# Visual Analysis of Patterns in Multiple Amino Acid Mutation Graphs

Olav Lenz, Frank Keul, Sebastian Bremm, Kay Hamacher and Tatiana von Landesberger

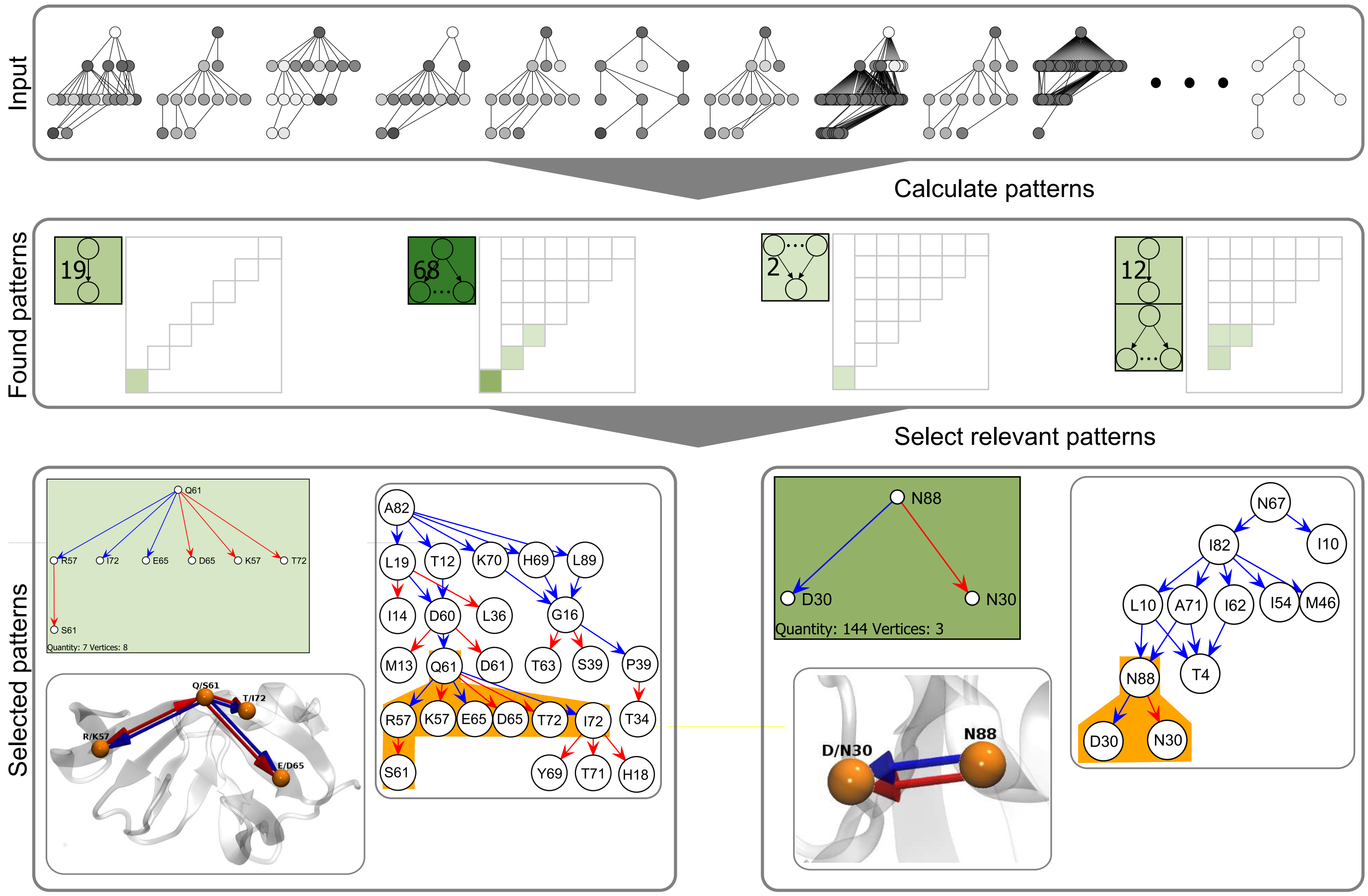


Fig. 1. The problem and the proposed solution for the visual analysis of patterns in mutation graphs: Top: A set of input mutation graphs. It is difficult to compare them and to identify common patterns. Middle: Visualization of the number of found patterns grouped by their structure. Bottom: User-selected relevant patterns can be examined in detail both in the input graph and in the 3D structure.

**Abstract**— Proteins are essential parts in all living organisms. They consist of sequences of amino acids. An interaction with reactive agent can stimulate a mutation at a specific position in the sequence. This mutation may set off a chain reaction, which effects other amino acids in the protein. Chain reactions need to be analyzed, as they may invoke unwanted side effects in drug treatment.

A mutation chain is represented by a directed acyclic graph, where amino acids are connected by their mutation dependencies. As each amino acid may mutate individually, many mutation graphs exist. To determine important impacts of mutations, experts need to analyze and compare common patterns in these mutations graphs. Experts, however, lack suitable tools for this purpose.

We present a new system for the search and the exploration of frequent patterns (i.e., motifs) in mutation graphs. We present a fast pattern search algorithm specifically developed for finding biologically relevant patterns in many mutation graphs (i.e., many labeled acyclic directed graphs). Our visualization system allows an interactive exploration and comparison of the found patterns. It enables locating the found patterns in the mutation graphs and in the 3D protein structures. In this way, potentially interesting patterns can be discovered. These patterns serve as starting point for a further biological analysis.

In cooperation with biologists, we use our approach for analyzing a real world data set based on multiple HIV protease sequences.

**Index Terms**— Biologic Visualization, Graph Visualization, Motif Search, Motif Visualization, Biology, Mutations, Pattern Visualization

## 1 Introduction

Pharmaceutical drug development strives to find effective drugs with high impact and low side effects. These drugs interact with a targeted protein (i.e., a sequence of amino acids) at specific positions of the amino acid chain. The interactions with the amino acid chain may

- *O. Lenz, S. Bremm and T. von Landesberger are with GRIS, TU Darmstadt. E-mail: name.surname@gris.tu-darmstadt.de*
- *F. Keul and K. Hamacher are with Computational Biology, TU Darmstadt. E-mail: surname@bio.tu-darmstadt.de*

*e-mail to: tvcg@computer.org.*

IEEE Symposium on Visual Analytics Science and Technology 2014
November 9-14, Paris, France
978-1-4799-6227-3/14/$31.00 ©2014 IEEE

induce indirect changes, e.g., induced mutations. The induced mutations may in turn induce further derived mutations, creating numerous evolutionary linked mutation events. These events may have a strong effect on large parts of a protein causing unwanted side effects, such as diminishing effectiveness of the drug. Therefore, the investigation of such evolutionary linked chain mutations plays an important role in drug development. Drug developers need to avoid addressing protein's positions with strong co-evolutionary dependencies [17].

The analysis of mutation chain events can also lead to new insights not only in drug development, but also in other fields of research, such as phylogenetic studies of our evolutionary history [4]. In this paper, we focus on the analysis of chained mutation events in evolutionary research. With this novel approach, biologists try to detect important patterns within mutation chain events of the HIV protease.

The mutation chain events can be represented by so-called "mutation graphs" (MGs) (see Fig. 2). Mutation graphs are labeled directed acyclic graphs, in which the nodes are amino acids located at particular protein's positions, e.g., G18: glycid at position 18. The directed edges reflect the mutation dependencies of the amino acids, i.e., inducing or repressing further mutations. Each mutation graph shows how the mutation of one amino acid (the root node on the top) induces a chain of the following mutations. As each amino acid in a protein might mutate individually, many mutation graphs per protein can emerge (see Fig. 1 top).

The biologists need to analyze all the mutation graphs in order to identify important mutation patterns. Patterns are parts of evolutionary chain reactions (subgraphs of MGs), which occur often or have a specific form or size (see Fig. 2). For example, large patterns can affect many positions in the protein. They imply a large influence on the protein. Thus, detecting patterns in the mutation graphs could allow biologists and drug developers to detect "evasive" mutations and can provide knowledge for new drug design strategies and target sites.

Many visual interactive tools have been developed for analyzing various types of data like genome or protein data [6, 7, 26, 30]. However, examining patterns in mutation graphs is a relatively novel field of research, therefore biologists and pharmacologists currently lack proper analytical tools. Especially, the available tools do not allow for visual analysis of patterns in multiple mutation graphs simultaneously.

We present a new approach for finding and exploring patterns in many mutation graphs. Our contributions and application benefits are:

- We present a fast pattern search algorithm, which finds frequent biologically relevant patterns in multiple mutation graphs (i.e., labeled directed acyclic graphs).
- We propose a novel interactive visualization system for the exploration of found patters from various perspectives: a global overview of the found patterns, a view on selected types of found patterns with additional information, as well as a view on locations of the found patterns in both mutation graphs and 3D protein structures. Note, the global overview is a new way of visualizing a large number of the found patterns.
- Our approach identifies frequent mutation patterns and key amino acids which are involved in multiple mutation chains. This helps the pharmacologist to estimate the side effects caused by a potential drug treatment.

Our software has been developed in close cooperation with biologists performing research in this field. We apply our new system on real-world data from the life science domain and evaluate its advantages for analyzing mutation chain events for an important protein in biomedicine (the HIV protease). The system extension and adaptation to biological use case and the evaluation have been conducted together with biological experts co-authoring the paper.

The paper is structured as follows. Section 2 provides details about the biological background and the tasks. Section 3 presents related work including pattern search and visual exploration of event propagation in networks. Section 4 gives a brief overview of our approach. Section 5 presents our pattern search algorithm, while Section 6 describes the interactive interface for pattern exploration. Section 7 demonstrates the approach in an use case with real world data. Section 8 concludes and outlines of possible future extensions.

## 2 Biologic Background and Tasks

Biologists wish to analyze the direct and implied effects of a mutation of amino acids in a protein using the so-called mutation graphs (MGs).

Mutation graphs are composed of amino-acids (nodes) connected by directed edges representing their co-evolutionary relationships. These relationships are of two types: inducing and repressing. This means, that an amino acid at one position in the protein can increase or decrease the occurrence of other amino acids at other positions (see blue (inducing) and red (repressing) edges in Fig. 2). Figure 2 shows a small example of several mutation graphs. On the left, Glycine on position 16 (G16) would induce a mutation to Lysin on position 70 (K70). This again would induce mutations to Argenine at the same position (R70) and Threonine at position 39 (T39).

Such inducing/repressing mutation reactions are calculated for each amino acid in a protein independently. This results in a set of mutation graphs (see Fig. 1 top). Thus, the biologists need to analyze hundreds of mutation graphs (see Sec. 7 as example of 246 MGs). The number of analyzed mutation graphs $N$ corresponds to the number of amino acids in a protein starting the mutation chain. As each amino acid can set off a mutation chain possibly affecting all amino acids in a protein, the analyzed mutation graphs can then altogether contain up to $N^2$ nodes.

The analyzed mutation graphs contain possibly interesting patterns (i.e., subgraphs or motifs), which need to be extracted and analyzed. A pattern is a subgraph of MG, which has specific meaning or occurs often in a dataset (see the highlighted areas in Fig. 2). Biologically, these patterns describe often occurring dependency routes corresponding to evolutionary trails and bottlenecks. Finding and analyzing of these patterns is in the focus of our co-authors from biology and, thus, also of this paper. Note that this kind of analysis is novel also in biologic research. Therefore, no ready to use tools are available to help finding and analyzing these patterns.

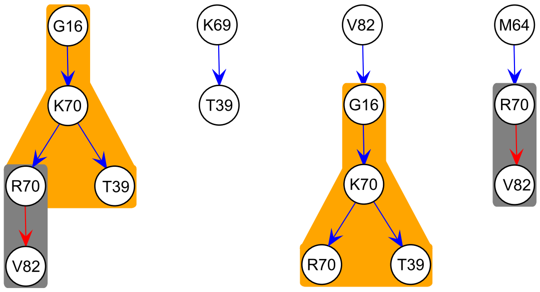


Fig. 2. Toy-example of four mutation graphs with potential patterns highlighted (colored background). The nodes represent amino acids, where the labels reflect their type and position. These graphs are calculated for each amino acid in the protein independently.

When analyzing patterns in mutation graphs, the following biologically relevant **tasks** emerge:

T1 **Distribution of all patterns**
Examining the distribution of all patterns is necessary for assessing of the magnitude of effects in mutation graphs. It summarizes which kinds of patterns were found in the data set and thus shows whether large or small patterns are present.

T2 **Identification of most prominent patterns**
Prominent patterns appear in many different mutation graphs. Highlighting frequent patterns is of special interest for the biologists. The frequent patterns give insight into "bottlenecks" of evolutionary events.

T3 **Identification of very complex or simple patterns**
Complex patterns explain higher order interdependencies. The analysis of complex patterns allows to inspect dependencies affecting large portions of the protein. While these large chain

reactions should be considered in drug development, targeting amino acids within these patterns can imply unfavorable results such as faster drug resistance mutations.

T4 **Structural localization of patterns**
The found patterns should not only be analyzed in isolation, but with respect to their location within the mutation graph and to their positions in the 3D representation of the protein as a physical object. Spatial position of the affected sites and their respective structural organization can give insight on the rationale behind the pattern. Position in mutation graphs allows researchers to identify patterns serving, e.g., as evolutionary bottleneck or as an entry to numerous subsequent mutations.

These tasks are the pivotal elements for the design and development of the analysis software described in the following section.

## 3 Related Work

There are many different publications and examples for visual interactive tools in the field of biology like [6, 7, 26, 30]. They focus on different areas then our paper (e.g., on genome). We present works closely related to our work: visual analysis of mutation chains (a kind of event propagation in networks). We reviews methods for finding patterns in networks and their visualization. We also discuss visual comparison of multiple graphs for identifying common subgraphs.

### 3.1 Pattern Search in Graphs

An exhaustive search for patterns (i.e., motifs) in a network is an NP-hard problem [8]. There are several algorithms for finding patterns in directed and undirected graphs. Recent reviews [37, 51] summarize and compare them. The available algorithms search for all unlabeled patterns in a network, which have pre-defined size $k$ ($k$ = number of nodes in a pattern). These tools include FANMOD [50], Mavisto [40], MFinder [27], [21], [24], MODA [16, 33]).

Available strategies for a speed-up the pattern search are: 1) better ways of building up larger patterns from previously found smaller subgraphs [24] 2) symmetry breaking, which eliminates the need for isomorphism checking [16, 33], or 3) sampling – heuristic approach [27, 50]. The above-mentioned algorithms find only patterns with a certain number of nodes or edges. They would not find all possible patterns with different sizes (esp., larger). This is a restriction for an exploratory analysis, which assumes no apriori knowledge about the dataset and its structure.

In our work, we use a labeled extension if the algorithm by Kuramochi [24]. We adapt this technique for labeled pattern search in many directed acyclic graphs. Our method proposes a fast combination of patterns. As we focus on an explorative analysis, we do not restrict the size of the patterns in advance.

### 3.2 Pattern Visualization in Graphs

The set of all patterns found in the network is commonly presented as a list of node link diagrams together with patterns' frequency as a number [25, 27, 40, 50]. These views were mainly developed for unlabeled patterns or patterns with few categories of nodes [25]. With the increasing number of different patterns found, a simple pattern list does not scale up. It requires a lot of scrolling. This problem is particularly severe for labeled graphs, which are in our focus. The number of possible labeled patterns is much larger then the number of unlabeled patters. The number of possible labeled patterns is a combination of the number of all possible pattern structures and the number of their possible label combinations (see Sec. 5.1 for details).

Another possibility of pattern visualization is highlighting of user-selected patterns in the network (e.g., [46]). For better readability of the pattern structure, specific pattern-based layouts have been developed [18, 22]. These technique do not scale up for large graphs or graphs with many patterns as each pattern needs to be shown individually.

Scalability of pattern exploration can be improved by pattern-based graph aggregation of node-link diagrams [14, 38, 46], iconic pattern presentation [25] or compression of adjacency matrices [13]. These methods decrease network size, however the aggregation of overlapping patterns, found also in our case, is problematic for all methods. Therefore, these techniques have only limited value added.

### 3.3 Visual Analysis of Event Propagation in Networks

Visual Analysis of event propagation in networks is a new topic, therefore few dedicated tools exist. Analysts mainly employ standard statistical software tools (e.g., Matlab) and charts available in standard visualization softwares (e.g., Pajek [3], or GraphViz [**?**]).

Three systems specialize on event propagation [31, 41, 45]. FNA (Financial network analysis) [41] focuses on analysis of financial transactions. It offers simple static visualization of networks. The users with good programming skills can run pre-defined simulations, however the tool does not provide methods for the exploration of results and their comparison. NBW (Network workbench tool) [31] is developed for analysis of gene regulation models – binary processes in networks. It offers an interface for defining model parameters and a static visualization of the result. It is not capable of comparing several inputs. Recently, von Landesberger et al. presented a system for visual analysis of contagion in networks [45]. The system offers simulation and interactive visualization of expansion processes. The resulting graphs are shown as node-link diagrams. The user has the possibility to select one or more nodes for analyzing their occurrence across the simulations. The system neither detects nor shows patterns. Moreover, this manual approach does not scale with the number of graphs. It is limited to a handful of graphs.

In sum, Event propagation analysis systems do not support finding and exploration of interesting patterns in many networks.

### 3.4 Visual Comparison of Multiple Graphs

Several visual comparison techniques also focus on exploration of commonalities among graphs [48].

Two compared graphs are usually shown next to each other, where the corresponding nodes or graph parts are highlighted. VisLink [10] and Holten et al. [19] link the same nodes of the two graphs. This technique does not show structural relationship (pattern) among highlighted nodes.

Archambault [2] identifies the common subgraphs of two graphs and then shows only the differences between them. Tu and Shen [43] create a unified graph of two trees and show it in a treemap. The unification however can produce large trees if the two compared graphs are very dissimilar. These methods are restricted only to two graphs. For several graphs, pairwise comparison leading to a quadratic number of comparisons would be required.

Several tree comparison techniques combine interactive visualization with highlighting of common subgraphs (e.g., [7, **?**, 29]. These techniques are however restricted to trees with common leaves, without labels in inner nodes. Therefore they are not suitable for our case.

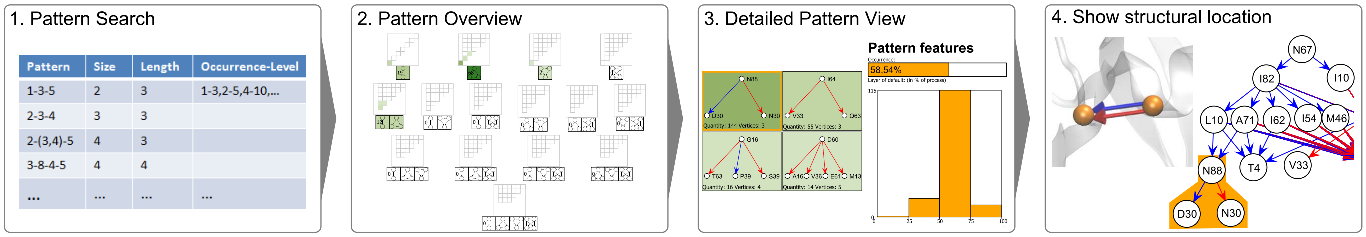
There are techniques for multiple graph comparison [15, 32, 47]. They only analyze global similarities without subgraph matching. They are not able to identify and show common patterns.

In general, current methods for visual graph comparison have restrictions (only pairwise comparison, only trees, only global similarity), which are not suitable for our use case.

## 4 Overview of Our Approach

Our approach combines a new specific algorithm for finding biologically relevant patterns in mutation graphs (MGs) with a novel interactive visual interface for pattern exploration (see Fig. 3). The interface shows the patterns on various levels of detail, addressing the scalability issue w.r.t. the number of patterns found and w.r.t. the number of graphs (MGs). The approach has been developed together with experts from biology so that it suits their analytical needs.

We first explain our pattern search algorithm (see Sec. 5) and then describe the interactive interface (see Sec. 6).



| Pattern | Size | Length | Occurrence-Level |
|---|---|---|---|
| 1-3-5 | 2 | 3 | 1-3,2-5,4-10,... |
| 2-3-4 | 3 | 3 | |
| 2-(3,4)-5 | 4 | 3 | |
| 3-8-4-5 | 4 | 4 | |
| ... | ... | ... | ... |

Fig. 3. Overview of our approach: (1) Pattern search results. (2) Overview of the number of patterns found. (3) Interesting patterns with additional information on their occurrence. (4) The location of selected patterns is examined directly in MGs and in 3D protein structures.

## 5 Patterns in Mutation Graphs and Pattern Search

### 5.1 Patterns in Mutation Graphs

Patterns are subgraphs of graphs with specific meaning or functional property [28] (see also Fig. 2). They are building blocks of networks, such as biological or financial networks.

This paper focuses on finding patterns in a set of $N$ mutation graphs (see Fig 1 top). Each MG has up to $N$ labeled nodes. MGs are directed acyclic graphs (DAGs) with a single root and unique node labels (see Fig. 2). A mutation graph has edges only between neighboring levels. These properties constrain the set of possible patterns.

We consider labeled patterns with unique labels (i.e., each node has a different label). Patterns are composed of several nodes, which can be on two neighboring levels (basic patterns) or on several levels (complex patterns). Figure 6 shows an example of a basic pattern (pink) and a complex pattern (orange). One pattern can be found in several MGs, however in each MG only once (due to uniqueness of labels).

**Pattern Labels** Labellings in mutation graphs and in their patterns are very important for the analysis, as amino acid position and type can give rise to varying effects. Labels, however, pose a scalability challenge to both pattern search and to the visualization. In fact, labels expand the number of possible patterns exponentially compared to unlabeled patterns. One unlabeled pattern with $k$ vertices can have the same structure as $\frac{|N|!}{(|N|-k)! \cdot k!}$ labeled patterns, where $|N|$ is the number of nodes in the original MG (see also Fig. 4).

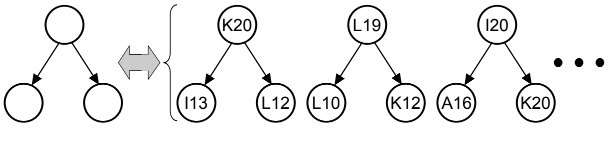


Fig. 4. One unlabeled pattern represents a large number of labeled patterns with the same structure.

**Basic Patterns** A basic pattern is a subgraph of MG, which stretches only between two neighboring levels (see the pink pattern in Figure 6). We focus on basic patterns with specific biological meaning for mutation expansion (see Fig. 5). This set was inspired by common patterns in biology and finance [28, 46, 47].

- *Line pattern* ($LP$, also called "single chain"): The mutation of one amino acid impacts only one other amino acid. These effects are desirable if a target amino acid should be influenced indirectly.
- *Fan-out pattern* ($FP$): The mutation of one amino acid influences numerous other amino acids. This might lead to unpredictable impact on the function of the whole protein.
- *Merge pattern* ($MP$): One target amino acid can be affected by multiple other amino acids. This might reveal potentially unstable amino acids.
- *Double-cross pattern* ($DCP$) This type combines the previous two types. Contained amino acids can have a significant influence on the mutation graph. However, as this networks are highly interconnected, they tend to be rather small, so that the resulting impact on the protein is limited and predictable.

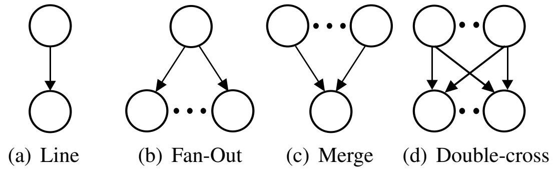

(a) Line (b) Fan-Out (c) Merge (d) Double-cross

Fig. 5. Types of biologically relevant basic patterns used in our work. Out of these 4, more complex patterns can be build.

We suppose, that the found basic patterns cannot be expanded to a larger pattern (e.g., from a line to a merge pattern).

**Complex Patterns** Complex patterns are a combination of two or more basic patterns (see Fig. 6). Two basic patterns may be combined only when the labeled root nodes of one pattern match the labeled leaves of the second pattern within the same MG. Biologically, complex patterns may represent multi-level dependencies between structurally distinct regions and can explain wide-spread evolutionary interactions.

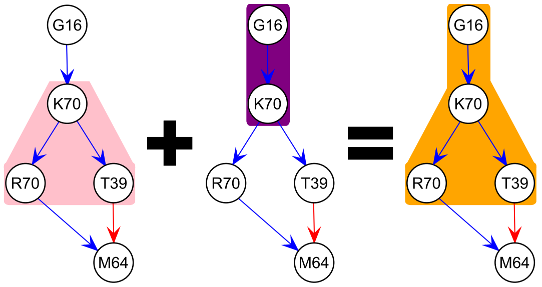


Fig. 6. Combination of two basic patterns (pink, Fan-Out and purple, Line) found in the same MG into a complex pattern (orange).

### 5.2 Pattern Search

In our work, we developed a special fast exhaustive pattern search algorithm for finding labeled patterns in the set of input MGs. In line with the requirements of biological analysis, the assumptions of the search algorithm are:

1. Unknown pattern size: We aim for exploratory analysis with no *a priori* assumption on maximum or minimum pattern size.
2. The number of graphs to analyze: The patterns need to be found in a possibly large set of mutation graphs. In a medium-sized data set, we need to analyze 400 MGs with up to 400 nodes each, leading to up to 160,000 nodes in total.
3. Special pattern types: We need to find biologically relevant patterns in mutation graphs (see Sec. 5.1).

Our algorithm is composed of two steps: 1) finding basic patterns and 2) iteratively combining them into larger complex patterns.

We first perform an exhaustive search for basic patterns (see Alg. 1) in the set of input mutation graphs $MGs$. For each node $v$ of a graph

$G \in MGs$, we check whether it is a part of a basic pattern $P$. The basic pattern search $getBasicPattern()$ uses the algorithm by Grochow et al. [16], as it has high performance for finding specific patterns. We store the found patterns $p$ in a set $foundBasicPatterns$. We also store in which MGs the pattern is located ($location(p) \rightarrow locBasicPattern$). This information is used for building complex patterns in the second step and for analyzing pattern occurrence.

**Algorithm 1** Find basic patterns

```
Require: $MGs$
Ensure: $foundBasicPatterns, locBasicPattern$
  $foundBasicPatterns \leftarrow \emptyset$
  $locBasicPattern \leftarrow \emptyset$
  for all $G \in MGs$ do
    for all $v \in G$ do
      for all $P \in \{LP, MP, FP, MP, DCP\}$ do
        $p \leftarrow getBasicPattern(v, G, P)$
        if $p \neq \emptyset$ then
          $foundBasicPatterns \leftarrow foundBasicPatterns \cup p$
          $locBasicPattern \leftarrow locBasicPattern \cup \{p, location(p)\}$
        end if
      end for
    end for
  end for
```

**Algorithm 2** Find complex Patterns

```
  $patterns \leftarrow foundBasicPatterns$
  $newPatterns \leftarrow foundBasicPatterns$
  $combPatterns \leftarrow \emptyset$
  $locPatterns \leftarrow locBasicPattern$
  for all $p1 \in patterns$ do
    for all $p2 \in newPatterns$ do
      if $p1 \neq p2 \wedge (|location(p1) \cap location(p2)| > 1)$ then
        if $(roots(p1) \cap leaves(p2) \neq \emptyset)$or $(leaves(p1) \cap roots(p2) \neq \emptyset)$ then
          $combp = combine(p1, p2)$
          $location(combP) \leftarrow location(p1) \cap location(p2)$
          $combPatterns \leftarrow combPatterns \cup combP$
          $locPatterns \leftarrow locPatterns \cup location(combP)$
        end if
      end if
    end for
    for all $cP \in combp$ do
      $removeSubPatterns(cP)$
    end for
    $patterns \leftarrow patterns - \{p1\}$
    $newPatterns \leftarrow newPatterns \cup combPatterns$
  end for
```

In the second step, the found labeled basic patterns $foundBasicPatterns$ are successively merged into larger and larger complex patterns until all maximal (largest) patterns have been found (see Alg. 2). During the search, location and frequency of all found complex patterns is stored. For faster calculation of complex patterns, we use the information on the location and the form of the base patterns found. Our algorithm relies on the uniqueness of node labels and the implied single occurrence of a pattern in a MG. Therefore, we do not need to go recursively through the graphs.

In our work, we wish to focus on larger patterns as they are more interesting then smaller patterns, if their have the same occurrence. Therefore, smaller patterns that form a larger pattern at the same locations (so-called "complete subpatterns") are removed in $removeSubPatterns$. This also reduces the number of patterns found for the further analysis. The constraint of same-location is important. Only if the subpattern occurs at the same locations as the larger pattern, then the existence of the smaller pattern does not bring any additional information to the user. An example is shown in Figure 7. The purple pattern is a complete sub-pattern of the orange pattern, its occurrence is redundant to the orange pattern. However, the pink pattern occurs more often then the orange pattern, so the removal of the pink pattern would lead to an information loss.

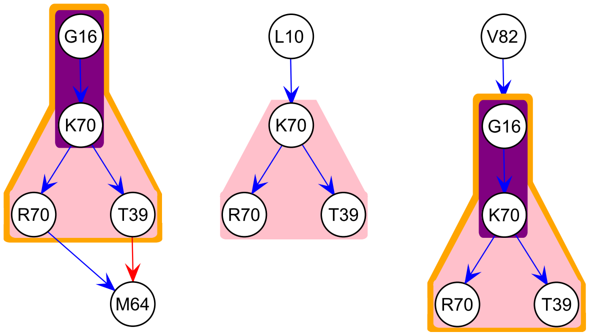


Fig. 7. Example for pattern removal: The purple pattern is removed as it is a full subpattern of the orange pattern. Both patterns occur simultaneously. The pink pattern occurs also in other MG, then orange pattern, so it carries extra information. Therefore, it is not removed.

**Theoretic complexity** The theoretic complexity of the pattern search is composed of two parts: 1) the search for basic patterns and 2) their combination. The basic pattern search combines breadth first search with the search for a specific basic pattern leading to the complexity of $O(|V| + |E|)^2 \cdot N$, where $|V|$ and $|E|$ are the maximum number of nodes and edges in the MGs. $N$ is the number of MGs. The second step has a worst case complexity of $O(log(|V|)+log(N)+p)$. This complexity is lower then that of common search algorithms $O(E^{k \cdot \frac{|V|!}{|V-k|!k!}} \cdot N)$ [1].

**Experimental runtimes:** We show the speed of our algorithm on mutation graphs of various numbers and sizes (see Table 1). The MGs were extracted from random inputs. We used a PC with Intel(R) Core(TM) i7 CPU 920 @ 2.67GHz, Java32bit.

| | p = 0.3 | | p = 0.4 | | p = 0.5 | |
|---|---|---|---|---|---|---|
| # MGs | # P | T(s) | # P | T(s) | # P | T(s) |
| 50 | 70 | 0.022 | 60 | 0.016 | 57 | 0.017 |
| 100 | 217 | 0.218 | 160 | 0.127 | 142 | 0.105 |
| 200 | 410 | 1.451 | 338 | 1.006 | 269 | 0.602 |
| 300 | 508 | 2.718 | 377 | 1.396 | 203 | 0.249 |
| 400 | 548 | 3.799 | 235 | 0.432 | 147 | 0.156 |
| 500 | 263 | 0.287 | 209 | 0.323 | 198 | 0.534 |

Table 1. Experimental runtimes. Average results (10 runs of each configuration). #P: the number of found patterns, T(s): the runtime in seconds. These results show that the runtime depends strongly on the number of found patterns, less then on the overall graph size.

## 6 Visual Exploration of Patterns

The visual interface (see Fig. 3) shows the found patterns (1) in several interactively linked views. The views were designed in cooperation with biologists so that they suit to their analytical needs.

The view guides the users through various levels of details (see Fig. 3). The number of found patterns (1) by type is shown in a coarse overview (2). Then the user explores advanced information on selected pattern types (2) and finally, she can analyze the location of the found patterns in the mutation graphs or in the 3D protein structure (3).

- *Pattern Overview*: This novel scalable view shows the number of patterns found in the dataset. It also allows to select certain types of patterns for further detailed inspection.

- *Detailed Pattern View*: It displays the user-selected patterns with additional analytical information (e.g., pattern's occurrence, or its position within MGs).
- *Pattern Locations*: These view show the locations of user-selected patterns in the mutation graphs and in 3D protein structures. The user can explore in detail, in which mutation graphs and where exactly the patterns are situated.

### 6.1 Pattern Overview

Pattern overview shows the number of found patterns by their composition in a compact scale-free representation (see Fig. 8). It allows the user to get an overview of which patterns are found and how often. It provides a first insight into the dataset.

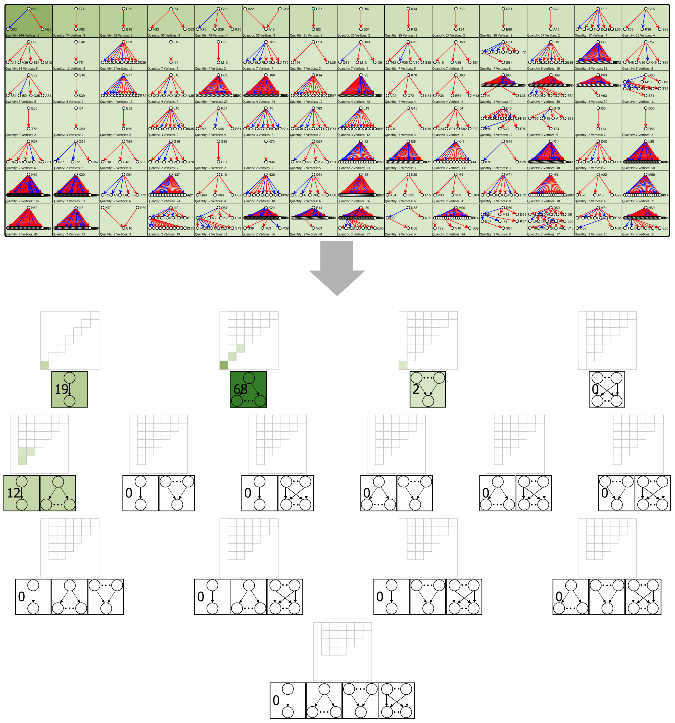


Fig. 8. Benefit of our Pattern Overview. Top: A large number of found patterns presented in a standard way. Patterns are difficult to compare and analyze. Bottom: Our pattern overview visualizes the same amount of information in a compact and structured way. The patterns are grouped by their structure. The number of found patterns is indicated by color from light to dark green (few to many patterns).

**Design:** This view was specially developed for scalability reasons, as the traditional approaches (esp. a list of patterns), did not scale up for our use case. It required a lot of scrolling, making a comparison of patterns difficult and time consuming.

Our idea was to compress the view on patterns in a way that the user still can see which patterns are found and how often. This was a difficult challenge. We first tried group pattern by their structure. For example, we grouped all line-line-merge patterns in one and all line-merge-line patterns into another group. This kind of grouping, however, still resulted in a large number of data to show, which increases exponentially with the number of basic patterns combined. Therefore, we decided to simplify the view even more.

The final approach groups the found patterns according to the types of combined basic patterns (see Fig. 9). We thereby disregard the order of combined patterns and their number. For example, line-merge and merge-line-line pattern would be together in one group line+merge. This simplification has one important advantage: the number of such groups is limited to 15 irrespective of the size and structure of the found patterns. There are 4 individual, 6 double, 7 tripple and 1 four pattern combination (see Fig. 8 bottom).

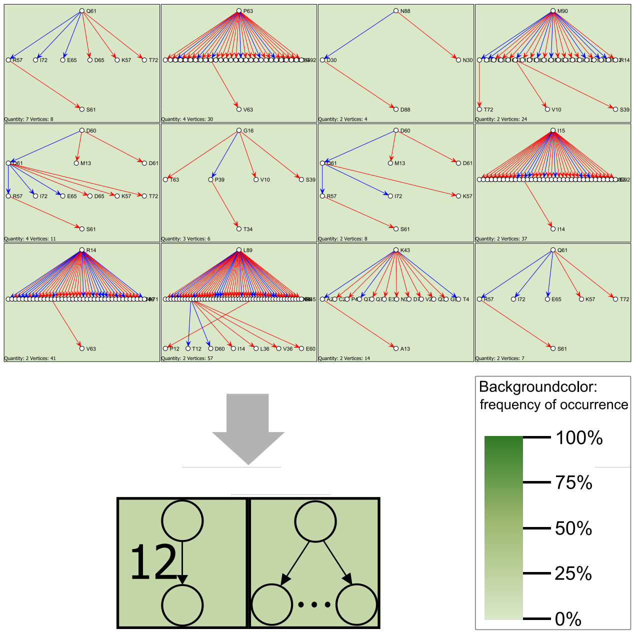


Fig. 9. Grouping patterns by their composition. The icon (bottom) shows the basic patterns, which are combined in the patterns (top). In this example the icon stands for all patterns which consists only of line and fan-out patterns. The number shows how many of this combinations were found (in this example 12).

The presented grouping allows us to create a scale-free visualization of the patterns. The pattern combinations are placed on the screen according to the number of combined patterns (see Fig. 8 bottom). As this compact representation hides inner structure of the patterns, we decided to design specific glyphs showing the user more insight into the structures and numbers of the found patterns (see Fig. 10).

Each glyph shows the number of found patterns in a group (icon) as well as more detailed information on the structure of patterns in the group (heatmap) (see Fig. 10).

- The icon shows the basic pattern types combine within a group (see Fig. 10 right). The color denotes the number of patterns found (light to dark green). The exact number is in the icon. Note that the color scheme can be changed on demand.
- Pattern heatmap (see Fig. 10 left) on the top shows the number of found patterns according to their structure. Each cell shows the number of patterns having a particular size (column) and a particular number of basic patterns combined (row). Green tone shows the number of found patterns. The white cells with gray outline show possible pattern combinations, where no patterns were found. The outer white areas correspond to impossible pattern type combinations (e.g., a larger size then the number of basic patterns combined).

**Interaction:** The user can use the heatmap for selecting interesting patterns for a detailed analysis (see Fig. 11). The user can select the patterns with a specific structure (a cell), with a specific size (row) or length (column). The selected patterns are shown either in the detailed view or are highlighted directly in the MGs.

### 6.2 Detail Pattern View

The detailed pattern view provides the user with important information on the exact form of the patterns as well as their distribution across the dataset (see Fig. 12). These views are important for the biological interpretation of the found patterns.

**Design:** The detailed pattern view has three linked parts:

The top left part of the detailed view shows the number of found patterns according to their type (see Fig. 12A). This provides a brief

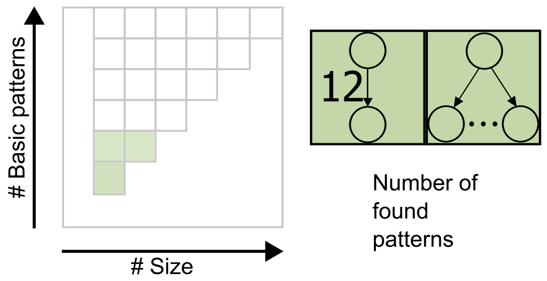


Fig. 10. Glyph showing the number of found patterns. Left: Heatmap shows the number of patterns according to their size and the number of combined basic patterns. Background color denotes the number of patterns found. Right: Icon shows the types of basic patterns combined within the group. The number and the background color indicates the number of found patterns. Color scheme: light to dark green meaning few to many.

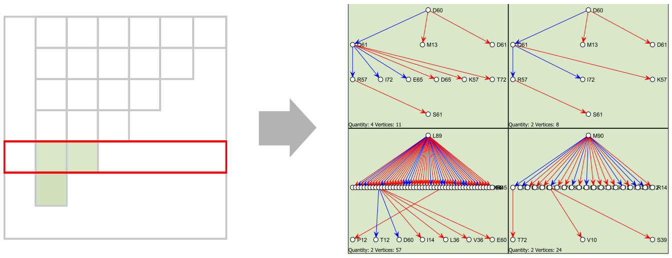


Fig. 11. Interaction with the overview: The user selects a set of patterns – a row in the heatmap – for their detailed inspection. In this example, the user selects all complex patterns, which consist of exactly three basic patterns and have the size of 3 or 4.

information on the number of patterns, which could be seen only in the overview.

The main part (see Fig. 12B) shows the exact form and labels of the found patterns as node link diagrams. The background color indicates the pattern frequency (light to dark green, few to many). This allows the user to inspect the patterns and extract their biological meaning. The list is sorted by user preferences: by frequency, size or number of basic patterns combined.

The information panel on the right (see Fig. 12C) shows analytical information on pattern occurrence and location of user-selected patterns. Pattern frequency is shown on the top as a bar. The full bar means occurrence in all MGs.

As requested by the users, we also included information on pattern location within MGs. The distribution of pattern locations within a MGs (hierarchy level) is shown as a histogram. It shows whether the pattern occurs rather at the beginning or at the end of the mutation chain. The patterns occurring at the beginning are more likely to cause multitude of pattern unrelated subsequent mutations, whereas patterns at the end of mutation chains can be viewed as evolutionary end points.

**Interaction:** The user can select one or more patterns in the main view for a detailed inspection in additional views (see Fig. 13). The selected patterns are highlighted by a colored border. Each pattern is assigned a different color. The colors are kept constant in all views. Note that we expect that the user focuses on a few patterns. If the user wishes to analyze more patterns, she can do so sequentially.

### 6.3 Pattern Locations Views

The most detailed views on exact locations of selected patterns in mutation graphs and in 3D protein structures were requested by the users. These views are needed for biological analysis of the data. The users wish to know exactly where the patterns are located in order to make inference from the data (see Fig. 13).

The view on the pattern locations within the graphs shows the filtered set of MGs – only those containing the selected patterns. This set is usually smaller then the input set of graphs. MGs are displayed as node-link diagrams using Sugyiama-style layout [42]. The selected patterns are highlighted using convex hulls. The hulls are inspired by the visualization of groups of nodes in graphs called bubble sets [11] and groups of points in scatterplots [39, 44]. For consistency, the hulls have the color assigned to the pattern in the detailed view.

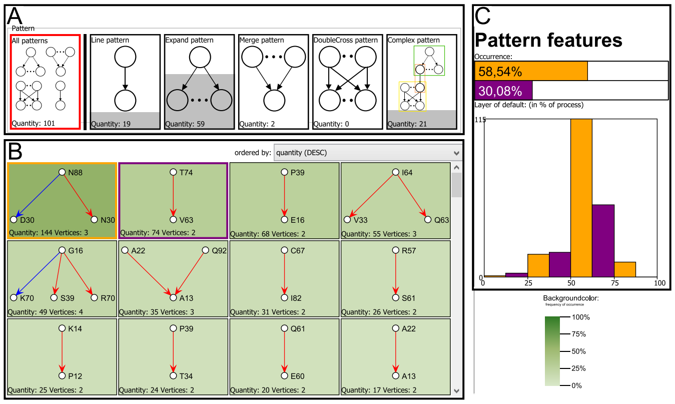


Fig. 12. Detailed Pattern View. A: Overview of the number of found patterns by basic types. B: The patterns found and their frequency indicated by background color. Dark green means many patterns found. C: Information panel showing additional analytical information for user-selected patterns.

The 3D structural view on the patterns shows the pattern as additional nodes and links within the protein structure. This view allows the biologist to examine also structural closeness of the mutated amino acids in the protein. It shows which part of the protein is affected by the mutation pattern and whether this pattern is compact or spreads across wide parts of the protein. 3D structures are visualized with VMD [20], as it is the common tool used in biological analysis. We extended the standard VMD view with the display of the found patterns. Figure 13 shows the patterns within the 3D structure as red and blue arrows between the corresponding amino acids.

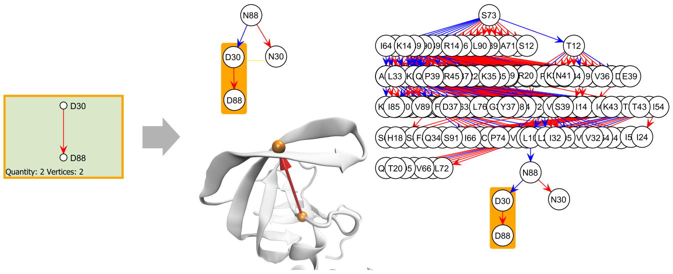


Fig. 13. Further exploration: For one or more selected patterns (left) in the Detailed Pattern View the pattern location can be shown in the Pattern Location View (right).

## 7 Application

We show the application of our tool to real world data as analyzed by biologists. The biologists were able to detect important patterns within mutation chain events of the HIV protease.

### 7.1 Motivation and Analytic Goal

The human immuno-deficiency virus 1 (HIV1) targets T helper cells of the immune system. After infection, the virus evntually destroys the target cells leading to the acquired immuno-deficiency syndrome (AIDS) and subsequent infection with HIV and AIDS associated diseases.

Modern therapies target a variety of essential HIV proteins such as the reverse transcriptase and the HIV protease (HIVP). While HIVP inhibition does not prevent infection of T helper cells, it prevents the formation of a functional virion and, in theory, should also stop HIV from proliferating within the infected organism. This may not be successful due to a high variability of HIV genes. These so-called viral quasi-species are capable of adapting to additional evolutionary pressure induced by drugs which thus leads to the evolution of drug-resistant viruses. This eventually increases the failure rate of a treatment.

A large number of HIVP data are available for analysis. The analysis based on such wide data base can lead to robust evolutionary insights. Especially, the analysis of patterns in the mutation chains can reveal causally determined evolutionary relationships beyond simple pairwise dependencies.

The new insights on mutation patterns could enable the design of new combined drug therapies targeting a multitude of connected residual positions to further limit HIVs elusiveness against long-term drug treatment.

### 7.2 Input Data

The input data are HIVP sequences. Each sequence has exactly 99 amino acids. The data were obtained from the HIV drug resistance database [36]. It provides data on more than $65,000$ HIVP sequences. The data is gained from both drug treated and untreated patients. As data curation, we omitted sequences with non-canonical amino acids, since they might indicate imprecise or wrong sequencing results, resulting in $34,747$ sequences.

The data for the analysis in this use case was pre-processed in two steps: first calculating the initial correlated mutation graph and then extracting the mutation chains in form of mutation graphs (MGs) for all amino acids in the protein.

As initial step, so-called correlated mutation graphs were constructed. We used the protocol by [4]. The number of minimal sequences present in any sub-alignment was set to 2. The frequency threshold was set to $\Delta f = 0.15$. We used unweighted graphs consisting of edges the revealed $p$-values of 0.05 or less. We then corrected for multiple-testing effects. Here, we decided to use the conservative Bonferroni-correction which avoids problems with dependent hypotheses on co-evolutionary signatures among the various higher-order correlations present in molecular evolution [49]. The number of tests was $n_t := N_a \cdot N_a \cdot N \cdot (N-1)$, where $N$ is the number of amino acid positions due to the lack of symmetry in amino acid pairs $(i,j) \leftrightarrow (j,i)$; and $N_a = 20$ is the number of naturally occurring of amino acid types. Thus, $n_t = 3,880,800$. We have therefore used an effective $p$-value threshold of $10^{-2}/n_t \approx 10^{-8}$ to construct the resulting directed unweighted graph. It contained 246 vertices and $3,233$ edges.

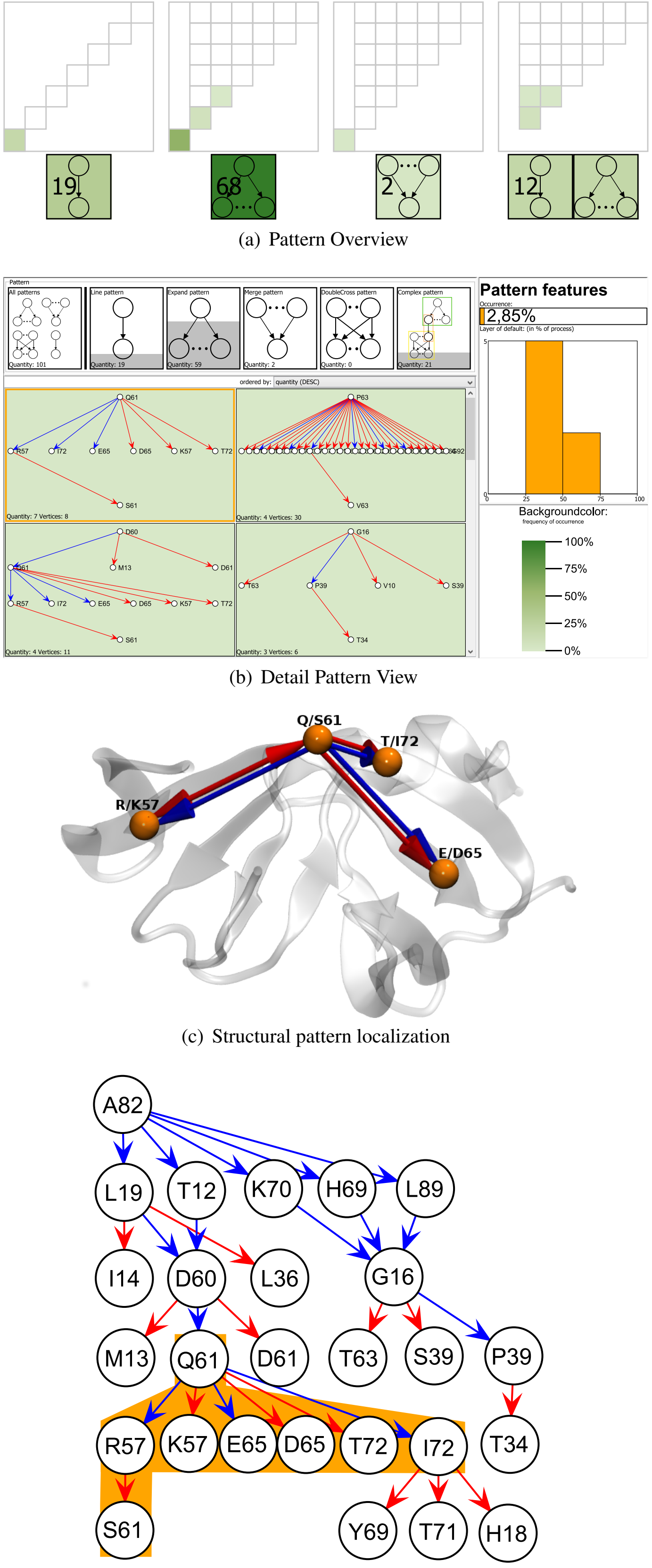
In the second step, we extracted the dataset of mutation graphs for the visual pattern analysis. It was calculated from the initial correlated mutation graph using mutation chain algorithm provided by the biologists. We got a set of 246 mutation graphs (MGs). The graphs had between 1 and 203 nodes, having from 0 to 610 edges each. This graph data set serves as input for the analysis of patterns.

Graph computation used R [35] and the iGraph [12].

### 7.3 Use Case

Our pattern search algorithm found 101 patterns in the input mutation graphs. The overview of all found patterns is shown in Figure 14(a). The most prominent group of patters is composed fan-out patterns (68, highlighted in dark green (**T1**). These fan-out patterns have varying number of levels (see green cells in the heatmap in Figure 14(a)). Mutation fan-out patterns with a larger number of levels can be seen as more complex interaction cascades, which could contain information on long ranging relationship networks.

Figure 14(a) shows many potentially relevant patterns, which should be analyzed in detail (**T2**). Due to page limitation, we show two prominent examples selected by the biologists.



(a) Pattern Overview

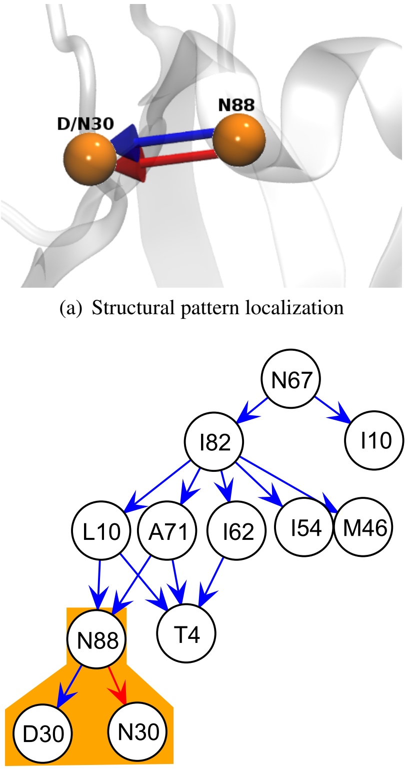
(b) Detail Pattern View

(c) Structural pattern localization

(d) Graph pattern localization

Fig. 14. Use Case workflow. Starting with a summary of all found patterns (a), selected pattern types are shown in the detailed pattern view (b). The user then selects most interesting patterns for exploration of their locations both in the 3D protein structure (c) and within their respective mutation graphs (d).

First, the expert focuses on complex patterns as they reveal an interplay of multiple biologically relevant interactions. Figure 14(b) highlights the most prominent pattern in the top left corner (see the pattern



(a) Structural pattern localization

(b) Graph pattern localization

Fig. 15. (a): Structural view of line pattern connecting position 88 to 30. (b) Position of a fan out pattern within a mutation graph.

highlighted with an orange rectangle). This pattern is composed of a fan-out and a line pattern. Interestingly this pattern occurs solely in the middle of mutation graphs (**T4**), neither at the beginning nor at the end of the mutation chain (see the orange histogram of occurrences on the right of the image). This implies that this mutation pattern can be thought as an evolutionary bottleneck for $2.85\%$, i.e. 7 out of 246, of all mutation graphs (see the relative number of pattern's occurences shown in Fig. 14(b), top right).

Biologically, the complex pattern highlighted in Figure 14(d) describes the influence of the occurrence of glutamine at position 61. This Gln61 is responsible for two different effects on three positions in the HIV protease (positions 57, 65 and 72) (**T3**). Hereby the first level influence on each of the three positions is both inductive and inhibitive. In detail, this means that at position 72 the occurrence of the hydrophobic isoleucine is positively influenced whereas the polar threonine is inhibited. Therefore, the polar uncharged Gln61 increases hydrophobicity at this position.

At position 57, a quite different effect can be observed. Here, Gln61 increases the probability of arginine and reduces the emergence of lysine. Since both arginine and lysine are positively charged, no change in net surface charge and polarity arises. Nevertheless, arginine with its guanidinium group can form more H-bonds than lysine and serves to increase the stability of the protein via additional salt bridge interactions [5] (**T4**). A similar effect can be observed at position 65 where the occurrence of the negatively charged glutamic acid is favored while also negatively charged aspartic acid can be observed less frequent. The only difference between these two amino acids is the length of the side chain. Furthermore, the bulkier side chain group of glutamic acid has a slightly smaller pKa value than aspartic acid. This could lead to altering responses to solvent interactions. Interestingly, second level effects are induced by the Arg57 which in turn influences position 61 by reducing the occurrence of serine and thus leading to mutual dependencies on both sites.

Altogether, Gln61 influences residues in the cantilever region (position 65 and 72) and the flaps of HIVP (position 57) hence forming an evolutionary pattern connecting these two regions. This region has already been shown to be functionally involved in the opening and closing of the flaps by the means of compression and extension [34] (**T4**). More recently, this unique property of the exo region near Gln61 has been used for HIVP inhibitor design [9, 23], further emphasizing the importance of understanding possible evolutionary dependencies.

The second biologically relevant example, is the most frequently observed pattern (see the pattern highlighted with an orange rectangle in Fig. 12). This pattern is observed in over $58\%$ of all mutation graphs (see Fig. 12 top right) (**T2**). A detailed view on 3D structure in Figure 15(a) shows that asparagine at position 88 is located in an $\alpha$-helical structure. This residue exerts an inhibitory and an enhancing evolutionary effect on position 30 (see Fig. 15) (**T4**). More exactly, the probability to find aspartic acid is increased, while structurally highly similar asparagine is disfavored. Due to the change in net charge at this site, the flexibility of the loop region connecting the HIVP active site (Asp25 – Thr26 – Gly27) with the subsequent $\beta$-sheet could be directly affected.

The above described patterns – the most frequent fan-out and complex patterns – serve as selected examples from the multitude of identified patterns. More patterns, which could not be shown due to space limitations could lead to further insights into how and why specific drug resistances arise, which drug interactions can be easily avoided by the HIVP or on general evolutionary dynamics in HIVP. In this way, our approach may enable researchers to dissect evolutionary dependencies in regions with high mutual information [17] and study future drug therapy approaches.

## 8 Conclusions and Future Work

We presented a new approach for the visual analysis of patterns in mutation graphs, which represent mutation chain reactions of amino acids in proteins. Many such graphs result from modifications of individual amino acids, e.g., through drug treatment. The finding and exploration of often occurring patterns in the mutation graphs is an important part of pharmacological drug development or evolutionary research.

Our approach combines an adapted algorithm for pattern search in multiple mutation graphs and an interactive interface for pattern exploration in several linked views. As the number of patterns can be large, we presented a novel overview of the number of found patterns grouped by structure. In this view, the user can select interesting pattern types for their detailed analysis and for the exploration of their locations in the mutation graphs and in 3D protein structures.

We have evaluated our system on real world data of HIV protease mutations. The results show interesting insights into mutations of this protein. These insights initiate future biological investigations.

In the future, we would like to extend our approach in several ways. The biologists would be interested in new ways of comparing patterns across several datasets. It would be also interesting to include additional biological meta-information (e.g., charge, secondary structure information, positions affected by drug application) in the analysis. From visualization point of view, we would like to develop an extended pattern-centric graph layout, which emphasizes the patterns found in acyclic directed graphs. Our approach could be applied to further application areas, which analyze chain reactions in networks, such as information cascades or disease spreading.

## Acknowledgments

The authors gratefully acknowledge financial support by the Deutsche Forschungsgemeinschaft under grants no. HA 5261/3-1 and LA 3001/1-1.

## References

[1] N. Alon, R. Yuster, and U. Zwick. Finding and counting given length cycles. *Algorithmica*, 17(3):209–223, 1997.

[2] D. Archambault. Structural differences between two graphs through hierarchies. In *Proceedings of Graphics Interface 2009*, pages 87–94. Canadian Information Processing Society, 2009.

[3] V. Batagelj, V. Batagelj, A. Mrvar, and A. Mrvar. Pajek - analysis and visualization of large networks. *Mathematics and Visualization: Graph Drawing Software*, pages 77–103, 2003.

[4] L. Bleicher, N. Lemke, and R. C. Garratt. Using amino acid correlation and community detection algorithms to identify functional determinants in protein families. *PLoS One*, 6(12):e27786, 2011.

[5] C. Borders, Jr, J. A. Broadwater, P. A. Bekeny, J. E. Salmon, A. S. Lee, A. M. Eldridge, and V. B. Pett. A structural role for arginine in proteins: multiple hydrogen bonds to backbone carbonyl oxygens. *Protein Sci*, 3(4):541–548, Apr 1994.

[6] S. Bremm, T. Schreck, P. Boba, S. Held, and K. Hamacher. Computing and visually analyzing mutual information in molecular co-evolution. *BMC Bioinformatics*, 11(1):330, 2010.

[7] S. Bremm, T. von Landesberger, M. He, T. Schreck, P. Weil, and K. Hamacher. Interactive visual comparison of multiple trees. In *IEEE Visual Analytics Science and Technology*, pages 31–40, 2011.

[8] S. Bruckner, F. Hüffner, R. M. Karp, R. Shamir, and R. Sharan. Torque: topology-free querying of protein interaction networks. *Nucleic acids research*, 37(suppl 2):W106–W108, 2009.

[9] M. W. Chang, M. J. Giffin, R. Muller, J. Savage, Y. C. Lin, S. Hong, W. Jin, L. R. Whitby, J. H. Elder, D. L. Boger, and B. E. Torbett. Identification of broad-based hiv-1 protease inhibitors from combinatorial libraries. *Biochem J*, 429(3):527–532, Aug 2010.

[10] C. Collins and S. Carpendale. Vislink: Revealing relationships amongst visualizations. *IEEE TVCG*, 13(6):1192–1199, 2007.

[11] C. Collins, G. Penn, and S. Carpendale. Bubble sets: Revealing set relations with isocontours over existing visualizations. *IEEE TVCG*, 15(6):1009–1016, Nov.-Dec.

[12] G. Csardi and T. Nepusz. The igraph software package for complex network research. *InterJournal*, Complex Systems:1695, 2006.

[13] K. Dinkla, M. Westenberg, and J. van Wijk. Compressed adjacency matrices: Untangling gene regulatory networks. *Visualization and Computer Graphics, IEEE Transactions on*, 18(12):2457–2466, Dec.

[14] C. Dunne and B. Shneiderman. Motif simplification: improving network visualization readability with fan, connector, and clique glyphs. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 3247–3256. ACM, 2013.

[15] M. Freire, C. Plaisant, B. Shneiderman, and J. Golbeck. ManyNets: an interface for multiple network analysis and visualization. In *Proc. of Int. Conf. on Human factors in computing systems*, pages 213–222, 2010.

[16] J. Grochow and M. Kellis. Network motif discovery using subgraph enumeration and symmetry-breaking. In *Research in Computational Molecular Biology*, volume 4453, pages 92–106. Springer, 2007.

[17] K. Hamacher. Relating sequence evolution of hiv1-protease to its underlying molecular mechanics. *Gene*, 422(1-2):30–36, Oct 2008.

[18] P. Holleis, T. Zimmermann, and D. Gmach. Drawing graphs within graphs. *J. of Graph Algorithms and Applications*, 9(1):7–18, 2005.

[19] D. Holten and J. Van Wijk. Visual comparison of hierarchically organized data. In *Computer Graphics Forum*, volume 27, pages 759–766, 2008.

[20] W. Humphrey, A. Dalke, and K. Schulten. Vmd: visual molecular dynamics. *J Mol Graph*, 14(1):33–8, 27–8, Feb 1996.

[21] N. Kashtan, S. Itzkovitz, R. Milo, and U. Alon. Efficient sampling algorithm for estimating sub-graph concentrations and detecting network motifs. *Bioinformatics*, 20:1746–1758, 2004.

[22] C. Klukas, F. Schreiber, and H. Schwöbbermeyer. Coordinated perspectives and enhanced force-directed layout for the analysis of network motifs. In *Asia-Pacific Symposium on Information Visualisation*, pages 39–48, 2006.

[23] J. Kunze, N. Todoroff, P. Schneider, T. Rodrigues, T. Geppert, F. Reisen, H. Schreuder, J. Saas, G. Hessler, K.-H. Baringhaus, and G. Schneider. Targeting dynamic pockets of hiv-1 protease by structure-based computational screening for allosteric inhibitors. *J Chem Inf Model*, 54(3):987–991, Mar 2014.

[24] M. Kuramochi and G. Karypis. An efficient algorithm for discovering frequent subgraphs. *Knowledge and Data Eng., IEEE Trans. on*, 16(9):1038 – 1051, sept. 2004.

[25] E. Maguire, P. Rocca-Serra, S.-A. Sansone, J. Davies, and M. Chen. Visual compression of workflow visualizations with automated detection of macro motifs. *IEEE TVCG*, 19(12):2576–2585, 2013.

[26] M. Meyer, T. Munzner, and H. Pfister. Mizbee: a multiscale synteny browser. *IEEE TVCG*, 15(6):897–904, 2009.

[27] MFinder and MDraw. `http://www.weizmann.ac.il/mcb/UriAlon/groupNetworkMotifSW.html` (accessed 5.6.2009).

[28] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network motifs: simple building blocks of complex networks. *Science Signalling*, 298(5594):824, 2002.

[29] T. Munzner, F. Guimbretière, S. Tasiran, L. Zhang, and Y. Zhou. Treejuxtaposer: scalable tree comparison using focus+context with guaranteed visibility. *ACM Trans. Graph.*, 22:453–462, July 2003.

[30] K. T. Nguyen and T. Ropinski. Large-scale multiple sequence alignment visualization through gradient vector flow analysis. In *IEEE Biological Data Visualization*, pages 9–16. IEEE, 2013.

[31] NWB Team. Network workbench tool. Indiana University, Northeastern University, and University of Michigan, 2006. http://nwb.slis.indiana.edu, accessed on 10.10.2012.

[32] T. Nye. Trees of trees: an approach to comparing multiple alternative phylogenies. *Systematic biology*, 57(5):785, 2008.

[33] S. Omidi, F. Schreiber, and A. Masoudi-Nejad. Moda: an efficient algorithm for network motif discovery in biological networks. *Genes & genetic systems*, 84(5):385–395, 2009.

[34] A. L. Perryman, J.-H. Lin, and J. A. McCammon. Hiv-1 protease molecular dynamics of a wild-type and of the v82f/i84v mutant: possible contributions to drug resistance and a potential new target site for drugs. *Protein Sci*, 13(4):1108–1123, Apr 2004.

[35] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2014.

[36] S.-Y. Rhee. Human immunodeficiency virus reverse transcriptase and protease sequence database. *Nucleic Acids Res.*, 31(1):298303, 2003.

[37] P. Ribeiro, F. Silva, and M. Kaiser. Strategies for network motifs discovery. In *IEEE Int. Conf. on e-Science*, pages 80–87. IEEE, 2009.

[38] L. Royer, M. Reimann, B. Andreopoulos, and M. Schroeder. Unraveling protein networks with power graph analysis. *PLoS computational biology*, 4(7):e1000108, 2008.

[39] T. Schreck and C. Panse. A new metaphor for projection-based visual analysis and data exploration. In *Electronic Imaging 2007*, pages 64950L–64950L. International Society for Optics and Photonics, 2007.

[40] F. Schreiber and H. Schwobbermeyer. Mavisto: a tool for the exploration of network motifs. *Bioinformatics*, 21:3572–3574, 2005.

[41] K. Soramäki. Financial network analysis (FNA), 2012. http://www.financialnetworkanalysis.com/, accessed on 15.10.2012.

[42] K. Sugiyama, S. Tagawa, and M. Toda. Methods for visual understanding of hierarchical system structures. *Systems, Man and Cybernetics, IEEE Transactions on*, 11(2):109–125, 1981.

[43] Y. Tu and H.-W. Shen. Visualizing changes of hierarchical data using treemaps. *Visualization and Computer Graphics, IEEE Transactions on*, 13(6):1286–1293, 2007.

[44] T. von Landesberger, S. Bremm, P. Rezaei, and T. Schreck. Visual analytics of time dependent 2D point clouds. In *Computer Graphics International*, pages 97–101, 2009.

[45] T. von Landesberger, S. Diel, S. Bremm, and D. W. Fellner. Visual analysis of contagion in networks. *Information Visualization*, 2013.

[46] T. von Landesberger, M. Görner, R. Rehner, and T. Schreck. A system for interactive visual analysis of large graphs using motifs in graph editing and aggregation. In *Proceedings of Vision Modeling Visualization Workshop*, pages 331–339, 2009.

[47] T. von Landesberger, M. Görner, and T. Schreck. Visual analysis of graphs with multiple connected components. In *Proc. IEEE Symp. on Visual Analytics Science and Technology*, pages 155–162, 2009.

[48] T. von Landesberger, A. Kuijper, T. Schreck, J. Kohlhammer, J. van Wijk, J.-D. Fekete, and D. Fellner. Visual analysis of large graphs: State-of-the-art and future research challenges. *Computer Graphics Forum*, 30:1719–1749, 2011.

[49] M. Waechter, K. Jaeger, D. Thuerck, S. Weissgraeber, S. Widmer, M. Goesele, and K. Hamacher. Using graphics processing units to investigate molecular coevolution. *Concurrency and Computation: Practice and Experience*, 26(6):1278–1296, 2014.

[50] S. Wernicke and F. Rasche. FANMOD: a tool for fast network motif detection. *Bioinformatics*, 22(9):1152–1153, 2006.

[51] E. Wong, B. Baur, S. Quader, and C.-H. Huang. Biological network motif detection: principles and practice. *Briefings in bioinformatics*, 13(2):202–215, 2012.