PolyLens: Software for Map-based Visualization and Analysis of Genome-scale Polymorphism Data

Ryhan Pathan
Department of Electrical Engineering
and Computer Science
University of Tennessee Knoxville
Knoxville, Tennessee 37996
Email: rpathan@utk.edu

***Abstract*—Currently available software for visualizing and interpreting large scale genomic sequence information in a geographic context is less than ideal because outputs are frequently not presented in an easily interpretable and user-friendly format. PolyLens, a map-based visualization tool, attempts to address this issue. Written in Java and R, it provides a self-contained and portable means for processing population genomic data, visualizing geographical distribution of lineages, and displaying allele distribution patterns. This paper details the implementation of the software, and, using a test-case genome-scale population data set consisting of 32 individuals of the species *Drosophila melanogaster* (common fruit fly) sampled from Africa and France, demonstrates its potential uses.**

I. Introduction

When working with large scale genomic sequence data, drawing meaningful conclusions can be a difficult and tedious task. The goal of the PolyLens project is to provide a portable and efficient tool to facilitate phylogeographical analysis and discovery, particularly within a spatial context. It achieves this goal by providing a highly interactive user interface that allows the user to generate queries on the genetic relatedness between organisms in a population. The results of these queries are then visualized in an easily interpreted map display, as well as in a compact chart-like display.

The inspiration for the tool stems primarily from FutureLens [1], a software for text visualization and analysis, shown in Figure 1. For a collection of documents each written by an author with a specific date, FutureLens allows the user to explore frequently occurring terms or patterns among documents. Connections between these frequent terms and the dates at which they appear in the documents can quickly be visualized and investigated. When one or a combination of terms is investigated, a graph of the percentage of documents containing the term versus time is also shown to the user. Also serving as inspiration was PhyloLens, an earlier prototype for PolyLens that provided proof of concept but lacked the general usability and robustness required to be a widely applicable tool. A screenshot of this prototype is provided in Figure 2.

II. Phylogeographic Data

Before discussing the implementation details of the project, it is first necessary to understand the data on which it operates. PolyLens was designed to work with population genomic data consisting of the following components: sample genomic sequences, location list, gene ID list, and stop list. Each of these are discussed briefly below.

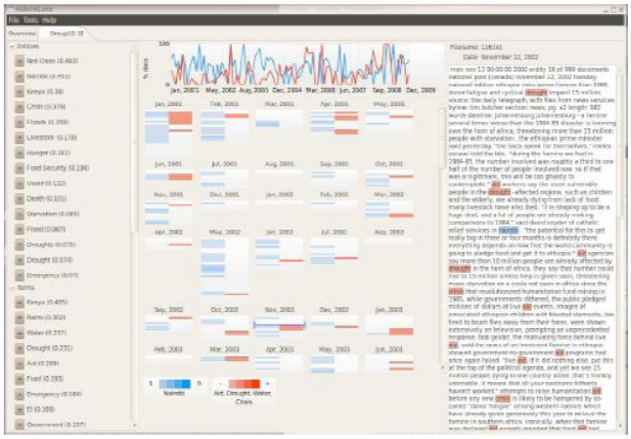

Fig. 1. Three-panel color display with timeline at the top.

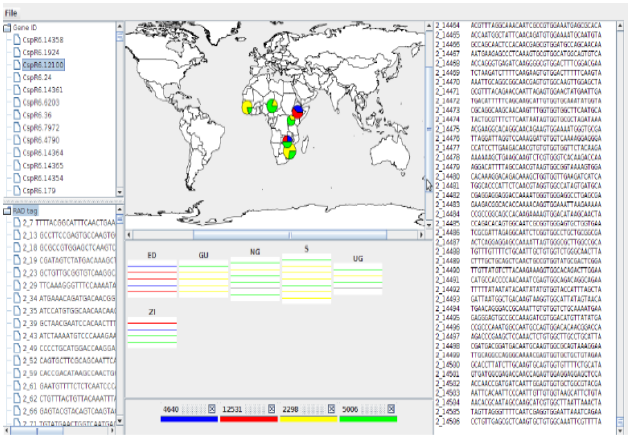

Fig. 2. The PhyloLens user interface.

A. Sample Genomic Sequences

The core of a PolyLens data set is a series of full DNA sequences sampled from various individual organisms. These genomes are subdivided into substrings of nucleotides that are long enough to be unique across multiple genomes, called

RADTags. In the *Drosophila* data set, this length was 38. These RADTags were then given a unique identifier to specify its location in the individual's genome. This locational identifier, or locus, was of the form:

$$organismID\_location$$

A portion of an example DNA sequence is provided in Figure 3.

```
12_0 ATCACCGAAATACCAAAATATGTGGTGTAACAGTCAAA
12_1 GTAACTTCAAAGCCAAGGCATGTGGTACAATACCATAA
12_2 GTTCGGTGATTGCCAAGCAATGTGGTTTGCAGATGATA
12_3 ATTAGGCGTTAGACAAAGACGGTGGACAAAGCCATATA
12_4 TTACAAGCCTAGACAAGCCCAGTGGCCCCTCCGGCAGA
12_5 TGGCTCTCTTGTGCAATTACTGTGGCGCACTCCGTAGA
12_6 TTGTATTAACCAACAATGTCGGTGGCAGCAGGGCAACA
12_7 CAATGTCGGGAAGCAATCAATGTGGCTTCATGTTTCGA
12_8 TTTATCTTCATTACAATTTTTGTGGTAGGAAACTGATA
12_9 AGAGATATAAAACCAATCGGAGTGGGGCCTCATTAAGA
12_10 GTAGCGCATTCATCAATGTCCGTGGCGCTTCCGTTGGA
12_11 AGGGCGGTGGACACAATGAATGTGGACACCATCAGGGA
12_12 GATGAACGCGTGTCAAGGGGTGTGGGGTGTGGTGGGTA
12_13 TTCGAATTAATTGCAACTCCAGTGGAGAAATGTGTCTA
12_14 AAACGGATAAATACAACTTATGTGGGAATGGCAAAATA
12_15 TATAAGGATGCTCCAAAAGGGGTGGGGCGGGGCGTGGA
12_16 TACTGTACTAGCGCAAGTACAGTGGAGCCAACAATAGA
12_17 TTATGTGCTAACACAAAGTGTGTGGATGCCCGAATGGA
12_18 ACAAATTTCAGGGCAAAAAATGTGGCTATGGGAAAGCA
12_19 CCCGGTACATTTCCAACAGAGGTGGGGTTGCTGCCAAA
12_20 AGTTCGTGAATCTCAAGGGCGGTGGACTCCTGGTGGAA
12_21 AAGACGAGCATCACAACGGGCGTGGTGGATCGCTCGGA
12_22 TTTATTTGTATTGCAAAGGAAGTGGCTCCTTAACTGCA
```

Fig. 3. A sample DNA sequence.

B. Location List

Each sampled organism is associated with a geographic location at which the sample was collected. Each entry in the location list specifies a latitude and longitude as well as the organism ID of the organism sampled there. Figure 4 is an example location list.

```
 6.98  39.18   1
 6.98  39.18   2
 6.98  39.18   3
 6.98  39.18   4
 6.98  39.18   5
10.70 -12.25   6
10.70 -12.25   7
10.70 -12.25   8
10.70 -12.25   9
10.70 -12.25  10
11.85  13.16  11
11.85  13.16  12
11.85  13.16  13
11.85  13.16  14
11.85  13.16  15
11.85  13.16  16
```

Fig. 4. Coordinates associated with each organism.

C. Gene ID List

The gene ID list consists of groupings of RADTags that represent different expressions of the same gene. Figure 5 shows a subset of the gene ID list from the *Drosophila* data set. Each gene ID is associated with a set of loci, which in turn map to RADTags. Note that multiple loci can map to the same RADTag.

```
CspR6.20480 21_7004,22_6804,31_6896
CspR6.20481 21_7016,27_6645
CspR6.20485 21_7269,30_7109,31_7125
CspR6.20486 21_7313,22_7091,31_7181
CspR6.20487 21_7434,28_7221
CspR6.20489 21_7487,22_7262,23_7567,30_7334
CspR6.20490 21_7694,23_7763
CspR6.20491 21_8110,30_7950
CspR6.20492 21_8528,27_8013,29_8238
CspR6.20493 21_8993,31_8752
CspR6.20495 21_9395,29_9068,30_9152
CspR6.20500 21_10174,29_9821
CspR6.20505 21_10724,23_10786,28_10372
CspR6.20509 21_12244,30_11892
CspR6.20512 21_12595,29_12179
CspR6.20513 21_12733,31_12389
CspR6.20514 21_12913,26_11509,28_12471
CspR6.20515 21_12991,23_13074,31_12643
CspR6.20518 21_13114,28_12653
CspR6.20519 21_13124,23_13204
CspR6.20521 21_13443,28_12962,30_13048
CspR6.20526 21_14282,30_13828
```

Fig. 5. Gene IDs mapped to corresponding loci.

D. Stop List

The stop list is a component borrowed from the text mining field. In text mining, high-frequency words such as *a*, *an*, and *the* merely serve as lexical filler and are relatively meaningless. In text mining algorithms, these words are often added to a stop list, or list of words to be excluded as noise. Likewise, some RADTags are so common across populations that they can be discarded in a similar fashion. Furthermore, on the other end of the spectrum, there are some RADTags that are so unique that they are irrelevant to relatedness queries. These can be removed from consideration as well. The stop list then, in a genetic context, is simply a list of RADTags to be excluded as irrelevant. Figure 6 shows a sample subset of a PolyLens stop list. Each RADTag is accompanied by a full listing of the organism loci at which it is present.

```
**AAAAATTCCCTTTCAAGTGTTGTGGCAAGGACATCCTA
1_8114, 2_8136, 3_8208, 4_8278, 5_8109, 6_8220, 7_7683, 8_8301, 9
_8327, 10_8043, 11_8346, 12_8220, 13_8032, 14_8108, 15_8176, 16_
8046, 17_8778, 18_8794, 19_8820, 20_8769, 21_8765, 22_8446, 23_
8820, 24_8058, 25_8061, 26_7801, 27_8226, 28_8492, 29_8454, 30_
8554, 31_8535
**AAAACATATGCAACAAGAGGTGTGGCCAAGACTCTCAA
1_353, 2_349, 3_349, 4_356, 5_341, 6_344, 7_329, 8_350, 9_356, 10
_358, 11_368, 12_344, 13_345, 14_317, 15_345, 16_331, 17_374, 18_
379, 19_381, 20_367, 21_369, 22_351, 23_359, 24_342, 25_340, 26_
353, 27_346, 28_360, 29_346, 30_361, 31_368
**AAAACGCTGTGACCAATTCAAGTGGAGTTTTCCGGCGA
1_7453, 2_7435, 3_7500, 4_7581, 5_7436, 6_7549, 7_7064, 8_7618, 9
_7626, 10_7379, 11_7656, 12_7549, 13_7356, 14_7448, 15_7486, 16_
7386, 17_8041, 18_8061, 19_8083, 20_8024, 21_7990, 22_7739, 23_
8077, 24_7384, 25_7376, 26_7129, 27_7528, 28_7771, 29_7742, 30_
7829, 31_7831
**AAAAGCAATATTACAATAACTGTGGTAGAATTGGCTCA
1_7377, 2_7354, 3_7421, 4_7498, 5_7359, 6_7474, 7_6993, 8_7545, 9
_7542, 10_7302, 11_7576, 12_7474, 13_7286, 14_7366, 15_7407, 16_
7306, 17_7954, 18_7983, 19_7995, 20_7942, 21_7912, 22_7664, 23_
7982, 24_7302, 25_7289, 26_7059, 27_7450, 28_7679, 29_7664, 30_
7738, 31_7749
**AAAAGGCATTTGGCAAATGAGGTGGAAATGCTCTCAAA
1_10505, 2_10607, 3_10727, 4_10773, 5_10609, 6_10708, 7_10048, 8_
10872, 9_10828, 10_10522, 11_10840, 12_10708, 13_10519, 14_10535,
15_10698, 16_10451, 17_11462, 18_11481, 19_11487, 20_11453, 21_
11448, 22_11031, 23_11533, 24_10476, 25_10579, 26_10228, 27_
10674, 28_11073, 29_11068, 30_11136, 31_11152
```

Fig. 6. A sample PolyLens stop list.

E. Data Relationships

It is also important to understand the relationships between the different data components. Each organism sample, com-

posed of a set of RADTags, is associated with one location, the coordinates at which the sample was collected. While a given RADTag is generally only encountered once in an individual, multiple occurrences are possible. Furthermore, a RADTag can occur in any number of samples. While the gene ID and stop list data are currently expressed in terms of organism loci, they are essentially mappings to RADTags as well. A gene ID is associated with a set of RADTags that are the different expressions of a gene. The stop list is a set of RADTags to be excluded from examination. This information is represented in an efficient manner in the entity relationship diagram shown in Figure 7.

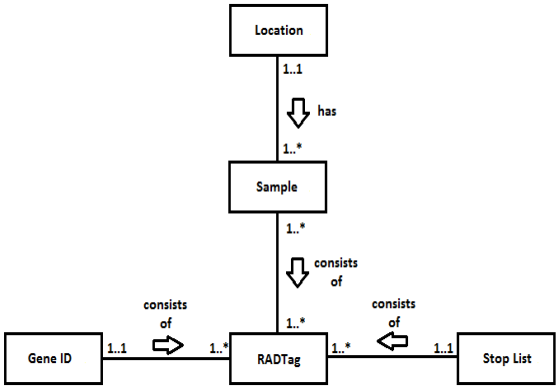

Fig. 7. An ER diagram describing the data relationships.

III. Implementation

The predecessor software, PhyloLens, while an effective proof of concept, did not provide the necessary efficiency, reliability, and polish to be put to practical use. Though there is still much room for improvement, PolyLens addressed many of these issues, resulting in a more robust and user-friendly tool. PolyLens was implemented in Java according to the Model-View-Controller (MVC) design, an architecture that attempts to separate data representation and the user's interaction with it. This architecture provides a highly modularized class structure and distribution of tasks, providing major benefits both in performance and further development. When a change is made to the underlying data model, only the relevant views need be updated. The PolyLens class structure is illustrated in Figure 8. The following sections discuss each of the major components, their responsibilities, and the features they provide.

A. Data Model

Due to the size and complexity of the data, the data model is divided into several subclasses. A series of data management classes are each tasked with parsing, maintaining, and error checking data of a certain type. Then, a master model class, PolyLensModel, serves as a mediator between the data managers and the other components of the software. Each of the data model components is discussed briefly in the following sections.

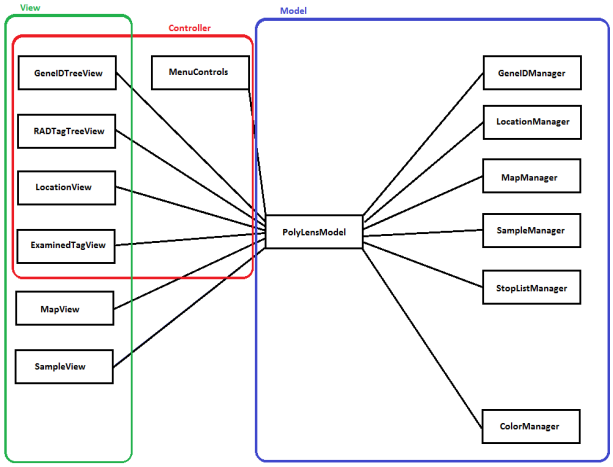

Fig. 8. PolyLens class structure.

1) GeneID Manager, Location Manager, Sample Manager, and Stop List Manager: Each of these manager classes is simply tasked with maintaining an in-memory mapping of data from each of the four data file types. The GeneIDManager class maintains a mapping from gene IDs to their corresponding RADTags. The LocationManager class maintains a mapping from organism IDs to their corresponding latitude and longitude pairs. The SampleManager class maintains both a mapping from loci to their corresponding RADTags and a mapping from RADTags to their corresponding loci. Finally, the StopListManager class maintains a map of all loci that are in the stop list. Each of these manager classes also maintains a list of files from which their data is collected. At parse time, each manager class also validates the data, excluding erroneous entries and generating informative error messages for the user. Each of these manager classes also maintains an update status that the master model class can poll to discover whether or not the data has changed.

2) Map Manager: The MapManager class serves as the interface between R and Java. R is a software environment for statistical computing and graphics that proved to be quite useful for this project. Specifically, the rworldmap library was used in conjunction with JRI, a Java/R interface, to generate the map images illustrating the geographic distribution of the RADTags selected for examination [3]. Given a frequency table populated with RADTag occurrences at each coordinate, rworldmap is capable of generating a series of pie charts drawn at the appropriate locations on a world map. Figures 9 and 10 show an example frequency table and its corresponding map. The MapManager class also supports the saving of frequency tables and maps to file for later examination.

3) Color Manager: The ColorManager class maintains a list of colors to be cycled through in the color coding of RADTags. This class was created because several components of the software must coordinate such that the same color is picked for a RADTag in each component. The master model class, PolyLensModel, cycles through one of eight

Latitude	Longitude	TAACATATACCTCCAAC	TAACATATACCTGCAAC	TAACATATACCTCCAAG	TAACATATACCTCCAACA
6.98	39.18	3	2	0	0
10.7	-12.25	1	3	1	0
11.85	13.16	0	2	2	1
-23.94	31.14	2	3	0	0
0.53	32.6	2	1	0	0
-16.54	28.72	0	2	0	2

Fig. 9. Distribution of RADTags by location.

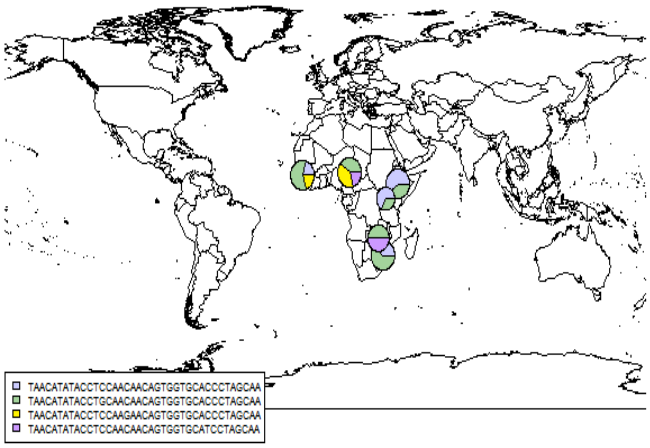

Fig. 10. An example map generated by rworldmap.

predetermined colors each time a RADTag is added to the list of examined RADTags. More colors can be added to the cycle, but it can be difficult to choose colors that are distinctly different from eachother, do not obscure the underlying text when used for highlighting, and are not irritating to the eye.

4) PolyLens Model: As mentioned earlier, PolyLensModel serves as a mediator between the various data management classes and the view and controller classes. Each of the view and controller classes are registered as observers with PolyLensModel when the application begins. Whenever an event occurs that modifies one of the data management classes, each observer is notified. Each observer then updates its display only if the data relevant to it has changed. Furthermore, the controller classes may generate these events through a series of data modification functions that also reside in PolyLensModel.

B. Views and Controllers

In the MVC design scheme, a view provides a visual representation of the underlying data model, while a controller allows the user to modify it. Unfortunately, it is not always practical to achieve separation between the two, as it may be desirable for some views to also serve as controllers. This was often the case in PolyLens. This section briefly details the functionality of each view, controller, and view/controller.

C. Gene ID Tree View

The GeneIDTreeView class provides the user with an organized list view of the available gene IDs. The gene IDs

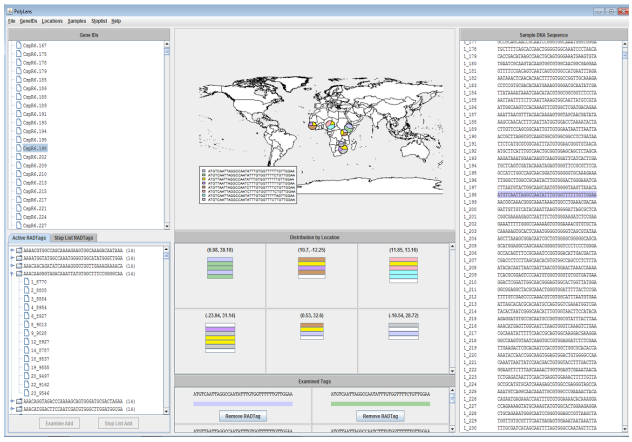

Fig. 11. The PolyLens interface as a whole.

are sorted alphabetically by their first component, and then numerically, allowing the user to quickly navigate to the gene ID they wish to examine. When a gene ID is selected for examination, all of its associated RADTags are added to the list of examined RADTags, generating an update event in several other view components. If a RADTag is not present in any of the sample organisms, a warning is issued to the user, and the RADTag is excluded. Figure 12 provides an example gene ID tree.

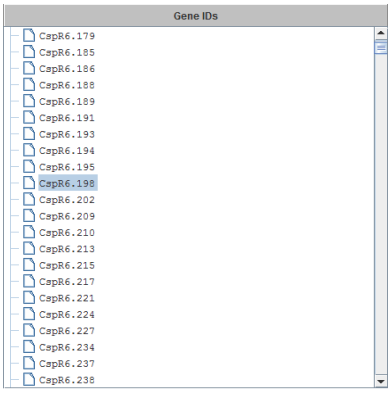

Fig. 12. The Gene ID Tree View.

D. RADTag Tree View

The RADTagTreeView class serves as an interface to all of the RADTags contained by the sample organisms, whether present in the stop list or not. The component was inspired by file system hierarchies used in operating systems. Both the active RADTags and the stoplist RADTags are represented by 'folders', with their associated loci represented by a 'file' within them. Furthermore, each RADTag is labeled with the number of organisms that it is present in. Then, the RADTags are displayed in list order, sorted first by frequency, then alphabetically. The loci within each folder is also sorted, first by organism ID, and then by location. This sorting was provided to facilitate the primary purpose of the component, the online

building of a stop list. Since it is often severely overrepresented and underrepresented tags that hinder relationship discovery, it made sense to make frequency the primary organization method for the RADTags. The component also allows the user to add RADTags to the list of examined tags independent of gene IDs, allowing more freedom in the exploration process. Figure 13 illustrates the active RADTag and stop list RADTag directory structures side by side.

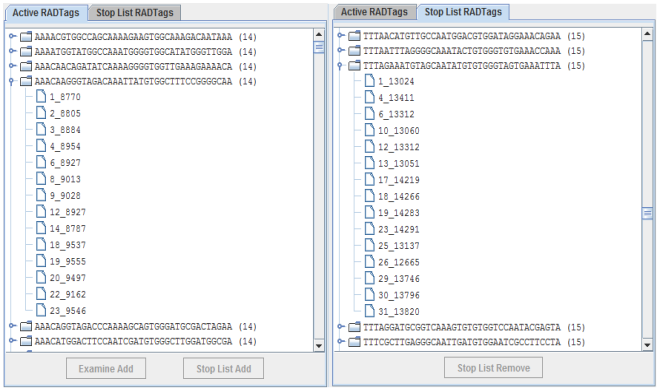

Fig. 13. The RADTag Tree View.

E. Map View

The MapView class simply displays the image generated by the rworldmap library, scaling it to the size allocated to the component. Whenever the list of examined RADTags is modified, the PolyLensModel class notifies the MapManager class to generate a new map. It then notifies the MapView that its current map is out of date. An example of this view was provided earlier in Figure 10.

F. Location View

The LocationView class displays the RADTag distribution in a chart-like fashion that is more organism-centric. Each organism is represented by a bar that is placed according to its geographic location. The bar is then color coded according to which of the examined RADTags it possesses, using the same color coding scheme used by the MapView. If a single organism possesses more than one of the examined RADTags, then the bar is subdivided. In this way, a more granular and specific view of the RADTag distribution is provided to the user. Furthermore, if the user mouses over a bar, a tooltip indicating the locus and RADTag being represented is displayed. This provides expedient access to information available from the other components. Figure 14 provides an illustration. When a user clicks on one of the sections of an organism bar, the SampleView displays the genome sequence document corresponding to that organism, with the corresponding RADTag highlighted.

G. Examined Tag View

The ExaminedTagView class displays for the user each of the RADTags currently being examined, as well as the color currently assigned to it. Furthermore, if the user mouses

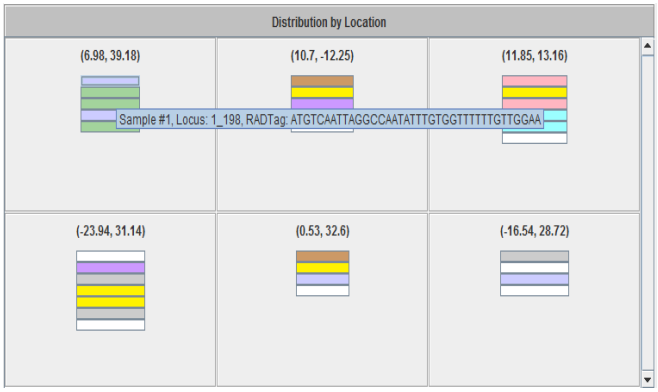

Fig. 14. The Location View.

over the RADTag, all of its corresponding loci are displayed, once again providing an expedient means to reaching that information. This component also allows the user to remove RADTags from examination. Figure 15 provides an illustration of this component.

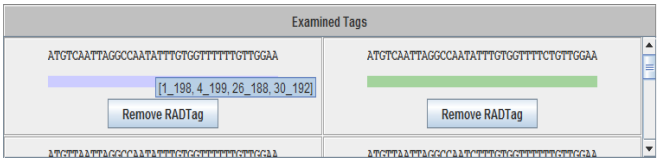

Fig. 15. The Examined Tag View.

H. Sample View

The SampleView class provides a display of the full genomic sequence for a selected sample organism. If the selected sample contains one or more of the currently examined RADTags, then the document highlights them in the color currently assigned to each. The user is free to navigate through the document manually, but automatic scrolling to examined RADTags is provided through interaction with the LocationView. This component can be seen in Figure 16.

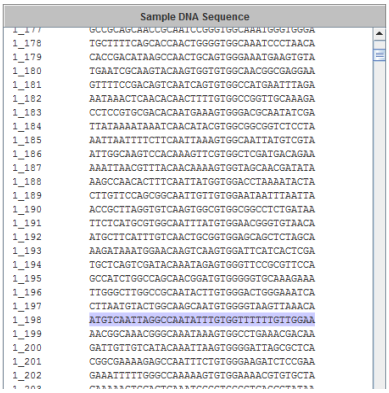

Fig. 16. The Sample View.

I. Menu Controls

The MenuControls class provides the user with a means for parsing data, removing data, and saving maps, frequency tables, and stop lists generated by their exploration.

IV. Results

A first attempt to initiate genome-scale comparisons of multiple individuals and populations involved a RADtag data set obtained from 32 individual genomes of *Drosophila melanogaster*. These genomes were obtained from 6 different locations within sub-Saharan Africa and Europe. The single European population consisted of 8 distinct French genomes, while each of 5 African populations were represented by 4-6 distinct genomes. In order to illustrate the potential usefulness of PolyLens, observe Figure 17, a query on the gene ID CspR6.16520. For this particular gene, a number of flies in France and Nigeria share a common allele that is not present in the other African countries. This could suggest a relatedness between the two populations. While the sample size is perhaps not large enough to say this with confidence, this example serves to illustrate the software's capability.

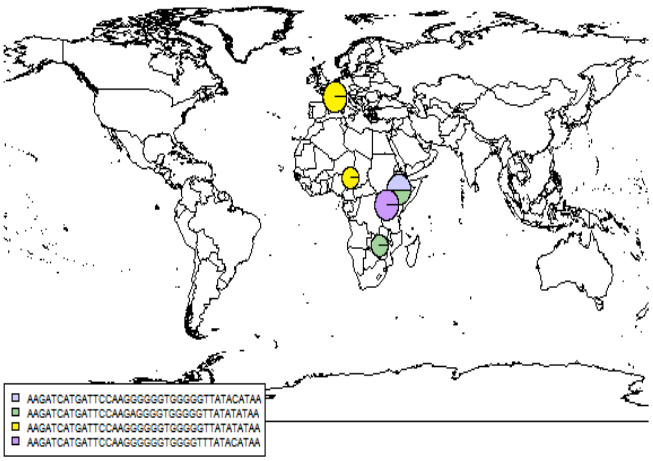

Fig. 17. Distribution for Gene ID CspR6.16520.

V. Future Work

While PolyLens provides many features for population genomic data exploration and pattern discovery, there is still room for improvement. The RADtag manipulation functions will be extended, and the merging or splitting of two or more RADtags will also be supported in future versions. Future improvements are also planned for the decoration of maps with pie charts. Future pie charts will be adjusted to accurately display information about groups of RADtags merged by the user as well as the complete absence of related RADtags. Another planned extension is to add in/out zoom capability to the maps. Additional improvements include the automatic generation and easy application of stop lists and the display/summary of clustering output.

References

[1] G. L. Shutt, A. A. Puretskiy, and M. W. Berry, "FutureLens Software for Text Visualization and Tracking," in *Proceedings of the Ninth SIAM International Conference on Data Mining: Text Mining Workshop*, 2009.

[2] G. Stuart, T. Gao, R. Pathan, and M. Berry, "PolyLens: Software for Map-based Visualization and Analysis of Genome-scale Polymorphism Data," *International Journal of Computational Biology and Drug Design*, to appear.

[3] A. B. South, "rworldmap: A New R Package for Mapping Global Data," *The R Journal*, 2011.