

---

## Research and Applications

# Development and validation of Houston Methodist Variant Viewer version 3: updates to our application for interpretation of next-generation sequencing data

Paul A. Christensen, Sishir Subedi, Kristi Pepper, Heather L. Hendrickson, Zejuan Li, Jessica S. Thomas, S. Wesley Long, and Randall J. Olsen

Department of Pathology and Genomic Medicine, Houston Methodist Hospital, Weill Cornell Medical College of Cornell University, Houston, Texas, USA

Corresponding Author: Paul A. Christensen, MD, Department of Pathology and Genomic Medicine, Houston Methodist Hospital, Weill Cornell Medical College of Cornell University, 6565 Fannin Street, M227, Houston, TX 77030, USA; pachristensen@houstonmethodist.org

Received 18 November 2019; Revised 21 January 2020; Editorial Decision 8 February 2020; Accepted 14 February 2020

### ABSTRACT

**Objectives:** Informatics tools that support next-generation sequencing workflows are essential to deliver timely interpretation of somatic variants in cancer. Here, we describe significant updates to our laboratory developed bioinformatics pipelines and data management application termed Houston Methodist Variant Viewer (HMOVV).

**Materials and Methods:** We collected feature requests and workflow improvement suggestions from the end-users of HMOVV version 1. Over 1.5 years, we iteratively implemented these features in five sequential updates to HMOVV version 3.

**Results:** We improved the performance and data throughput of the application while reducing the opportunity for manual data entry errors. We enabled end-user workflows for pipeline monitoring, variant interpretation and annotation, and integration with our laboratory information system. System maintenance was improved through enhanced defect reporting, heightened data security, and improved modularity in the code and system environments.

**Discussion and Conclusion:** Validation of each HMOVV update was performed according to expert guidelines. We enabled an 8× reduction in the bioinformatics pipeline computation time for our longest running assay. Our molecular pathologists can interpret the assay results at least 2 days sooner than was previously possible. The application and pipeline code are publicly available at <https://github.com/hmvv>.

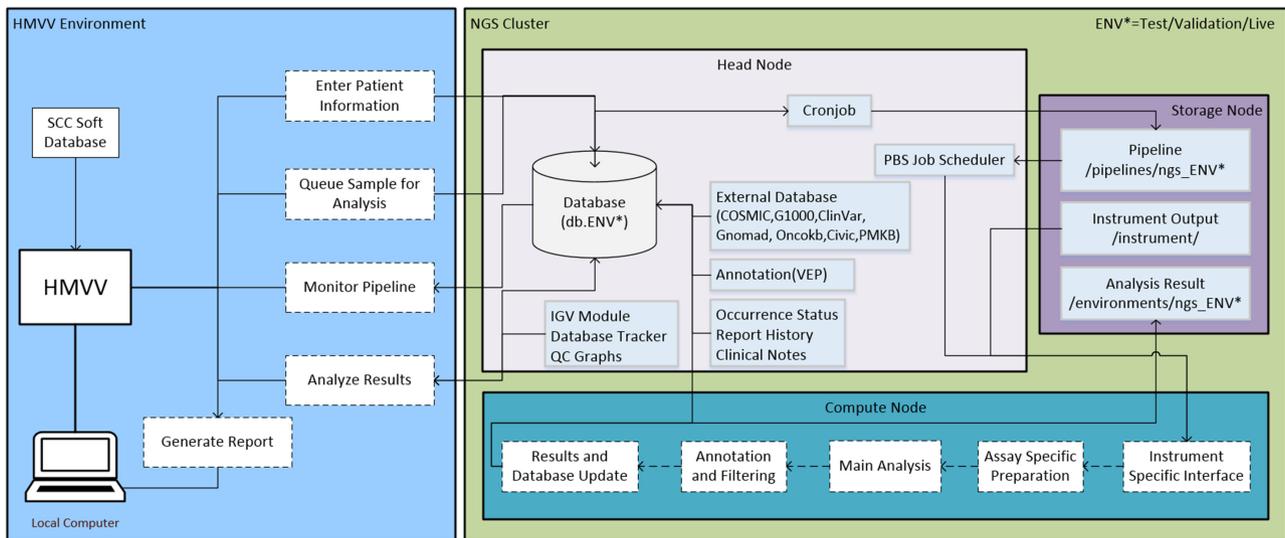
**Key words:** computational biology, high-throughput nucleotide sequencing, pathology, molecular, informatics, clinical laboratory information systems

---

## BACKGROUND AND SIGNIFICANCE

In 2017, we described Houston Methodist Variant Viewer (HMOVV),<sup>1</sup> our lab-developed next-generation sequencing (NGS) bioinformatics tool used to facilitate NGS data analysis and reporting. At that time, the application code was made available at <https://github.com/hmvv>. Over the subsequent 1.5 years, our development

team implemented significant enhancements to the analysis pipelines (computationally expensive data analysis scripts<sup>2</sup>) database design and graphical user interface (GUI). This manuscript describes the implemented updates that have enabled workflow improvements, reductions in possible transcription errors, and reductions in computational processing time for our bioinformatics pipelines.



**Figure 1.** Architecture diagram of the infrastructure components and data flow for bioinformatics processing of NGS samples. The HMVV environment consists of the application running on the user's workstation. The NGS Cluster consists of the head processing node, the storage node, and compute nodes.

### Use case for HMVV

HMVV was designed to assist our laboratory with the management of NGS sample and variant data (both pipeline generated and user generated). It provides functions to assist with interpretation of variants by collating relevant information from public databases. It integrates with Integrate Genomics Viewer (IGV) for drill-down into the alignment data for each potential variant call. Finally, it automates otherwise manual processes that would be performed by a laboratory technologist, including partial integration with our Laboratory Information System (LIS). While commercial packages were available for purchase at the time of our initial evaluation, we decided they did not adequately meet our needs and decided to build HMVV in house.

### HMVV background and overview

HMVV consists of Linux scripts (pipelines), a MySQL database and Java application that run on a computing cluster in our institution's data center (Figure 1). The pipelines either (1) annotate NGS variant calls (VCF files) generated by the vendor instrument and server or (2) process raw sequencing files (BCL) through an entire informatics pipeline (BCL → FASTQ → Alignment → Variant calls → Annotated variant calls). The purpose of each pipeline is to prepare a list of potential somatic variants from cancer specimens for review and final determination by a molecular pathologist. The functions of the MySQL database and Java application are to organize the sample information and variant calls into a user interface that makes interpreting the variant calls easier for the pathologist. The hardware infrastructure includes a storage server for storing the NGS data, a head node for supporting the MySQL database and user interactions within the Java application, and the compute nodes for executing the pipeline scripts. We refer you to the original manuscript for further details.<sup>1</sup>

## METHODS AND RESULTS

We collected feature requests and workflow improvement suggestions from all end-users of HMVV 1.0 (V1). The end-users included molecular pathologists, pathology residents and fellows, medical

technologists and medical laboratory technologists, bioinformaticians, and the molecular diagnostics laboratory manager. The feedback data were used to engineer modifications and feature enhancements to the HMVV V1 application. The changes aimed to increase efficiency for the molecular laboratory staff and directors, improve access to available variant databases, decrease manual processes, and improve turn-around-time. We iteratively implemented these improvements over 1.5 years in five versions of HMVV (V2.0, V3.0, V3.1, V3.2, and V3.3). At the time of writing this manuscript, the most current version of the application was HMVV V3.3 and will be referred to in abbreviated form (V3).

No ethics approvals were required for this manuscript.

### Architecture changes

We redesigned the overall architecture of HMVV V1 to improve modularity, testability, security, performance, and usability. The impact of these architecture changes included creation of three distinct environments (production, validation, and development), harmonization of end-user workflows, concurrent bioinformatics processing of samples, enhanced system security, improved re-use of common code across different pipelines, improved error handling, and database updates. The overall architecture consists of a computing cluster that contains a head node and multiple compute nodes, a storage node, and the end-user's application (Figure 1).

### Multiple environments

Based on the Development, Testing, Acceptance, and Production (DTAP-street) approach of software development,<sup>3</sup> we designed three different application environments to accommodate various stages of the application development and deployment. The development/test environment is used by the development team to implement and test new features and troubleshoot programming defects. The validation environment is used to perform an appropriate validation study of each new version. The live environment is used for sequence analysis and variant interpretation.

Each environment consists of three modules—a bioinformatics pipeline, data analysis files, and database. The pipelines module consists of scripts and configuration files. The data analysis module

QueueID	sampleID	RunID	sampleName	assayID	instrumentID	status	Status update time	program	progress
917	3186	476	neuro	neuro	proton	queued	2019-11-04 10:13:50	0/3	0%
916	3185	476	neuro	gene50	proton	queued	2019-11-04 10:13:36	0/3	0%
915	3184	476	neuro	gene50	proton	queued	2019-11-04 10:13:30	0/3	0%
914	3183	0397	neuro	heme	nextseq	queued	2019-11-04 10:13:12	0/6	0%
912	3181	476	neuro	neuro	proton	PipelineCompleted	2019-11-04 10:13:06	Complete	100%
913	3182	0397	neuro	heme	nextseq	queued	2019-11-04 10:13:00	0/6	0%
911	3180	476	neuro	neuro	proton	RunningEP	2019-11-04 10:12:31	1/3	55%
910	3179	476	neuro	gene50	proton	RunningEP	2019-11-04 10:12:31	1/3	55%
909	3178	476	neuro	gene50	proton	RunningEP	2019-11-04 10:12:31	1/3	55%
908	3177	476	neuro	gene50	proton	RunningEP	2019-11-04 10:10:26	1/3	55%
907	3176	476	neuro	gene50	proton	RunningEP	2019-11-04 10:10:25	1/3	55%
904	3173	0397	neuro	heme	nextseq	Alignment	2019-11-04 10:09:59	2/6	93%
905	3174	476	neuro	gene50	proton	PipelineCompleted	2019-11-04 10:07:37	Complete	100%

**Figure 2.** In the Monitor Pipelines window, the progress of currently running and recently finished pipelines are shown. The green background indicates a completed pipeline, yellow background indicates a running pipeline, and red (not shown) indicates a pipeline that encountered an error and needs to be investigated. The “program” column indicates the major step in analysis for that pipeline, while the “progress” column indicates an estimated completion percentage based on historical runtimes for samples from that assay and instrument.

consists of files generated by the pipeline scripts. The database module consists of sample information and annotated variant calls imported from the pipeline result files.

### Harmonization of end-user workflows and concurrent processing of samples

In HMVV V1, a technologist used a command-line interface to launch the bioinformatics analysis pipeline and monitor its successful completion. The pipeline executed each sample consecutively; no analysis jobs were run concurrently. After pipeline completion, the technologist used the HMVV GUI to enter the sample information and load the analysis results into the database.

In V3, the analysis pipeline launch was linked to HMVV sample entry, thus eliminating a possibly error-prone manual data entry step in the workflow (launching the pipeline via command line). The application simultaneously adds new sample information to the database and subsequently queues it for analysis using Terascale Open-source Resource and QUEUE Manager (TORQUE).<sup>4</sup> By using TORQUE on a compute cluster, multiple analysis jobs are run in parallel rather than individual jobs run sequentially. The number of concurrent samples is adjusted by configuring pipeline specific queues, each with different compute and memory requirements. Throughout the analysis, the pipeline updates the application database regarding its status. Following completion of the analysis, the pipeline automatically imports the annotated variant data into the application database, and the results become available for end-users. Additionally, an analysis completion notification email is sent to the technologist who launched the job.

### System security

Removing the command line access requirement for analysis job submission enabled creation of a faceless account to own the code, own the analysis data, and run the pipeline scripts. This design improved the permission controls and general security of the application by isolating account roles and supporting a restricted access control strategy. The faceless account also facilitates standardization of folder and file permissions and security restrictions.

### Pipeline code re-use and error handling

The analysis pipelines were redesigned to improve code re-use, readability, and maintenance. Various independent modules and sub-modules were built for different instruments, assays, and analysis functionality. Each sample is processed through a common checkpoint for identifying its instrument and assay type, and then routed to a specific instrument interface script. Similarly, once instrument-/assay-specific tasks are completed, each sample passes through

another common database update interface script that enters the required information into the HMVV database.

In the HMVV V1 pipelines, samples were analyzed sequentially. Our V3 design enabled the application and pipelines to process each sample concurrently. This approach provided an opportunity to utilize maximum available computing resources using parallel processing. We implemented error checking functionality at every major step to identify processing failures. Processing to the next step continues if all the necessary files were created in an appropriate format. In case of a failure at one of the checkpoints, the pipeline will terminate and display an error message on the monitoring dashboard of HMVV (Figure 2).

All scripts and script variables were renamed to keep consistency across the three environments. Common code was re-factored into subroutines with parameters, and environment variables were implemented throughout the pipelines. Together, all these changes improve maintenance, feature enhancements, and portability of the code.

### Database updates

We updated various elements of the database to improve performance and accommodate the architectural changes. We indexed key database fields which resulted in substantial reduction in variant list load times. We redesigned tables using database normalization principles and updated field naming and data definitions to better reflect the type and size of data stored in those columns. Similarly, we added new columns and tables to support new features in the application, such as a narrative of the sample diagnosis, trainee-entered variant annotation draft, variant annotation history, and pipeline progress and status logs. Finally, we updated previously used databases (ClinVar,<sup>5</sup> G1000,<sup>6</sup> and COSMIC<sup>7</sup>) and added new clinically relevant databases (gnomAD,<sup>8</sup> OncoKB,<sup>9</sup> CIViC,<sup>10</sup> and PMKB<sup>11</sup>) To ensure version control, copies of each database were maintained locally; after a version upgrade, we updated the application references without removing historical database versions.

### Workflow improvements changes in the GUI

We implemented improvements throughout HMVV to streamline the workflows for all users and prevent errors, including sample entry, pipelines monitor, and quality metric monitoring.

### Sample entry

We improved duplicate sample detection during sample entry (Figure 3). The technologist enters a sample into HMVV by typing data associated with a sample, including the instrument and assay identifiers, sample demographic information, and sample characteristics.

**Figure 3.** In Sample Entry, the technologist selects the Instrument, types the instrument generated RunID, and clicks “Find Run” (A). The list of samples will populate the SampleName Combo Box (B). If the sample was already entered, the sample information will populate the read-only fields (D). If the sample is new, the fields are editable. Scanning the sample barcode into the Barcode field (C) will populate the sample information for new samples.

Basic	VEP	Cosmic	G1000	ClinVar	Gnomad	OncoKB	Civic	PMKB	reported	IGV	Load IGV	gene	exons	type	HGVS	altFreq	g1000	gnomad	cosmicID	occurrence	annotation	somatic
✓									chr1:115256530			NRAS	3/7	smv	ENST00000369535.4:c.181C>A	10.916	0		COSMS80	0	Annotation	Somatic
✓									chr12:25398281			KRAS	2/6	smv	ENST00000256078.4:c.38G>A	14.85	0		COSM1140132, COSM532	3	Annotation	Somatic
✓									chr2:209108317			IDH1	6/10	smv	ENST00000415913.1:c.532G>A	8.012	4.932	4.864	COSM97131	16	Annotation	Somatic
✓									chr7:55241707			EGFR	18/28	smv	ENST00000275493.2:c.2155G>A	24.55			COSM6252	3	Annotation	Somatic

**Figure 4.** In the Mutation List, the Pathologist reviews the variants identified by the bioinformatics pipeline. In the Basic tab, the Load IGV checkbox (D) is selected for the creation of an alignment file that is transmitted from the server to the client when the “Load IGV” button (B) is pressed. The IGV column hyperlinks (C) use the IGV socket protocol to navigate IGV to the genomic coordinate for that variant. Historical reports (A) can be loaded from the LIS. The variant and gene level annotation window can be accessed with the Annotation hyperlink (E).

In V1, duplicate entries were detected only after a technologist entered all the relevant information and clicked the submit button. In V3, the duplicate is detected as soon as the unique identifiers are selected. This approach provides immediate feedback to the technologist that a duplicate sample identifier was chosen, which could be an indicator of user error.

We created a field to capture a specimen narrative history description. This field contains a free-text description summarizing the specimen diagnosis and indications for molecular testing. These data are most frequently entered by the molecular pathology fellow and are reviewed at the time of our consensus conference.

Integration with LIS generated barcodes in V3 is done through utilization of a Java Database Connectivity (JDBC) read-only connection to the LIS database. During sample entry, this connection is used to obtain information from a LIS barcode. This connection enables fast and accurate sample entry by eliminating manual transcription.

### Pipelines monitor

We designed a new module in HMVV V3 to monitor the progress of the analysis pipelines (Figure 2). The pipeline monitoring feature provides up to date information of all active analyses and a 7-day history of previous samples analyses. We can obtain a general overview or detailed information that includes the various steps of the analysis and associated timestamps. For actively running analysis

jobs, a progress bar estimate is displayed; the estimate is based on the average run time for historical samples from the same instrument and assay. When a pipeline failure occurs, the monitoring system helps to identify and troubleshoot the errant step.

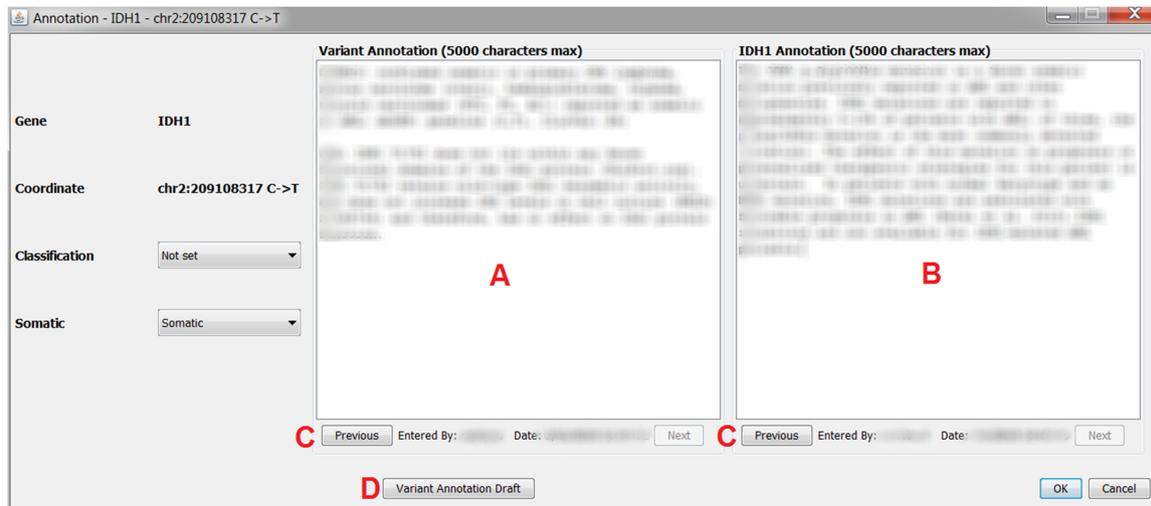
### Quality metric monitoring

The presentation of amplicon read depth data was improved from a long multi-line string of text to a structured and sortable table containing columns for the amplicon name and read depth. In addition, the percent of amplicons below the lab determined threshold is now automatically calculated and displayed.

We implemented a chart that trends the variant allele frequency for multiple variants expected to be present in the positive controls analyzed with each assay run. This trend is used as one metric to indicate the overall stability and quality of the assays.

### IGV integration

Workflow improvements in V3 were gained by enhancing the interactions between HMVV and IGV (Figure 4). Improvements include the file transfer implementation, reduction in the alignment file size, and genomic coordinate navigation. The three improvements to the HMVV-IGV integration enabled the molecular pathologists to more efficiently review variants, especially during the consensus conference when every variant is visually inspected in a group setting.



**Figure 5.** The variant annotation window enables variant level (A) and gene level (B) annotations. All versions of the entire annotation history are available (C). The variant annotation draft (D) is specific for the variant on that sample and provides a way for a trainee to enter a preliminary interpretation.

IGV provides a socket-based interface to support control from an external program.<sup>12</sup> We previously implemented the protocol to load alignment files (BAM) into IGV; however, the load command provided a Hypertext Transfer Protocol (HTTP) link to the BAM file stored on the storage server. This design resulted in significant latency in genomic coordinate navigation. In V3, we copy the BAM file from the server to the user's local computer running IGV. Although there is an initial file transfer delay, the subsequent navigation within IGV is rapid.

While implementing the above feature, we noted that the molecular pathologist was most likely to investigate a few select coordinates in the alignment file. We implemented an approach that creates a new BAM file on the server for only the specific variants selected, then transfers that file over the network. These changes reduce the BAM file size from several GBs to a few MBs, resulting in file transfer times of only a few seconds over a secure wireless network.

Although HMMV V1 supported the loading of the BAM file into IGV, the molecular pathologist had to copy and paste a row of data from HMMV into a text editor, select the genomic coordinate from the copied text, and paste the coordinate into the search bar of IGV. In V3, we implemented hyperlinks for each variant that automatically direct IGV to navigate to the genomic coordinate. The hyperlinks use the same socket-based interface used for loading BAM files. This enhancement made the workflow of investigating details of the variant calls much quicker and reduced the possibility of user error.

### Variant annotations

HMMV generates a draft text report containing the variant details and in-house annotations for all variants the molecular pathologist chooses to include in the final report (Figure 5). We enhanced the variant annotation feature to better allow trainees to document their interpretation of the variant data. These improvements include the creation of a variant level annotation history, gene level annotation history, variant annotation draft for trainees, and an additional column in the variant list to display the assignment of a variant's pathogenicity.

V1 only stored the most recent draft of a variant or gene annotation that carried forward across all instances of that variant. Now,

the entire history of the annotation is stored and visible, along with the user identification and entry date. These enhancements enabled a workflow for trainees to make changes to the variant and gene annotations without the risk of prior information being overwritten.

Another new feature is the variant annotation draft. While the variant and gene annotations persist for each instance of the variant, the variant annotation draft provides a mechanism for a trainee to enter their interpretation of the variant, which can be edited during consensus review, for that specific sample.

The third improvement to the annotation feature is the inclusion of the molecular pathologist's assignment of a variant's pathogenicity into the variant list. While this information was previously available in the Annotation Window, it was not visible in the variant list table. Presenting this information in the table provides a quicker cue to the pathologist regarding the institutional knowledge about all variants detected in the sample.

### Variant filtering

We added new variant filters to the HMMV Filter Panel. The new filters include the gnomAD Allele Frequency and Variant Effect Predictor (VEP) impact prediction. The new and previously included filters (G1000 global allele frequency, variant occurrence count, variant allele frequency, and read depth) allow molecular pathologists to select the list of variants that require further investigation.

The bioinformatics pipeline was originally designed to apply specific filters to the variant calls based on laboratory validated criteria. In rare circumstances, the pathologists desired to review the entire list of variant calls, but V1 did not support this workflow since the filtered list was imported into the applications database. In V3, all variant calls are imported into the database and can be loaded into the application, but only the filtered variants are visible in the default setting.

### Enhanced exception handling

HMMV is written in the Java programming language. If the code encounters an error, then it creates an exception. An exception contains a message about the error that was encountered, as well as information about the context of that error. In V1, only the error message was displayed. When these occurred, the users would send

a screenshot of the error to the developers along with a description of the steps that produced the error. While helpful, the developers found that a detailed context of the error was needed for adequate troubleshooting. V3 now presents both the error message and the context message. It includes a link to a defect report website designed for submission and tracking. This enhancement allows for better reporting and resolution of application defects.

### LIS integration

As noted earlier, we utilized a direct JDBC connection to our LIS (SCC Soft) database (Oracle). The connection incorporated barcode scanning to the sample entry workflow, thus reducing opportunities for manual data entry errors. After implementing V3, we retrospectively reviewed manually entered data from V1 and identified a 6.5% data entry error rate. In V3, the error rate was reduced to 0.9%, and in all cases, the data were manually entered rather than using a barcode. The LIS integration also natively integrates the specimen case history into HMMV.

### External database tracking dashboard

We implemented a dashboard of all external databases used in HMMV along with their version and release date information. This window displays the version status for each external database. The status consists of a number (months passed since its release date) and a color code (green = 0–1 year, yellow = 1–2 years, red  $\geq 2$  years). This dashboard enables convenient monitoring of external database ages and plan for appropriate upgrades.

### Sample search

V3 supports searching for samples based on the sample number, in addition to the other fields previously available (assay, lab order number, and run date).

### Application website

A user-friendly, internally hosted website was developed to provide access to all information related to the application. The website is used to download the application and release notes. It provides a defect reporting module for users to submit a description and supportive documentation for any issues with the application. The administrators are automatically emailed when new issues are submitted.

### Data migration

Due to significant changes in the database structure, and in order to maintain appropriate historical records of the previous live environment, we copied the database data from V1 into a new live environment.

## DISCUSSION

### Clinical validation

According to regulatory requirements and expert guidelines,<sup>13</sup> each iterative HMMV update included a validation plan that to ensure accuracy of analysis results, database information, data migration, and application functionality. As described in the system design section, all activities related to validation were conducted in a separate validation environment, which consisted of necessary code and database updates from the test environment. In a typical HMMV update validation, we retrospectively analyzed 60 patient samples proportionately distributed among different assays and compared the

results from the previous version. Data points expected to be unchanged between the two versions, such as total amplicons, total variants, variants match, total unique genes, and annotation classification were compared. To validate the external database updates, we analyzed all samples with both versions of the database and tabulated the matched and unique information. The results were further confirmed with the database source. Finally, internal database updates, data migration, and data integrity checks were conducted to ensure successful data transfer between the old and new live environments. An automated validation program, independent from the HMMV system, was created in the Python programming language to conduct these validations. Separate user workflow scenarios were developed for each feature and thoroughly tested to identify any defects in the system.

### Laboratory process improvement

The implemented changes reduced opportunities for manual error. The technologist performs their work relating to the bioinformatics pipelines using one harmonized application. We eliminated the possibility of manual errors in typing the Linux commands that launch pipelines. All status monitoring is done within HMMV. Barcode support and duplicate sample checks reduce opportunities for transcription and sample entry errors.

From the trainee perspective, HMMV V3 enables increased opportunity for active participation in the creation of the annotated report. A preliminary interpretation is entered into HMMV and presented directly from the application during a consensus conference attended by technologists, bioinformaticians, trainees, and physicians. During the consensus conference, BAM files are visualized using IGV from a laptop with a wireless network connection (as opposed to requiring a high-bandwidth wired connection). A similar workflow for presentation at tumor boards or other venues is also possible.

### Laboratory throughput improvement

Parallel processing (versus sequential processing) of samples on our compute cluster reduced processing time from sample entry to the final sample completing for all assays. For our longest running assay, the HMMV V1 average total runtime for consecutive samples was 48 h (range 30–72 h). The runs consisted of 11 samples on average (7–20 samples), and they could be further delayed by unexpected restarts of the client computers, which occurred almost weekly during regular IT security updates. In V3, the runs complete in 6 h on average (range 3–10 h) with similar run volume (10 samples on average, range 7–16 samples), resulting in an overall 8 $\times$  reduction in the analysis compute time. In addition, the V3 runs are never interrupted since the analysis is no longer dependent on the client computer. These changes made it possible for our molecular pathologists to interpret the assay results at least 2 days sooner than was previously possible.

### Comparison to IGV

HMMV and IGV are complementary in the suite of tools used to interpret NGS data. We use IGV to examine the sequencing base reads and alignment in the region of variant calls of interest. HMMV is used to manage and visualize the sample-specific metadata, the list of variants called by the pipeline, and integrate external databases for each of those variants. We have implemented the External Control of IGV functionality within HMMV to make these tools even more complementary.

## CONCLUSION

Due to the high cost and lack of satisfaction with available commercial options for NGS testing in cancer, we chose (as others have done<sup>14–17</sup>) to develop our own bioinformatics pipeline and data visualization tool. Although this application was previously described in 2017, development over the past 1.5 years has produced marked improvement in the feature set, stability, and portability of HMVV. In addition to the HMVV GUI application code that was previously made available (<https://github.com/hmvv/hmvv>), we have now published the pipeline scripts for our assays ([https://github.com/hmvv/ngs\\_pipelines](https://github.com/hmvv/ngs_pipelines)). We strongly believe that laboratory developed applications such as HMVV V3 should be made publicly available to the biomedical community.

To replicate our system from a technical perspective, an institution would need a Linux server with a MySQL database and various freely available applications installed (bwa,<sup>18</sup> samtools,<sup>19</sup> bedtools,<sup>20</sup> picard,<sup>21</sup> varscan,<sup>22</sup> VEP,<sup>23</sup> and TORQUE<sup>4</sup>). Adequate storage space for sequencing data is needed. The server would either need to function as an analysis compute node or be able to launch compute jobs to another computing cluster. The client computers would need Java version 1.8 or newer. From a governance and regulatory perspective, policies and procedures for data access, data backup, and all other regulatory requirements are recommended. While some level of technical expertise is needed to implement this infrastructure, the overall cost may be lower than purchasing and maintaining a commercial system.

## CONFLICT OF INTEREST STATEMENT

None declared.

## AUTHOR CONTRIBUTIONS

All authors made substantial contributions to the application features and workflow designs. PC and SS architected the system and wrote the application code. All authors participated in drafting and revising the manuscript.

## REFERENCES

- Christensen PA, Ni Y, Bao F, *et al.* Houston Methodist Variant Viewer: an application to support clinical laboratory interpretation of next-generation sequencing data for cancer. *J Pathol Inform* 2017; 8: 44.
- Roy S, LaFramboise WA, Nikiforov YE, *et al.* Next-generation sequencing informatics: challenges and strategies for implementation in a clinical environment. *Arch Pathol Lab Med* 2016; 140 (9): 958–75.
- Lubsen Z, Wijnholds G, Rigal S, Visser J. *Building Software Teams*. Sebastopol, CA: O'Reilly Media, Inc.; 2016.
- Staples G. TORQUE resource manage. In: proceedings of the 2006 ACM/IEEE conference on supercomputing, 2006. Tampa, FL: ACM, pp. 8.
- Landrum MJ, Lee JM, Riley GR, *et al.* ClinVar: public archive of relationships among sequence variation and human phenotype. *Nucleic Acids Res* 2014; 42 (D1): D980–5.
- 1000 Genomes Project Consortium, Auton A, Brooks LD, *et al.* A global reference for human genetic variation. *Nature* 2015; 526 (7571): 68–74.
- Tate JG, Bamford S, Jubb HC, *et al.* COSMIC: the catalogue of somatic mutations in cancer. *Nucleic Acids Res* 2019; 47 (D1): D941–7.
- Lek M, Karczewski KJ, Minikel EV, *et al.*; Exome Aggregation Consortium. Analysis of protein-coding genetic variation in 60,706 humans. *Nature* 2016; 536 (7616): 285–91.
- Chakravarty D, Gao J, Phillips SM, *et al.* OncoKB: a precision oncology knowledge base. *JCO Precis Oncol* 2017; 2017: 1–16.
- Griffith M, Spies NC, Krysiak K, *et al.* CIViC is a community knowledge-base for expert crowdsourcing the clinical interpretation of variants in cancer. *Nat Genet* 2017; 49 (2): 170–4.
- Huang L, Fernandes H, Zia H, *et al.* The cancer precision medicine knowledge base for structured clinical-grade mutations and interpretations. *J Am Med Inform Assoc* 2017; 24 (3): 513–9.
- Controlling IGV through a Port. Secondary Controlling IGV through a Port. <https://software.broadinstitute.org/software/igv/PortCommands>. Accessed July 12, 2018.
- Aziz N, Zhao Q, Bry L, *et al.* College of American Pathologists' laboratory standards for next-generation sequencing clinical tests. *Arch Pathol Lab Med* 2015; 139 (4): 481–93.
- Kang W, Kadri S, Puranik R, *et al.* System for informatics in the molecular pathology laboratory: an open-source end-to-end solution for next-generation sequencing clinical data management. *J Mol Diagn* 2018; 20 (4): 522–32.
- Roy S, Durso MB, Wald A, Nikiforov YE, Nikiforova MN. SeqReporter: automating next-generation sequencing result interpretation and reporting workflow in a clinical laboratory. *J Mol Diagn* 2014; 16 (1): 11–22.
- Crowgey EL, Stabley DL, Chen C, *et al.* An integrated approach for analyzing clinical genomic variant data from next-generation sequencing. *J Biomol Tech* 2015; 26 (1): 19–28.
- Woste M, Dugas M. VIPER: a web application for rapid expert review of variant calls. *Bioinformatics* 2018; 34 (11): 1928–9.
- Li H, Durbin R. Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics* 2009; 25 (14): 1754–60.
- Li H, Handsaker B, Wysoker A, *et al.*; 1000 Genome Project Data Processing Subgroup. The sequence alignment/map format and SAMtools. *Bioinformatics* 2009; 25 (16): 2078–9.
- Quinlan AR, Hall IM. BEDTools: a flexible suite of utilities for comparing genomic features. *Bioinformatics* 2010; 26 (6): 841–2.
- Picard toolkit. Broad Institute, GitHub repository 2019. <http://broadinstitute.github.io/picard/>. Accessed July 18, 2019.
- Koboldt DC, Chen K, Wylie T, *et al.* VarScan: variant detection in massively parallel sequencing of individual and pooled samples. *Bioinformatics* 2009; 25 (17): 2283–5.
- McLaren W, Gil L, Hunt SE, *et al.* The Ensembl variant effect predictor. *Genome Biol* 2016; 17 (1): 122.