Biological Image Indexing for Content-Based Retrieval of Drug Effects in Phenotypic Screening Data of Macroparasites

Ahmed Gater[1] and Rahul Singh[1,2,†]

[1]*Department of Computer Science, San Francisco State University,* [2]*Center for Discovery and Innovation in Parasitic Diseases, University of California, San Francisco, CA*

Abstract

Phenotypic-screening involves systematically assessing the therapeutic effects of a set of molecules by exposing entire disease systems to them and observing, through imaging, the effects of the compounds. Phenotypic assays typically generate hundreds of thousands to millions of images. An unmet challenge in this setting is to identify similar phenotypic effects caused by molecules, which may potentially be structurally different. While phenotypes can be compared using their feature vectors, real-time querying of these data sets becomes a challenging task because of the size of the data sets and the high dimensionality of the feature vectors. In this paper, we present an indexing approach that seeks to address this problem and allows efficient query-retrieval of phenotypic drug effects.

1. Introduction

Neglected tropical diseases (NTDs), especially those caused by helminths, constitute some of the most common infections of the world's poorest people. The major diseases within this class include schistosomiasis, lymphatic filariasis, and onchocerciasis. Amongst the helmintic diseases, the impact of schistosomiasis is especially significant, and impacts more than 200 million people, with an additional 800 million at risk. The World Health Organization (WHO) considers schistosomiasis a disease for which new treatments are urgently needed.

One of the first steps in modern drug discovery involves screening compounds for lead identification. The goal is to identify molecules that show efficacy against the disease and can be considered for further optimization/study. Screening can either be target-based or phenotypic. In the former case, a purified protein target is used in in-vitro settings to find the compounds that interact with it. In the latter case, the entire disease system is involved. For example, in the context of parasitic diseases, a drug is administered to the live parasite and then its time-course effects are monitored. Such effects, or *phenotypes*, can be expressed in terms of changes in the shape, appearance, motion, or behavior of the parasites. Phenotypic screens are inherently holistic – that is, such screens may identify drugs whose efficacy derives from one or more of unknown molecular mechanisms of action. The information captured in phenotypic assays is complex, encompassing multiple time-varying phenotypes. While the complete richness of such data is yet to be fully exploited, there is evidence that, phenotypic assays are significantly more effective than target-based screens [1]. While the concept of automated whole organism phenotypic screening against parasitic diseases is relatively new, starting with the seminal work of Singh *et al.* [2], a number of results have been recently published [3-5].

The ability to compare phenotypes is a significant challenge that impedes utilizing the full richness of the data arising from phenotypic assays of complex macroparasites. Were this critical problem to be solved, a scientist could seek out, for instance, compounds that induce similar phenotypes regardless of their chemical structure. Such ability could be very powerful, among others, in identifying compounds that lead to similar therapeutic effects even if they belong to different structural classes – something that is very difficult to do today. Figure 1 shows an example of discovering such kind of knowledge through searching a database of parasite images in order to retrieve the parasites exhibiting similar phenotypes to a query (parasite).

Phenotypic assays involving macro-parasites typically involve multiple genetically heterogeneous individuals and test a number of compounds. The output of a typical screening campaign can therefore yield very large image data sets. For example, the data set used in this paper involves 500 thousands images of parasites exposed to different drugs. Even larger data sets with hundreds of millions of images are also possible. Furthermore, each of these images is represented by a high-dimensional feature vector.

† Corresponding author. E-mail: rahul@sfsu.edu

1063-7125/14 $31.00 © 2014 IEEE
DOI 10.1109/CBMS.2014.19

While, phenotypes can be compared by matching their vector space representations, the large size of image datasets and high dimensionality of the data poses a significant obstacle for executing real-time comparisons. While the existing indexing techniques have proven to be inefficient for high-dimensional space [6], we propose in this paper a high-dimensional indexing strategy that outperforms the existing methods in this problem context.

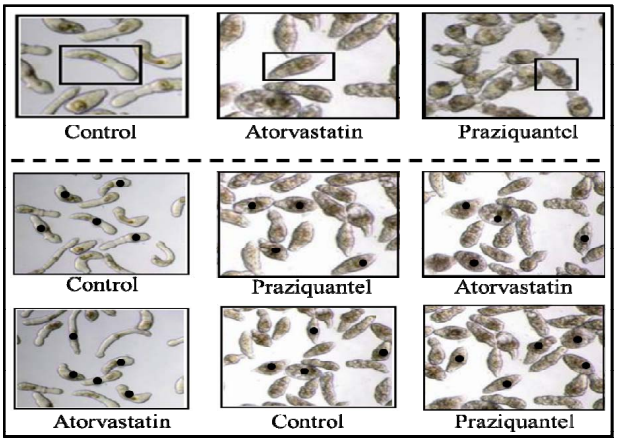

Figure 1 Example of searching a database using three parasite images (shown surrounded by a rectangle) along with the agents to which they were exposed. Examples of content-based retrievals corresponding to the queries are shown below with dots indicating the parasites found to be similar by our approach. The figure presents examples where the retrieved data may be from parasites exposed to a different compound/condition than the query.

2. Parasite Detection, Representation and Matching

In our approach, the image data is first segmented [8], and then the shape, appearance, and motion features of the parasites are captured and summarized as quantized multidimensional distributions called histograms [3]. Given a collection of normalized histograms $H=\{h_1, ..., h_L\}$, our goal is to develop a search technique that allows quickly retrieving all the histograms which are within a given distance to a query. In this paper, we use the term coefficient to refer to a value of a given bin, and $h_i[j]$ refers to the coefficient of the j^{th} bin of a histogram h_i. A traditional approach used to assess the similarity of two histograms h_l and h_i is based on the notion of minimum histogram intersection, which is formally defined as:

$$S_H(h_i,h_l) = \sum_{j=1}^{N} \min(h_i[j],h_l[j]) \quad (1)$$

The search technique we propose in this paper is an approximate method in the sense that the similarity computed in the index space is not necessarily equal to that computed in the raw histogram space. Thus, in order to avoid missing qualified matches, the similarity measure defined in the index space (S_I) has to be an upper bound of S_H as defined in Eq. (2). The index-based similarity measure proposed by us conforms to this property (proof omitted for brevity).

$$\forall (h_i,h_l) \in H \times H, S_I(h_i,h_l) \geq S_H(h_i,h_l) \quad (2)$$

3. Hierarchical Subspace Histogram Indexing

Our indexing technique consists of three steps, namely: *histogram approximation*, *histogram mapping*, and *hierarchical partitioning*. The first step consists of approximating the coefficients of all the histograms in the database, at each bin, by a reduced set of values. The approximation is done by dividing the coefficient space of each bin into equidistant sub-intervals whose bounds are considered as the representative values of all the bin coefficients that are within their range. The partitioning of coefficient space of a bin j is an ordered set of intervals $I_j=\{[l_{j,1},u_{j,1}[, [l_{j,2},u_{j,2}[, \ldots, [l_{j,M},u_{j,M}]\}$, where the upper bound of one interval is equal to the lower bound of the next interval, i.e. $u_{j,k} = l_{j,(k+1)}$. Thus, given a bin space partitioning I_j of a bin j, a coefficient $h_i[j]$ of histogram h_i is associated with the interval $[l_{j,m},u_{j,m}[$, when $l_{j,m} \leq h_i[j] < u_{j,m}$.

In the second step, each histogram is mapped to a new representation, called signature, which is derived as follows: each bin coefficient of a histogram is mapped to the identifier of the sub-interval to which it belongs in that bin. Formally, given a partitioning of the bins, a histogram h_i is mapped to a signature $\hbar_i$ where: $\hbar_i[j]=Id(g(h_i[j]))$, where $Id([a..b])$ denotes the identifier of the interval $[a..b]$ and $g(h_i[j])$ denotes the identifier of the interval to which $h_i[j]$ is assigned at the j^{th} bin. Table 1 shows a set of histograms and their signatures.

Finally, the signatures of the histograms are hierarchically partitioned using subspace clustering. The signatures are first divided into several clusters each of which is defined by a set of signatures it covers along with a set of bins and their interval identifiers involved in obtaining that cluster. That way, the signatures covered by a particular cluster have the same interval identifiers in the bins defining that cluster. For instance, the signatures shown in Table 1 are divided into 2 clusters C1 and C2 (see Figure 2), where C1 is obtained using the bins 2 and 3 with interval identifiers 5 and 11 and covers the histograms h_1 to h_8; and C2 is obtained using bins 2, 3 and 4 with interval identifiers 8, 9, and 13 respectively. Each cluster is then recursively divided into sub-clusters using the bins that are not involved in any of its ascending clusters**.** For instance, the signatures covered

by C1 are clustered; resulting in cluster C1.1, which covers signatures h_1 to h_4; and cluster C1.2, which covers signatures h_5 to h_8. The hierarchical clustering ends either when no bins are available that allow further divisions into sub-clusters or if a cluster has a very small number of signatures left in it. This recursive process leads to a tree structure where child clusters explore new bins which can improve the discrimination of the contents of their parents. It should be noted that the internal nodes store only the bins and the interval identifiers, while the leaf nodes store both the histogram identifiers and the remaining bins necessary to complete the similarity computation. The clusters at each level are computed using a modified version of the Proclus algorithm [9].

Table 1: collection of 10 histograms over 4 bins. The signatures are obtained with the partitioning{[0.1, 0.35[, [0.35, 0.6 [, [0.65, 0.85]} for bin 1, {[0,0.2[,[0.2,0.4[,[0.4, 0.6]} for bin 2, {[0, 0.2[, [0.2, 0.4[, [0.4, 0.6]} for bin 3, and {[0, 0.15[, [0.15,0.35[, [0.35,0.5]} for bin 4.

Hist. Id	Histogram Content	Signature
h_1	(0.65, 0.15, 0.1, 0.1)	(3,4, 7, 10)
h_2	(0.6, 0.2, 0.06, 0.14)	(3, 4, 7, 10)
h_3	(0.7, 0.07, 0.08, 0.15)	(3,4, 7, 11)
h_4	(0.65, 0, 0.1, 0.25)	(3,4, 7, 11)
h_5	(0.75, 0.2, 0.13, 0.12)	(2,4, 7, 10)
h_6	(0.57, 0.18, 0.11, 0.14)	(2,4, 7, 10)
h_7	(0.4, 0.15, 0.05, 0.3)	(2, 4, 7, 11)
h_8	(0.35, 0.15, 0.1, 0.25)	(2, 4, 7, 11)
h_9	(0.2, 0.55, 0.15, 0.1)	(1, 6,7, 10)
h_{10}	(0.4, 0.45, 0.1, 0.05)	(1, 6,7, 10)

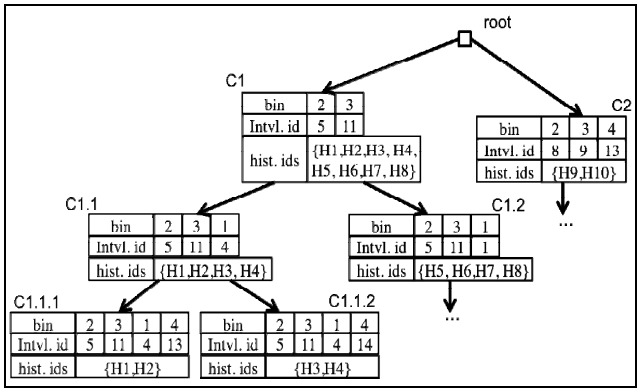

Figure 2: Index tree generated using the signatures obtained from the data set shown in Table 1.

4. Histogram search

We define a similarity measure S_I over index structure that upper bounds the S_H measure. Given raw query q and an indexed histogram h_i, their similarity in the index space is given by:

$$S_I(q,h_i) = \sum_{j=1}^{N} \min(q[j], UB(g(h_i[j]))) \quad (3)$$

where $UB([a,b])=b$. To prove that $S_I \geq S_H$, it suffices to prove that each term of the sum of S_I is greater than or equal to its corresponding term in the sum of S_H. Since we have $UB(g(h_i[j])) \geq h_i[j]$ for each $1 \leq j \leq N$, the inequality $min\ (q[j],\ UB(g(h_i[j]))) \geq min(q[j],h_i[j])$ holds for each j, proving that S_I is a upper bound of S_H.

The search algorithm retrieves all the histograms stored in the index having a similarity greater than a threshold *MinSim*. The similarity is computed by traversing, level by level, the index tree and incrementally evaluating the S_I measure using the bins defining each tree node. The similarity in each node is obtained by summing up the similarity accumulated by its parent and the similarity induced by the bins defining that node. For instance the similarity of node C1.1 in Figure 2 is computed by summing up the similarity induced by the bin 1 and the similarity accumulated by its parent node C1. The similarity of C1 is in turn computed by summing up the similarity induced by the bins 2 and 3, and the similarity inherited from its parent, which is the root node (equal to 0).

The incremental nature of our algorithm allows safely pruning the internal nodes that cannot be part of the final result. Indeed, the sub-tree rooted at a node is not explored if, given the partial similarity of that node and the bins used so far to compute it, the histograms covered by that node cannot have a similarity greater than the query threshold *MinSim*. Assume that the similarity of a tree node C is S_c, the set of remaining bins is B_r, and h_i is a histogram covered by C. The similarity between h_i and a query q can be rewritten as follows: $S_I(q,h_i) = S_C + S_R$, where,

$$S_R(q,h_i) = \sum_{j \in B_r} \min(q[j], UB(g(h_i[j]))) \quad (4)$$

The maximum value of $S_R(q,h_i)$ corresponds to the case where all the minimums derive from the query q, i.e.

$$S_R(q,h_i) \leq \sum_{j \in B_r} q[j] \quad (5)$$

Consequently, a sub-tree rooted at a node C can be pruned when:

$$S_C + \sum_{j \in B_r} q[j] \leq MinSim \quad (6)$$

5. Experiments

We evaluated the effectiveness of our indexing technique using a collection of 500 thousands images of the parasite *Schistosoma mansoni* generated from phenotypic drug-screening experiments. Due to space constraints, we only present results obtained using the grey scale histograms of the parasites. Our experiments were conducted by fixing the number of clusters to 10, the query threshold to 0.8, and by varying the number of intervals used to build the indices from 10 to 100. We ran 100 queries over each index. All the experiments were conducted on a laptop with an Intel

Core i5 processor, 2.5 GHz, 4 GB memory, running Mac OSX and SUN Java Virtual Machine version 1.7.

The first experiment evaluates the tightness of S_I, which is defined by: $T = S_H/S_I$. The approximation is better when the tightness is close to 1. Figure 3 shows the evolution of the tightness according to the number of intervals (the bold line). As we can see, the tightness exceeds 0.98 as the number of intervals becomes greater than 20. This demonstrates the effectiveness of the approximation strategy. The second experiment evaluates the effectiveness of our indexing technique with respect to the quality of the matches. We compare the results of our algorithm against those found by an exhaustive algorithm using the *F*-measure (*FM*). The results are shown in Figure 3 (dashed line). As expected, the quality of the results improves with the increase of the number of intervals. This is because of the fact that with the increase in the number of intervals, the tightness of S_I increases, which reduces the number of false positives. Result quality can also be improved by using more features.

The third experiment studies the execution time as a function of the number of intervals. We compare the results of our algorithm against those of the VA-File technique [6]. The results, reported in Table 2, show that our algorithm outperforms the VA-File technique by a mean factor of 3.7. This is due to the hierarchical structure of our indexing structure that incrementally computes the similarity between the query and a set of histograms, which allows us to prune the search space and obtain lower execution times.

Table 2**:** Evolution of the execution time (sec) according to the number of intervals of our algorithm and VA-File

Intervals	10	20	30	40	50	60	70	80	90	100
Proposed method	**1.7**	**1.9**	**2.6**	**2.9**	**3.6**	**3.7**	**3.8**	**4.1**	**4.1**	**4.1**
VA-FILE	8.4	9.9	10.9	11.7	11.8	12.2	12.3	12.7	12.8	12.9

6. Conclusions

In this paper, we have presented an effective hierarchical indexing approach that upper bounds the minimum histogram intersection measure and can be used for content-based retrieval of drug-induced phenotypes in assays. The effectiveness of the technique is demonstrated on a collection of 500 thousands images represented obtained from drug screening assays of the parasite *S. mansoni*. Comparative studies indicate the effectiveness and speed of the proposed method and underline its promise.

7. Contributions and Acknowledgements

RS formulated the problem and the solution strategy. The indexing method was developed by AG and RS. AG conducted the experiments whose results were analyzed by AG and RS. The paper was written by RS and AG.

This research was funded in part the NIH grant 1R01A1089896 and the Center for Computing in Life Sciences at San Francisco State University. The authors thank C. Caffrey and B. Suzuki for the assay data and D. Asarnow and S. Dubrovskiy for proof-reading the manuscript.

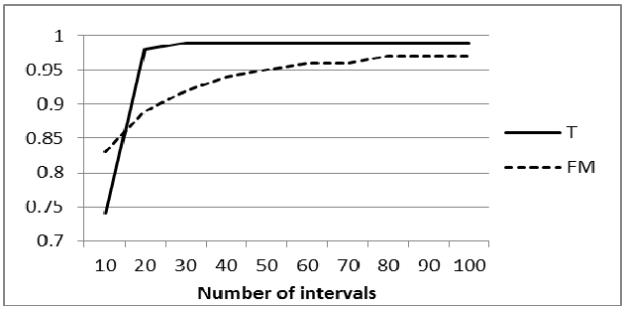

Figure 3: Change in the tightness (T) and FM as a function of the number of intervals used to build the index structure

8. References

[1] D. C. Swinney and J. Anthony, "How were new medicines discovered?," *Nature Reviews Drug Discovery*, vol. 10, no. 7, pp. 507–519, Jun. 2011.

[2] R. Singh, M. Pittas, I. Heskia, F. Xu, J. H. McKerrow, and C. Caffrey, "Automated Image-Based Phenotypic Screening for High-Throughput Drug Discovery", *IEEE Symposium on Computer-Based Medical Systems*, pp. 1-8, 2009

[3] H. Lee, *et al*., "Quantification and clustering of phenotypic screening data using time-series analysis for chemotherapy of schistosomiasis," *BMC Genomics*, 2012, 13.

[4] R.A. Paveley, *et al.* (2012) Whole Organism High-Content Screening by Label-Free, Image-Based Bayesian Classification for Parasitic Diseases. *PLoS Negl Trop Dis* 6: e1762.

[5] C. Marcellino, J. Gut, K. C. Lim, R. Singh, J. McKerrow, J. Sakanari, "WormAssay: A Novel Computer Application for Whole-Plate Screening of Macroscopic Parasites", *PLoS Neglected Tropical Diseases*, Vol. 6(1):e1494, 2012

[6] R. Weber, H.J. Schek, and S. Blott. A quantitative analysis and performance study for similarity-search methods in highdimensional spaces. *Proc. Inter. Conf. on Very Large Data Bases*, 1998, 194-205.

[8] D. Asarnow and R. Singh, "Segmenting the Etiological Agent of Schistosomiasis for High-Content Screening", *IEEE Transactions on Medical Imaging*, Vol. 32, NO. 6, 2013, pp. 1007-1018

[9] C.C. Aggarwal, J.L. Wolf, P. S. Yu, C. Procopiuc, and J.S. Park. Fast algorithms for projected clustering. *Proc. ACM Inter. Conf. on Manage. of Data*, 61-72, 1999.