iHAT: interactive Hierarchical Aggregation Table

Corinna Vehlow∗
VISUS
University of Stuttgart

Julian Heinrich†
VISUS
University of Stuttgart

Florian Battke‡
Integrative Transcriptomics
University of Tübingen

Daniel Weiskopf‡
VISUS
University of Stuttgart

Kay Nieselt‡
Integrative Transcriptomics
University of Tübingen

ABSTRACT
In the search for single-nucleotide polymorphisms (SNPs), genome wide association studies have become an important technique for the identification of associations between genotype and phenotype of a diverse set of sequence-based data. In this work, we present a methodology for the visual assessment of SNPs using interactive hierarchical aggregation techniques combined with methods known from traditional sequence browsers and cluster heatmaps. Our prototype tool iHAT supports the visualization of multiple sequence alignments, associated metadata, and hierarchical clusterings. Moreover, data-type dependent colormaps and aggregation strategies as well as different filtering options support the user in finding correlations between sequences and metadata. Similar to other visualizations such as parallel coordinates or heatmaps, iHAT is aimed at exploiting the human pattern-recognition ability for spotting patterns that might indicate correlation or anticorrelation. Together with its interactive features and a database backend for fast data retrieval, we consider iHAT as a prototype for a visual analytics system for genome-wide association studies.

Keywords: genome wide association studies, hierarchical aggregation, phenotype-genotype correlation.

Index Terms: H.5.m [Information Systems]: Information Interfaces and Presentation—Miscellaneous; H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval—Information filtering; H.5.2 [Information Interfaces and Presentation]: User Interfaces—Graphical user interfaces (GUI); J.3 [Computer Applications]: Life and Medical Sciences—Biology and genetics;

1 INTRODUCTION
Genome wide association studies (GWAS) are used to study the variation of genes between individuals (the genotype), and their association with a variety of complex traits (the phenotype), e.g. diabetes, heart disease, or arthritis. GWAS have become an established method to alleviate the identification of genetic risk factors of diseases, as they make use of recent technologies that allow a rapid and cost-effective analysis of genetic differences. Within the last five years, many single-nucleotide polymorphisms (SNPs) could be identified with the help of GWAS, implicating hundreds of loci for common traits [13]. The huge amount of data produced by GWAS implies a great challenge for data analysis and visualization. In this paper, we use interactive hierarchical aggregation together with a sequence alignment viewer as a tool for the visual analysis of correlations between sequence data and associated metadata.

∗e-mail: corinna.vehow@visus.uni-stuttgart.de
†e-mail: julian.heinrich@visus.uni-stuttgart.de
‡e-mail: florian.battke@uni-tuebingen.de
¶e-mail: daniel.weiskopf@visus.uni-stuttgart.de
§e-mail: julian.heinrich@visus.uni-stuttgart.de
¶e-mail: corinna.vehlow@visus.uni-stuttgart.de

IEEE Symposium on Biological Data Visualization
October 23 - 24, Providence, Rhode Island, USA
978-1-4673-0002-5/11/$26.00 ©2011 IEEE

Sequence alignment viewers use tables to represent large datasets and color coding for the illustration of data attributes. This concept is similar to that of the heatmap (mostly used for large-scale expression data) [3, 26]. While table-based visualizations are very useful for getting an overview of a dataset and visually finding patterns that might be difficult to spot with automatic methods, there are two drawbacks of sequence alignment viewers or heatmaps: First, the patterns that emerge depend on the order of rows and columns. For time-series expression data and sequence alignments, column-order is usually fixed and hence does not pose any problem. Where applicable, row and/or column order can be partly defined (i.e. within groups) using hierarchical clustering. Heatmaps therefore often include a dendrogram visualizing the result of hierarchical grouping of rows and/or columns. Second, the number of data items that can be visualized is restricted by the number of pixels that are available. As a consequence, many datasets produced today cannot be visualized in a single image using the traditional tools. To mitigate the problem, most implementations provide a scrolling mechanism that enables the user to adapt the region of interest. However, scrolling the view does not provide an overview and hence contradicts the information seeking mantra [18]. While aggregation is used in sequence alignment viewers as a tool for zooming, none of the existing visualizations facilitate the aggregation of multiple sequences in order to reduce the amount of data items that have to be rendered.

Similar to traditional heatmaps, our tool iHAT consists of a table-based visualization of both primary sequence data and metadata with an attached dendrogram. In contrast to other tools that include table-based visualizations, iHAT allows the user to apply different colormaps and aggregation strategies. Both depend on the type of data for every column, i.e. nominal, ordinal, ratio, and interval data are aggregated based on different methods and mapped onto different (single- or multi-hue) colormaps. Rows can be aggregated using either the attached hierarchy or according to interactive selection. To find associations between the genotype and phenotype, the aggregation process is guided by the metadata representing the phenotypes.

We demonstrate our techniques with multiple sequence alignments of DNA and amino acid sequences with attached phenotype metadata. For the analysis of DNA data, we use the IEEE VAST 2010 Challenge Data [8]. For proteins, we use amino acid sequence data of the neuraminidase protein of 15 H5N1 influenza virus samples [10]. Using our aggregation techniques, correlations between sequence positions and metadata become apparent.

2 RELATED WORK
Heatmaps represent matrices of data values in the form of two-dimensional tables whose tiles are colored with respect to the underlying data values. The idea of a semi-graphic table that displays values of a data table by levels of shading is quite old, first used in 1873 by Loua to visualize social statistics [26]. Within the cluster heatmap, the matrix view is expanded by trees for the row and column association matrices. This approach of appending trees
to the rectangular matrix was introduced by Gower and Digby in 1981 [26]. Eisen et al. [3] developed a system that applies clustering to genome-wide expression data from DNA microarray hybridization. They used the average-linkage method to arrange genes with respect to some similarity in their patterns of gene expression. As a result, all elements are assembled within a dendrogram, which is appended to the rows of the similarity matrix to indicate the nature of relationships computed among genes. The matrix contains the similarity scores for all pairs of genes, mapped to colors. However, their tool does not allow the user to (interactively) change the derived hierarchy or to aggregate clusters within the hierarchy.

Another example of the cluster heatmap is the hierarchical clustering explorer (HCE) by Seo and Shneiderman [17]. Besides a matrix view and dendrogram, the HCE provides the user with a two-dimensional scatterplot showing the expression profile of genes. The scatterplot is bidirectionally linked to the matrix view via brushing and linking. Seo and Shneiderman used clustering to perform microarray data analysis and identify genes with similar profiles and thus likely similar function. The hierarchy of genes is constructed iteratively by pairing genes with the most similar expression profiles in each step. Similar to other cluster heatmaps, the hierarchy is displayed as a dendrogram. Dynamic queries/filtering can be applied to hide uninteresting clusters or highlight interesting ones. In contrast to iHAT, the HCE does neither support interactive construction of hierarchies nor aggregation of subhierarchies.

As the size of datasets increases, it is more and more difficult to analyze them and keep track of the overview. Hierarchical aggregation is an important and successful method to reduce the amount of data. A data-driven aggregation enables the user to reveal patterns within the data. For the purpose of visual inspection, it is necessary to summarize large amounts of data effectively, making visual representations of these data more scalable and less cluttered. Elmqvist and Fekete [5] developed a model for building, visualizing, and interacting with multiscale representations of information visualization techniques through the use of hierarchical aggregation. They pointed out that the visual representation of the aggregate should convey information about the underlying data. Furthermore, the aggregation method should depend on the underlying data type. Based on the discussion of available visualizations, which already include hierarchical aggregation, they derived a set of general guidelines for the design and implementation of aggregation techniques.

Matrices, displayed in cluster heatmaps, can visualize information and data structures of various types. The Zoomable Adjacency Matrix Explorer (ZAME), developed by Elmqvist et al. [4], represents a graph in the form of an adjacency matrix. The three main components of ZAME include the hierarchical aggregation, automatic reordering of components, and accelerated rendering. The hierarchical clustering of edges is defined by a pyramid of recursive edge merging. Exactly four children are aggregated into a new internal node, which is not flexible enough to create hierarchies of rows. The aggregation is performed differently depending on the underlying data type, which might be numerical, nominal, or categorical. Attributes of aggregates like average, minimal, or maximum values are mapped to visual attributes, e.g. with a histogram.

While heatmaps use colored matrices to illustrate data values of a table, sequence viewers use them to show aligned sequences of nucleic acids or amino acids. Color is employed to indicate the type of nucleic acid or amino acid, or it represents some attribute of the alignment. There are many sequence viewers with different extents of functionality. Tools like JBrowse [19] and the human genome browser at UCSC [11] only serve as sequence viewing software, whereas other tools like ClustalW/ClustalX [23], Jalview [25], CINEMA (Colour INteractive Editor for Multiple Alignments) [14], and STRAP (Interactive Structure based Sequences Alignment Program) [6] also include capabilities for the visualization of (multiple) sequence alignments. The Integrative Genomics Viewer (IGV) [16] further supports the import and visualization of microarrays and genomic annotations. SeaView [7] also enables the user to construct and investigate phylogenetic trees of the alignments. Unfortunately, none of these tools facilitate the aggregation of sequences. In order to use the screen space efficiently, one could use the Table Lens approach by Rao and Card [15] as a focus and context technique to have a closer look at some sequences while fading others into the background.

Similar to other tools like the traditional heatmap or the hierarchical clustering explorer, our tool consists of a table view linked with an aligned dendrogram. The key difference to other tools is that it supports the interactive construction of trees from arbitrary data. Furthermore, it allows one to aggregate flexible numbers of rows. Several functionalities were implemented to support the user during the aggregation process. These will be described in more detail in the next sections.

3 TERMINOLOGY

In this paper, we present a method for the visualization of hierarchical multivariate data. To be precise, we consider multivariate data as a set of $N$ samples (rows), each comprising the same number of values (columns) from a set of $M$ variables. Each variable has a scale type [21], which can be one of:

- **Nominal**: Nominal data has neither ordering nor metric. Only the equality operation ($\equiv$) is defined for values on this scale.
- **Ordinal**: Ordinal data defines an ordering. In addition to the operations allowed for nominal data, the order operator ($<$) is defined.
- **Interval**: Interval data is measurable on an interval scale. In addition to the operations defined for ordinal data, the difference operator ($-$) is defined.
- **Ratio**: For data on a ratio scale, the equality of ratios can be established in addition to differences.

Column $C_j$ contains all values of variable $j$ and row $R_i$ contains all values of the sample $i$ (see Figure 1). The value of a cell at row $i$ and column $j$ can now be addressed using either $R_{i,j}$ or $C_{j,i}$. Samples can further be aggregated into a tree $T = (V,E)$ with vertices $V$ and edges $E$, resulting in hierarchical multivariate data. Using the terminology from Elmqvist and Fekete [5], our multivariate samples are data items that can be grouped into aggregate items. While...
both data and aggregate items are represented by a vertex $v \in V$, data items define the set $L = \{v \in V | \text{succ}(v) = \emptyset\}$ of leaf nodes and aggregate items define the complement $I = V \setminus L$ of interior nodes plus the root node. Aggregate items can also be grouped, such that the root node denotes the set of all data items. Note that for aggregation, aggregates are first resolved to their components, before the set of leaves is aggregated.

4 IMPLEMENTATION

Our tool is implemented in Java. Data accessibility is ensured via database access, which allows us to perform queries efficiently (for the purpose of filtering and sorting). A file-based backend is included for datasets.

iHAT serves two purposes: the exploration of hierarchical multivariate data (in combination with some related metadata) and the interactive generation of hierarchies via stepwise aggregation. Depending on the underlying data, the colored matrix (see Section 4.1) can be too large to be displayed completely on screen. To avoid extensive scrolling and to obtain an immediate impression of the relevant information, different techniques are offered to reduce the number of matrix rows and columns displayed. These include aggregation of rows (Section 4.2) and filtering of columns (Section 4.3). Furthermore, the number of pixels allocated to each cell can be increased/decreased, thereby decreasing/increasing the number of cells that can be displayed at the same time.

4.1 Color Coding

iHat maps values to color depending on their type. As the appropriate colormap greatly depends on the data that is visualized [2], we adopted general design principles from literature [24] for the different scale types:

- For nominal variables, the size of the colormap should reflect the number of unique values within all samples. Hence, it would be desirable to use a set of colors that facilitate qualitative separation of the unique values. However, this approach does not scale well for large numbers of unique variables. Depending on the study, users are not able to robustly distinguish more than about 10 different colors.

- For ordinal variables, it would be desirable to use a colormap that represents the order of values.

- For ratio and interval variables, a colormap that allows one to accurately infer distances from colors should be used.

While these principles clearly apply to single or “flat” variables, the mapping of colors to hierarchical data introduces new challenges.

For the application to nucleic acid and amino acid sequences, iHAT offers different colormaps used by tools like ClustalX [23], Jalview [25], Lesk [12], or the Nucleic Acid Database [1]. In addition, we developed a novel colormap for amino acids following the Venn diagram [22] grouping of amino acids based on their properties, only considering the groups formed by the three main properties: hydrophobicity, size, and polarity (see Figure 2). Based on these properties and their intersections, the Venn diagram divides amino acids into seven groups. Amino acids are thus colored with respect to the group they belong to, where each group is assigned a color. All amino acids within the same group are mapped to slight variations of the respective color for the group they belong to (see Figure 3). At that, we tried to maximize the differences of color within the group. The developed color scheme helps the user in getting an immediate impression of the biochemical properties of amino acids within the sequences.

For metadata values, we follow the same general approach. Metric colors are colored using a single-hue colormap with varying saturation and value. For nominal columns we adapt the number of different hues to the number of classes contained in the respective column and map the relative frequency of the consensus (the most frequent child item) to saturation and value. This way, the color scheme is used to visualize the (un-)certainty of the consensus.

4.2 Hierarchical Aggregation

A table is used to render the visual representations of multivariate samples (visual data items) while the data hierarchy is visualized with a dendrogram attached to the rows of the table (see Figure 4). For multivariate data without existing hierarchy, we create a tree of height one, where every sample (row) is a child node of the root and a leaf node of the tree.

iHAT implements bottom-up aggregation: a hierarchy can be constructed in a stepwise manner through the aggregation of a set of selected samples (rows that represent leaves in the aggregation tree) or aggregates (rows that represent internal nodes). Hence, several consensus rows can also be joined into a new consensus row. Interactively constructed trees can be exported (in Newick format) and imported again for further investigation. The dendrogram itself can be visualized as (left-to-right) node link diagram. To reduce the number of rows and compare subclasses of the hierarchy, internal nodes can be collapsed to show a consensus row or expanded to show all underlying samples of the aggregate individually.
Aggregate items, i.e. internal nodes of the tree, are visualized using the same visual mapping as for leaf nodes (data items). Hence, a row in the table may represent the multivariate data of any node in the aggregate tree. To achieve a consistent representation of data samples, the values for interior nodes have to be computed from the nodes’ children. Then, the same color mapping used for leaf nodes can also be applied to interior nodes. The computation of aggregate values depends on several factors:

- **Variable type**: Aggregation operations may not be applicable to all types of variables. For instance, a mean value can only be computed for ratio and interval data, and not for nominal or ordinal data.

- **Underlying distribution**: The aggregate value should represent the most important aspect of the distribution of aggregated values, if the entire distribution cannot be represented by the aggregate. Thus the choice of algorithm depends on the semantics of the distribution, i.e. the data that is visualized. As an example, consider normally distributed data for which the mean or the variance of the distribution are important descriptors that one might use as the aggregate value. For more complicated distributions (e.g. with multiple modes) or non-metric data, other aggregation techniques might be more suitable.

- **Number of visual channels**: The choice of aggregation algorithm also depends on the number of visual channels that can be used to map the aggregated value. A visual channel can be any variable that defines the visual appearance of the underlying geometry, such as area of the cell, length of a line, radius, texture function, color, etc. Again, consider normally distributed data, which is fully described by only two metric values (the mean and the variance). If the visual aggregate is defined by two or more variables, the distribution can be visualized. Besides the limited number of visual channels, the visualization might be further constrained by the layout leading to the need to restrict the number of aggregation variables, which ultimately results in loss of information.

In our case of a tabular layout of visual items and visual aggregates, we use the color channel to convey information about the distribution of items. Depending on the color space, the color channel can be split into further variables such as hue, saturation, and value or red, green, and blue, which gives more degrees of freedom for the design of visual aggregates. However, as a simple mapping of aggregate variables to these color changes very likely interferes with the coloring principles outlined in Section 4.1, we use the following data-type dependent strategies to assign aggregate values to colors.

### 4.2.1 Nominal Data

For nominal data, we use multi-hue bivariate colormaps to indicate class membership and map saturation and value to the relative frequency of the consensus. We use HSV colorspace [20] to choose the final color: The hues required to distinguish classes can be chosen by distributing all classes over the range of available hues. This strategy enables one to use saturation and value as an indication for the uncertainty of the most frequent child item, where colors of maximum saturation and low value denote certain values. However, while this approach can easily be automated, it does not scale well for large numbers of classes. At least, the color scheme for amino acids (introduced in Section 4.1) allows the user to easily differentiate between groups of amino acids, where the differences within a group are standing out less.

### 4.2.2 Ordinal Data

Ordinal data is treated similar to nominal data with respect to aggregation strategies and color mapping because colormaps for ordered data highly depend on the semantics of the data. While we use a discrete color table for the first dimension (the ordinal value), for the second dimension (the uncertainty) we use the same strategy as for nominal values.

### 4.2.3 Ratio Data

Following the design principles for ratio and interval data (see Section 4.1), we are interested in conveying quantitative information using the color channel. Data on a ratio scale is aggregated computing the mean value. Different colormaps exist that ensure that the equivalence of distances of ratios and intervals is perceived correctly. We map ratio values to an univariate single-hue colormap, where the ratio value determines saturation and value.

### 4.2.4 Interval Data

Interval data is mapped to color equivalent to ratio data, after the interval data have been converted into ratio data using the minimum and maximum.

### 4.3 Sorting and Filtering

The samples (rows) can be sorted with respect to selected metadata items. If several metadata items are used for sorting, this results in a nested sorting, which is a useful feature to interactively construct a hierarchy of samples.

Columns can be filtered to hide uninteresting information. Reasonable filtering options should always be based on the underlying data. As our application targets sequences of nucleic acids or amino acids (as samples), currently implemented filtering options were designed to hide columns that are too homogeneous or too noisy. iHAT supports semi-automatic filtering of columns, based on the following characteristics:

- **Number of symbols**: The number of different symbols (nucleic or amino acids) is determined, only considering symbols that exceed a given minimum frequency in the respective column. Only columns are shown whose number of symbols lies in a specified interval of interest. This supports the process of revealing associations between the genotype and phenotype.

- **Missing symbols**: Only columns with less than a given percentage of unknown symbols (i.e. gaps in the sequence) are shown. Columns that contain mostly gaps (resulting from the
alignment) do not contain any information that helps the user find correlation with the phenotype (metadata) and can therefore be hidden. While unique insertions or deletions may convey a difference in phenotype, they should at least occur in a certain percentage of the underlying population to allow statistically meaningful conclusions.

- **Noise:** When searching for associations between genotypes and phenotypes, we are interested in finding columns that show differences between the phenotypes, while being mostly uniform within each phenotype. By using a row-order dependent noise filter, we try to hide columns that violate this assumption, i.e. columns that do not match the sorting based on metadata: We count all row indices \( i (1 \leq i < N) \) where the symbol \( R_{i,j} \) differs from the symbol \( R_{i+1,j} \) and hide all columns where the percentage of such indices is above a given threshold.

- **Prior knowledge:** Users can supply a list of columns of interest (determined by an external method, e.g. some correlation or other statistical method) and only show those columns.

## 5 Results

To demonstrate the functionality and usefulness of iHAT, we used it for the analysis of nucleic acid sequences and amino acid sequences with associated metadata. Hence, rows represent sequences, columns represent alignment positions, and cells contain nucleic (resp. amino) acids, or metadata of scale type ratio, interval, nominal, or ordinal. In the matrix view, each position is colored either by nucleic acid (or amino acid) or attribute value. Depending on the scale type, different color schemes are used (as described in Section 4.1).

One of the main features of iHAT is the aggregation of rows (here sequences). As sequences are of nominal type, the nucleic acid (amino acid) of the aggregated or **consensus** sequence at position \( i \) is chosen as the one with largest frequency (i.e. the mode), giving rise to the color value in the respective cell. The relative frequency of the nucleic acid or amino acid in the consensus (i.e. the degree of conservation in the alignment) is mapped to saturation and value. For ratio values within the metadata, the mean value is taken as the consensus.

When using filtering of columns and sorting and aggregation of rows based on some metadata in combination with color mapping, column specific patterns emerge that facilitate the detailed analysis of correlation between nucleic acid (amino acid) sequences and metadata (e.g. phenotype data). To unclutter the matrix view and to improve the visual pattern matching, the labels (for nucleic acids, amino acids or attribute values) can be hidden on demand (see Figure 4).

### 5.1 DNA

For the analysis of DNA data, our general approach is to associate genotype (sequence) with phenotype data (metadata) with the help of the matrix based alignment view. We used the IEEE VAST 2010 Challenge Data [8] (mini challenge 3: genetic sequences) to demonstrate the analysis. The dataset consists of 58 sequences with 1403 nucleic acids.

![Figure 5](image.png)

(a) Sequences are sorted and aggregated based on metadata score.
(b) Sequences are sorted and aggregated based on drugResistance.
(c) Sequences are sorted and aggregated based on metadata symptoms.
(d) Sequences are sorted and aggregated based on metadata mortality.

Figure 5: Aggregation creates new consensus rows (which are assigned unique numerical labels starting with 'A'). Sorting these consensus rows by different metadata columns reveals correlations: complications/drugResistance correlates with positions 160 and 789, and drugResistance also correlates with positions 21 and 78 (b). Correlations also emerge between symptoms/mortality and positions 78 and 1086 as well as between symptoms and mortality themselves (c,d).

We converted the ordinal data into ratio data according to:

- symptoms \( \in \{\text{Mild}=0.0, \text{Moderate}=0.5, \text{Severe}=1.0\} \)
- mortality \( \in \{\text{Low}=0.0, \text{Medium}=0.5, \text{High}=1.0\} \)
- complications \( \in \{\text{Minor}=0.0, \text{Major}=1.0\} \)
- drug resistance \( \in \{\text{Susceptible}=0.0, \text{Intermediate}=0.5, \text{Resistant}=1.0\} \)
- at risk vulnerability \( \in \{\text{Low}=0.0, \text{Medium}=0.5, \text{High}=1.0\} \).

For a detailed explanation of the metadata types and their values, we refer to [8]. Based on the ratio values, we computed a score by summing over all metadata values per sequence, which in turn served as additional metadata column. Sorting and aggregation based on these metadata types showed that there are correlations between phenotypes and specific positions within the sequence (see Figure 5).
There is a strong correlation between complications and SNPs at sequence positions 160 and 789, and a slight correlation with positions 841, 945, and 1086. Some of these columns also correlate with drug resistance, particularly SNPs at position 160, 789, and 841 seem to be the reason for a drug resistance, whereas SNPs at position 21 and 78 seem to correlate with drug susceptibility (see Figure 5b). The symptoms and mortality annotations correlate best with SNPs at sequence position 1086, but also correlate with each other (see Figure 5c,d). The derived score correlates with column 841, and the at risk vulnerability is correlated with position 945.

The analysis showed that sorting of DNA sequences by metadata reveals patterns of equal saturation/value gradient between nucleic acid columns (genotype) and metadata (phenotype).

### 5.2 Protein

To show the application of iHAT to protein sequences, we used sequence data for the neuraminidase protein of 15 H5N1 influenza virus samples [10]. The sequences were aligned using ClustalW [9] and loaded into iHAT together with the respective strains’ virulence strengths (classified as low, intermediate, or high). The complete alignment comprises 450 columns (Figure 6A). We first apply a filter to show only those columns that contain at least two different amino acids, each at least present in 10% of the samples, i.e. in more than one sequence, drastically reducing the number of columns to inspect (Figure 6B). We then sorted the sequences according to the virulence annotation and created new internal nodes in the aggregation tree by aggregating all strains with low virulence into one group and aggregating the remaining intermediate and high virulence strains into another group (Figure 6C). Collapsing the aggregation nodes results in our final alignment of two consensus sequences. From this alignment, we can clearly see that column 28 (T vs I), 38 (K vs Q), and 203 (T vs I) are correlated with the strength of virulence (Figure 6D). In the original publication, the correlation of column 28 with lethality in mice was experimentally validated.

### 6 Conclusion

The huge amount of data produced by GWAS implies a great challenge for data analysis and visualization. In particular, scalability and pattern matching problems need to be addressed. Hence, we developed iHAT, which is a framework for generic data. iHAT serves the visual analysis of correlations between samples and associated metadata using interactive hierarchical aggregation in combination with a sequence browser.

Our usage scenarios showed that it is particularly useful for the exploration of genomic data, especially if phenotype information is available. iHAT allows the user to aggregate rows interactively, where metadata (phenotype information) can be used to guide this process. The aggregation of rows guided by metadata turned out to be helpful in revealing patterns from a multiple sequence alignment that might have their origin in SNPs related to the phenotype(s) under consideration. Furthermore, the program can be used to find correlations between mutations within amino acid sequences and some traits (phenotypes).

With iHAT we present a tool that transforms the problem of correlating genotype with phenotype to a visual pattern matching task. Starting from an overview of the aligned sequences, followed by filtering of uninformative sites and subsequent computation of consensus sequences for chosen subgroups, patterns emerge.
7 Future Work

There are several issues that we want to address in future work. Most importantly, the scalability with respect to the size of the dataset as well as the size of aggregate items has to be investigated. While scalability should not be an issue from a computational point of view as heatmaps can be computed efficiently, the ability of human investigators to visually recognize patterns in increasingly large datasets is worthy of further study. From our experience, we can assume that solely looking at large complex GWAS data cannot lead to an answer, but a combination of sorting and statistics with viewing is helpful, especially for the purpose of aggregation.

For the analysis of large GWAS data, the integration of analytic methods will be very beneficial and is planned for future work. iHAT already supports importing hierarchies (phylogenetic trees) produced by other tools. A direct connection to tools or methods producing computational statistics, e.g., for SNP analysis, would be of high value for the analysis process.

The sorting capabilities of iHAT could be extended to offer sorting based on columns/rows that show interesting patterns in addition to the already existing sorting based on metadata columns. For this feature, we have to consider that the use of both sorting schemes at the same time might lead to conflicts or overlaps.

We also want to add the possibility to aggregate both rows and columns to the general aggregation framework to allow for one additional hierarchy. In order to aggregate and create hierarchies of columns, it should be possible to sort columns based on some criteria. At the moment, iHAT only allows one to interactively rearrange them.

References


