**5. Research Design and Methods (12 pages maximum; separate PDF attachment)**

**Challenge Area:** *06: Enabling Technologies***Challenge Topic:** *06-HG-101  
New computational and statistical methods for the analysis of large data sets from next-generation sequencing technologies.*

**The Challenge and Potential Impact**

**The principal stakeholders in our study are large sequencing centers, genomics research groups, funders of genomics research, clinical diagnostics laboratories and a significant number of consumers of health care services.** Genomics plays a part in nine of the Ten Leading Causes of Death in the United States (<http://www.cdc.gov/genomics/faq.htm>). Genomics funding by government and nonprofit organizations averaged $2.9 billion a year from 2003-2006, of which the United States accounted for 35%, half of which was provided by the NIH (National Institutes of Health) (1).Rapidly falling NextGen sequencing costs mean that, even given a reduction in funding due to the current global economic downturn, the demand for NextGen analysis services will continue to rise and the use of NextGen technologies will continue to spread into other research communities. As these technologies develop, they will present greater data infrastructure demands and new bioinformatics challenges. **The increased adoption of NextGen sequencing among genomics research groups and clinical diagnostics labs must be accompanied by advances in high-throughput analysis, strategies for mitigating systemic bias, and new sequence assembly and downstream analysis pipelines.**

This section describes the particular challenges of NextGen technologies, applications and bioinformatics in more detail and discusses the anticipated impact of solutions provided by this study.

**Specific Aim 1: Develop and implement an optimized NextGen assembly workflow**

In order to appreciate the particular problems and challenges of NextGen sequence analysis, we must first grasp the particular strengths and weaknesses of the different sequencing technologies. The current mainstream NextGen platforms produce millions of short (50bp – 400bp) sequence reads. Each of the three main platforms, namely Illumina/Solexa, Roche/454 and ABI/SOLiD, have their own inherent problems, including significant sequencing error rates and systematic errors. Large sequencing organizations such as genome centers, academic core facilities and commercial contract-sequencing enterprises across the globe have already adopted this NextGen technology (Figure 1) and smaller labs and molecular diagnostics facilities participating in growing numbers. A common refrain among adaptors of this technology is that the downstream bioinformatics analyses are often poorly understood and underestimated.

Alongside the rollout of NextGen sequencing platforms, third generation sequencing technologies are being developed to sequence single DNA molecules faster and cheaper with streamlined samplepreparation. Real-time sequencing by synthesis is being developedby VisiGen (<http://www.visigenbio.com>) and Pacific Biosciences(<http://www.pacificbiosciences.com>). Pacific Biosciences is due to launch commercially in 2010 and has a mean DNA synthesisrate of approximately 4 bases per second, with a maximum read length of 4,000 bp.Also in development is sequencing based on sensing the bases of DNA molecules passed through nanopores (~5 nmin diameter). Different methods are being tested to create nanopores, including inorganicmembranes (solid-state nanopores), genetically engineered protein channelsby Oxford Nanopore Technologies(<http://www.nanoporetech.com>), polymer-based nanofluidicchannels, and a combinationof nanopores with sequencing by hybridization by NABsys (<http://www.nabsys.com>).

The currently available bioinformatics tools fall into four categories: reference aligners, de novo assemblers, variant-discovery tools and alignment viewers. Among the reference aligners are Eland (GAPipeline v0.30, Illumina), Mira, Genomics Workbench (CLC Bio), Seqman NGen (DNAStar), NextGene (Soft Genetics), MAQ and Shrimp. De novo assemblers include Edina, EULER-SR, SHARCGS, SSAKE,Velvet, and SOAPdenovo. Some NextGen statistical data-analysis tools are also available, such as JMP Genomics. **Despite the growing number of NextGen assembly/alignment tools, obtaining an accurately assembled sequence contig is still a very challenging problem.** The current tools vary widely in terms of data volume capacity (e.g., bacterial versus human data sets), number of reads aligned/assembled, error rates and bias, all of which may lead to suboptimal assemblies. Moreover, little is known about the comparative performance of the available tools because the few available performance statistics are based mainly on results from different non-human data sets (e.g., phage, bacteria, yeast). Therefore, achieving an unbiased comparison between different assemblers is difficult even before considering how they perform on human sequence data.

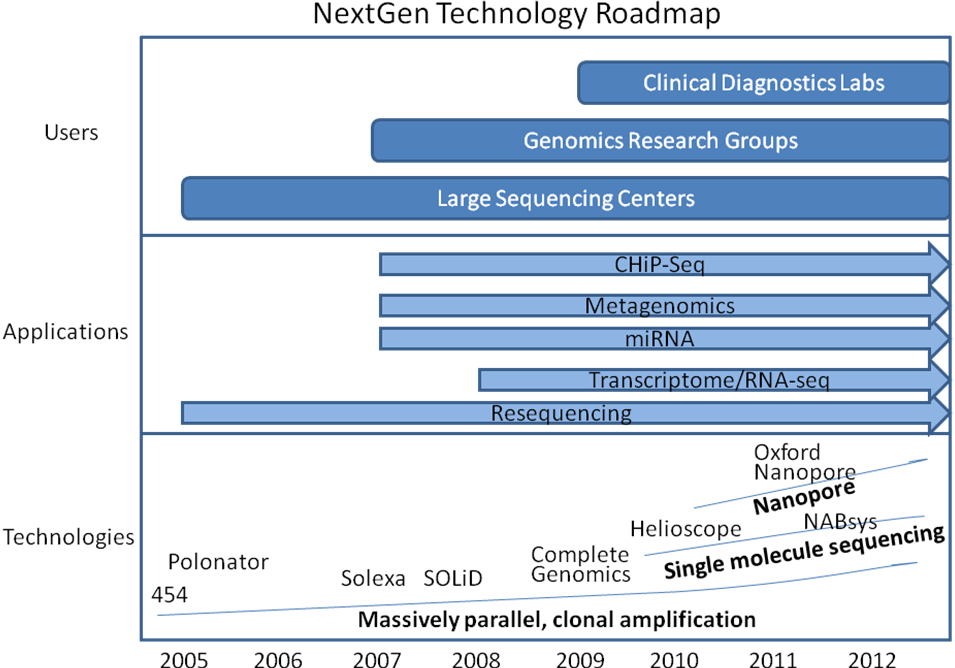


Figure 1. NextGen sequencing technology roadmap: the spread of uptake of NextGen sequencing and growth of applications parallels the development of new sequencing technologies. Data outputs are projected to increase rapidly as sequencing costs fall due to the rollout of new technologies.

Particularly in the early phases of the development of NextGen technology while many competing algorithms vie for supremacy, scientific publications will require comparisons of results using several different sequence analysis algorithms. **An objective, performance-based comparison of NextGen bioinformatics tools is an important step towards mitigating systemic bias in next generation data analysis. It will also lower the entry barrier allowing biomedical researchers to answer more penetrating questions and in less time.** An optimized NextGen assembly workflow will allow users to improve their work efficiency and the quality of their results. Customizable pipelines will also help meet the bioinformatics challenges faced by researchers at the cutting edge of life science exploration. Dynamic pipeline configuration coupled with high performance computing will enable researchers and other end users to rapidly develop and adapt different approaches to solving particular problems.

Another anticipated outcome will be the hastening of the transition to a mature technology, with fewer bioinformatics applications used for a wide range of applications. Extensive and objective comparisons between assembly/alignment tools will also serve to inform biomedical users which tools are more suitable for particular applications and data sets. A platform that is flexible enough to allow routine comparisons between the outputs of emerging algorithms for as-yet unknown NextGen bioinformatics challenges will be a valuable resource for research groups using genomics technologies and diagnostics labs alike.

**Specific Aim 2: Develop and implement NextGen genomic variation and expression analysis workflows**

Sequencing only the cDNA or transcribed portion of the genome focuses the analysis by reducing the size of the sequencing target space and also reduces costs. Expression profiling (a.k.a. RNA-seq or transcriptome analysis) is an increasingly popular application of NextGen sequencing (2) that has been shown to be robust and sensitive in comparison to five microarray platforms. NextGen sequencing also identified antisense transcription, which microarrays cannot detect, in 51% of all genes. In yeast, NextGen expression analysis has demonstrated a larger, more complex transcriptome than had been expected (3). An estimated 74.5% of the non-repetitive yeast genome was shown to be transcribed, as were many overlapping genes, alternative initiation codons and upstream open reading frames of yeast genes were demonstrated using short reads to generate a high-resolution map of the genome. Transcriptomes for mouse brain, liver and skeletal muscle were mapped by NextGen deep sequencing (4), providing a digital measure of the presence and prevalence of transcripts from known and previously unknown genes. RNA standards were used to quantify transcript prevalence and to test the linear range of transcript detection, which spanned five orders of magnitude.

Alongside the profound impact of NextGen applications in basic research, NextGen sequencing is also being adopted by clinical diagnostics laboratories for applications requiring deep sequence coverage and high-sensitivity such as rare HIV drug resistant variant detection (5). As the focus in human genetics research has shifted to genome wide association studies for the identification of genes involved with complex, multi-gene diseases, there is an increasing need for comprehensive diagnostic evaluations of SNPs as common or rare variants in multiple genes**.** At the base-pair level, NextGen sequencing has been shown to be highly suitable for high-throughput SNP acquisition using novel algorithms such as PolyBayes (6).

At the level of large-scale genomic variation, somatically acquired genomic rearrangements have been implicated in the development of various cancer phenotypes. The feasibility of using NextGen sequencing for the systematic, genome-wide characterization of rearrangements in human cancer genomes has already been demonstrated. The first high-resolution map of human genome structural variation revealed complex and large-scale structural variation in the form of insertions, deletions and inversions from a few thousand to millions of base pairs in length. NextGen sequencing has been used to characterize 306 germline structural variants and 103 somatic rearrangements to the base-pair level of resolution (7). **Improvements in the discovery of large scale genomic variations using NextGen sequencing will likely have a deep impact on the study of their involvement in cancer.**

Copy number variations are another form of important large-scale variation: CNVs of 100 kilobases and greater contribute substantially to genomic variation between normal humans (8, 9) however their contribution to human phenotype remains unclear, and they are difficult to measure. Microarray-based approaches for detecting CNVs depend on microarray signal intensity differences to predict regions of variation and cannot detect inversions. Before the advent of CNV prediction based on NextGen sequencing, only a small fraction of CNV base pairs had been determined at the sequence level. NextGen CNV mapping allows the discovery of cancer-causing genes in genomic regions that show recurrent copy-number alterations (gains and losses) in tumor genomes (10). **Advances in CNV prediction and characterization can be expected to have a far-reaching impact in the study of their involvement not only in the characterization of different disease phenotypes, but also in the context of human phenotypic variation.**

Developing high-throughput approaches for the analysis of the mechanisms whereby genomic variation can cause disease is a major genomics research challenge. To the extent that gene expression is a proxy of disease phenotype, NextGen sequencing can be used provide evidence of the relationship of sequence variation and phenotypes to disease. One approach is to use transcriptome QTL (quantitative trait locus) mapping to identify chromosomal regions containing sequence variants that cause variations in downstream expression. However, one issue with quantitative trait mapping is that of determining an appropriate threshold value for declaring significant QTL effects. Another approach is to target SNPs of interest in particular individuals early in the design stage of a study or to target specific pathways when analyzing genome-wide data but this has the drawback of selection bias. SNPs and CNVs have been associated with gene expression, but in an uncorrelated way. In a study of individuals in the International HapMap project, SNPs captured 83.6% of the total detected variation in expression levels of 14,925 transcripts and CNVs captured 17.7%, although the signals from the two types of variation had little overlap (11). Non-parametric machine learning techniques have been applied to narrow down groups of SNPs that best capture phenotypic variation using information metrics to select SNPs (12) and greedy screening (13). In yet another approach, allelic imbalance in gene expression has been found in 20–50% of genes tested in human brain, liver and kidney samples with no obvious reason evident from the SNP itself (14) and a pipeline for screening allele-specific expression has been proposed (15). These and other new approaches and tools for characterizing the relationship between genomic variation and phenotype or disease will improve our understanding of diseases and may lead to ways to treat and prevent them. Bioinformatics fills the space between NextGen sequencing technology and the interpretation of the data by the medical community to ultimately make an impact on human health.



Figure 2. An overview of workflow and viewer functionalities of currently available workflow tools and genome viewers shows that no existing tool provides all of the required functionality for an end-to-end solution.

**Specific Aim 3: Development of a NextGen workflow and visualization tool**

As bioscience becomes increasingly a quantitative analysis activity, workflow tools enable users to accomplish two main tasks: 1) automating well-defined, repetitive processes and 2) exploring data with ad-hoc analyses (16). **The few studies on common bioinformatics tasks) and usability of bioinformatics tools (17) identified an unmet need among bioscience researchers for workflow-based tools.** There are over 200 major Internet biological data sources however these sites are mostly simple GUIs (Graphical User Interfaces) with limited data compatibility between them despite the fact that researchers often need to combine the outputs of multiple sites to generate bioinformatics analyses. **These resources may be underexploited if users feel too much time is used navigating the sites, selecting among appropriate sources, downloading and uploading files. Workflow tools can solve this problem by generating large and complex systems from collections of programs, data sources and even structured data services. However, the majority of bioinformatics workflow tools only partially realize the potential of the available data and application resources (Figure 2).** With notable exceptions (18), there has been limited progress in connecting different sites with the client as the intermediary. Another approach is to use a central site as a service directory lookup such as BioMOBY and TAVERNA with the limitation that service discovery relies upon the accurate and uniform description of biological data types and relations between them, for which there is no commonly-accepted ontology (19-21) or language, despite some developments (22).

Data visualization and interpretation become paramount as the bioinformatics challenge shifts from mastering the basic tools to gaining biological insights from huge amounts of data. Three commercial software packages by DNAStar, SoftGeneticsand CLC Bio provide data viewers that allow the user to see read alignments, coveragedepth, genome annotations, and variant analysis. However, they lack the capability for viewing data sets as large as a whole human chromosome and show poor performance even on sub-chromosome data sets. The three major publicly available genome viewers, UCSC Genome Browser, Ensembl genome browser and GBrowse, are based on the traditional client-server model where the user’s requested data is reloaded as an image file delivered from the server. Java-based applications such as Apollo are more interactive but lack a concerted approach to data sharing. Most do not allow the user to filter the displayed data set based on biological criteria, although some newer applications such as IGV (http://www.broad.mit.edu/igv/) allow for limited filtering of the displayed features. Figure 2 lists the capabilities that a fully functional workflow tool and genome viewer must possess and shows the gaps in functionality of the currently available tools. We propose to provide all of this functionality in an integrated workflow tool and genome viewer.

**The Approach**

We propose to address the abovementioned bioinformatics challenges, namely the development of 1) improved NextGen sequence assembly workflows, 2) optimized genomic variation and expression workflows, and 3) a NextGen workflow and visualization tool with the following approaches:

**Specific Aim 1: Develop and implement an optimized NextGen assembly workflow**

We propose carrying out an objective evaluation of current NextGen assemblers/aligners using artificial data sets based on human biological samples in which each read’s position is known *a priori* in order to accurately compare results between different algorithms. Performance criteria will be established before testing based on the particular difficulties of assembling short reads derived from human genomic material. One or more optimized assembly workflows maximizing the performance criteria will be created as push-button tools to generate assemblies and associated quality control information. These workflows will also be customizable by the biologist/researcher to suit particular desired quality metrics or to meet any necessary tradeoffs between different quality metrics.

A combination of different assemblies may provide more reliable estimates of genetic aberrations by flagging dubious assembly regions that are not represented in a majority of the different assemblies. Conversely, regions that are matched identically by a majority of the different algorithms might be accorded greater confidence with regard to their predicted SNPs, indels and breakpoints. We will attempt to prove or disprove these hypotheses in the second step of specific aim 1 by using artificial reads generated from approximately 100 human sequence samples selected from the SRA (Short Read Archive <http://www.ncbi.nlm.nih.gov/Traces/sra/sra.cgi>). **A comparison study of this scale has never before been attempted.** Samples from individuals with sequences from all three NextGen sequencing platforms will be selected based on coverage depth, the presence of family member samples and the availability of additional genomic analyses such as microarrays in order to detect anomalies and discriminate between differing sequence assembly results.

**Specific Aim 2: Develop and implement NextGen genomic variation and expression analysis workflows**

To meet the need for comprehensive diagnostic evaluations of genomic variation in multiple genes, we propose the development of discovery and annotation workflows for three kinds of variation: SNPs, CNVs and large-scale genomic variation (chromosomal indels, inversions and dislocations). We will conduct three consecutive studies using biological samples from several human diseases to determine optimal methods for the prediction, filtering and verification of these different genomic variations. **The novelty of our approach lies in the fact that these workflows will include defined quality control/assurance algorithms, and the use of pedigree information and statistical techniques for predicting heterozygote/homozygote calls.** We will develop a methodology for estimating erroneous SNP calls and predicting homozygote and heterozygote genotypes using Roche/454 reads derived from Nimblegen exome capture samples (Figure 3). The initial experimental subjects are an extended family of 8 individuals, including 4 with a heritable neurodegenerative disorder. Initial SNP filtering and annotation will be carried out using read quality, read coverage, presence of read coverage in other sample members, presence in dbSNP, HapMap and the Venter and Watson genomes. We will validate our approach with a sample of 1,000 individuals on a smaller subset of captured genomic regions. **This analysis is being carried out in partnership with the MIHG (Miami Institute for Human Genomics). Further joint studies with the MIHG are anticipated involving the analysis of SNPs, large-scale variation and copy number variation in metabolic disorders, cancer, and cardiovascular disease.**

We propose enabling the integration into the variation analysis workflows of existing algorithms for determining the relationship between genomic variation and disease. These include GWAS studies and combinations of SNP, CNV and gene expression data with disease phenotypes in different human tissues and for different disease states. Many of these methods have not yet been applied to NextGen data and comparing the results of several methods will allow us to confirm associations between cis- or trans-acting genomic variations, expression phenotypes and disease.

In addition to the variation workflows, we propose the development of an optimized workflow for expression analysis. **This workflow will be integrated with current expression analysis packages such as ERANGE (4) and will include new expression analysis algorithms to improve quantization of transcript counts (Figure 3).** One approach that will be evaluated for inclusion in the workflow is the use of ‘standard gene sets’ or other transcribed regions with stable copy numbers across tissue types that can be used to calibrate the relative expression levels of genes and estimate absolute copy numbers per cell between different tissue samples.

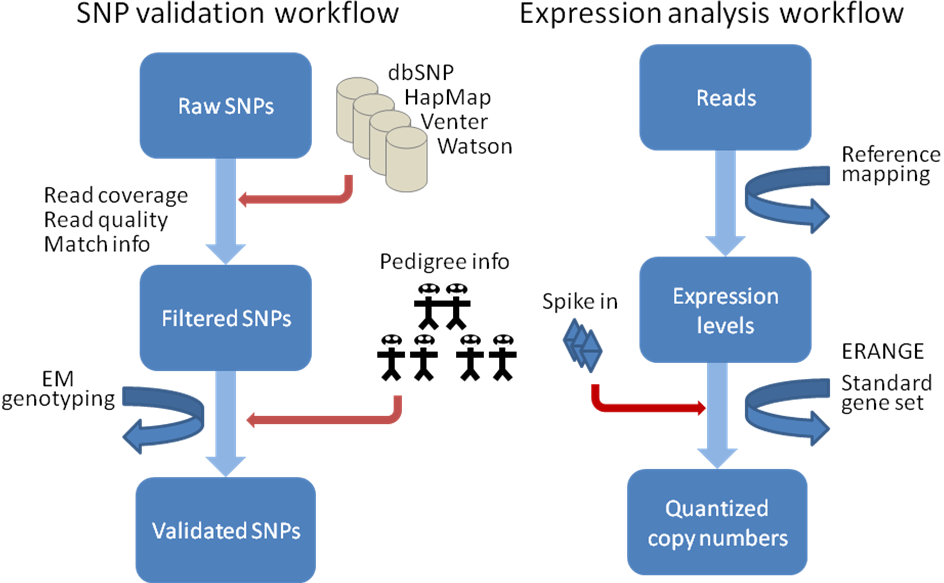


Figure 3. Prototype SNP validation and expression analysis workflows.

**Another unique feature of our approach is that, as part of the reporting and visualization of results, data filters will be designed based on user requirements to extract result subsets and provide genome-level views of the results integrated with external genomic features.** Results will also be exportable to downstream analysis applications (Cytoscape, Genespring, R, etc.).

**Specific Aim 3: Develop a NextGen workflow and visualization tool**Based on the requirements in Aims 1 and 2, we propose the implementation of a novel bioinformatics tool providing end-to-end integrated NextGen analysis workflows, reporting and real-time visualization of huge genomic data sets. The tool, named Aqwa (Automated Query and Workflow Agent), will provide pre-optimized workflows for NextGen assembly, genomic variation and expression analysis, and will also allow users to create their own customized workflows. **The software development process will implement a user-centric approach including extensive user testing at each project milestone and intervening iterations. Our emphasis on user testing and user-centric development, which differs from all previous bioinformatics workflow tools, is designed to ensure that the user interface is as intuitive as possible.**

Aqwa is designed so that only a basic familiarity with web page navigation and drag and drop user interfaces is required of the user. **Unlike other existing workflow systems, Aqwa will also support the addition of new Linux-platform applications in a “plug ‘n play” and scalable manner in order to flexibly adapt to changing computational needs and rapidly evolving bioinformatics challenges.** The following list of requirements and functional criteria encompasses the functionality envisaged for Aqwa.

**Functional Requirements**

1. Low barriers to usage
   1. Web access
   2. User-friendly, intuitive interface
   3. “Plug ‘n play” applications for rapid deployment
   4. Searchable project annotations
2. Workflows
   1. Predefined workflows (transcriptome, variation annotation, gene networks, file conversion utilities, ID conversion utilities, etc.)
   2. Customizable workflows
   3. Drag ‘n drop workflows
   4. Persistent data and workflow configurations
   5. Loops, conditional branching
3. Reports - customizable report extraction from workflow output
4. Views
   1. Customizable genomic views of report data
   2. Interactive display with rich context menu
   3. Integrated, extensible genomic features
   4. Multiple feature views – nucleotide level to aggregate high-level view
   5. Fast view update
   6. User can filter view based on data
   7. Genomic feature-level annotation by user
5. Sharing
   1. User-defined groups with customizable permissions
   2. Workflow, report and view sharing among groups
6. Input/Output and execution
   1. Import external biological data and genomic features into workflow
   2. Integration with external software (e.g., Cytoscape, R, GeneSet Analyzer)
   3. Programmatic remote access (API and Web Service)
   4. Cluster execution
   5. Grid execution
7. Maintain state (action history)
8. Data management – direct user access to input and output files

Information regarding data provenance is retained in the system to identify the source of data throughout the workflow such as the owner, author application, creation and modification dates, and content type. In addition, a log is kept of all project changes and updates. The user can also annotate the project at all levels and search these annotations. Aqwa is similar to the laboratory notebook paradigm employed by the BCJ (Bioinformatics Computational Journal) workflow tool but has a wider the range of functionality. **Aqwa’s functional requirements largely encompass those for a proposed ‘genome wiki’ (23) intended to facilitate cooperative genome annotation by a community of experts, reflecting Aqwa’s utility to the wider genomics community as an accurate, continually updated source of genome annotation.** Aqwa’s genome viewer functionality incorporates the AJAX-enabled JBrowse genome viewer (<http://jbrowse.org>) to provide a fast, fluid and responsive genome browser interface (Figure 4).



Figure 4. Aqwa’s Jbrowse-based allows drag ‘n drop feature selection and real-time zooming and panning.

From the user’s perspective, designing customizable workflows can be a daunting prospect due to the complexity of inputs for bioinformatics applications. Help information is displayed for each application linked to the University of Miami’s online bioinformatics information services portal (<http://bio.ccs.miami.edu/ibis>) and an automatic syntax checker ensures that application inputs are sufficient and correct. In addition, each application object contains methods that automatically derive its input arguments from the resources and outputs of preceding applications in the workflow. In the web interface, each application in a workflow is represented as a block with required input parameters and outputs (Figure 5).



Figure 5. Workflows are created by dragging applications from the left toolbar into the center pane. Users can choose to use the default automated argument settings or manually configure applications in the right pane.

For advanced users, Aqwa provides a file manager which allows direct access and drag ‘n drop manipulation of the file system of each workflow (Figure 6), rather than solely an abstract data interface such as in the BCJ. This is an optional function, which provides the benefits of fine control of workflows for power users without the burden of increased complexity for less advanced users.



Figure 6. Aqwa’s file manager allows drag ‘n drop manipulation of workflow file systems.

The various activities involved in Aqwa’s development are detailed in the following summarized software design document.

**Architectural design**

Guided by the software engineering strategy of *separation of concerns* for ease of development and maintenance, the system is essentially embodied by the Model-View-Controller design pattern. The Model represents the data objects of the system (i.e., database tables on the server), the View is the display the user sees representing the Model and the Controller takes care of processing user-initiated events like mouse clicks and key presses to change the Model and the representation of the Model in the View (Figure 7). The Controller is composed of several MVC components that interact with a single Model to accomplish specific tasks such as authentication, project management and workflow. By adhering to a loose coupling between the system components, this component-based approach promotes ease of maintenance and reuse of architectural elements. Each controller component is represented at the file level by a plugin directory containing Javascript class files, templates and other resources required to perform one particular function. The core files of the Controller, found in the plugins/core directory, complete the task of loading all of the required modules, which can then load the rest of the system. This involves, for each plugin:

1. Locating the plugin descriptor files 'info.json' in the plugin directory
2. Determining whether the requirements for the plugin are already loaded
3. Loading the plugin
4. Registering the plugin and its version number

Once a plugin is loaded it may request information from the backend using asynchronous dojo.xhrGet/xhrPost requests. The additional resources included in each plugin (templates, css files, json files, images, etc.) represent the 'View' portion of the application.

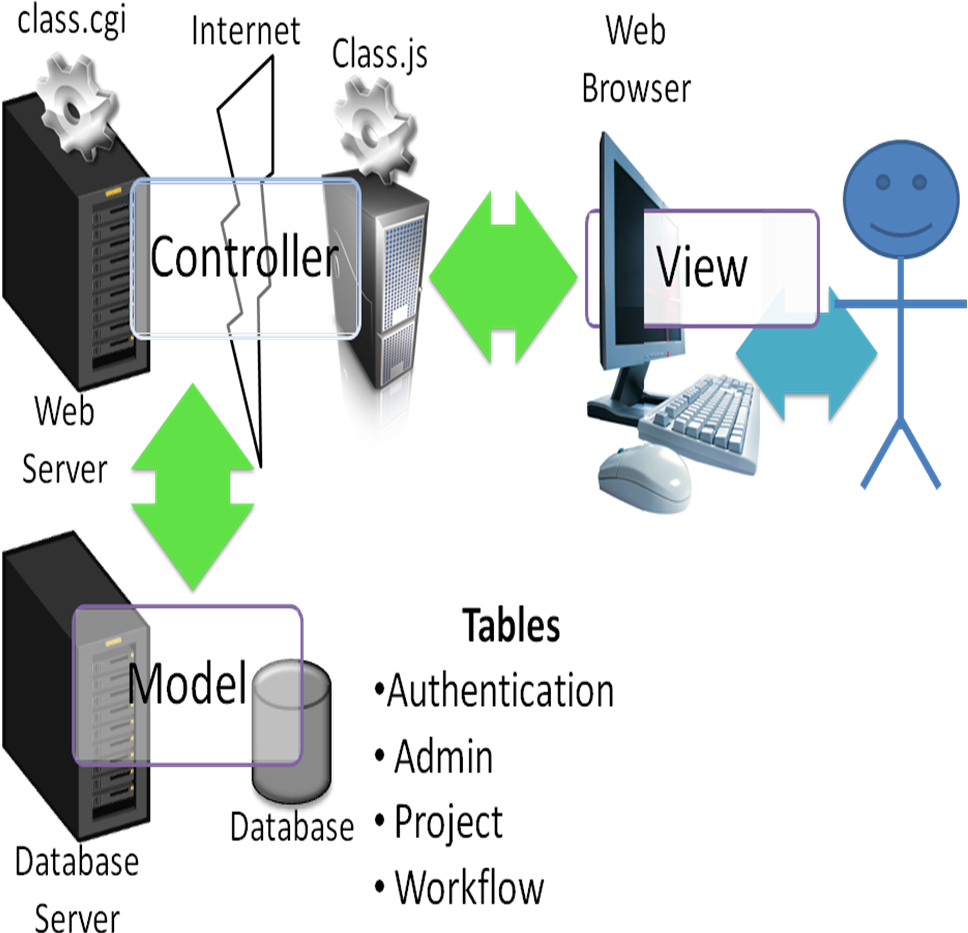


Figure 7. Breakdown of Model-View-Controller architectural design. The user interacts with the View, sending AJAX messages to the Web server via the particular MVC component (e.g., ‘Workflow.js’ represents the browser-side Controller component and it communicates with the server-side Controller component ‘workflow.cgi’). The Controller effects any necessary transactions with the Model (i.e., changes in database tables) and changes the state of the View accordingly.

**User Interface (View)**

The central metaphor of the user interface is that of a project folder containing one or more workflows (customized or standard). Each workflow contains one or more applications and their input data. Each application has inputs, resources (which are similar to inputs but are not specified as command line arguments) and outputs (Figure 4). The user interface is a web application implemented in Javascript and based on the Dojo Javascript toolkit (<http://www.dojotoolkit.org>) which contains a rich assortment of web application components and utilities. Implementing the View as a web interface also allows for the eventual incorporation of three-dimensional graphical viewers directly inside the View for molecular dynamics simulation outputs. In addition, the user will be able to customize the View by specifying the default viewer for a particular data type. When the user elects to view a file with the specified data type, a copy of the file is downloaded to the client machine and the associated application is invoked with this input.

**Data design (Model)**

The database tables residing on the server and their relationships make up the Model. These include tables for user authentication, project ownership and permissions, workflow content and execution status, and data provenance. Conversion of data into a unified data model in a standard format, such as in Pegasys, is not part of the functional requirements of the system, although the system will present different export and processing options for data based on its type. Individual applications are represented by JSON, similar to the resource definition XML files commonly used in systems such as GPIPE/PISE and Pegasys. The data design is distinct from the API (Application Programming Interface), which is presented as a web service with relatively stable inputs and outputs.

**Low-level design (Controller)**

The Controller, or so-called ‘business logic’ of the application, which interacts with the View and Model is composed of Javascript classes on the client and corresponding Perl modules on the server. The choice of Perl as the backend programming language was influenced by its large user community, abundance of bioinformatics tools such as BioPerl (http://www.bioperl.org), ease of accessibility for novice programmers and wide range of applications from text manipulation to system administration.

The system will use existing or novel syntactic structures and algebraic operators for describing bioinformatics workflows to achieve fully customizable workflows with forks, conditional statements and loops, and construct treelike workflows composed of multiple workflows linked together with logical commands. JSON (JavaScript Object Notation) is used as the data interchange format because, unlike XML, JSON-encoded objects need no additional parsing to define them at the object level, which allows them to be easily chained together in complex workflows. Following the evaluation of existing workflow control methods and their applicability to the Aqwa system, a core set of operators will be implemented in the release version of Aqwa.

The first two versions of Aqwa will support execution on the local server or on a cluster using the PBS (Portable Batch System) queue scheduler. An abstraction layer separates the workflows and the execution method for ease of extension in later versions to execution on a grid and eventually as web service requests. The latter two execution modes will require a more sophisticated pipe component and conditional operator due to the need to check for service availability before jobs can be dispatched for execution on a remote host. Failure management of jobs executed on a local server or cluster is accomplished by job monitoring scripts and wrappers to distinguish between error and normal exit modes. Failure management of jobs executed as web services or on a grid may require the development of additional tools.

The project uses an iterative/incremental development model, starting with a simple implementation of the basic functions (workflow, data management, reporting and genome view) and iteratively enhancing at each build with design modifications and new functional capabilities until the release version. Each iteration includes an examination of both the functional and quality requirements, the latter defined by user feedback and user testing (17) and interaction with other stakeholders. The alpha version will be used by a selected group of ‘power users’ within the University of Miami and the source code will be freely available for academic users. For the beta release, the user group will be expanded to all NextGen data users within the University of Miami and registered external users. Extensive user testing will be carried out at regular intervals and the results of the tests will be used to inform any additions or changes to the system’s functional requirements.

**Timeline and Milestones**

The project timeline and milestones are shown in Figure 8, with projected stage start and end dates in months. The project has four stages – the inception, elaboration, construction and transition stages. The inception stage (-3 to 0 months) has already been completed. This stage included identifying the stakeholders, determining the project requirements and implementation options, and conducting preliminary studies for specific aims 1 and 2. This stage also involved the preparation of this research plan and a detailed timeline.

In the elaboration stage (0 to 6 months) we will elaborate Aqwa’s project design and architecture, and develop prototype assembly, genomic variation and expression analysis workflows. The majority of the work for specific aim 1 will be done in this stage and work will begin in earnest on specific aim 2. Prototypes of the developing workflows will be implemented in the Aqwa alpha version (Milestone 1) at the midpoint of this stage. Significant risks will also be identified and risk mitigation procedures put in place and the first of six, four-month user testing cycles will begin in this stage.

During the one-year construction stage, the software design will be further refined through interaction with testers into the Aqwa beta version (Milestone 2). Specific Aim 1 will be completed at the midpoint of the construction stage and specific aim 2 will be completed at the end of the stage. Three user testing cycles will be carried out and the bulk of the software implementation work will be accomplished during this stage.

The transition stage will begin with the Aqwa gamma release (Milestone 3), which will incorporate all of the analysis workflows in developed in specific aims 1 and 2 and will be thus complete enough to transition to the user community as the release version. The goal of this stage is to ensure that the requirements have been met to the satisfaction of the stakeholders. During this stage, all remaining user and developer documentation will be completed and any defects will be identified and corrected.

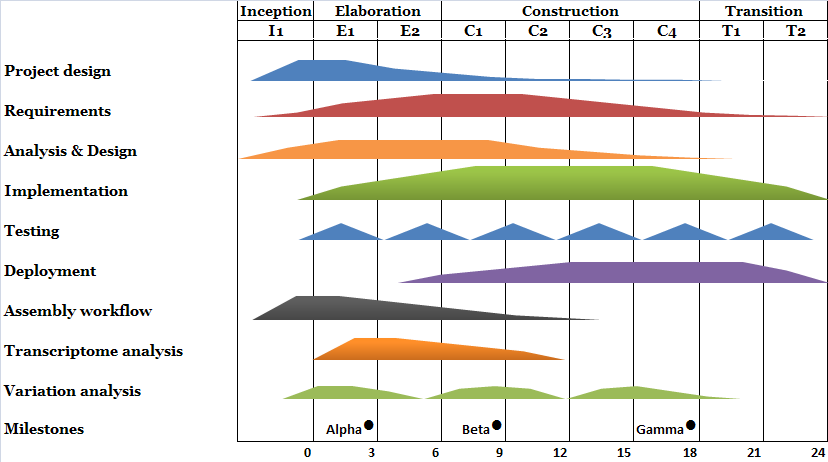


Figure 8. Project timeline and milestones by month from project start. Following a 3-month project Inception stage (currently underway), user testing will take place at regular intervals during the 6-month Elaboration, 1-year Construction and 6-month Transition stages.

As part of the inception stage, preliminary data for all three specific aims have been obtained:

**Specific Aim 1: Develop and implement an optimized NextGen assembly workflow**

We investigated the performance of the following short read assembly tools: Eland (GAPipeline v0.30, Illumina), Velvet v0.7.16, Mira v2.9.25, Genomics Workbench (CLC Bio) v1.2, Seqman NGen (DNAStar) 1.1, NextGene (Soft Genetics) 1.0 and MAQ v 0.6.8. The input data sets were: human mtDNA, human whole-genome mRNA, E. coli DNA, Herpes simplex and bacteriophage PhiX. The assemblers were assessed for assembly speed and capacity in terms of the maximum number of reads that can be effectively assembled using relatively high-end computer hardware. Assemblies produced by the different programs were compared and a consensus determined based on read identity and divergence from the relevant reference sequence. There were significant differences in the sequence capacities of the different reference alignment and *de novo* short read assembly tools, and the reference aligner outputs showed significant differences in reads matching against the reference sequence in particular locations. For human data, Seqman NGen, Genomics Workbench and NextGen showed better performance in terms of the number of reads assembled however this may result in less accurate contigs.

**Specific Aim 2: Develop and implement NextGen genomic variation and expression analysis workflows**

The Nimblegen exome capture of an eight-person pedigree sample set has been successfully concluded and sequencing of up to 15-fold average coverage is currently underway. Based on the initial shallow sequencing data, the framework of a SNP validation workflow has been completed, including filtering based on read quality, read depth and presence in dbSNP followed by the incorporation of pedigree information such as the presence/absence of non-SNP calls at identical chromosomal locations in parent-offspring trios. We will shortly evaluate the use of an expectation-maximization approach to predicting heterozygotes and homozygotes. In the final stage of this experiment, we will validate our approach with a sample of 1,000 individuals on a smaller subset of the captured genomic regions.

**Specific Aim 3: Develop a NextGen workflow and visualization tool**A working prototype of Aqwa (pre-alpha) has been deployed, providing 80% of the core functionality of the project Aqwa alpha version. Power users are currently being recruited for the initial testing phase to assess the usability of the interface and the suitability of the current requirements specification.