Resource Project Research Plan

Specific Aims

Aim 1. Develop, maintain, and extend software for the display and analysis of genomics resources.

Aim 2. Build and maintain genome browsers and related resources for species of biomedical interest.

Aim 3. Integrate data from the scientific community that help interpret the functions of various genome regions.

Research Strategy

Progress report

From its original focus on the early drafts of the human genome, the Genome Browser database now offers genomic data for 96 organisms, many with multiple assemblies. Fewer than 3 months after the Genome Reference Consortium (GRC) produced the latest GRCh38 human assembly in December 2013, we created and released the equivalent hg38 assembly browser that included 261 alternate-locus scaffolds, centromeres, and mitochondrial genome, as well as several annotation tracks. In addition to the new hg38 assembly, we also released 65 new or updated assembly browsers for other organisms, including a quickly produced browser for the Broad Institute's Jun. 2014 assembly of the Ebola virus in response to the 2014 African outbreak.

The human and mouse genomes are the most heavily annotated of the entire genome set. During the current grant cycle to date, we have added 92 new tracks to the latest human assemblies, updated 29 existing tracks, and "lifted" 7 tracks from the hg19 assembly to hg38 coordinates to increase the robustness of the annotation set on the hg38 assembly. The latest human assemblies include a broad range of annotations: conservation and evolutionary comparisons, gene models, regulation, expression, epigenetics and tissue differentiation, variation, phenotype, literature, and disease associations. During this same time period, the mouse browsers acquired 30 new tracks and 8 updates. We also added 65 new tracks to browsers on species other than mouse and human. We have implemented efficiency mechanisms that automatically update tracks based on external sources as soon as new data is released, thus reducing staff time and funding needed to produce these updates. We currently apply this automation to 12 annotation tracks. We incorporate the GRCh38 and GRCm38 patches, haplotypes, and alternate sequence from NCBI as they become available, displayed as a separate track aligned with the reference human or mouse assembly. In addition, the browser serves as a platform for a host of select tracks from the ENCODE and GTEx projects, which are funded by other mechanisms. Table 1 summarizes genomic data highlights during the period from 2012 to the present.

Genome browsers and comparative genomics
New GRCh38/hg38 assembly browser with centromere annotations, several annotation tracks, and three conservation tracks: 7-vertebrates, 20-primates, and 100-species.
65 new or updated assembly genome browsers.
Ebola virus assembly browser and portal.
Ten sets of comparative tracks: Human – 7 species, 20 species, 100 species (2 sets); Neandertal – 20 species; Mouse – 60 species; Rat – 13 species; Tarsier – 20 species; C. elegans – 26 species; Ebola virus – 160 sequences.
Assembly hubs to allow users to view their own genome assemblies in our browser.
Alignment tools bake-off: multiz pipeline vs. Cactus.
Snake display for viewing Cactus multiple alignments.
Data imported from researchers and consortia
Automated update process for many frequently updated tracks.
Streamlined GenBank process.
ENCODE 2 data for hg19 and mm9. Selected ENCODE data for hg38.

Addition of more than 45 public track hubs (including 118 species and strains not hosted locally).
Genotype-Tissue Expression (GTEx) v6 data and new displays.
Addition of data from Peptide Atlas database.
Display of gene haplotypes from the 1000 Genomes data as amino acid or DNA variants.
Import of manually curated data from Protein Families (PFam) database.
Human genetic polymorphism and disease association
Notable new and updated tracks: GWAS Catalog, SNPs, DGV, Genome Variants, Segmental Duplications, 1000 Genomes Phase 3, EVS, ExAC, Coriell CNVs, DECIPHER, OMIM, ClinGen, ClinVar, GeneReviews, HGMD, Lens Patents, LOVD, UniProt Variants.
Two types of publications tracks: sequences found in publications and mapped to genomes using BLAT; identifiers found in publications.
Sequences found searching worldwide patent applications, then mapped to several genomes using BLAT.
Sequences found from searching 40 billion web pages, then mapped to several assemblies using BLAT.
Imported data from Catalogue Of Somatic Mutations In Cancer (COSMIC).
Gene models for human and model organism genomes
UCSC Genes sets for human and mouse.
GENCODE genes models as the default gene set in GRCh38/hg38 and available gene set on hg38 and mm10.
Regular updates of CCDS, Ensembl, Augustus, GENCODE, RefSeq, tRNA, and TransMap annotations on human genome and other organisms.
Other gene sets added or updated on human, mouse or model organisms: GeneID, IKMC, SGP evidence-based gene predictions.

Table 1. Significant data releases and updates during the 2012-present grant period.

We collaborate with many high-throughput data producers and computational biologists worldwide to incorporate their data. Our partner groups and consortia include the ENCODE Consortium, the Genotype-Tissue Expression (GTEx), the California Institute for Regenerative Medicine (CIRM), the Roadmap Epigenomics Mapping Consortium, the 1000 Genomes Project, and the Clinical Proteomic Tumor Analysis Consortium (CPTAC).

Major additions and enhancements to the Genome Browser tools and underlying data support during the current grant period include the following. See Appendix 1 for a complete list of resources produced by this project over its lifetime.

- Variant Annotation Integrator (VAI) tool, which annotates variant calls with predicted functional effects on protein-coding genes and regulatory regions. Given a set of variants uploaded as a custom track, the VAI returns the predicted functional effect (e.g., synonymous, missense, frameshift, intronic) for each variant.
- Assembly data hubs, an extension of the track data hub functionality that allows users to annotate sequences for which we do not host an assembly database.
- Data Integrator tool, which provides a simple, flexible interface for combining data from up to 5 tracks in the Genome Browser database, as well as custom tracks and hub tracks.
- Genome Browser in a Box (GBiB), a virtual machine version of the Genome Browser with a small memory footprint that is easily installed on a user's own laptop. GBiB includes the UCSC Genome Browser software, all required utilities, and a basic set of human genome annotation data. Additional annotation data can be loaded on demand from UCSC via the Internet or can be downloaded to the local machine for faster access. Individuals can also view and manipulate their own local data in the Genome Browser through the use of custom tracks, thus allowing them to view private or very large data files on their own hard disk without the need to upload the files to the UCSC server.
- Genome Browser in the Cloud (GBiC), a product that sets up the UCSC Genome Browser on a user's virtual machine or native Linux server. The virtual machine can run in a data center of one of the many

cloud service providers. GBiC is useful for individuals who are unable to use our GBiB virtual machine solution due to technical limitations, or who have already migrated IT services to external cloud providers.

- Various security updates to prevent malicious attacks on the Browser website.
- Performance enhancements to speed up the track display, especially when viewing large chromosomal regions.
- Addition of full mirror sites in Europe and Asia to address the latency experienced by users in those geographical locations.

A notable recent addition to the Genome Browser, the multi-region display configuration, allows the user to condense the view to show only specific regions, such as exons, genes, or user-defined BED regions, while hiding the extraneous intergenic, intronic or other unwanted regions from view, thus extending the usefulness of the Genome Browser for displaying expression, proteomic, and exome sequencing results (see Fig. 1 in Overall component). For human assemblies NCBI35 (hg17) and later, the multi-region view also supports the replacement of a section of the reference genome with an alternate haplotype chromosome, allowing the user to visualize the haplotype in the general context of the reference chromosome.

In the Specific Aims below, we describe in detail our proposed intentions for extending, enhancing and maintaining the Genome Browser software tools and data sets during the next grant renewal period. See the Management, Dissemination and Training Core component for a discussion of our plans for training people to use the resource.

Aim 1. Develop, maintain, and extend software for display and analysis of genomics resources

Significance

As costs decrease, genome sequencing has become an important diagnostic tool in understanding the diseases of many pediatric and oncology patients. We expect this trend to accelerate rapidly over the course of the next five years, and reasonably anticipate that millions of patients will be sequenced. Sequence interpretation is not a simple task. It is essential to have software tools that distill the data, place it in the context of what is known, and present the results in both a visual form as well as a format that can be used for further computation. These tools and their underlying databases must respond quickly even under conditions involving huge data sets and a large number of simultaneous users. They must support data views that range from individual bases up through entire genomes, and they must meet reasonable security expectations.

The Genome Browser software has met these needs for the last sixteen years as we continually adapt to new data types, larger volumes of data and users, and new computing platforms and displays. In addition to the software visible on the Genome Browser website, our many tools with Unix command-line interfaces play a key role in the genomic analysis pipelines at institutions throughout the world. During the next grant period, we plan to expend more effort on this aim than any other in this proposal.

Innovation

As discussed in the Overall component, innovation must be balanced with discretion in a mature project such as the Genome Browser, to maintain the usefulness of the tools as science and technology advance, while at the same time preserving the reliability and stability of the tools for our users. Our proposed software innovations address six aspects of this aim: a) support visualizations to interpret personal genomes, b) develop displays that aggregate increasing volumes of data, c) support chromatin conformation capture data and other non-local interactions, d) improve search capabilities, e) build a genome browser version targeted at mobile devices such as tablets and smartphones, and f) build browsers and displays for single cell data. We also plan numerous smaller improvements.

a) Support visualizations to interpret personal genomes

As sequencing capability becomes ubiquitous and more information is available for interpreting it, medical diagnosis in general and, in particular, diagnoses for birth defects and cancer are moving from microarrays to whole-exome and whole-genome sequencing. The visualization and integration tools of the Genome Browser are an ideal venue for researchers and clinicians who want to share their own data and access public

aggregate data to assist in interpreting their results. We plan three main Genome Browser enhancements to better support the display of personal genomes.

First, when visualizing the sequence of a single individual, the display will include separate haplotype lines representing each copy of the genome (typically two copies, except for the male sex chromosome or in areas of copy number variation). As long-read (nanopore) technology improves, phasing information will become increasingly available, and it will be used to group SNPs on haplotype lines. Non-reference SNP alleles will be marked on the appropriate haplotype line to make heterozygosity more apparent. We will indicate when phasing information is not available, or where breaks in phasing occur. Copy number variation will be readily apparent as an increased or decreased numbers of lines.

Second, we will develop a trio display that shows genomes from a child and parents (Fig. 1), making it easier to spot *de novo* mutations and the parent of origin in inherited but rare mutations. This display will build on top of the single individual display described above, showing the parents as nearby tracks, and will handle cases where data is available only from a single parent. When the trio display functionality is stable, we plan to extend it to encompass larger family groups as well, showing the family tree relationships to the left of the data tracks.

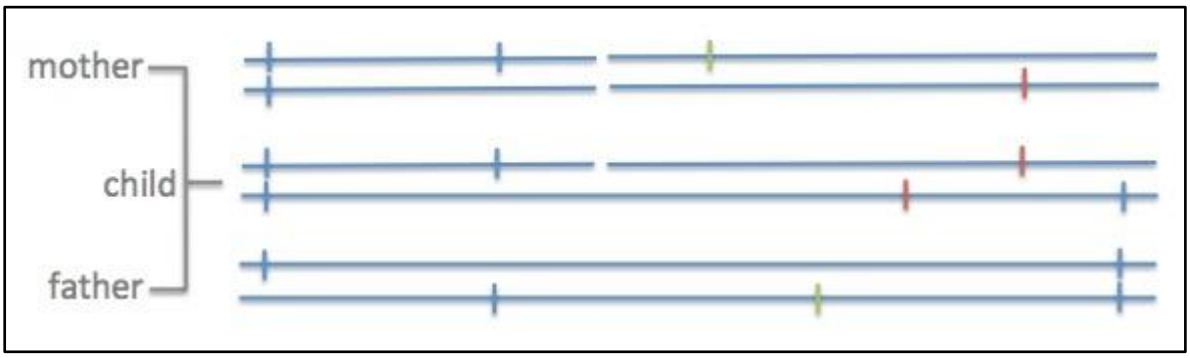

Figure 1. Proposed data display for trio data. In this case the display facilitates the identification of a likely *de novo* mutation on the lower haplotype line of the child and a small deletion shared between mother and child.

Third, we will develop a view that shows an individual tumor display alongside the somatic tissue display, highlighting copy number variation, loss of heterozygosity, and sites of chromosomal rearrangement in the tumor. This will be similar to the trio display, but with two groupings (somatic and tumor) rather than three. As with the personal display, copy number variations will be shown as an increased or decreased number of lines relative to the normal diploid two lines. Chromosomal fusions and breaks will be indicated with breaks in the haplotype line and glyphs at the site of the breaks.

All personal genome displays will support exome sequencing when used in conjunction with our new multi-region exon-only display (Overall component, Fig. 1). We plan to add support for gene-oriented views of personal genomes in the Gene Sorter, allowing users to make lists of disrupted genes, filtered and sorted on various criteria. We will also support personal genomes through our planned enhancements to the Variant Annotation Integrator tool, e.g., allowing the successive removal of shared variants from cohorts of controls and affected individuals.

Over the past four years, we have taken several measures to ensure a reasonable level of security and authentication for our website and tools. Because the bulk of information available on the Genome Browser website has typically resided in the public domain, our security efforts have focused primarily on ensuring that the web servers and the data behind them are not compromised by hacking efforts, and that users can expect a reasonable level of security when viewing private data through sessions, Genome Browser in a Box, or browser mirrors. Moving forward, we plan to support the privacy of personal genomic data in three ways: by continuing to test and increase the security of data that is uploaded by users to our site, by continuing to make it easier to set up mirrors of our site behind medical firewalls, and by developing a method to restrict track hub access to authorized users.

b) Develop displays that aggregate increasing volumes of data

Several of our website improvements in recent years have been driven by the need to accommodate the sheer amount of data resulting from the success of modern high-throughput sequencing methods. We plan three main site enhancements to handle increasing data volumes. First, we will increase the speed of our existing code where possible to minimize performance impacts when handling larger data sets. Second, we will continue to develop methods that efficiently compute and display summary properties, such as clusters and counts of overlapping features for data sets made up of discrete entities (e.g., SNPs and RNA alignments), and min, max, mean, and quartile displays for data sets with numerical values. Third, we plan to develop methods to interactively select subsets of large data sets based on both the metadata properties (such as age, sex, disease, organ, or tissue) and the properties of the data itself (such as selecting samples based on the expression level of a gene of interest, or samples similar or dissimilar to a particular sample).

c) Support chromatin conformation capture data and other non-local interactions

Chromatin cross-linking techniques (chromatin conformation capture, 3C and 5C) provide information about interactions between genomic regions brought into proximity by higher-order folding and packing in the nucleus. These interactions have been difficult to represent in the linear arrangement of the Genome Browser coordinate system. We plan to develop methods that allow the side-by-side display of regions located in different areas of the genome. In particular this will enable the same-screen visualization of regions that are in close proximity in three-dimensional space (as determined by chromatin conformation capture type experiments), but not necessarily close to each other or even on the same chromosome in a two-dimensional representation. The details page associated with an item in the conformation capture track will show the coordinates and strength of the interactions between the regions, and will link to a browser window with multi-region mode enabled to showcase these regions.

We also plan to utilize the multi-region view mode to effectively visualize in a single display sets of genes related in a number of ways, such as gene families, genes in a pathway, and genes that are part of a multi-protein complex. We will link to these multi-region views from the appropriate sections of the gene details pages, and will allow the creation of custom gene sets for this purpose using the Gene Sorter.

d) Improve search capabilities

Access to large amounts of data is useless without a means for narrowing the view to only those data subsets of particular interest. The existing search mechanisms on our website use state-of-the-art technology, allowing searches across multiple words or multiple fields simultaneously and automatically expanding partial words in the query. Unfortunately, this search technology does not extend to much of the information on track hubs.

We have identified three primary areas in which we would like to expand the search capabilities on our site. First, we plan to extend the search of remotely hosted track hub data to encompass gene names and other item names within tracks on a track hub for genome position searches. As we add more metadata to tracks (Aim 1 in Resource Informatics component), we will include it in track searches. Second, we will extend the search capabilities on both local and remotely hosted data by adding synonym tables, which will be particularly useful for genes that are known by multiple names. We will also accommodate slight misspellings in search terms by looking for minor variants of the words. Third, we would like to enhance our track search mechanism for finding relevant tracks in both locally hosted and hub databases.

e) Build a genome browser targeted at mobile devices

Mobile devices such as smartphones and tablets are being used for a growing number of tasks previously performed on a computer or laptop, particularly within the younger demographic of our user base. To accommodate this trend, we plan to build a version of the Genome Browser suitable for the smaller screens and limited keyboard capabilities of these devices. This adaptation will require several innovations, e.g., making the track and position controls on the main graphical display visible only on demand and supporting scrolling and zooming via swipes and two-finger movements. It will require experimentation (described in the Approach section below) to determine the best method of implementation.

f) Build browsers and displays for single cell data

The ability to sequence single cells via microfluidics and droplet barcode methods has been an exciting innovation of recent years. Even micro-dissected tissues are typically a collection of multiple cell types, so that RNA and epigenetic signals represent a mixture. This can hide etiologies of diseases that affect only specific cells, and in general can inhibit our understanding of human biology.

Clustering is a crucial step in understanding single cell data. Tools such as Seurat/t-SNE (Fig. 2) group together cells with similar signals into clusters, and provide lists of genes that are differentially expressed significantly and can be used as markers for the clusters. This process feeds back onto itself: cells can be sorted on certain markers, and cells sharing these markers can be clustered again to create subtypes. We plan to develop interactive tools for browsing through clusters and subclusters of single cells, which we will then apply to selected high-quality, large-scale, published data sets. We will enable our users to visualize their own data sets as well.

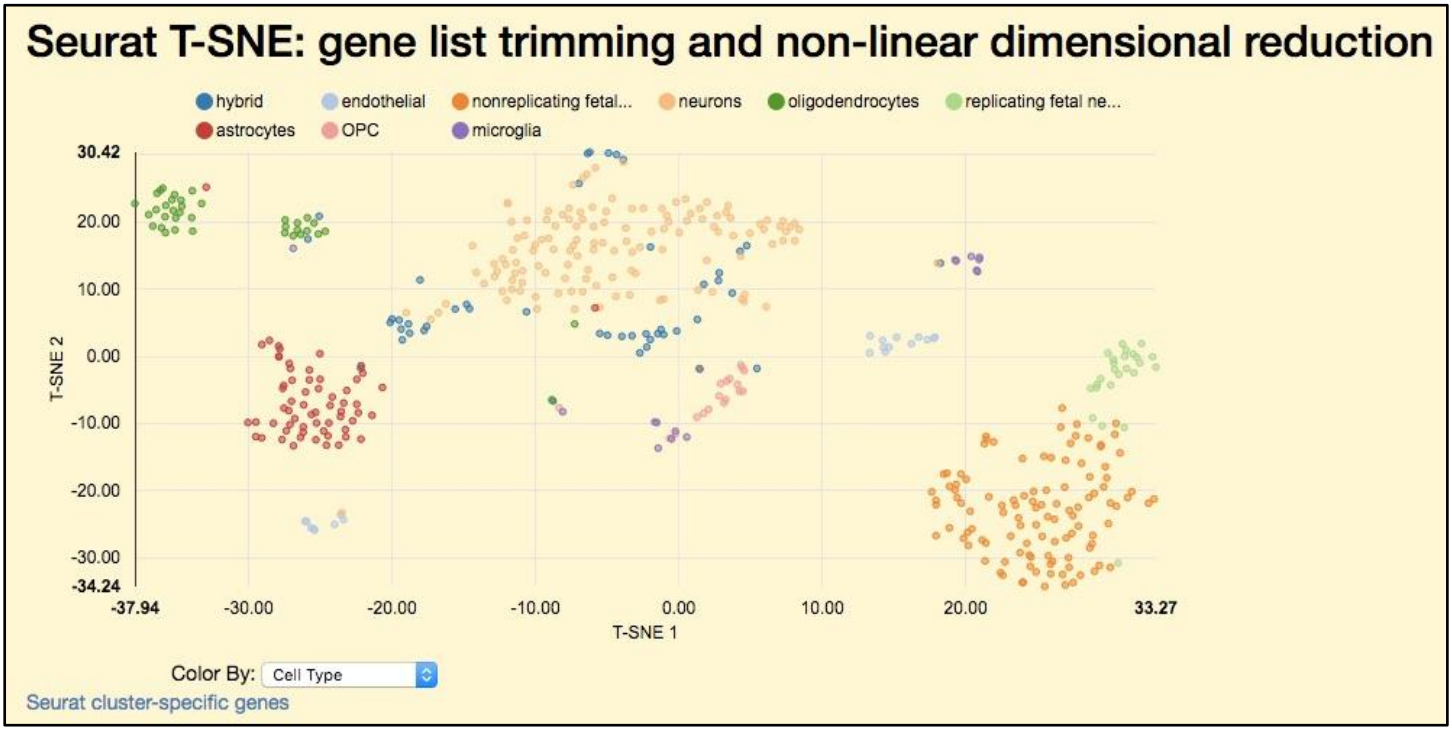

Figure 2. Single cell RNA-seq data from fetal and adult brains (1) filtered by Seurat (2) and clustered by t-SNE (3) as displayed on the CIRM Stem Cell Hub developed at UCSC. Each dot represents a single cell. The dots can be colored by various metadata terms, in this case by the cell type assignments provided by the Quake lab at Stanford University.

Other smaller improvements

We have plans for many other small improvements and new features too numerous to describe in detail in this document. These include a body map display that allows users to view the origin of a sample, an option to BLAT a sequence across all organisms, lollipop displays that visually integrate multiple types of metadata for variants (Fig. 3), improvements and simplifications to the complex configuration mechanisms for composite tracks, support for filtering options on more tracks, and selective updating of the site to a more modern user interface.

Approach

In a mature software base such as ours, much of the infrastructure has already been developed; thus, adding a new feature that fits within an established pattern, such as a new gene track in the Genome Browser, is relatively easy. However, adding new functionality that falls outside these patterns can require the modification of a large amount of existing code, in addition to the new code required to implement the feature itself. Fundamental to the art of software engineering is the ability to separate systems into components that interact only loosely through well-defined interfaces, and to discern when it is better to implement a new feature by extending an existing component or by creating a new one. We plan to implement the innovations listed in

items *a-d* of this aim through incremental expansion of the existing software. The innovations described in items *e* and *f* will require the development of a significant amount of new code coupled only loosely to the existing code base.

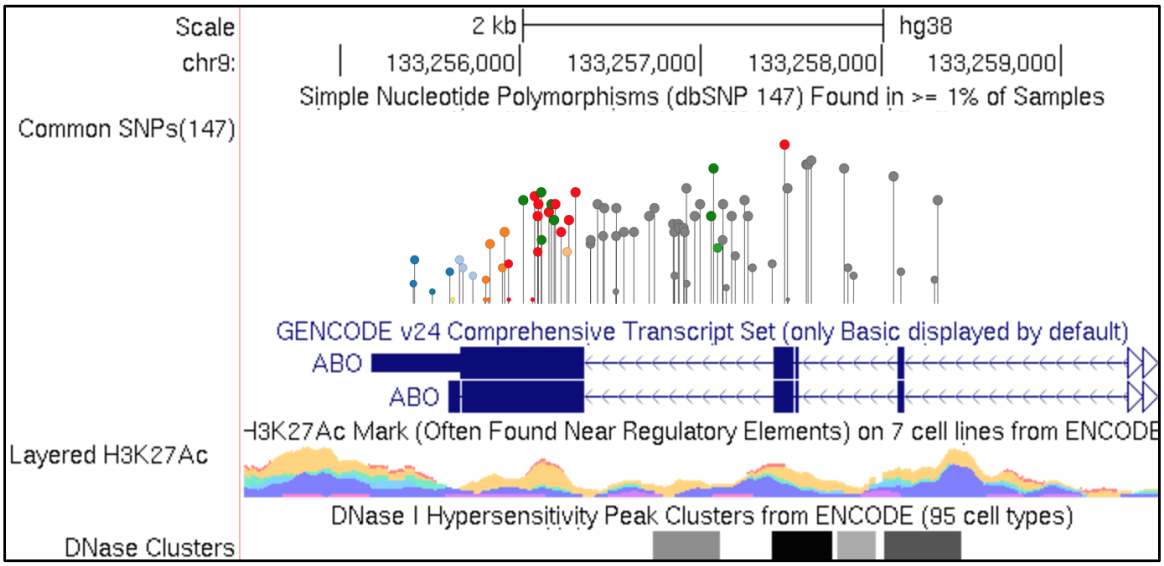

Figure 3. Mockup of a possible lollipop display for SNPs. In this image, the color of the lollipop head corresponds to the type of variant (e.g., red = non-synonymous) and the height of the lollipop stem indicates the minor allele frequency.

a) Support visualizations to interpret personal genomes

Our first step in adding support for this display will be to improve the visualization of individual genomes. We plan to develop a new track type and the corresponding C code module to display the phasing and copy number variation information we currently parse out of Variant Call Format (VCF) files. Next, we will use our existing composite track mechanism to combine tracks from trios or somatic/tumor pairs into a larger unit. Finally, we will replace our generic mechanisms for creating composite tracks with code specific to this task, which will support such enhancements as displaying family trees on the left-hand side of the track graphic and adding optional filters that allow the user to restrict the display to *de novo* mutations in child or tumor data.

b) Develop displays that aggregate increasing volumes of data

Improving the aggregation of large data sets requires efficient data computing and storage, displays that can show more than just a single data point for a region, and methods for selecting which data to aggregate. We have noted that users often wish to see min, max, mean, median, and quartile information over an entire large data set. To implement this efficiently, we will precompute and store these values when loading the data set. To compute statistics on data subsets of interest only to specific users, we will implement efficient C code and perform the minimum calculations needed for the range of data shown in the browser window. The BigWig and BigBed data formats contain information summarized at various view levels, thus these calculations and the data fetched from each individual data track are minimal even for large windows because the browser can fetch data from the appropriate zoom level rather than fetching data for each base. We would like to develop regional zoom-level summaries for other data types as well.

We plan to expand our current composite track mechanism to show an aggregate display of all selected tracks to augment or replace the occasionally long stack of individual tracks. We will allow users to show or hide tracks in bulk based on search criteria or their own preferences, rather than requiring tracks to be turned on or off individually or according to a pre-selected hierarchy.

c) Support chromatin conformation capture data and other nonlocal interactions

Much of the underlying work for the chromatin conformation capture and other non-local displays was completed during the development of the multi-region mode that supports the exon-only display. We plan to streamline the process by which users specify multiple regions, primarily by adding a few new buttons and links to the browser interface that let the user generate a temporary BED file that is passed to the multi-region engine.

d) Improve search capabilities

We plan to improve our search capabilities to include synonyms and accommodate misspelled query terms. Our existing search capabilities are driven largely by a two-level inverted word index, managed by our *ixixx* C module and utilities, that associates a list of words with each item being indexed, such as genes for a gene index, tracks for a track index, hubs for a hub index, etc. To improve the search, we will construct synonym lists that are included with the original words when we build the indices. When a search term does not correspond to any words in the index, we will also look for other words that would match with minor spelling changes to the query word. Since the total number of unique words is relatively small, there are several existing algorithms, similar in nature to DNA alignment algorithms, which can efficiently look for these imperfect matches.

We also plan to extend our support of *ixixx* indexes to track hubs to increase their searchability. The tools for building these indices are mature and fast, and should add little overhead to hub developers who typically already use our tools to generate BigWig and BigBed files. The two-level *ixixx* search is quite efficient when measured by its rate-limiting step, which is the number of round-trip queries to the remote track hub required to find an item. In particular, the higher-level index is small enough to transmit quickly and then remain in local cache, thus only a single continuous section of the lower-level index is searched for each word in the query. Generally an *ixixx* search requires just one round trip per word in the query, plus one additional round trip the first time an index is used (the duration of a round trip is typically less than a quarter of a second).

e) Build a genome browser targeted at mobile devices

To develop the mobile device genome browser, we will rely on several newer programming languages in addition to our existing C code base. The mobile browser will fetch data using JSON/REST-based APIs (described in Aim 1 in the Resource Informatics component). The code to render the data into an image will execute on the mobile device, rather than the server, which will have the benefits of offloading work from our server and allowing greater flexibility in developing more interactive displays, since time-consuming round-trip requests to the server will be eliminated.

In developing code that executes on a mobile device, we will have to consider portability issues, because the code must work on a variety of mobile devices rather than just a single server. Two different development paths are under consideration. The first would require writing code that executes in web browsers on mobile devices, which conform to the Javascript/Canvas/HTML5 standards. On the surface, this has the advantage of requiring only a single set of code for all devices, since theoretically the devices should all follow the same standard. Unfortunately, in practice the interpretation of the standards varies significantly not only across mobile devices, but also among web browsers on the same device. Alternatively, we could write code using the native development kit for each mobile device. Writing separate native code for iOS (Apple products) and Android (most other products) has proven easier for many software developers than writing generic web-based code, and allows the developer to take advantage of capabilities in native apps, such as local file access, that are not available to code executing in web browsers.

Because the development of a mobile device browser is a major long-term investment for us, we plan to take a prototyping approach. We will first develop minimal browser implementations in both HTML5/Canvas/Javascript and iOS that are just capable of displaying BigBed and BigWig data from track hubs, and will subject both versions to our QA process. If, as we suspect, the iOS native code is significantly easier to write and test, we will further develop that prototype, and then port the resulting application to Android. Because many software companies now specialize in efficiently porting applications between iOS and Android, we plan to contract this work to an external team. Unlike our standard software development work, in which engineers must tackle a genomics learning curve to become effective, the porting work should not require this specialized knowledge since the iOS application will serve as an extremely detailed specification for developing the Android version.

f) Building browsers and displays for single cell data

Several of the software engineers in our group, including the PI, have acquired extensive experience with single cell data through their involvement with stem cell genomics projects funded by the California Institute of Regenerative Medicine (CIRM). The web pages developed for this effort (e.g., the graphic shown in Fig. 2)

already interface with the Genome Browser in several ways, though unfortunately the bulk of the CIRM work remains locked behind medical firewalls. We plan to leverage our CIRM work to build single cell browsers on public data accessible through genome.ucsc.edu. The CIRM project, which extends through June 2017, will fund the majority of the single-cell browser development.

Aim 2. Build and maintain genome browsers and related resources for species of biomedical interest

Significance

The genome of an organism is fundamental to its basic biology, and understanding the human genome in particular is fundamental to understanding medicine. Beyond its biomedical significance, the genome also serves as a basic framework for presenting the research efforts of a broad range of scientists in a unified fashion. The integrated view provided by the UCSC Genome Browser allows many of these research findings to be mapped directly to the genome or associated with particular genes in the genome.

The primary focus of our website is on vertebrate genomes, with an emphasis on primates, animals used in scientific research, and animals that help extend coverage of the vertebrate phylogenetic tree, which in turn strengthens the power of the associated multiple genome alignments. We also support the major non-vertebrate model organisms of fly, worm, and yeast, in conjunction with the associated model organism databases. Our “assembly hub” system enables third parties to add their own custom genome sequence and annotation files for browsing at genome.ucsc.edu without requiring the support of the UCSC staff.

The browsers for the human and mouse genomes include a rich variety of annotation tracks, many developed outside of UCSC. Most of the tools on the Genome Browser website can be used to explore a large number of our supported organisms. Of particular interest to our advanced users, the Table Browser and Data Integrator tools provide access to the data underlying the Genome Browser graphical display in a variety of formats (e.g., tab-separated text) that are suitable for further computation. These tools allow the user to logically combine data from multiple tracks, supporting such queries as “find all common SNPs in regions that are DNAse hypersensitive but not near promoters in cardiomyocytes”. The Table Browser also provides convenient access to a well-documented schema describing all the tables in our database and how they relate to one another.

In this aim we will focus on the tracks that are computed locally at UCSC and are available on all, or nearly all, of the 96 genomes we now display, including the GenBank (4), mRNA, and EST alignments; RefSeq (5) and Ensembl (6) curated gene models; Augustus (7) computed gene models; multiple genome alignments and conservation score mappings of human and/or mouse genes to other organisms; self alignments to annotate duplicated regions; and RepeatMasker results for annotating transposons. Annotation tracks generated by external groups are discussed in Aim 3.

Innovation

We intend to make it easier to build browsers and related tools on new genomes, improve tool performance, and maintain tool operability even as the operating systems, databases, and libraries on which they are built evolve over time. We plan to map and compute annotations on patches issued between major releases of the human and mouse genomes to increase the usefulness of the patches to the scientific community. Browser users will be able to view the patches in place on the corresponding reference genome or continue to display the unpatched version of the assembly, the latter of which is of particular value for in-progress projects initiated in the original coordinate system of the major release.

We also plan to evaluate new multiple genome alignment software as it becomes available. Our existing pipeline is based on the lastz, chain, net, and multiz tools. A few years ago we evaluated the multiple alignment and ancestral genome reconstruction software called Cactus (8,9), which is developed by an active UCSC research group with funding external to this grant. Cactus performed well on primates, but showed less sensitivity for more distant species than our current alignment approach; therefore, we decided to continue to use our existing pipeline. Approximately every two years we plan to evaluate other multiple genome alignment software packages by choosing a target set of genomes and making the sequence available to other groups, such as Cactus and Ensembl, that wish to have their software considered. The software will be ranked using a variety of criteria: how well the protein structures are preserved, how well the duplicated regions are covered, how well the orthologs and paralogs are separated, and with spot checks by biological experts in randomly

chosen regions as well as specifically conserved, duplicated, and highly diverging regions. We will switch to a different pipeline if one proves superior to our current methods.

Approach

We plan to continue our relatively conservative approach in executing the work outlined in this aim. The Genome Browser is a CGI script written in C that uses data stored in a MySQL database and in various genomics file formats. The C code produces a web page in HTML format that includes PNG format images and a modest amount of JavaScript, which displays on the user's web browser. The C, MySQL, and file technologies are all very stable. HTML and JavaScript continue to evolve, but at a relatively modest pace compared to the past. We do not anticipate major difficulties in maintaining the operability of the core of the Genome Browser display code. We should be able to improve the website speed by moving to newer, faster hardware and selectively tuning performance in a few C modules that have become bottlenecks.

We will continue to automate the process of building the databases and files needed to instantiate a Genome Browser on a new assembly. This automation is feasible now that genome assemblies released by NCBI are of uniform format. The bulk of this automation will be performed using Unix shell scripts configured with extensive error-checking to prevent problems due to input errors or transient computer glitches from propagating to the public site.

We also plan to automate the pipeline that recomputes UCSC-generated annotation tracks to incorporate the GRC patches and haplotype additions to the human, mouse, and other GRC-supported reference vertebrate genomes. The recomputation will be done primarily by rerunning the scripts that produced the initial annotations on the new larger data set that includes the patches. Fortunately our "near best in genome" mapping approach allows mRNA alignments and other annotations to map to more than one place, a necessity for coping with the ambiguities caused by the many near-identical segmental duplications in the human genome. This same approach works with preserving annotations across nearly identical patched and unpatched versions of the sequence, as well as shared sequences within divergent haplotypes.

In addition to recomputing tracks that we map ourselves, we will refine our automated methods for converting third-party annotations to cover new haplotypes and patches through the extension of our "LiftOver" tool. LiftOver functions well in mapping annotations between different human genome assemblies within the framework of the originally released chromosomes, but does not have the logic needed to cope with alternative haplotypes and patches. Adding this logic should be a detailed, but straightforward, task. We also plan to extend the "alternative haplotype" configuration of the multi-region display to let users view patches and new alternative haplotypes in addition to the haplotypes included in the original release of the reference assembly.

We plan to continue to use our existing pipeline to produce multiple alignments of up to a few hundred vertebrate species. This is the most computationally intensive aspect of our project. We plan to keep the web and database servers, which are in continuous use, on computers based at UCSC. However, we plan to move the computation of the multiple alignments, which tends to come in bursts, into the cloud rather than maintaining sufficient capacity on our own computer clusters to produce a multiple alignment in a reasonable amount of time.

Aim 3. Integrate data from the scientific community that help interpret the functions of various genome regions

Significance

While the annotation tracks computed at UCSC (Aim 2) provide a useful foundation, they are enriched tremendously by data integrated from other projects and from the scientific community at large. On the human genome in particular, the vast majority of our annotation data sets are contributed by external groups.

We import significant information (e.g., gene models) from the model organism databases and link back to these databases from browsers we build on the respective model organisms, as well as from the orthology links on our human genome gene pages. Having the model organisms available on the UCSC Genome Browser increases the visibility of the work on these organisms, and also provides a common informatics environment across species for researchers working on multiple organisms.

Because different scientists tend to describe the same observation in different ways, it can be difficult to analyze large databases unless the descriptive vocabularies are controlled in some way. We therefore promote and use the work of standards groups working on controlled vocabularies. In particular the Gene Ontology (10) vocabularies are used extensively in our site – on the gene description pages, in the filters and columns of the Gene Sorter, and in our database tables. We also use Sequence Ontology (11) terms in our GFF output and internal database columns, where appropriate.

Though we expend considerable effort importing third-party data, the scientific community collectively produces far more useful genomics information than we can wrangle into our databases. We have two mechanisms by which external users can add data to the browser without the involvement of UCSC staff: custom tracks and track hubs. Custom tracks are useful for smaller data sets and typically employed for private use, although they can be shared. Track hubs are better suited for larger, shared data sets. Individuals or groups can request that we add their track hubs to a public list we maintain. We perform a light quality assurance check on these hubs, which predominantly involves running established compliance verification scripts and encouraging good documentation. See Aim 1 in the Resource Informatics component for more information on track hubs.

Innovation

We plan to continue integrating new data releases from projects that we currently support, as well as incorporating new data from selected projects and papers recommended to us by our users, our scientific advisory board, and our funding agency. When possible we will automate updates for new releases. We plan to work closely with the ENCODE and GTEx consortia in particular to ensure that their rich data sets are well represented in the Genome Browser. We will display and compute summary tracks and statistics for these two projects that will be highly visible in the browser. We will import the “catalog” tracks produced by the ENCODE Data Analysis Center (DAC) and apply summary tools we developed in our role as the ENCODE 2 Data Coordination Center (DCC). We plan to work with the ENCODE 3 and GTEx DCCs, as well as DCCs associated with other major projects, to help them learn how to use track hubs that can be updated with the detailed underlying data as it becomes available. We will to publicize these track hubs on our public hubs list.

In the upcoming grant cycle we anticipate a sharp increase in the amount of human variation data and human genotype/phenotype correlations. We will track the work of groups producing and collecting such data, including the 100,000 Genomes Project (12), SNPedia (13), OMIM (14), ClinVar (15) and dbSNP (16,17). We will integrate de-identified data into our public browser, and will coordinate with dbGAP (18) to allow their authorized users to access the same specific, identifiable, or otherwise private data sets in a secure manner on our site as well.

Approach

We term the process of integrating new data into the browser “wrangling.” A good wrangler has the social skills to communicate well with collaborators in external labs or consortia, the scientific background to understand and describe the data, and the computer skills to organize and, if necessary, reformat the data for our systems. The wrangling process is described in more detail in the second aim of the Resource Informatics component.

Though each scientific data set is unique, we have developed a large variety of tools that help the wrangling process: those that convert among many common and less common genomics file formats; those used for analyzing and reformatting XML files, SQL databases, spreadsheets, JSON, and a wide variety of text formats; and those that check fields against controlled vocabularies. Still, many aspects of wrangling resist automation and thus will continue to require a hands-on approach, such as interactions with the data producers, handling quirks in the data set, reading scientific papers, and producing descriptive text with a minimum of technical jargon that helps a broad section of our user base understand the data.

Milestones

Aim	Description of task	Grant Year				
		1	2	3	4	5
1	Improve search capabilities.					
1	Create displays for chromatin conformation capture, etc.					
1	Create visualizations for personal genomics data.					
1	Develop displays that aggregate large sets of data.					
1	Create a lollipop display for variant data.					
1	Phased trio and somatic vs. cancer personal genome tracks.					
1	Build Genome Browser app for tablet and mobile devices.					
1	Support data from single cell sequencing.					
2	Build genome browsers for vertebrate and model organisms.					
2	Compute conservation tracks on new and existing species.					
2	Build and display internally computed annotation tracks.					
2	Continual performance tuning to ensure quick response times.					
2	Map tracks to patches in human and mouse assemblies.					
2	Allow patches to be swapped into the graphical display.					
2	Extend LiftOver tool to map to alternative haplotypes and patches.					
2	Compare multiple genome alignment software pipelines.					
2	Automate mapping and recomputation of tracks onto patches.					
3	Build gene sets from model organism databases.					
3	Integrate data from significant papers.					
3	Integrate data from the GTEx project.					
3	Build summary tracks from selected ENCODE 3 data sets.					
3	Allow secure access to data sets for dbGAP authorized users.					
3	Integrate data from other consortia (TBD by SAB and users).					
3	Integrate data from the ENCODE 4 encyclopedia.					

Table 2. Approximate timing and level of effort of specific tasks within this Research Project component. Darker colors indicate greater effort over the given year.