read names (#3589)
- update README with correct path to install_R_packages.R #3601 (#3602)
- HostAlignmentReadFilter and PSScorer use only identity scores and exp... (#3537)
- Fixed alt-allele count in AllelicCountCollector and changed unspecified alleles in AllelicCount to N. (#3550)
- Fix bad version check in manage_sv_pipeline.sh (#3595)
- Use a handmade TestReferenceMultiSource in tests instead of a mock. (#3586)
- Repackage ReadFilter plugin tests (#3525)
- BamOut in M2 WDL and unsupported version with NIO for SpecOps Team (#3582)
- Changed the path for posting the test reports
- updates sv manager and cluster creation scripts to utilize dataproc cluster timed self-termination feature (#3579)
- Implemented watershed algorithm for finding local minima in 1D data based on topological persistence. (#3515)
- Reduce number of output partitions in PathSeqPipelineSpark (#3545)
- add gathering of imprecise evidence links and extend evidence intervals to make links coherent in most cases (#3469)
- Refactor PrimaryAlignmentReadFilter to PrimaryLineReadFilter (#3195)
- Update ReadFilters documentation (#3128)
- Changes in BwaMemIntegrationTest to avoid a 3-4 minutes runtime. (#3563)

- Make error informative for non-diploid family likelihoods #3320 (#3329)
- TableFeature javadoc and more tests (#3175)
- Re-enable ancient BED test in IndexFeatureFile. (#3507)
- add external evidence stream for CNVs (#3542)
- clip M2 alleles before emitting in case some alleles were dropped (#3509)
- Docs for M2 filtering (#3560)
- Fix static test blocks and @BeforeSuite usages to prevent excessive code execution when tests aren't included in a suite. (#3551)
- hide prototyping tools in sv package from help message (but still runnable if knowing their existence) (#3556)
- Add support for running tools with omitFromCommandLine=true (#3486)
- Adds utility methods to ReadUtils and CigarUtils. (#3531)
- Cpx SV PR serisers, part-3 (#3457)

▶ **Assets** 3

Sep 6, 2017

lbergelson 4.beta.5 `0ef81ef` Compare

4.beta.5

Pre-release

Small release, includes highlights include an update to our `BWA-MEM` version, an experimental `PythonScriptExecutor` and an important bugfix for `ValidateVariants -gvcf` mode

Note: this still includes snapshot dependencies that prevent us from releasing to Maven central.

Complete change list:

- Make directory name unique for BucketUtilsTest#testDirSizeGCS to avoid unwanted test interaction. (#3547)
- Simple PythonScriptExecutor. #3501 (#3536)
- Fix BucketUtils#dirSize on GCS. #3437 (#3539)
- code duplication in read pos rank sum and its allele-specific version #1882 (#2657)

- validatevariants -gvcf fix (#3530)
- Added `GetSampleName` as stopgap until we have named parameters (#3538)
- Pair HMM docs (#3433)
- Fix MissingReferenceDictFile exception constructor. #3492 #2922 (#3524)
- Extend ReadsPipelineSpark to run HaplotypeCallerSpark (#3452)
- Updates bwamem-jni depedency to 1.0.2 and adds the possibility of aligning singletons to BwaEngine classes. (#3474)
- Structural Variant Context (#3476)

▶ **Assets** 3

Aug 25, 2017

droazen 4.beta.4 38707bf Compare

4.beta.4

Pre-release

Highlights of this release include fixes to the GATK4 `HaplotypeCaller` to bring it closer to the output of the GATK3 `HaplotypeCaller` (although many of these fixes still need to be applied to `HaplotypeCallerSpark`), fixes for longstanding indexing and CRAM-related bugs in htsjdk, bash tab completion support for GATK commands, and many improvements to `Mutect2` and the SV tools.

A docker image for this release can be found in the `broadinstitute/gatk` repository on dockerhub. Within the image, cd into `/gatk` then run `gatk-launch` commands as usual.

Note: Due to our current dependency on a snapshot of `google-cloud-java`, this release cannot be published to maven central.

Changes in this release:

- `HaplotypeCaller`: a number of important updates and fixes to bring it closer to GATK 3.x's output (most of these fixes apply only to `HaplotypeCaller`, not `HaplotypeCallerSpark`) (#3519)
  - reduce memory usage of the `AssemblyRegion` traversal by an order of magnitude

  - create empty pileup objects for uncovered loci internally (fixes occasional gaps between GVCF blocks as well as some calling artifacts)
  - when determining active regions, only consider loci within the user's intervals
  - port some additional changes to the GATK 3.x `HaplotypeCaller` to GATK4
  - fix bug with handling of the `MQ` annotation
- Added bash tab completion support for GATK commands (#3424)
- Updated to `Intel GKL` 0.5.8, which fixes bug in AVX detection, which was behaving incorrectly on some AMD systems (#3513)
- Upgrade `htsjdk` to 2.11.0-4-g958dc6e-SNAPSHOT to pick up an important VCF header performance fix. (#3504)
- Updated `google-cloud-nio` dependency to 0.20.4-alpha-20170727.190814-1:shaded (#3373)
- Fix tabix indexing bugs in htsjdk, and reenable the `IndexFeatureFile` tool (#3425)
- Fix longstanding issue with CRAM MD5 slice calculation in htsjdk (#3430)
- Started publishing nightly builds
- Performance improvements to allow MD+BQSR+HC Spark pipeline to scale to a full genome (#3106)
- Eliminate expensive `toString()` call in `GenotypeGVCFs` (#3478)
- `ValidateVariants` gvcf memory optimization (#3445)
- Simplified `Mutect2` annotations (#3351)
- Fix MuTect2 INFO field types in the VCF header (#3422)
- SV tools: fixed possibility of a negative fragment length that shouldn't have happened (#3463)
- Added command line argument for IntervalMerging based on GATK3 (#3254)
- Added 'nio_max_retries' option as a command line accessible option for GATK tools (#3328)
- Fix aligned PathSeq input getting filtered by WellformedReadFilter (#3453)
- Patch the `ReferenceBases` annotation to handle the case where no reference is present (#3299)
- Honor index/MD5 creation for HaplotypeCaller/Mutect2 bamouts. (#3374)
- Fix SV pipeline default init script handling (#3467)
- SV tools: improve the test bam (#3455)
- SV tools: improved filtering for smallish indels (#3376)
- Extends BwaMemImageSingleton into a cache, BwaMemImageCache, that can... (#3359)
- Try installing R packages from multiple CRAN repos in case some are down (#3451)
- Run Oncotator (optional) in the CNV case WDL. (#3408)

- Add option to run Spark tests only (#3377)
- Added a .dockerignore file (#3418)
- Code cleanup in the sv discovery package (#3361) and fixes #3224
- Implement PathSeq taxon hit scoring in Spark (#3406)
- Add option to skip pre-Bwa repartitioning in PSFilter (#3405)
- Update the GQ after PLs get subset (#3409)
- Removed the explicit System.exit(0) from Main (#3400)
- build_docker.sh can run tests again #3191 #3160 (#3323)
- Minor doc fixes #3173 (#3332)
- Use ReadClipper in BaseQualityClipReadTransformer (#3388)
- PathSeq adapter trimming and simple repeat masking (#3354)
- Add scripts to manage SV spark jobs and copy result (#3370)
- Output empty VQSLOD tranches in scatterTranches mode if no variant has VQSLOD high enough for requested threshold (#3397)
- Option to filter short pathogen reference contigs (#3355)
- Rewrote hapmap autoval wdl (#3379)
- fixed contamination calculation, added error bars to output (#3385)
- wrote wdl for Mutect panel of normals (#3386)
- Turn off tranches plots if no output Rscript is specified (for annotation plots) (#3383)
- `Mutect2` wdls output the contamination (#3375)
- Increased maximum copy-ratio variance slice-sampling bound. (#3378)
- Replace --allowMissingData with --errorIfMissingData (gives opposite default behavior as previously) and print NA for null object in VariantsToTable (#3190)
- docs for proposed tumor-in-normal tool (#3264)
- Fixed the git version for the output jar on docker automatic builds (#3496)
- Use correct logger class in MathUtils (#3479)
- Make ShardBoundaryShard implement Serializable (#3245)

▶ **Assets** 3

EXHIBIT 1069.141

Jul 26, 2017

droazen 4.beta.3 b49f1c4 Compare

4.beta.3

Pre-release

This release contains a number of bug fixes and improvements. Highlights include a fix for intermittent failures/timeouts when accessing data in `Google Cloud Storage` (GCS), new and improved active-region detection for `Mutect2`, and a new `VariantRecalibrator` argument to allow the tool to scale better. See the full list of changes below. Most of the major known issues listed in the release notes for 4.beta.1 still apply, with the exception of the "intermittent GCS failures/timeouts" issue, which is now resolved.

A docker image for this release can be found in the broadinstitute/gatk repository on dockerhub. Within the image, cd into `/gatk` then run `gatk-launch` commands as usual.

Note: Due to our current dependency on a snapshot of `google-cloud-java`, this release cannot be published to maven central.

Changes in this release:

- `GATK engine`: Move to `google-cloud-java` snapshot with more robust retries, and set number of retries/reopens globally. This fixes the intermittent "all retries/reopens failed" error when accessing data on GCS (Google Cloud Storage). See issue #2749
- `Mutect2`: Implemented a new algorithm for active-region detection, reducing spurious active regions by almost 50%
- `Mutect2`: Filter artifacts that arise from apparent-duplicate reads
- `Mutect2 WDL`: `Oncotator` is now being told the case and control sample names explicitly in the WDL. The Oncotator code for inferring this could yield incorrect answers in some cases. See issue #3343

- `FilterByOrientationBias` : We discovered that it is impossible to guarantee a FDR threshold of all the variants when one artifact mode had high oxoQ and the other had low. We have changed the tool to guarantee the FDR threshold within each artifact mode, rather than for all variants. For more details, see issue #3344
- `FilterByOrientationBias` : Summary table was not being populated properly. That has been fixed. See issue #3309
- `VariantRecalibrator` : Add argument to pre-sample data for VQSR model building (and also recalibration) to reduce memory usage for production pipeline. See issue #3230
- Fix a stack overflow issue at high depths in the strand artifact annotation. See issue #3317
- `GenomicsDBImport` : add `--readerThreads` argument for multi-threaded vcf pre-loading. Improves performance of the tool by ~30% in our tests.
- `ValidateVariants` : port gvcf validation option from GATK3
- Polish up `PathSeq` and add pipeline tool
- Fix error message describing how to set the `GATK_STACKTRACE_ON_USER_EXCEPTION` property
- `Mutect2FilteringEngine` : correct `MEDIAN_BASE_QUALITY_DIFFERENCE_FILTER` and `MEDIAN_MAPPING_QUALITY_DIFFERENCE_FILTER` filter names
- `Mutect2 WDL` : gave `ProcessOptionalArguments` a leaner docker
- `GATK4 Docker Image` : changed the landing directory for the docker image to be `/gatk` instead of `/root`
- `Travis CI` : fixed test report not being uploaded to GCS
- `Travis CI` : removed non-docker unit and integration tests, which were redundant

▶ **Assets** 3

Jul 20, 2017

droazen 4.beta.2 a21447f Compare

4.beta.2

Pre-release

This is a bug fix release primarily aimed at fixing some issues in the `Mutect2` WDL. The major known issues listed in the release notes for 4.beta.1 still apply.

A docker image for this release can be found in the broadinstitute/gatk repository on dockerhub. Within the image, cd into `/gatk` then run `gatk-launch` commands as usual.

Changes in this release:

- `Mutect2 WDL` : corrected the ordering of `FilterMutectCalls` relative to `FilterByOrientationBias` . `FilterByOrientationBias` should always be run after all other filters, since (by design) it is trying to keep a bound on the FDR rate. See issue #3288
- `Mutect2 WDL` : added automated extraction of bam sample names from the input bam files, using samtools. This should be viewed as a temporary fix until named parameters are in place. See issue #3265
- `FilterByOrientationBias` : fixed to no longer throw IllegalStateExceptions when running on a large number of variants. This was due to a hashing collision in a sorted map. See issue #3291.
- `FilterByOrientationBias` : non-diploid warnings have been set to debug severity. This should reduce the stdout. As a side-effect, this should address/attenuate a comment in issue #3291.
- `VcfToIntervalList` : added ability to generate interval list on all variants, not just the ones that passed filtering. Please note that this change may need to be ported to Picard. Added an automated test that should fail if this mechanism is broken in the GATK. See PR #3250
- `CollectAllelicCounts` : now inherits from LocusWalker, rather than custom traversal. This reduced the amount of code. See issue #2968 (and PR #3203 for some other changes)
- Added experimental (and unsupported) tool `CalculatePulldownPhasePosteriors` at a user request. See issue #3296
- Implement `PathSeqScoreSpark` and `PathSeqBwaSpark` tools, and update `PathSeqFilterSpark` and `PathSeqBuildKmers` tools
- Many changes to `Mutect2` Hapmap validation WDL
- `GatherVcfs` : support block copy mode with GCS inputs
- `GatherVcfs` : fix crash when gathering files with no variants
- `AlleleSubsettingUtils` : if null likelihoods, don't add to likelihoods sums (fixes #3210)
- SV tools: add small indel evidence
- SV tools: several FASTQ-related fixes (#3131, #2754, #3214)
- SV tools: always use upstream read when looking at template lengths
- SV tools: fix bugs in the SV pipeline's cross-contig ignore logic regarding non-primary contigs

- SV tools: switch to dataproc image 1.1 in `create_cluster.sh`
- SV tools: `FindBreakEvidenceSpark` can now produce a coordinate sorted Assemblies bam
- Bait count bias correction for `TargetCoverageSexGenotyper`
- `CountFalsePositives` : fix so it a) does not return garbage for target territory and b) returns a proper fraction for false positive rate
- Specify UTF-8 encoding in implementations of `GATKRead.getAttributeAsByteArray()`
- GATK engine: fix sort order when reading multiple bams
- Fix `GATKSAMRecordToGATKReadAdapter.getAttributeAsString()` for byte[] attributes
- Fix various issues that were causing Travis CI test suite runs to fail intermittently

▶ **Assets** 3

Jun 28, 2017

droazen 4.beta.1 c43f93e Compare

4.beta.1

Pre-release

This release brings together most of the tools we intend to include in the final GATK 4.0 release. Some tools are stable and ready for production use, while others are still in a beta or experimental stage of development. You can see which tools are marked as beta/experimental by running `gatk-launch --list`

A docker image for this release can be found in the broadinstitute/gatk repository on dockerhub. Within the image, cd into `/gatk` then run `gatk-launch` commands as usual.

Major Known Issues

- GCS (Google Cloud Storage) inputs/outputs are only supported by a subset of the tools. For the 4.0 general release, we intend to extend support to all tools.
    - In particular, GCS support in most of the Spark tools is currently very limited when not running on Google Cloud Dataproc.
    - Writing BAMs to a GCS bucket on Spark is broken in some tools due to #2793

- `HaplotypeCaller` and `HaplotypeCallerSpark` are still in development and **not** ready for production use. Their output does not currently match the output of the GATK3 version of the tool in all respects.
- Picard tools bundled with the GATK are currently based off of an older release of Picard. For the 4.0 general release we plan to update to the latest version.
- CRAM reading can fail with an MD5 mismatch when the reference or reads contain ambiguity codes (#3154)
- The `IndexFeatureFile` tool is currently disabled due to serious Tabix-index-related bugs in htsjdk (#2801)
- The `GenomicsDBImport` tool (the GATK4 replacement for `CombineGVCFs`) experiences transient GCS failures/timeouts when run at massive scale (#2685)
- CNV workflows have been evaluated for use on whole-exome sequencing data, but evaluations for use on whole-genome sequencing data are ongoing. Additional tuning of various parameters (for example, those for `PerformSegmentation` or `AllelicCNV` in the somatic workflow) may improve performance or decrease runtime on WGS.
- Creation of a panel of normals with `GermlineCNVCaller` typically requires a Spark cluster.
- The `SV tools` pipeline is under active development and is missing many major features which are planned for its public release. The current pipeline produces deletion, insertion, and inversion calls for a single sample based on local assembly of breakpoints. Known issues and missing features include but are not limited to:
    - Inversions and breakpoints due to complex events are not properly filtered and annotated in some cases. Some inversion calls produced by the pipeline are due to uncharacterized complex events such as inverted and dispersed duplications. We plan to implement an overhauled, more complete detection system for complex SVs in future releases.
    - The SV pipeline does not incorporate read depth based information. We plan to provide integration with read-depth based detection methods in the future, which will increase the number of variants detectable, and assist in the characterization of complex SVs.
    - The SV pipeline does not yet genotype variants or provide genotype likelihoods.
    - The SV pipeline has only been tested on Spark clusters with a limited set of configurations in Google Cloud Dataproc. We have provided scripts in the test directory for creating and running the pipeline. Running in other configurations may cause problems.

▶ **Assets** 3