Published in final edited form as:

IEEE Comput Intell Mag. 2012 November ; 7(4): 35–45. doi:10.1109/MCI.2012.2215124.

An Analysis Pipeline with Statistical and Visualization-Guided Knowledge Discovery for Michigan-Style Learning Classifier Systems

Ryan J. Urbanowicz, Ambrose Granizo-Mackenzie, and Jason H. Moore

Dartmouth College, USA

Abstract

Michigan-style learning classifier systems (M-LCSs) represent an adaptive and powerful class of evolutionary algorithms which distribute the learned solution over a sizable population of rules. However their application to complex real world data mining problems, such as genetic association studies, has been limited. Traditional knowledge discovery strategies for M-LCS rule populations involve sorting and manual rule inspection. While this approach may be sufficient for simpler problems, the confounding influence of noise and the need to discriminate between predictive and non-predictive attributes calls for additional strategies. Additionally, tests of significance must be adapted to M-LCS analyses in order to make them a viable option within fields that require such analyses to assess confidence. In this work we introduce an M-LCS analysis pipeline that combines uniquely applied visualizations with objective statistical evaluation for the identification of predictive attributes, and reliable rule generalizations in noisy single-step data mining problems. This work considers an alternative paradigm for knowledge discovery in M-LCSs, shifting the focus from individual rules to a global, population-wide perspective. We demonstrate the efficacy of this pipeline applied to the identification of epistasis (i.e., attribute interaction) and heterogeneity in noisy simulated genetic association data.

1. Introduction

Learning classifier systems (LCSs) [1] are a rule-based class of algorithms that combine machine learning with evolutionary computing and other heuristics to produce an adaptive system. The goal of LCS is not to identify a single best model or rule, but to create a cooperative set of rules that collaborate to solve the given problem. This feature makes LCSs appealing for application to complex multifactor problem domains such as function approximation [2], clustering [3], behavior policy learning [4], and classification/data mining [5]. Data mining in bioinformatics problems can be particularly challenging, involving large, noisy, and complex problem landscapes.

1.1 Bioinformatics Problem

One bioinformatics domain where these shortcomings clearly impact the utility of LCS is within human genetics. Single nucleotide polymorphisms (SNPs) are single loci in the DNA sequence where alternate nucleotides (i.e., alleles) are observed between members of a species or between paired chromosomes in an individual. In a typical genetic association

study, researchers look for differences in SNP allele frequencies between a group of individuals with the disease of interest, and a matched group healthy controls.

As geneticists strive to identify “markers for” and “causes of” common complex disease, it has become increasingly clear that the statistical and analytical techniques, well suited for Mendelian studies, are insufficient to address the demands of common complex diseases [6, 7]. Epistasis (gene-gene interaction) and heterogeneity are two of several phenomena reviewed by [8] recognized to hinder the reliable identification of predictive genetic markers in association studies. *Epistasis* refers to the interaction between multiple genetic loci wherein the effect of a given loci is masked by one or more others. *Heterogeneity*, referring to either *genetic* heterogeneity (locus and allelic) or *environmental* heterogeneity, occurs when individual (or sets of) attributes are independently predictive of the same phenotype (i.e., class).

While the detection and modeling of epistasis has received a great deal of attention [9, 10, 11], methods for dealing with heterogeneity are lagging behind. From a computer science prospective, the problem of heterogeneity is similar to a latent or “hidden” class problem. While the disease status of each patient is already known, the individuals making up either class would be more accurately sub-typed into two or more “hidden” classes, each characterized by an independent predictive model of disease.

Presently, the most common strategy for dealing with heterogeneity involves some form of data stratification in order to identify more homogeneous subsets of patients [8]. This is in line with the standard epidemiological paradigm which seeks to identify the single best disease model within a given dataset. However, the efficacy of stratification is limited by the availability, quality, and relevance of the variables which determine these strata. Additionally, stratification represents a reduction in sample size, leading to an inevitable loss in power.

Genetic association studies typically consider many attributes which turn out to be “non-predictive,” i.e., they fail to confer any meaningful predictive ability. The primary goal in such studies is feature selection, i.e., to discriminate between potentially useful predictive attributes and these extraneous ones. It is also typical for association studies to exhibit a great deal of noise, masking the signal of predictive attributes. Noise makes it impossible to develop a rule-set that classifies with perfect accuracy on testing data (i.e., data which the algorithm it has not yet seen). Noise may come from a number of sources including missing variables (genetic or environmental) and errors in the collection and analysis of the data. In this paper we consider the functionality of LCS from the perspective of noisy problems such as the one described above.

1.2 Learning Classifier Systems

Since LCSs break from the “best model” paradigm, and evolves a solution comprised of a cooperative set of rules, it offers an intuitive strategy for addressing the heterogeneity problem without data stratification. Within the LCS community there is an impressive diversity of algorithm architectures and modifications that have been implemented with the intention of optimizing prediction accuracy, run time, generalization, solution compactness

and solution comprehensibility within the context of different problem domains. The infancy of LCS research saw the emergence of two founding classes of LCSs, referred to as the Michigan (M-LCS) and Pittsburgh (P-LCS) styles. The M-LCS is characterized by a population of rules with a GA operating within or between individual rules and an evolved solution represented by the entire rule population. Alternatively, the P-LCS is characterized by a population of variable length rule-sets (where each rule-set is a potential solution) and the GA operates between collective rule-sets. Recently, both Michigan and Pittsburgh-style LCSs were applied to the detection and modeling of simulated genetic disease associations in the presence of epistasis and heterogeneity [12, 13]. These evaluations demonstrated the potential of LCSs to address our unique problem and identified the strengths and weaknesses of using either a Michigan or Pittsburgh-style LCS on these types of complex, noisy problems. Two distinct differences favored P-LCSs in terms of knowledge discovery and interpretability: (1) the size of the evolved solution (i.e., a small number of rules), and (2) the optimality of rule generalizations. However, power analyses in P-LCSs indicated that they struggled to reliably learn precise rules (i.e., rules which were optimally generalized) [13], a sign of over-fitting. Analyses in [12] indicated that over-fitting tends to occur even more dramatically in M-LCSs. However, preliminary strategies for estimating power globally in M-LCSs suggested that global patterns could be reliably identified within M-LCSs despite over-fitting at the level of individual rules [12]. Of further interest, for both LCS styles, there is a lack of any established significance testing strategy for the evolved solution.

1.3 Knowledge Discovery in LCSs

Knowledge discovery has been described as the “non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data” [14]. Efforts to improve knowledge discovery in P-LCSs or similar genetics based machine learning (GBML) algorithms are synonymous with improving overall performance (i.e., accuracy, generality, speed, and solution compactness), since these algorithms explicitly strive to evolve a concise rule-set that can be directly inspected and interpreted [15, 16, 5]. As in P-LCSs, knowledge discovery in M-LCS is traditionally achieved through manual inspection of rules. Manual inspection simply involves an examination of the best rules in the population in order to determine which attributes (SNPs), with specific states (genotypes) are predictive of class (disease status). Due to the number of rules in M-LCS solutions, [17] suggested ranking rules by numerosity (number of rule copies in the population) and then inspecting those with the highest numerosities in order to identify key parts of the solution.

A largely ubiquitous goal for both LCS styles is to simultaneously optimize both accuracy and rule generality. However, these goals can be at odds with one another especially in the presence of noise. For example, in typical M-LCSs, despite implicit generalization pressures, fitness is ultimately determined by rule accuracy. Thus as long as some non-predictive attribute confers a slight accuracy advantage in the training set, a rule that includes it will be chosen over a more appropriately generalized rule. This is the driving force behind over-fitting. Over-fitting often indicates that the system is learning structure that is idiosyncratic to the training set and therefore generalization is occurring sub-optimally. Without prior knowledge of the problem complexity or structure, achieving the

ideal balance between accuracy and generalization may be impractical or even impossible. With that in mind, it seems unreasonable to expect M-LCSs to automatically evolve optimal individual rules for interpretation. The task then becomes: what reliable attribute patterns can we derive from the rule population?

For M-LCSs, existing strategies aimed at facilitating knowledge discovery focus on reducing the size of the rule population by rule compaction [18, 19, 20, 21] or condensation [22, 23], or alternatively, by modifying the rule representation [24, 23]. These strategies are all in-line with the classic paradigm of knowledge discovery wherein the goal is to identify explicit rules for interpretation that are accurate, and maximally general.

Kharbat, Odeh and Bull took a somewhat different approach to M-LCS rule compaction (rule-dependent as opposed to data-dependent) in order to extract minimal and representative rules from the original rule-set [25, 26, 27]. Rules were clustered based on similarity and these clusters were used to generate aggregate average rules and aggregate definite rules presenting the common characteristics of the cluster. These resulting aggregate rules are then interpreted by an expert seeking knowledge discovery. This strategy reflected a more global perspective of rule-set evaluation looking for common patterns in a large population of rules. In this study, we sought to extend this line of global thinking.

Our goals for applying an M-LCS to noisy single-step data mining problems include (1) distinguishing predictive from non-predictive attributes with statistical confidence, (2) identifying optimally generalized and accurate rule association patterns, (3) facilitating the identification of, and discrimination between, interaction and heterogeneity. These goals are critical for making M-LCS a viable data mining tool for our bioinformatics problem of interest. In the present study we laid out an analysis pipeline to address these goals. We applied this pipeline to a complex simulated genetic association dataset embedded with heterogeneity and epistasis as a demonstration of its efficacy. Our globally-minded strategy introduces novel statistics, a strategy for significance testing, and the application of visualizations to guide knowledge discovery in M-LCSs. Also, unique to gene association studies, we demonstrated that this analysis pipeline facilitates the identification and characterization of heterogeneity and epistasis.

2. Methods and Results

In this section we describe (1) the M-LCS algorithm and parameters used in this investigation, (2) a description of the simulated dataset investigated which models our target bioinformatics problem, (3) the results of manual inspection on this data, and (4) a step by step explanation of the analysis pipeline with corresponding results.

2.1 UCS

M-LCSs, often varying widely from version to version, generally possess four basic components; (1) a population of rules or classifiers, (2) a performance component that assesses how well the population of rules collectively explain the data, (3) a reinforcement component that distributes the rewards for correct prediction to each of the rules in the population, and (4) a discovery component that uses different operators to discover new

rules and improve existing ones. Learning progresses iteratively, relying on the performance and reinforcement components to drive the discovery of better rules. For a complete LCS introduction and review, see [1].

UCS, or the sUpervised Classifier System [28], is based largely on the very successful XCS algorithm [17], but replaces reinforcement learning with supervised learning, encouraging the formation of best action maps and altering the way in which accuracy, and thus fitness, is computed. UCS was designed specifically to address single-step problem domains such as classification and data mining, where delayed reward is not a concern. The implementation of UCS applied in this study is the same as the one used in [12] upon which we had previously performed a parameter sweep in an attempt to optimize major run parameters. With this as our basis, we adopted mostly default parameters with the exception of 200,000 learning iterations, a population size of 2000, tournament selection, uniform crossover, subsumption, and a v of 1. v has been described as a “constant set by the user that determines the strength [of] pressure toward accurate classifiers” [29], and is typically set to 10 by default. A low v was used to place less emphasis on high accuracy in this type of noisy problem domain, where 100% accuracy is only indicative of over-fitting. Also, as in [12], we employ a quaternary rule representation, where for each SNP attribute, a rule can specify genotype as (0, 1, or 2), or instead generalize with “#,” a character that implies that the rule does not care about the state of that particular attribute. Note that when evaluating UCS over the entire training or testing datasets, discovery mechanisms were disabled such that the rule population remained constant.

2.2 Simulated Dataset

In both [12] and [13], datasets were generated that concurrently modeled epistasis and heterogeneity as they might simultaneously occur in a single nucleotide polymorphism (SNP) genetic association study. All datasets were generated using a pair of distinct, two-locus epistatic interaction models, both utilized to generate instances (i.e., case and control individuals) within a respective subset of each data set. Two-locus epistatic models were simulated without Mendelian/main effects, as penetrance tables. In total, each simulated data set contained 4 predictive attributes, where only two attributes were predictive within a respective subset of the data. Additionally, each data set contained 16 randomly generated, non-predictive attributes, with minor allele frequencies randomly selected from a uniform distribution ranging from 0.05 to 0.5.

In this study we demonstrated the efficacy of our analysis pipeline using one of these aforementioned datasets that concurrently models epistasis and heterogeneity. Minor allele frequencies of all 4 predictive attributes were 0.2, heritabilities for both underlying models were 0.4, relative model architecture difficulty was “Easy,” the ratio of samples representative of either model was 50:50, and the dataset sample size was 1600. The heritability of a model indicates the proportion of variation that can be attributed to the attributes in the model. Any model heritability less than 1 indicates the presence of noise (i.e., lower heritability corresponds to greater noise). This dataset offers a reasonable approximation of how a complex disease association pattern might appear within a balanced dataset comprised of sick and healthy subjects. Evaluation of this dataset in [12], with an

implementation of the UCS algorithm, yielded significant power to detect the underlying predictive attributes. This dataset was chosen to provide a clear example of this pipeline analysis. If this were a real biological investigation, our goal would be the identification of predictive attributes, without making any assumptions about the number of attributes involved or knowing whether their association followed patterns of epistasis or heterogeneity. Genetic or environmental attributes identified in this manner would be investigated further to determine whether they are causal variants or merely markers of disease.

As a negative control for this study, we repeated the pipeline analysis on a second dataset within which all attributes were non-predictive. Specifically, this dataset is a class-permuted version of the dataset described above.

2.3 Manual Inspection

To highlight the need for our proposed analysis pipeline we began with an example of manual inspection within the rule population evolved by UCS on the noisy target dataset considered in this study. Table 1 displays the top 10 rules identified by UCS after 200,000 learning iterations. This rule population is the same one applied in section 3.0.3 for visualization. Half the samples in the dataset were generated with a predictive epistatic interaction between attributes 'X0' and 'X1,' while the other half were generated with a different epistatic interaction between attributes 'X2' and 'X3.' All other attributes were randomly simulated as non-predictive. Optimally generalized rules would strictly specify one of these two pairs of attributes (i.e., X0, X1 or X2, X3), but no others. Examining the top 10 rules in Table 1, we can see that this is never the case. We do see that the correct attribute pairs often occur together in these top rules, however one or two other non-predictive attributes tend to be specified as well. Specification of these non-predictive attributes affords the rule higher accuracy in the training data (all top rules in Table 1 have 100% accuracy). Scanning down the complete ordered list of rules, we finally observed an optimally generalized rule 43rd down on the list, however without already knowing the true association pattern we would have no way of identifying that rule as optimal. So how do we separate the attributes that are reliable, and those which are the product of over-fitting in a noisy environment?

3. Analysis Pipeline

Our proposed analysis pipeline includes the following steps: (1) run the M-LCS algorithm with 10-fold cross validation (CV) on the dataset, (2) run a permutation test with 1000 permutations, (3) confirm significance of testing accuracy, (4) identify significant attributes and significantly co-occurring pairs of attributes, (5) train the M-LCS algorithm on the entire dataset, (6) generate a clustered heat-map of the rule-population, (7) generate a network depicting attribute co-occurrence, and (8) combine statistical results with visualizations to interpret and generate hypothesis for further exploration and validation. While we utilized UCS in this study and focused on a very specific data mining problem, this pipeline could be expanded to knowledge discovery in any M-LCS applied to a single step data mining problem.

3.0.1 Run the M-LCS

This section details steps 1 and 2. First we employed a 10-fold CV strategy in order to determine average testing accuracy and account for over-fitting. The dataset is randomly partitioned into 10 equal parts and UCS is run 10 separate times during which 9/10 of the data is used to train the algorithm, and a different 1/10 is set aside for testing. We averaged training and testing accuracies over these 10 runs. Next we set up our permutation test. A permutation test involves repeating the analysis on variations of the dataset (with class status shuffled) in order to determine the likelihood that the observed result could have occurred by chance. We chose to use the permutation test since we do not know the chance distribution of our statistics ahead of time. We generated 1000 permuted versions of the original dataset by randomly permuting the affection status (class) of all samples, while preserving the number of cases and controls. For each permuted dataset we ran UCS using 10-fold CV. In total, permutation testing requires 10,000 runs of UCS. We performed this analysis using “Discovery,” a 1372 processor Linux cluster.

3.0.2 Significance Testing of M-LCS Statistics

This section details steps 3 and 4. First, and foremost, we confirmed that our average testing accuracy from Step 1 is significantly higher than those obtained by random chance. We utilized a typical one-tailed permutation test with a significance threshold of $p < 0.05$. To determine p for a test statistic (in this case average testing accuracy) calculate the test statistic for each of the 1000 permuted, CV analyses. If the true test statistic from Step 1 is greater than 95% of the 1000 permuted runs, you can reject the null hypothesis at $p < 0.05$. Here, the null hypothesis is: the observed value of the statistic could have likely occurred by chance. If the true test statistic is greater than all of the 1000 permuted runs, a minimum p of 0.001 is achieved. Our analysis on the aforementioned dataset yielded a significant testing accuracy of 0.701 ($p = 0.001$).

If average testing accuracy is not significantly high, this suggests that M-LCS was unable to learn any useful generalizations from the data, and there is little reason to progress with the rest of this analysis pathway. Failure to obtain a significant testing accuracy suggests either the absence of a useful generalization within the data, or that the run settings for the M-LCS were inappropriate for the detection of an existing generalization (e.g. a larger population size or greater number of learning iterations may be required).

Once a significant testing accuracy is confirmed we used the permutation test analysis to identify attributes in the dataset that show significant importance in making accurate classification. Here we describe novel population state metrics (test statistics describing the state of the rule population) for making such an inference from the rule population: (1) Specificity Sum (SpS) and (2) Accuracy Weighted Specificity Sum (AWSpS). The SpS is identical in principle to our previously described power estimation strategy [12]. The premise for this statistic states that if an M-LCS is learning useful generalizations while training on the data, attributes that are important for making correct classifications will tend to be specified more frequently within rules of the population. Alternatively, attributes that are not useful for making correct classifications will tend to be generalized (#/don't care symbol used). SpS is calculated separately for each attribute, and is simply the number of

rules in the population that specifies a value for that attribute (as opposed to having a '#'). This calculation takes rule numerosity into account, since numerosity represents the number of copies of that rule presently in the population. As an alternative to SpS, we also consider AWSpS, which simply weights the SpS by the respective rule accuracies. Like SpS, AWSpS is calculated separately for each attribute, and takes numerosity into account. Table 2 gives a simple example in which both SpS and AWSpS are calculated from a hypothetical population consisting of 4 rules. Notice how SpS and AWSpS scores for attributes 'X1' and 'X4' stand out from the rest.

The intuition behind using AWSpS vs. SpS, is based on the idea that: at any given learning iteration, it is possible for a non-predictive attribute to be specified frequently in rules of the population just by chance. Consider the hypothetical situation in which a predictive attribute and non-predictive attribute happen to be specified with equal frequency. We would expect that, overall, rules specifying the predictive attribute would tend to have higher accuracies than rules with the non-predictive attribute. Therefore, by weighting specification frequency by accuracy, we would be more likely to correctly favor the predictive attribute as important to classification and avoid false positives. Conversely, by chance, non-predictive attributes could be specified along with predictive attributes in a highly accurate rule. In this case, the non-predictive attributes might receive a parasitic boost from the overall high accuracy of the rule, potentially encouraging a false positive. Because of this dilemma we considered both statistics.

Again, we used permutation testing to determine the significance of attributes using the SpS and AWSpS metrics. For each attribute we separately calculated SpS and AWSpS over the 10 CV rule populations trained in Step 1. For each attribute we compared this sum to the 1000 respective sums from permutation testing to determine whether the true SpS or AWSpS is significantly higher than we would expect by chance. Significance values are calculated as previously described for each attribute. This step provides us with a statistically justified strategy for discriminating between predictive and non-predictive attributes in an M-LCS rule population.

Table 3 organizes the SpS and AWSpS statistic results for our simulated dataset. The SpS and AWSpS for our four predictive attributes (i.e., X0, X1, X2, X3) are significantly higher than would be expected by chance with sums dramatically larger than non-predictive attributes. Notice that with the AWSpS statistic, two non-predictive attributes barely make the significance cut-off. However their AWSpS values are dramatically lower than the predictive attributes. Investigators could also approach this analysis with a two-tailed permutation test (with a significance cut-off of $p < 0.025$), and ask the questions: which attributes are specified more frequently, and which attributes are specified less frequently than we would expect by chance? Here, such an analysis would have precisely identified the correct 4 predictive attributes that are significantly over-specified with no false positives, and in addition would identify 6 of the non-predictive attributes as having been significantly underspecified, suggesting that the M-LCS learned to avoid specifying these attributes.

The last population state metric we introduce, considers pair-wise attribute co-occurrence within rules. We consider this statistic in order to evaluate attribute interactions as well as

help discriminate between interactions and heterogeneity. We calculate co-occurrence for every non-redundant pair-wise combination of attributes in the dataset. For a dataset with 20 attributes (such as the one we examine here), we calculate 190 co-occurrence values. We calculated co-occurrence as follows: for every pair of attributes we go through each of the 10 CV rule populations and sum the number of times that both attributes are concurrently specified in a given rule. In the example from Table 2, the Co-occurrence Sum (CoS) for the attribute pair (X1, X2) would be 2 since the pair only cooccurs in rule 3, and the rule has a numerosity of 2. The significance of co-occurrence scores are determined as before using the results of permutation testing for each pair of attributes.

Table 4 organizes the top CoS statistic results for our target dataset (only significant CoSs with sums greater than 3000 are displayed). 43 out of the 190 CoSs were identified as significantly higher than by chance. Each of these significant pairs had at least one predictive attribute represented. Seeing that we found the four predictive attributes to be significantly over-represented, it follows that co-occurrence pairs including at least one of these attributes would turn up more frequently than by chance. Of particular note, we found that our two modeled, epistatic pairs of predictive attributes yielded the two highest CoSs. Below these two pairs, there is an immediate drop-off in the magnitude of CoS (almost halved). Also note that the magnitude of these two highest CoSs are well above the SpSs for all non-predictive attributes given in Table 3.

While additional analysis could examine higher order co-occurrence between all 3-way combinations of attributes (or beyond), we can use pair-wise analysis to infer higher order attribute interactions. For example, if a predictive, 3-way interaction existed in the data (between X0, X1, and X3) we would expect similar pair-wise CoSs between attribute pairs (X0, X1), (X1, X2), and (X0, X2). Using this logic we could hypothesize, given the results in Table 4, that attribute pairs (X0, X1), and (X2, X3) each represent an interacting pair, but since the 4 other pair-wise combinations of these attributes have about half the co-occurrence we might suspect heterogeneity as opposed to a 3 or 4-way interaction. Section 3.0.4 will offer a visualization of these co-occurrence results.

3.0.3 Visualization—Heat-Map

As indicated by [30], the “visualization of classification models can create understanding and trust in data mining models”. While visualization strategies are not entirely new to the LCS field, to date they have been applied only to track learning progress within the search space [31, 32]. The key difference here is that visualization is applied directly to knowledge discovery for the identification of global attribute generalizations.

This section details steps 5 and 6. All M-LCS runs up to this point have been dedicated to obtaining test statistics and making statistical inferences. Our first step towards visualization is to train the M-LCS on the entire dataset. M-LCS algorithms are typically very adaptive with a tendency to maintain some level of diversity in the population as it attempts to search for better and better rules. As a result we would expect that some proportion of the rules will be poor classifiers with useless generalizations. As previously mentioned, rule compaction [18, 19, 20, 21] and condensation [22, 23] algorithms offer a method of eliminating useless rules and compacting the size of the rule population. While beyond the scope of the present

study, such an algorithm could be used at this point in the analysis pipeline in an attempt to remove some of these useless, “noisy” rules. However, based on preliminary observations that explored rule compaction algorithms, we would caution the reader that a dramatic reduction in the size of the rule population may be counter-productive to successfully identifying global patterns.

Our next step includes re-encoding the rule population. The objective of the heat-map visualization is to discriminate predictive attributes from non-predictive attributes and to look for patterns of attribute interaction and heterogeneity. Therefore we encoded each rule such that any specified attribute is coded as a 1 while a ‘#’ is coded as a 0. Additionally, we expanded our rule population such that there are N copies of each rule reflecting respective numerosities. Similar to ordering by rule numerosity within manual inspection, this step draws greater attention to attribute patterns within rules having a higher numerosity. The last processing step before visualization is to apply a clustering algorithm to the coded and expanded rule population. Clustering a population of M-LCS rules for rule compaction was pioneered in [26, 27]. In this study we employed agglomerative hierarchical clustering using hamming distance as the distance metric. Clustering is performed on both axes (i.e., across rules and attributes). Both clustering and 2D heat-map visualization was performed in R using the *hclust* and *gplots* packages, respectively.

Finally, we applied 2D heat-maps to highlight global attribute patterns for knowledge discovery in M-LCSs. Figure 1 gives our 2D heat-maps visualizing the rule population. In Figure 1(a), the only clearly discernible pattern is the relative high specification of the four modeled attributes (X0, X1, X2, and X3) on the left. These four columns stand out as having more yellow color than the others. After clustering, the utility of this visualization become much more apparent. While portions of Figure 1(b) are relatively noisy, we see two dominant rule patterns emerge, one involving attributes X0 and X1, and the other involving attributes X2 and X3. Note how hierarchical clustering separates these independent epistatic models, suggesting the presence of a heterogeneity vs. higher order interactions. If a four way interaction was modeled, we would instead expect all four attributes to cluster together.

Apart from the 2D heat-map, we re-purposed a 3D heat-map visualization tool that can accommodate up to 6 dimensions of information and allows users to interactively explore the 3D visualization space. This software was adapted to bioinformatic visualization in [33] using Unity3D (<http://unity3d.com>). Beyond obtaining a global perspective of the rule population, this software could also be used to facilitate the identification of particularly interesting rules from the rule population by adding rule parameters such as class, accuracy, fitness, numerosity, and/or action set size as potential dimensions of the visualization. Users can assign 3 spatial dimensions along with bins and color (top and sides of bars). Figure 2 illustrates this 3D visualization, and offers a simple example of how these extra dimensions may be utilized.

3.0.4 Visualization—Co-Occurrence Network

This section details step 7. The network visualization may be generated using either the M-LCS run from step 5, or from the 10 CV runs from step 1. We use the 10 CV runs, since they include statistical analysis and summation over multiple runs. To generate our co-

occurrence network we use Gephi (<http://gephi.org/>) an open source graph visualization software. Using the 190 CoSs calculated above, we generated an adjacency matrix in a format consistent with Gephi requirements. Using Gephi, we generated a fully connected, undirected network, where nodes represent individual attributes, the diameter of a node is the SpS for that attribute, edges represent co-occurrence, and the thickness of an edge is the respective CoS. Additionally, Gephi offers a built-in function to filter edges from the network based on edge weight. Users can progressively filter out edges representing smaller CoSs in order to identify/highlight dominant patterns.

Figure 3 gives the co-occurrence network illustrating the 190 CoSs calculated in section 3.0.2. Filtering the network is a largely subjective process since the ideal filter threshold to capture the true underlying pattern is not known ahead of time. However by adjusting the filter threshold the investigator can isolate dominant patterns and exclude those less likely to be of interest. In Figure 3, images A through C show the network with a progressively more stringent edge weight filter. In each, the strong co-occurrence between (X0, X1) as well as between (X2, X3) can be observed, illustrating the interaction between these respective attribute pairs. In addition, the diameter of the nodes draws attention to the same four predictive attribute. Figure 3 (c) accurately captures the underlying relationships modeled in this simulated dataset. More specifically, we observed interaction between pairs (X0, X1) and (X2, X3) but independence between the pairs, as would be expected given the embedded heterogeneity. Notice the relationship between Figures 1 and 3. The network visualization efficiently summarizes the key features of the heat-map.

3.0.5 Guided Knowledge Discovery

This section details step 8. The final step in our analysis pipeline involves the merging of all empirical and subjective observations made thus far in order to characterize useful rule generalizations, and generate hypotheses for further investigation. Thus far, we have (1) successfully identified our four predictive attributes, significantly differentiated them from “noise” attributes, (2) identified strong, significant co-occurrence between both epistatic attribute pairs, but observed a dramatic drop in co-occurrence for other pair-wise combinations of the four predictive attributes (evidence against a higher order interaction), (3) observed separate clustering of the significant attribute pairs in the 2D heat-map (evidence of heterogeneity), and (4) observed two dominant attribute co-occurrence pairs (evidence of interaction within each pair, and heterogeneity between them). Together these observations correctly reflect the predictive attributes, interactions, and heterogeneity embedded in this complex simulated data. If only manual inspection had been utilized, we would have several additional irrelevant attributes to consider as candidates, we would lack statistical confidence to back up our claims, and it would be difficult or impossible to get a clear picture of the true underlying patterns of association (see Table 1).

However, we do not wish to discount manual inspection, only address its shortcomings. Our global rule population analysis may be supplemented by, or used to direct traditional rule inspection wherein the relationship between class and a specific attribute states is sought. This pipeline is intended to guide hypothesis generation and further investigation. It

may also be expanded, using the first pass as a filter to identify significant attributes and a subsequent analysis which only include those attributes that were found to be significant.

3.0.6 Negative Control Analysis

As mentioned in section 2.2 we repeated the analysis described above on a dataset with no predictive attributes for comparison. As expected, this analysis confirmed that the average testing accuracy of 0.502 was not significant ($p = 0.497$). Without a significant testing accuracy, subsequent analysis of SpS, AWSpS and CoSs is meaningless. However for the purposes of this study we report on these analyses for reference. SpS and AWSpS analyses each identified a single attribute (X13) as significant (p-values of 0.034 and 0.048 respectively) using a onetailed test. However, with a two-tailed test, none of the attributes make the significance cut-off. CoS analysis identified 16 out of 190 attribute pairs as significant. However, the magnitude of these CoSs are well below even the smallest of SpSs in this secondary analysis (approximately 3x smaller). Since the goal of the SpS, AWSpS and CoSs significance tests are to determine which elements are over-represented in the context of the set, it is not surprising that by chance we observe some significant findings. However they may be immediately discounted in the absence of a significantly high testing accuracy which reflects whether any generalizable patterns were identified. This analysis represents a successful negative control for the pipeline.

4. Conclusions

While M-LCS algorithms are inherently powerful, flexible, and adaptive, the complexity and ambiguity of the solutions they may evolve are deterrents for application to many data mining tasks. This study introduces a complete analysis pipeline for global knowledge discovery in M-LCS rule populations. In the course of this work we have (1) introduced novel population state metrics (SpS, AWSpS, and CoS) for global rule-population characterization, (2) adapted significance testing to M-LCS evaluation, and (3) adapted visualization strategies to facilitate the identification of rule patterns. We have demonstrated that this strategy may be applied successfully to a challenging genetic association problem involving epistasis and heterogeneity. We correctly discriminated between predictive and non-predictive attributes, characterize interactions, and make observations which correctly point to underlying heterogeneous patterns. To the best of our knowledge, this is the first attempt to apply significance testing to knowledge discovery in LCSs. We hope that this will encourage the application of LCSs problem domains in which significance testing is required.

While this pipeline offers a promising pathway for interpretation, this type of analysis is limited by substantial computational expense in running the MLCS algorithm along with the necessary permutations and cross validation. Given current computational technology, this makes analysis of large-scale genetic datasets impractical. M-LCSs and the analysis pipeline introduced here would be better suited to candidate studies with a refined set of potentially predictive attributes. In future studies, we plan to apply this pipeline to real world genetic association analysis in search of novel gene-disease relationships and evidence of heterogeneity in study samples. Beyond UCS and the bioinformatic problem considered in this work, this pipeline has the potential for application to any M-LCS tasked with a single

step data mining problem. Even if the proposed pathway as a whole is not applicable or practical for all problem domains we expect that certain components, such as our visualizations, will be more universally relatable.

Acknowledgments

We thank Douglas Hill for his expertise in managing the 3D heat-map software and Christian Darabos for his expertise in network visualizations using Gephi. This work was supported by NIH grants AI59694, LM009012, and LM010098.

References

1. Urbanowicz R, Moore J. Learning classifier systems: A complete introduction, review, and roadmap. *J Artif Evol Appl*. Sept. ;2009 2009:736398-1–736398-25.
2. Butz M, Lanzi P, Wilson S. Function approximation with XCS: Hyperellipsoidal conditions, recursive least squares, and compaction. *IEEE Trans Evol Comput*. June ;2008 12(3):355–376.
3. Shi, L.; Shi, Y.; Gao, Y. Clustering with XCS and agglomerative rule merging. *Proc 10th Int Conf Intelligent Data Engineering Automated Learning*; 2009; p. 242-250.
4. Howard, G.; Bull, L.; Lanzi, P. Self-adaptive constructivism in neural XCS and XCSF. *Proc 10th ACM Annu Conf Genetic Evolutionary Computation*; 2008; p. 1389-1396.
5. Bacardit J, Burke E, Krasnogor N. Improving the scalability of rule-based evolutionary learning. *Memetic Comput*. 2009; 1(1):55–67.
6. Shriner D, Vaughan L, Padilla M, Tiwari H. Problems with genome-wide association studies. *Science*. June ;2007 316:1840–1842. [PubMed: 17600199]
7. Eichler E, Flint J, Gibson G, Kong A, Leal S, Moore J, Nadeau J. Missing heritability and strategies for finding the underlying causes of complex disease. *Nature Rev Genet*. Jun.2010 11:446–450. [PubMed: 20479774]
8. Thornton-Wells T, Moore J, Haines J. Genetics, statistics and human disease: Analytical retooling for complexity. *Trends Genet*. 2004; 20(12):640–647. [PubMed: 15522460]
9. Cordell H. Epistasis: What it means, what it doesn't mean, and statistical methods to detect it in humans. *Human Molecular Genet*. 2002; 11(20):2463–2468.
10. Ritchie M, Hahn L, Moore J. Power of multifactor dimensionality reduction for detecting gene-gene interactions in the presence of genotyping error, missing data, phenocopy, and genetic heterogeneity. *Genetic Epidemiol*. 2003; 24(2):150–157.
11. Moore J, Gilbert J, Tsai C, Chiang F, Holden T, Barney N, White BC. A flexible computational framework for detecting, characterizing, and interpreting statistical patterns of epistasis in genetic studies of human disease susceptibility. *J Theoret Biol*. 2006; 241(2):252–261. [PubMed: 16457852]
12. Urbanowicz, R.; Moore, J. The application of Michigan-style learning classifier-systems to address genetic heterogeneity and epistasis in association studies. *Proc 12th ACM Annu Conf Genetic Evolutionary Computation*; 2010; p. 195-202.
13. Urbanowicz, R.; Moore, J. The application of Pittsburgh-style learning classifier systems to address genetic heterogeneity and epistasis in association studies. *Proc 11th Int Conf Parallel Problem Solving Nature*; 2011; p. 404-413.
14. Fayyad, U.; Piatetsky-Shapiro, G.; Smyth, P.; Uthurusamy, R. *Advances in Knowledge Discovery and Data Mining*. Palo Alto, CA: AAAI Press; 1996.
15. Llorá, X.; Reddy, R.; Matesic, B.; Bhargava, R. Towards better than human capability in diagnosing prostate cancer using infrared spectroscopic imaging. *Proc 9th Annu Conf Genetic Evolutionary Computation*; 2007; p. 2098-2105.
16. Bacardit, J.; Garrell, J. Bloat control and generalization pressure using the minimum description length principle for a Pittsburgh approach learning classifier system. *Proc Int Conf Learning Classifier Systems*; 2007; p. 59-79.
17. Wilson S. Classifier fitness based on accuracy. *Evol Comput*. 1995; 3(2):149–175.

18. Wilson, S. Compact rulesets from XCSI. Proc 4th Int Workshop Advances Learning Classifier Systems; 2002; p. 65-92.
19. Fu, C.; Davis, L. A modified classifier system compaction algorithm. Proc Genetic Evolutionary Computation Conf; 2002; p. 920-925.
20. Dixon, P.; Corne, D.; Oates, M. A ruleset reduction algorithm for the XCS learning classifier system. Proc Int Conf Learning Classifier Systems; 2003; p. 20-29.
21. Gao Y, Huang J, Wu L. Learning classifier system ensemble and compact rule set. Connect Sci. 2007; 19(4):321–337.
22. Kovacs, T. Cog Sci Res Centre. Univ. Birmingham; Birmingham, U.K: 1997. XCS classifier system reliably evolves accurate, complete and minimal representations for Boolean functions. Tech. Rep. CSRP-97-19
23. Lanzi, P. Mining interesting knowledge from data with the XCS classifier System. Proc Genetic Evolutionary Computation Conf; 2001; p. 7-11.
24. Butz, M.; Lanzi, P.; Llorá, X.; Goldberg, D. Knowledge extraction and problem structure identification in XCS. Proc Int Conf Parallel Problem Solving Nature; 2004; p. 1051-1060.
25. Kharbat, F.; Bull, L.; Odeh, M. Mining breast cancer data with XCS. Proc 9th Annu Conf Genetic Evolutionary Computation; 2007; p. 2066-2073.
26. Kharbat F, Odeh M, Bull L. New approach for extracting knowledge from the XCS learning classifier system. Int J Hybrid Intell Syst. 2007; 4(2):49–62.
27. Kharbat, F.; Odeh, M.; Bull, L. Knowledge discovery from medical data: An empirical study with XCS. Proc Int Conf Learning Classifier Systems Data Mining; 2008; p. 93-121.
28. Bernadó-Mansilla E, Garrell-Guiu J. Accuracy-based learning classifier systems: Models, analysis and applications to classification tasks. Evol Comput. 2003; 11(3):209–238. [PubMed: 14558911]
29. Orriols-Puig, A.; Bernadó-Mansilla, E. Revisiting UCS: Description, fitness sharing, and comparison with XCS. Proc Int Conf Learning Classifier Systems; 2008; p. 96-116.
30. Seifert, C.; Lex, E. A novel visualization approach for data-mining-related classification. Proc 13th Int Conf Information Visualisation; 2009; p. 490-495.
31. Butz, M. MEDAL Rep 2007008. Jul. 2007 Documentation of XCSFjava 1.1 plus visualization.
32. Smith, R.; Jiang, M. MILCS in protein structure prediction with default Hierarchies. Proc 1st ACM/SIGEVO Summit Genetic Evolutionary Computation; 2009; p. 953-956.
33. Moore, J.; Lari, R.; Hill, D.; Hibberd, P.; Madan, J. Human microbiome visualization using 3D technology. Proc Pacific Symp Biocomputing; 2011; p. 154

Figure 1.

Heat-map visualizations of the evolved M-LCS rule population. (a) illustrates the raw rule population after encoding and expansion. Each row in the heat-map is 1 of 2000 rules comprising the population. Each column is one of the 20 attributes. Four of these (X0, X1, X2, and X3) were modeled as predictive attributes. (b) illustrates the same population after hierarchical clustering on both axes. According to the attribute dendrogram, each pair of interacting attributes (X0, X1) and (X2, X3), modeled in this data, cluster together best of all attributes.

Figure 2.

3D visualization for the identification of interesting rules. In this figure, specified attributes are blue, while ‘#’ attributes are yellow. The height of attributes within each row/rule is the product of the SpS for that attribute and the numerosity of the rule.

Figure 3.

Co-occurrence networks. (a) Illustrates the fully connected network before any filtering is applied. The diameter of a node is the SpS for that attribute, edges represent co-occurrence, and the thickness of an edge is the respective CoS. (b) The network after filtering out all CoSs that did not meet the significance cut-off. (c) The network after filtering out all but to the two highest CoSs.

Manual rule inspection of our UCS rule population trained on the entire simulated dataset. Rules (R's) are ordered by numerosity (Num.). To save space we have left out SNPs (X's) which were generalized over these top 10 rules. The accuracy of each rule listed in the table was 100%.

TABLE 1

	X0	X1	X2	X3	X4	X6	X7	X8	X10	X11	X15	X18	CLASS	NUM.
R1	#	#	0	1	#	#	1	#	#	#	#	1	1	10
R2	1	0	1	0	#	#	#	#	#	#	#	#	1	9
R3	#	0	1	1	#	#	#	#	0	#	1	#	0	9
R4	2	1	#	#	#	#	1	#	#	#	#	#	1	6
R5	1	1	#	#	2	#	#	#	#	0	#	#	0	6
R6	0	1	0	1	#	#	#	#	#	#	#	#	1	6
R7	#	#	1	0	#	0	#	#	0	#	1	#	1	6
R8	1	0	#	#	#	#	#	2	#	#	#	#	1	5
R9	#	#	1	1	#	0	#	#	0	#	#	1	0	5
R10	#	#	1	1	#	#	#	#	0	#	2	#	0	5

TABLE 2

Example calculation of SpS and AWSpS within unordered hypothetical rules.

	X1	X2	X3	X4	CLASS	NUMEROSITY	ACCURACY
R1	2	#	#	1	0	5	0.73
R2	#	0	#	2	1	1	0.51
R3	2	0	#	1	0	2	0.88
R4	#	0	1	#	1	1	0.62
SpS	7	4	1	8			
AWSpS	5.41	2.89	0.62	5.92			

Table 3

SpS and AWSpS results.

ATTRIBUTES	SpS	p-VALUE	AWSpS	p-VALUE
X0	10885	0.001*	7589.49	0.001*
X1	11359	0.001*	7936.43	0.001*
X2	10569	0.001*	7369.84	0.001*
X3	10150	0.001*	7114.25	0.001*
X4	3863	0.999	2482.56	0.888
X5	3240	1	2090.05	1
X6	5217	0.737	3446.47	0.18
X7	5484	0.915	3647.67	0.336
X8	4429	0.95	2927.85	0.482
X9	5334	0.985	3569.25	0.484
X10	5907	0.414	3948.81	0.04*
X11	5725	0.414	3933.61	0.037*
X12	5273	1	3518.87	0.761
X13	4443	1	2854.43	0.996
X14	3709	1	2391.91	0.978
X15	5108	0.916	3425.64	0.355
X16	3613	1	2299.56	1
X17	3639	1	2302.01	0.992
X18	5933	0.629	4004.96	0.085
X19	4591	1	2940.28	0.999

TABLE 4

Top co-occurrence sum results.

ATTRIBUTE PAIRS		CoS	p-VALUE
X0	X1	8060	0.001*
X2	X3	7373	0.001*
X1	X2	4223	0.001*
X0	X2	4079	0.001*
X1	X3	3974	0.001*
X0	X3	3829	0.001*
X1	X11	3621	0.001*
X1	X7	3574	0.001*
X0	X11	3540	0.001*
X2	X10	3485	0.001*
X0	X7	3462	0.001*
X3	X10	3392	0.001*
X1	X18	3379	0.001*
X0	X18	3264	0.001*
X1	X15	3255	0.001*
X0	X15	3035	0.001*
X2	X12	3020	0.051*
X2	X18	3016	0.001*