SUPPLEMENTARY MATERIALS: LCS-DIVE: AN AUTOMATED RULE-BASED MACHINE LEARNING VISUALIZATION PIPELINE FOR CHARACTERIZING COMPLEX ASSOCIATIONS IN CLASSIFICATION

Robert Zhang
Institute for Biomedical Informatics
University of Pennsylvania
Philadelphia, PA, 19104
robertzh@wharton.upenn.edu

Rachael Stolzenberg-Solomon
Division of Cancer Epidemiology and Genetics
National Cancer Institute
Shady Grove, MD, USA
rachael.solomon@nih.gov

Shannon M. Lynch
Cancer Prevention and Control
Fox Chase Cancer Center
Philadelphia, PA, USA
shannon.lynch@fccc.edu

Ryan J. Urbanowicz
Institute for Biomedical Informatics
University of Pennsylvania
Philadelphia, PA, 19104
ryanurb@upenn.edu

April 26, 2021

1 Methods: LCS-DIVE

1.1 Input Data Specifications and Options

LCS-DIVE takes a single tabular dataset as an input (.csv or .txt). Since LCS-DIVE conducts internal CV, all instances should be made available in this dataset apart from any that might be further witheld for downstream model validation or replication analysis. The dataset must be (1) fully numeric, (2) constitute a binary or multi-class classification problem with a class outcome column, and (3) the first row of the dataset must contain column labels, e.g. feature names, outcome label, etc. In addition to feature and outcome columns, the dataset can also include 3 optional columns not used in training: (1) an 'instance label column', to uniquely identify each instance (2) 'true group column', to label instances by their true subgroup in simulated heterogeneous problems allowing for downstream analysis of successful subgroup identification, and (3) 'match label column', for matched cross validation if applicable to the given dataset. Matched cross validation ensures that instances that were matched to account for covariates in a given dataset analysis are kept together within CV partitions. Aside from these constraints, the system is versatile, reflecting the versatility of the original ExSTraCS algorithm. Missing feature data is accepted, and features can be either real or discrete-valued.

1.2 Scikit-ExSTraCS Hyperparameters and Logistics

Within LCS-DIVE, users can specify five key scikit-ExSTraCS hyperparameters to fine tune training (Table 1). All other ExSTraCS hyperparameters are left at their default values as described in [10], as they are commonly viewed as a stable and effective hyperparameter configuration within LCS algorithms. Users can optionally run scikit-ExSTraCS outside of LCS-DIVE and tune any and all hyperparameters.

Table 1: Key LCS-DIVE parameters

Parameter Name	Default Value	Description
–cv	10	# of CV Partitions
–iter	16000	# of Learning Iterations
–N	1000	Maximum micropopulation size
–nu	1	Standard nu hyperparameter
–fssample	1000	Feature Evaluation sample size: see next section

Further, LCS-DIVE has been extended such that it can load modeling outputs generated by scikit-ExSTraCS previously. This way users can simply run the remaining three phases of LCS-DIVE on phase 1 modeling that had been previously conducted. This allows LCS-DIVE to be applied more flexibly once it has been determined that ML modeling has yielded a predictive model worth interpreting further.

After training, for each CV split, LCS-DIVE saves the trained ExSTraCS model containing (1) feature tracking (FT) scores for each training instance and (2) the respective rule population. It also saves a record, for each testing instance, of the model vote for each class, along with predicted class and true instance class. This allows the model to be applied later as a probability machine.

1.3 FT Clustering

1.3.1 Seaborn Clustering

We used Seaborn's clustermap method to cluster FT scores as well as the rule population. This method applies hierarchical clustering. We chose to use hierarchical clustering against other clustering methods, such as k-means clustering, because it gives more flexibility to adjusting the number of discovered clusters. This is crucial in our case, since the true number of clusters is unknown. To perform clustering, we chose the Pearson correlation metric to find the distance between instances and the Ward method to find the distance between hierarchical clusters. While we do not claim these metrics to be the optimal way to cluster FT scores in all contexts, we believe these metrics to be reasonable because they consistently yielded the most visually and functionally desirable clusters in datasets where we knew the 'true' subgroups of each instance compared to other metrics Seaborn offered.

1.3.2 Method to finding statistically significant clusters

To find statistically significant clusters of FT scores and rules, we used the Monte Carlo based method described in Kimes Liu's 2017 paper 'Statistical Significance for Hierarchical Clustering'. In brief, the method works from the root of the dendrogram downwards. At each node of the dendrogram, the 'average' cluster instance is computed by taking the mean and standard deviations of each feature for the b instances in that cluster. Then, a size b gaussian sample of instances is generated using the average instance and feature standard deviations. This generated sample of instances is again clustered with Pearson correlation and the Ward method, and the Ward distance between the two largest found clusters is computed. This sampling is done 100 times at each node. If the actual Ward distance at the node was larger than the randomly sampled Ward distance more than $1 - 0.05 * \frac{N_{j-1}}{N-1} * 100$ percent of the time, where N_j equals the number of instances at the given node, and N equals the total number of instances, then the cluster is significant and we can perform the same procedure on the next two sub-clusters.

For each clustermap, LCS-DIVE automatically generates a summary of FT clusters that includes FT sums, class imbalance, average testing accuracy, and true group makeup (if applicable) of the instances in each cluster. This can be applied to further interrogate the instance makeup of discovered clusters.

1.4 Rule Encoding for Visualization

After merging the *n* rule sets, individual rules are converted into corresponding binary strings with length equal to the number of features and an encoding of 1 or 0 is applied for specified or ignored, respectively. LCS rules maintain a *numerosity* parameter that maintains the number of virtual copies maintained in the rule population for that rule. Rules that gain and retain a larger numerosity are considered to be more successful and stable. Rule numerosity also plays a role in LCS model prediction (where a higher numerosity equals a larger class vote) [9]. As described in [7] rules with numerosity m are copied to appear m times in the binary rule set. This way rules with higher numerosity have greater influence on the rule set interpretation.

2 Methods: Datasets

2.1 MUX Benchmark Dataset Generation

We generated our MUX datasets by randomly sampling the 2^n possible instances. For the 6-bit MUX problem, where only 64 instances exist, we generated 500 instances by sampling instances more than once. 10-fold CV was used for training and testing evaluation.

Further, we labelled each instance with a 'true group' label that indicates the instance's correct homogeneous subgroup. For example, in the 6-bit MUX, all instances with address bits 00 were given the same label, all instances with address bits 01 were given the same label, etc. for a total of 4 distinct labels. These labels weren't used as a feature used in training, but were given to LCS-DIVE afterwards to evaluate the pipeline's performance in characterizing heterogeneity.

2.2 GAMETES Simulated SNP Dataset Generation and Run Parameters

All SNP features generated have three possible values coded as (0, 1, or 2) corresponding to genotypes (e.g. AA,Aa,aa). A recently expanded version of this software is available at (https://github.com/UrbsLab/GAMETES), which not only capable of generating purely epistatic SNP models and datasets, but univariate, additively combined, and heterogeneously combined data as well. We simulated 21 SNP datasets (see Table 2 using the GAMETES software package. For simplicity, for each data scenario, the easiest variation of model architecture was selected for dataset generation as described in [8]. Unlike MUX problem heterogeneity, heterogeneity simulated in GAMETES datasets introduces a stochastic overlap between instance subgroups such that, by chance, a proportion of instances may reflect the association of more than one underlying model comprising the heterogeneity. As a result, this form of heterogeneity simulation introduces additional noise and complexity into the simulated data making it harder to achieve optimal testing accuracy. One can think of the MUX heterogeneity as being deterministic, and the GAMETES heterogeneity as being probabilistic.

Table 2: Characteristics of the 21 GAMETES Simulated Datasets

Dataset ID	Underlying Association	Predictive Features	# of Models	Model Ratio	Model Heritability	Instances
1 & 2	univariate	1	1	NA	1 & 0.4	1600
3 & 4	additive	2	2	equal	1 & 0.4	1600
5 & 6	additive	4	4	equal	1 & 0.4	1600
7 & 8	2-way pure epistasis	2	1	NA	1 & 0.4	1600
9 & 10	3-way pure epistasis	3	1	NA	1 & 0.2	3200
11 & 12	additive (2-way epistasis + univariate)	4	3	equal	1 & 0.4	1600
13 & 14	additive (2, 2-way epistasis)	4	2	equal	1 & 0.4	1600
15 & 16	heterogeneity (univariate)	2	2	equal	1 & 0.4	1600
17 & 18	heterogeneity (univariate)	4	4	equal	1 & 0.4	1600
19 & 20	heterogeneity (2, 2-way epistasis)	4	2	equal	1 & 0.4	1600
21	heterogeneity (2, 2-way epistasis)	4	2	75 : 25	0.4	1600

Each of these datasets had: (1) 1600 instances (with the exception of Datasets 9 and 10 which had 3200 instances) (2) 20 features (3) Each feature having possible values 0, 1, 2 (4) A binary outcome variable (5) Balanced cases and controls

These 21 datasets were all run with identical LCS-DIVE parameters: (1) 200k learning iterations (with the exception of Datasets 9 and 10 which had 500k learning iterations) (2) N = 2000 (3) MultiSURF sample size = 1000 (4) nu = 10 for clean problems, nu = 1 for noisy problems (5) Testing was done over 10 cross validation partitions

2.3 Pancreatic Cancer Datasets and Run Parameters

The three derived case-control datasets examined here were were previously described in detail and preliminarily analyzed with a rigorous ML prediction pipeline [11]. When running LCS-DIVE on these 2 datasets, we used the same parameters as was used for the simulated datasets: 200k learning iterations, N = 2k, nu = 1, MultiSURF sample size = 1000. 10-fold CV was applied as usual when running LCS-DIVE.

2.3.1 PLCO Full Study Population

The PLCO cohort is derived from the PLCO screening trial, which is a randomized multicenter trial in the United States (Birmingham, AL; Denver, CO; Detroit, MI; Honolulu, HI; Marshfield, WI; Minneapolis, MN; Pittsburgh, PA; Salt Lake City, UT; St. Louis, MO; and Washington, DC) with 152,810 men and women ages 55 to 74 at baseline, that sought to determine the effectiveness of early detection procedures for prostate, lung, colorectal, and ovarian cancers on disease-specific mortality [2]. Details of the study methods have been previously described [2]. The PLCO screening arm included approximately 77,000 men and women. Study recruitment and randomization began in November 1993 and was completed in July 2001. Data on demographics, health history, diet and other lifestyle factors (intervention arm only), were collected from self-administered questionnaires at baseline. A second self-administered dietary questionnaire was distributed to the intervention and control arms of the trial between 1998 and 2005 to provide additional dietary data [5]. Cancer cases were identified by self-report in the annual mail-in survey and cohort participants were also linked to cancer registries and the National Death Index. Medical and pathology records were obtained if possible and cancer cases confirmed by study staff [4]. In this study, we included incident primary pancreatic adenocarcinomas [International Classification of Diseases, ICD-O-3 code C250-C259 or C25.0-C25.3, C25.7-C25.9] diagnosed between 1994 and 2014. In total, 800 confirmed pancreatic adenocarcinoma cases were identified across both study arms for this analysis.

2.3.2 Pancreatic Cancer Case-Control Subsets

We derived two case-control analytic data sets (P1 and P2) within the PLCO study each defined by different a) case-control selection criteria, b) bias considerations, and c) available features (see Table 3). These datasets include all confirmed pancreatic cases diagnosed in the full cohort (n=800) and only healthy controls with available genotyping data from previous genome-wide association studies (n=4298) [12, 6, 1]. Thus, P1 and P2 introduce selection bias due to control selection. In particular, control instances included more males (85%) and smokers than would have been expected by chance because these control sets were used for GWAS involving smoking and male-dominated cancer (e.g. lung cancer). Thus, beyond selection bias, these control sets can also introduce confounding. P2 is very similar to P1 but adds 14 dietary features which include a much higher frequency of missing values in cases, i.e. 19% in contrast with controls at 2%. P2 is considered to be the most biased sample in this study as it includes the biases from P1 along with this missing data.

Table 3: Case-control subsets derived from the Full PLCO cohort

Dataset	Cases	Controls	Features	Major Bias Considerations
Dataset 1	800	4298	24	Sample selection bias
Dataset 2	800	4298	38	+ Dietary feature missingness

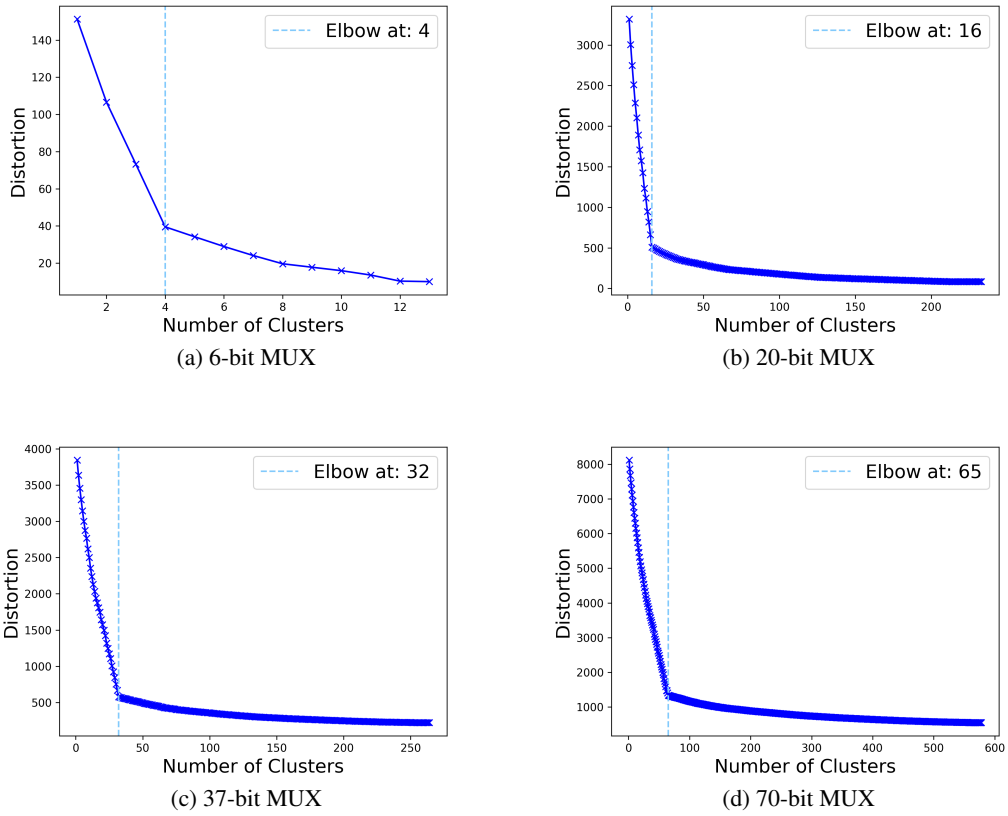
The set of universal risk factors examined in our two target datasets include: (1) established risk factors that are often matched upon and that are either innate (i.e. age, gender, race/ethnicity) or related to study variability (i.e. center, randomization year), (2) established modifiable pancreatic cancer risk factors; i.e. cigarette smoking (yes/no, former smoker, duration/years smoked, pack-years smoked), current body mass index (BMI), diabetes (yes/no), and pancreatic cancer family history (yes/no), (3) other factors that have been variably associated with pancreatic cancer; i.e. education status, BMI at age 20 and age 50, any family history cancer (yes/no), gallbladder and liver disease (yes/no), and (4) risk factors that have not been associated with pancreatic cancer; i.e. marital status, occupation, medication use aspirin (yes/no), ibuprofen (yes/no), daily dose of aspirin and ibuprofen.

Further, dataset P2 includes *dietary factors* that have been variably associated with pancreatic cancer including; glycemic index, glycemic load, and total daily intake of carbohydrates, energy, alcohol, fat, folate, red meat, protein, cholesterol, and calcium. In particular, we wanted to determine whether the dietary exposures could improve prediction models. This entire collection of candidate risk factors is based on literature review [3].

3 Results

3.1 MUX Results

Notably, each clustered FT subgroup includes both cases and controls that were correctly predicted with a respective subset of features. Figure 1 shows the DIVE generated elbow plots for these problems (excluding 11-bit, which appears in the main paper).



(a) 6-bit MUX

(b) 20-bit MUX

(c) 37-bit MUX

(d) 70-bit MUX

Figure 1: Elbow Plots of 6, 20, 37, and 70-bit MUX datasets.

3.1.1 MUX Rule Population Interpretation Example

Figure 2 shows the clustermaps and networks for the 6 bit MUX problem.

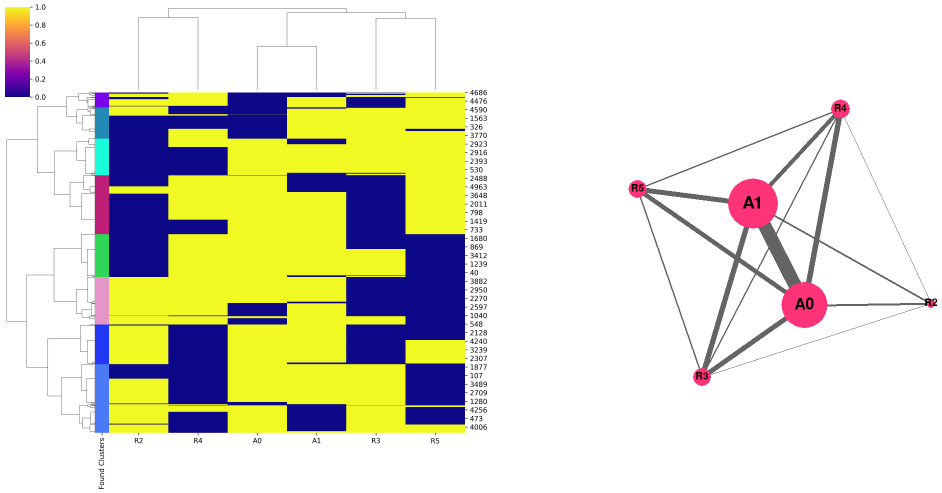
Compared to FT score clustermaps, the clustermaps from the rule population are not as informative: they are much noisier without having applied rule compaction. While the elbow plot from the FT clustermap recommended the correct number of clusters, the elbow plot for this clustermap does not. Regardless, some patterns can still be distinguished, which makes this method potentially valuable when FT scores are not available. The network diagrams are much more useful. The epistatic relationships between the address bits are very apparent. However, given that the network diagrams are an aggregate view of the model, they are unable to capture the importance of register bits. In addition, in real world problems, it's not easy to discriminate epistasis based on the network diagram's edges alone. Features can co-occur in a rule for many reasons other than epistasis (e.g. by coincidence), which confounds the interpretation of edge thickness.

3.2 GAMETES Simulated Dataset Model Performance Results

We summarize the automated results across all 21 SNP datasets including average balanced testing accuracy, and the elbow-method automatically suggested number of 'found' clusters (see Table 4).

For D1-D6, clean datasets (D1,D3,D5) yield scikit-ExSTraCS models with perfect testing accuracy, and those with noise (D2,D4,D6) achieve reduced but expected accuracies. Note that in generating additive models, GAMETES reduces the amount of noise in the resulting datasets reflected by the increasing balanced testing accuracy in going from the univariate to 2 additive and 4 additive model datasets. This is because even if one predictive feature value is noisy, the other predictive features in that instance would still point to the correct phenotype, minimizing the effect of the noisy value.

For D7-D14, clean datasets (D7,D9,D11,D13) yield scikit-ExSTraCS models with mostly perfect testing accuracy, and those with noise (D8,D10,D12,D14) again achieve reduced but expected accuracies.



(a) 6-bit Clustermap

(b) 6-bit Feature Cooccurence Network

Figure 2: Rule Population Clustermaps and Attribute Cooccurence Networks of the 6-bit problem

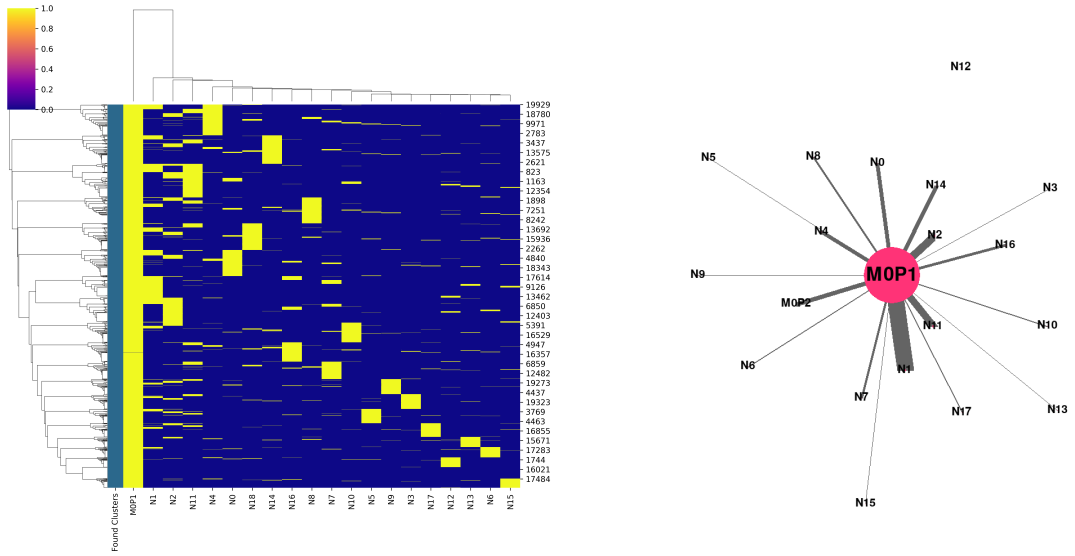
For D15-D21, clean datasets (D15,D17,D19) yield scikit-ExSTraCS models with higher testing accuracies while those with noise (D16,D18,D20,D21) achieve reduced accuracies as expected. Notice that the process of generating heterogeneity, even in clean data, makes it more difficult to achieve 100% testing accuracy in the resulting datasets. This is because, unlike in the MUX problems, there is no feature or set of features that indicate which other heterogeneous set of features are relevant to make an accurate prediction for a given instance.

Table 4: LCS-DIVE results summary for the 21 GAMETES dataset scenarios

Dataset ID	Underlying Association	Predictive Features	Model Her.	True Clusters	Found Clusters	Balanced Accuracy
D1	univariate	1	1	1	6	1.0
D2	univariate	1	0.4	1	6	0.846
D3	additive	2	1	1	7	1.0
D4	additive	2	0.4	1	5	0.875
D5	additive	4	1	1	6	1.0
D6	additive	4	0.4	1	10	0.944
D7	2-way pure epistasis	2	1	1	6	1.0
D8	2-way pure epistasis	2	0.4	1	8	0.798
D9	3-way pure epistasis	4	1	1	10	1.0
D10	3-way pure epistasis	4	0.1	1	16	0.669
D11	additive (2-way epistasis + univariate)	4	1	1	10	1.0
D12	additive (2-way epistasis + univariate)	4	0.4	1	4	0.948
D13	additive (2, 2-way epistasis)	4	1	1	3	0.999
D14	additive (2, 2-way epistasis)	4	0.4	1	5	0.913
D15	heterogeneity (univariate)	2	1	2	3	0.984
D16	heterogeneity (univariate)	2	0.4	2	4	0.684
D17	heterogeneity (univariate)	4	1	4	8	0.939
D18	heterogeneity (univariate)	4	0.4	4	3	0.673
D19	heterogeneity (2, 2-way epistasis)	4	1	2	5	0.764
D20	heterogeneity (2, 2-way epistasis)	4	0.4	2	4	0.680
D21	heterogeneity (2, 2-way epistasis)	4	0.4	2	6	0.723

3.2.1 D1 Rule Population Visualization Example

Figure 3 shows the clustermap and network for the 1 feature main effect without noise problem.



(a) Clustermap (b) Network

Figure 3: Rule Population Clustermap and Network of Clean 1 Feature Main Effect Problem

The predictive power of the M0P0 feature is clearly shown by the clustermap and network. However, noise is prevalent in the clustermap, which make it very difficult to discriminate noise from heterogeneity. This is in large contrast to the FT clustermap which shows the main effect without other misleading markers.

Since this rule population method can only poorly characterize the most basic pattern (1 feature main effect), we discovered that adding complexity (e.g. noise, epistasis, heterogeneity) makes its visualizations less informative.

3.3 GAMETES Heterogeneous Association Results: Elbow Plots

Figure 4 shows the corresponding LCS-DIVE generated FT elbow plots for D15 - D20.

3.4 GAMETES D21 Results

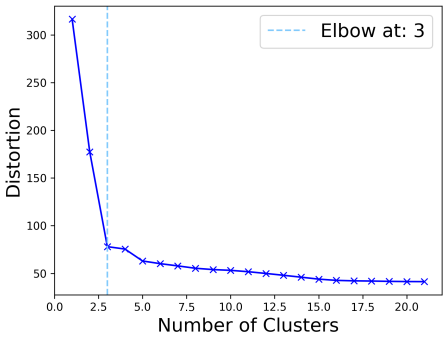
D21 constitutes a noisy, heterogeneous dataset with the two 2-way models combined in an imbalanced way (i.e 75:25 heterogeneous model ratio where one model is applied to 75% of instances and the other to 25% of instances). FT clustermap and corresponding elbow plot is given in Figure 5. We would expect to see 3 clusters of instances similar to the balanced heterogeneous dataset. However, we only see 2 clear ones. The cluster that emphasizes Model M1's features, and darkens Model M0's features, is missing. Since instances of the M0 model are much more prevalent, many rules in the rule population would have M0P0 and M0P1 specified. This means that for instances belonging to the second model, M0P0 and M0P1 are still imprinted on their FT scores, and aren't darkened. Even if M0P0 and M0P1 weren't specifically predictive for that given instance, their high specificity throughout the rule population leads them to be emphasized anyway.

3.5 FT signatures in GAMETES Datasets: Comparison and Further Discussion

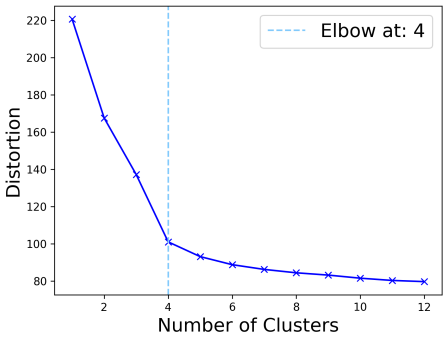
In the following subsections we highlight some of the similarities and subtle differences between some of the FT signatures for simulated GAMETES datasets.

3.5.1 GAMETES D6 vs. D18

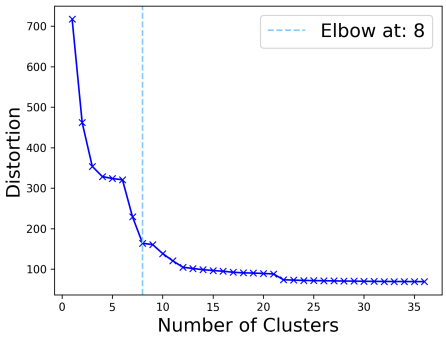
Figure 6 directly compares FT clustermaps for D6 and D18 wherein the goal was to distinguish 4 additively combined features from 4 heterogeneously combined features in noisy simulations. The most obvious distinction was that the additive combination (D6), yielded a large cluster where all 4 predictive features were informative while this was not the case for D18. Further, for D6, most of the other clusters emphasized 3 out of 4 features as being predictive, while for D18 there was a mix of 1,2,3, or all 4 features appearing informative within a variety of subgroups.



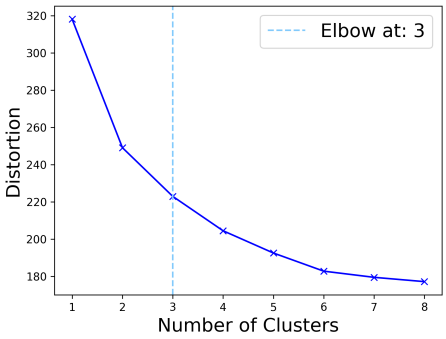
(a) D15: Clean 2-feature heterogeneous univariate



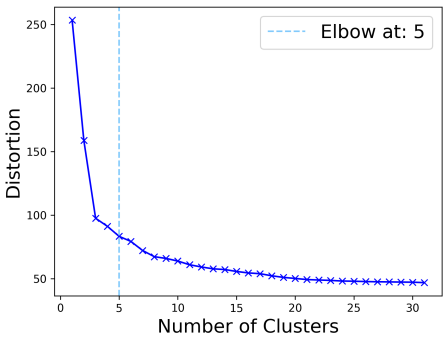
(b) D16: Noisy 2-feature heterogeneous univariate



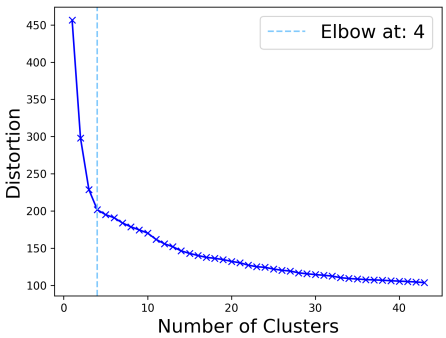
(c) D17: Clean 4-feature heterogeneous univariate



(d) D18: Noisy 4-feature heterogeneous univariate

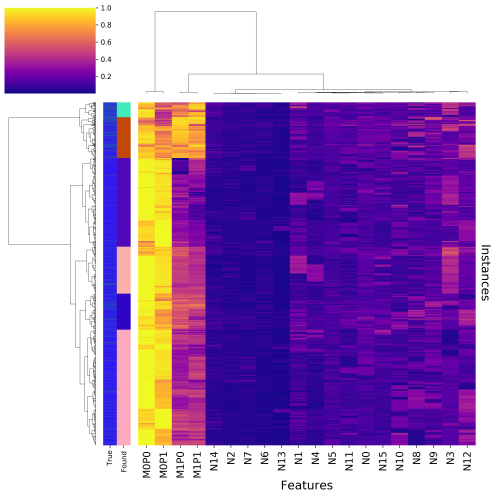


(e) D19: Clean 2 sets of 2-way Epistasis, heterogeneous

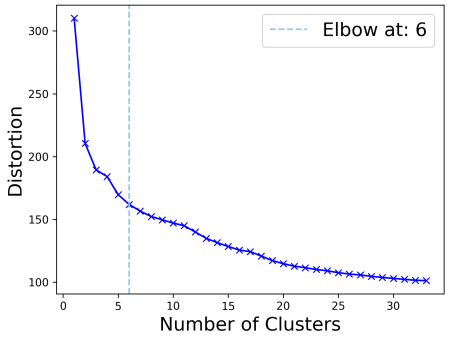


(f) D20: Noisy 2 sets of 2-way Epistasis, heterogeneous

Figure 4: Elbow Plots of D15 - D20 GAMETES datasets.

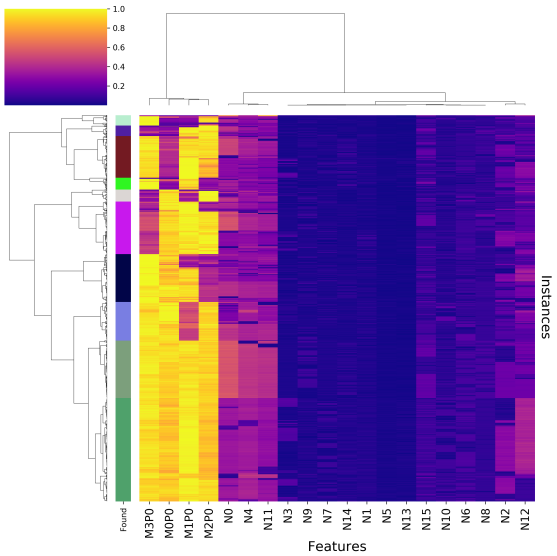


(a) D21: Noisy Imbalanced Heterogeneous Epistasis, Clustermap

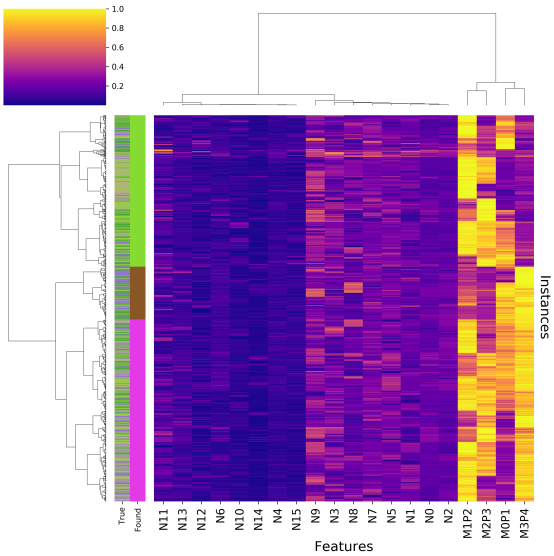


(b) D21: Noisy Imbalanced Heterogeneous Epistasis, Elbow Plot

Figure 5: D21: Noisy Imbalanced Heterogeneous Epistasis



(a) D6: Noisy 4-feature additive univariate

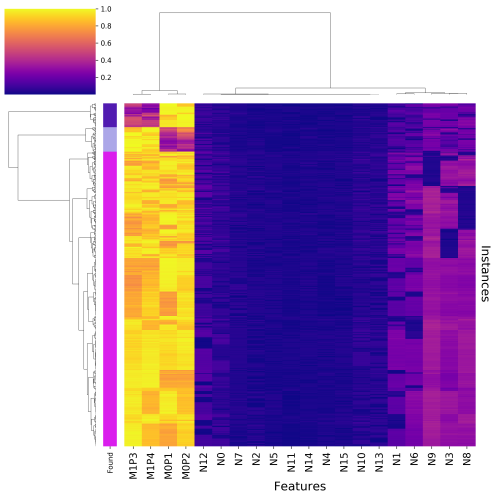


(b) D18: Noisy 4-feature heterogeneous univariate

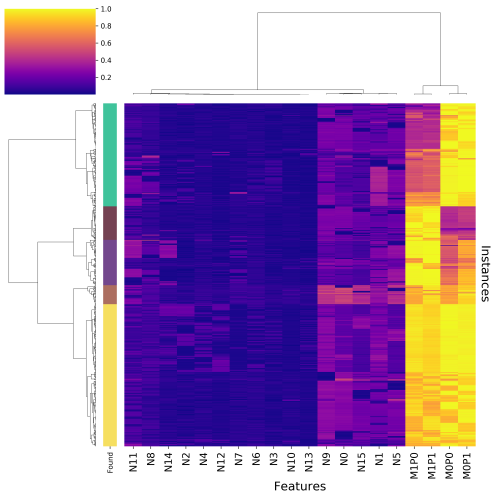
Figure 6: Comparing D6 and D18 clustermaps: Additive vs. heterogeneous univariate associations

3.5.2 GAMETES D14 vs. D20

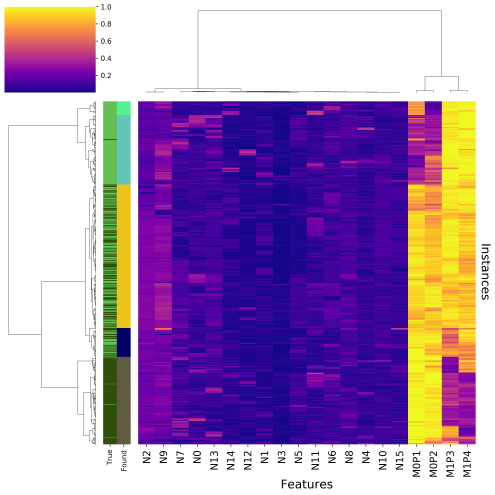
Figure 7 directly compares FT clustermaps for datasets D14 and D20 (subplots on the right) wherein the goal was to distinguish and additive combination of 2-way epistasis from a heterogeneous combination of 2-way epistasis in noisy simulations. Included in this figure are D13 and D19 which are the clean dataset counterparts for each scenario to provide further context. Again the major, albeit subtle, distinction between D14 and D20 is that in D14 we observe a larger stronger FT cluster where all four predictive features appear important (i.e the lowest cluster in Figure 7B). However, one could easily argue that the FT signatures between D14 and D19 are indistinguishable, i.e. they both have a large cluster where all four predictive features are important, and two additional clusters where only one of the epistatic feature pairs are important. In this case it's important to be aware of the LCS model testing accuracy (i.e. it was 91% for D14 and 76% for D19), where overlapping heterogeneous associations are much harder to achieve high testing accuracies than for additive ones. Knowing the model testing accuracy thus can better inform interpretation of the FT signatures using the simulation scenarios in this work as a reference. While it may not always be possible to definitively distinguish additive from heterogeneous associations (with FT signatures alone), we can leverage (1) individual cluster analyses, (2) LCS model testing accuracy, and (3) targeted analysis of the instances within 'found' clusters using the found-cluster-tagged datasets output by LCS-DIVE to conduct follow up characterization of the LCS models.



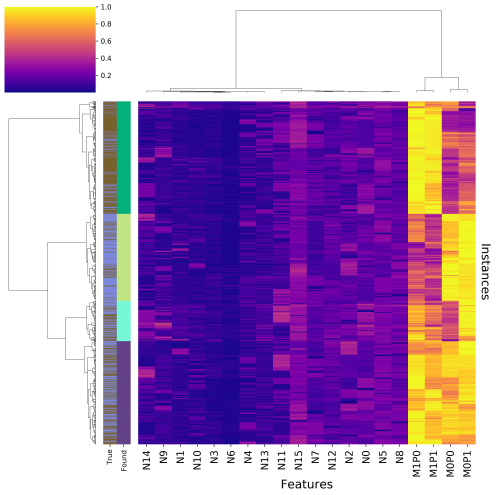
(a) D13: Clean additive (2, 2-way epistasis)



(b) D14: Noisy additive (2, 2-way epistasis)



(c) D19: Clean 2 sets of 2-way Epistasis, heterogeneous



(d) D20: Noisy 2 sets of 2-way Epistasis, heterogeneous

Figure 7: Comparing D14 and D20 clustermaps: Additive vs heterogeneous epistatic associations. D13 and D19 included for context

3.6 LCS-DIVE and Rule Population Analysis Evaluation Times

Table 5 summarizes the average scikit-ExSTraCS training time, FT score analysis time, and rule population analysis time for all of the datasets using LCS-DIVE. The simulated datasets are labelled D1 through D21, and the pancreatic cancer datasets are labelled P1 and P2.

Table 5: Experimental Run Times For All Datasets

Dataset	Average scikit-ExSTraCS Training Time (s)	FT Cluster and Viz. Time (s)	Rule Cluster and Viz Time (s)
6-bit MUX	13	27	139
11-bit MUX	48	168	652
20-bit MUX	636	946	3979
37-bit MUX	3343	1107	39111
70-bit MUX	24743	4712	TIMEOUT
D1	508	68	747
D2	1208	88	759
D3	606	82	752
D4	927	48	4978
D5	621	77	667
D6	980	101	2498
D7	552	67	5230
D8	1065	114	5048
D9	1433	200	5215
D10	3167	220	4322
D11	561	84	1970
D12	875	39	793
D13	996	39	1196
D14	615	102	1737
D15	996	76	3979
D16	773	35	863
D17	1190	113	4156
D18	1164	24	869
D19	1166	72	1873
D20	1298	124	1578
D21	1244	106	870
P1	1568	440	844
P2	1877	288	894

3.6.1 Fitting Time

Model fitting time is a function of feature evaluation time (using MultiSURF) and training time (using ExSTraCS). MultiSURF has time efficiency $O(n^2)$ where $n =$ number of instances in multiSURF subset. For all datasets, except the 70 bit MUX, anywhere from 450 to 1000 instances were used. However, the 70 bit MUX used 9000 instances, which significantly increased fitting time. Training time is mostly a function of N and the number of learning iterations, which explains the similar training times for the simulated and real world datasets (since they used the same MultiSURF subset size, the same N, and the same number of learning iterations). Training time for the 37 bit MUX, which also used 200k learning iterations, was larger due to the larger N.

Overall, training time for the datasets are within reasonable bounds to achieve a good testing accuracy. In real world scenarios, feature selection using MultiSURF would typically take a disproportionate amount of time, if many instances are used, so the subset size should be minimized if possible.

3.6.2 FT and Rule Population Analysis Time

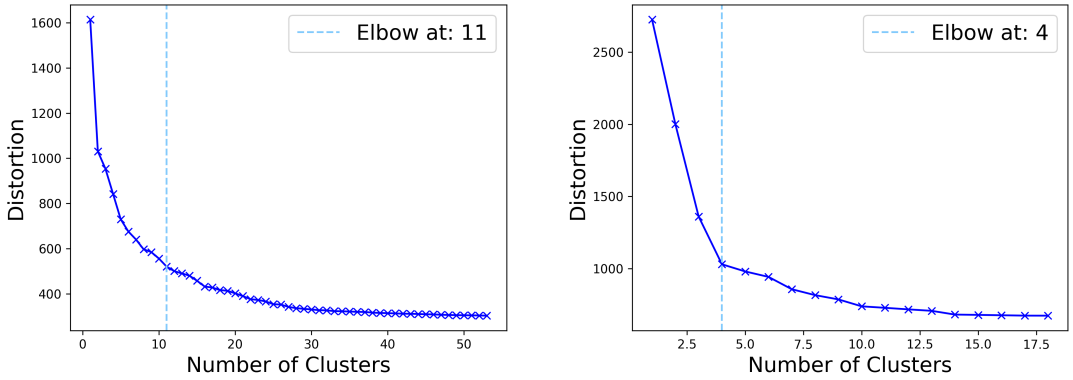
FT Analysis Time is a function of how many instances and features are in the dataset, while Rule Population Time is a function of micropopulation size, CV count, and feature count. In our experimentation, since we typically used a smaller number of instances (most 1600 instances, max 30k instances), FT population time was very reasonable: most finished within a few minutes. Meanwhile, rule population times exploded at times due to the high CV count and N.

FT analysis seems to perform in reasonable time (<1 hr) for small to medium sized datasets, such as those tested in this paper.

3.7 Pancreatic cancer

3.7.1 FT clustermaps

Figure 8 shows the elbow plots for P1 and P2.



(a) P1

(b) P2: Adds Dietary features to P1

Figure 8: FT Elbow Plots of Pancreatic Datasets P1 and P2

The LCS-DIVE recommended FT clustermap with 11 clusters for P1 is given in Figure 9.

Figure 10 gives the FT clustermap for the P2 dataset (both recommended and chosen clusters were 4). The cluster findings were very similar to that as in P1, however here a number of (newly added) dietary features yield the strongest secondary FT signatures within the large cluster. Here we identified 4 clusters where clusters pink, purple, green, brown yielded testing accuracies of 0.0142, 0.0788, 0.9893, 0.8983 respectively. Again the bottom two clusters have a similar signature to the bottom two clusters identified for the P1 dataset.

3.7.2 Rule Population Visualizations

Lastly, we present rule population visualizations for P2 as a final example. Figure 11 and Figure 12 shows the rule clustermap and network, respectively, for this real world problem. The rule clustermap is again difficult to interpret without rule compaction. However, the network diagram cuts through a lot of the noise and shows the strength of the sex and race features. In general, this visualization is a decent model characterization. However, compared to the FT score derived characterization, it lacks nuance.

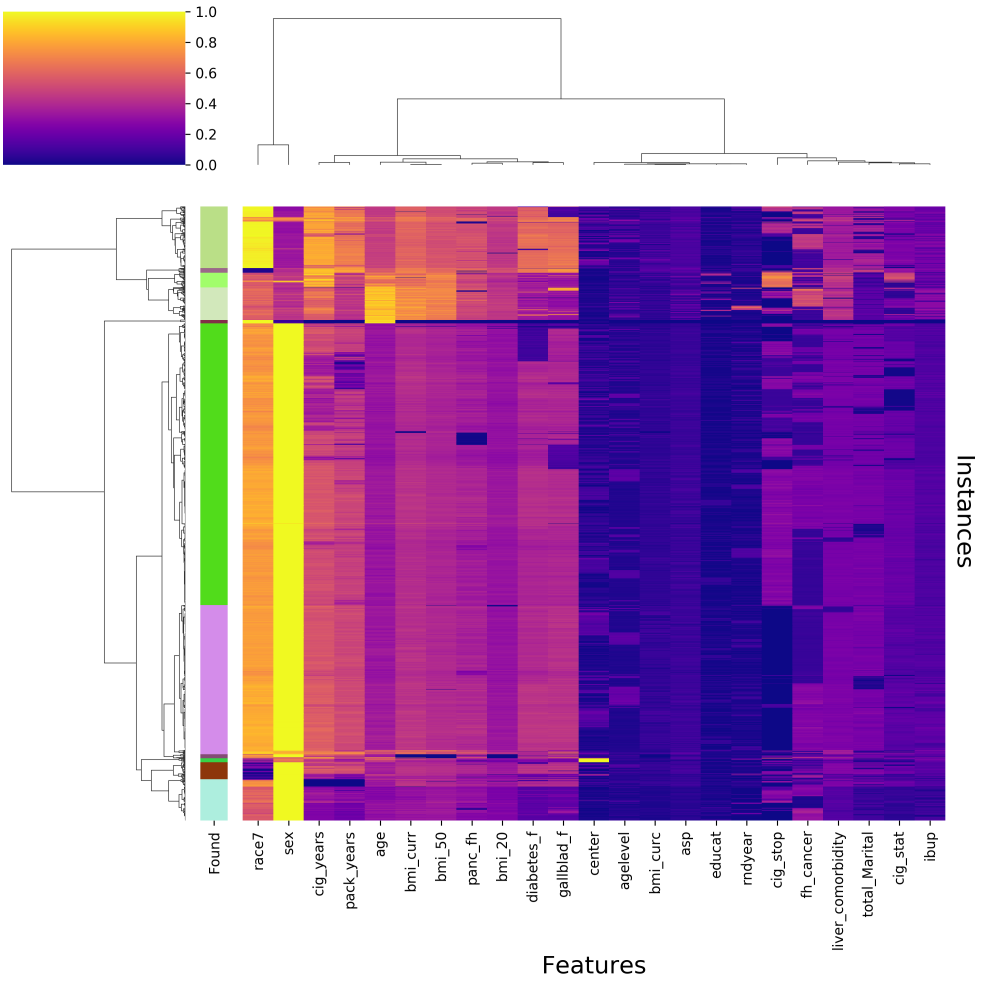
References

[1] M. T. Landi, N. Chatterjee, K. Yu, L. R. Goldin, A. M. Goldstein, M. Rotunno, L. Mirabello, K. Jacobs, W. Wheeler, M. Yeager, et al. A genome-wide association study of lung cancer identifies a region of chromosome 5p15 associated with risk for adenocarcinoma. *The american journal of human genetics*, 85(5):679–691, 2009.

[2] P. C. Prorok, G. L. Andriole, R. S. Bresalier, S. S. Buys, D. Chia, E. D. Crawford, R. Fogel, E. P. Gelmann, F. Gilbert, M. A. Hasson, et al. Design of the prostate, lung, colorectal and ovarian (plco) cancer screening trial. *Controlled clinical trials*, 21(6):273S–309S, 2000.

[3] R. Z. Stolzenberg-Solomon and L. T. Amundadottir. Epidemiology and inherited predisposition for sporadic pancreatic adenocarcinoma. *Hematology/Oncology Clinics*, 29(4):619–640, 2015.

[4] R. Z. Stolzenberg-Solomon, C. C. Newton, D. T. Silverman, M. Pollak, L. M. Nogueira, S. J. Weinstein, D. Albanes, S. Männistö, and E. J. Jacobs. Circulating leptin and risk of pancreatic cancer: a pooled analysis from 3 cohorts. *American journal of epidemiology*, 182(3):187–197, 2015.



(a)

Figure 9: P1 Pancreatic Cancer FT clustermap with LCS-DIVE auto-recommended number of clusters

[5] A. F. Subar, D. Midthune, M. Kulldorff, C. C. Brown, F. E. Thompson, V. Kipnis, and A. Schatzkin. Evaluation of alternative approaches to assign nutrient values to food groups in food frequency questionnaires. *American journal of epidemiology*, 152(3):279–286, 2000.

[6] G. Thomas, K. B. Jacobs, M. Yeager, P. Kraft, S. Wacholder, N. Orr, K. Yu, N. Chatterjee, R. Welch, A. Hutchinson, et al. Multiple loci identified in a genome-wide association study of prostate cancer. *Nature genetics*, 40(3):310, 2008.

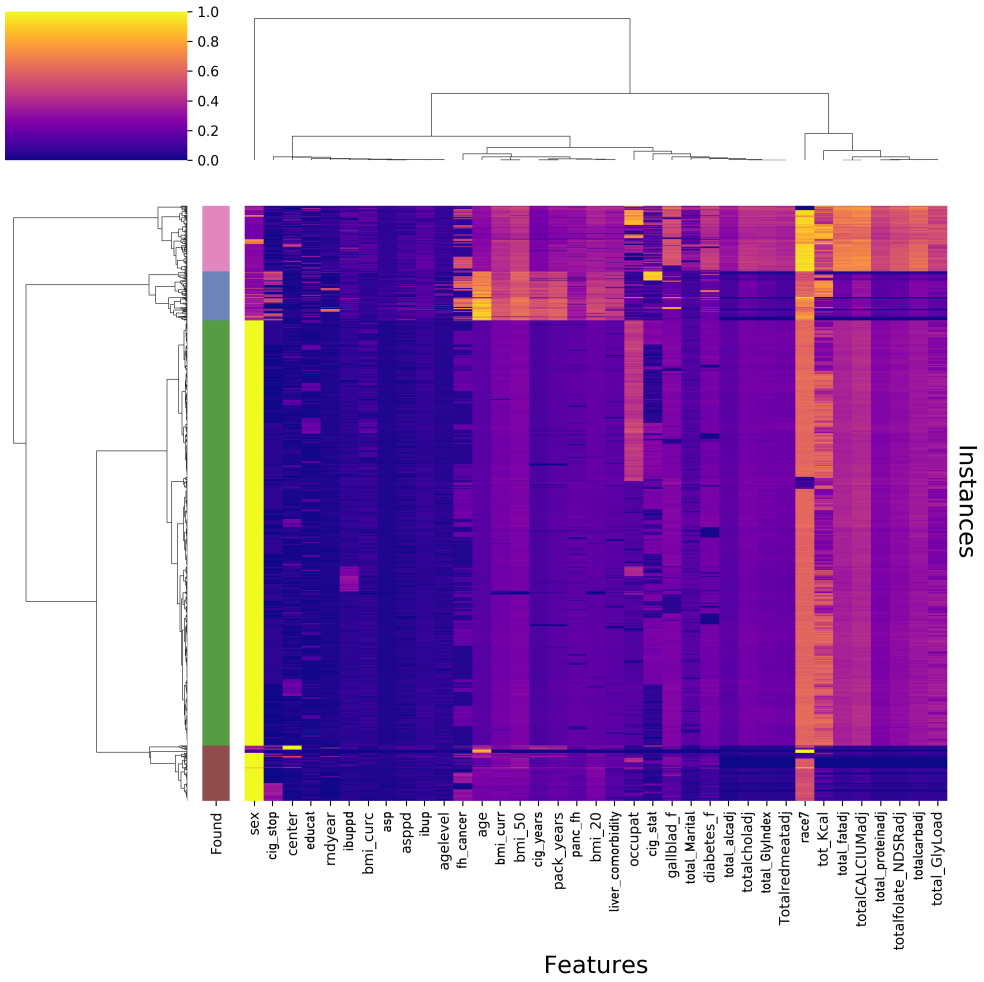
[7] R. J. Urbanowicz, A. Granizo-Mackenzie, and J. H. Moore. An analysis pipeline with statistical and visualization-guided knowledge discovery for michigan-style learning classifier systems. *IEEE computational intelligence magazine*, 7(4):35–45, 2012.

[8] R. J. Urbanowicz, J. Kiralis, J. M. Fisher, and J. H. Moore. Predicting the difficulty of pure, strict, epistatic models: metrics for simulated model selection. *BioData mining*, 5(1):15, 2012.

[9] R. J. Urbanowicz and J. H. Moore. Learning classifier systems: a complete introduction, review, and roadmap. *Journal of Artificial Evolution and Applications*, 2009, 2009.

[10] R. J. Urbanowicz and J. H. Moore. Exstracs 2.0: description and evaluation of a scalable learning classifier system. *Evolutionary intelligence*, 8(2-3):89–116, 2015.

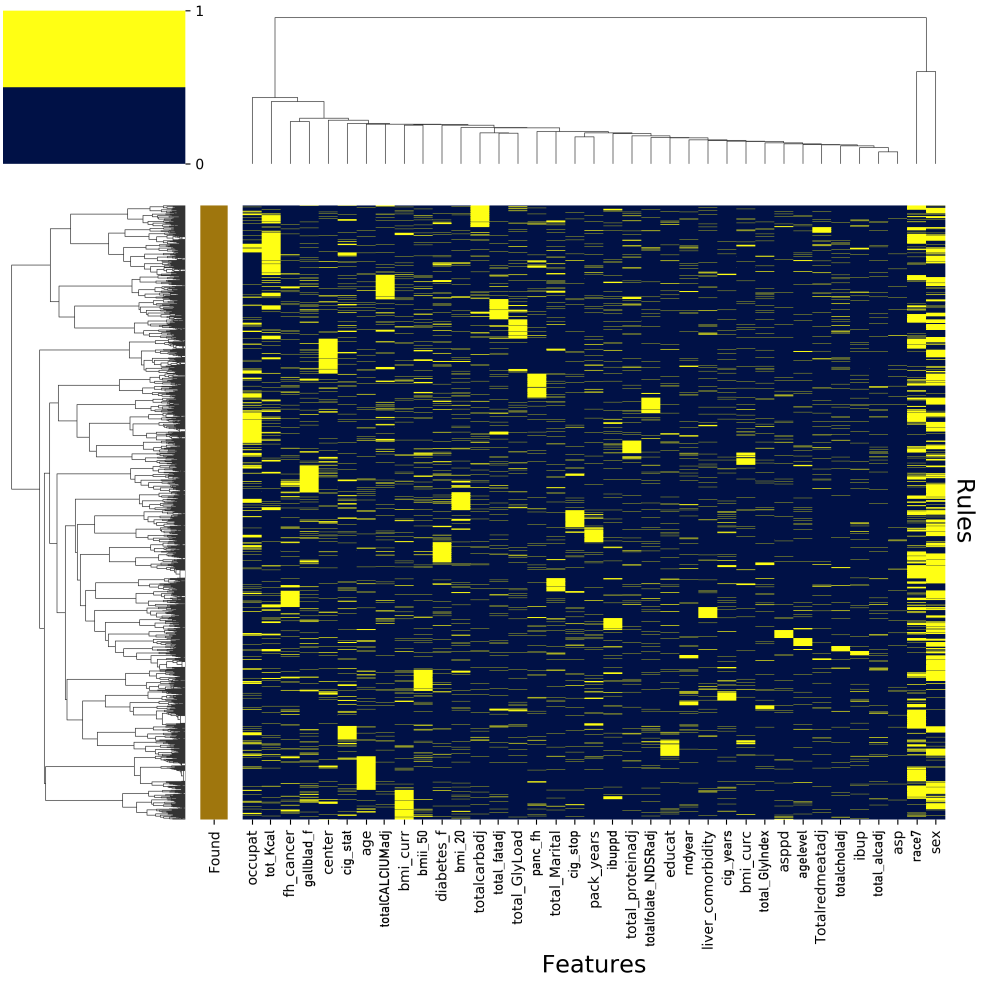
[11] R. J. Urbanowicz, P. Suri, Y. Lu, J. H. Moore, K. Ruth, R. Stolzenberg-Solomon, and S. M. Lynch. A rigorous machine learning analysis pipeline for biomedical binary classification: Application in pancreatic cancer nested case-control studies with implications for bias assessments. *arXiv preprint arXiv:2008.12829*, 2020.



(a)

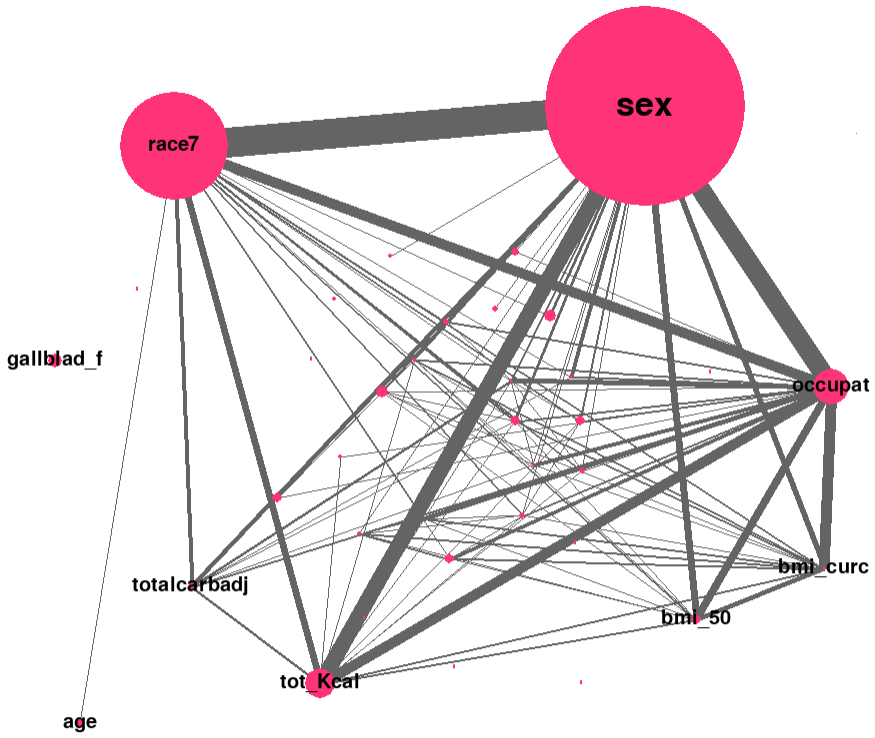
Figure 10: P2 Pancreatic Cancer FT Clustermap

[12] B. M. Wolpin, C. Rizzato, P. Kraft, C. Kooperberg, G. M. Petersen, Z. Wang, A. A. Arslan, L. Beane-Freeman, P. M. Bracci, J. Buring, et al. Genome-wide association study identifies multiple susceptibility loci for pancreatic cancer. *Nature genetics*, 46(9):994–1000, 2014.



(a) Clustermap

Figure 11: P2: Rule Population Clustermap of the dietary feature pancreatic cancer dataset



(a) Network

Figure 12: P2: Rule Population Network of the dietary feature pancreatic cancer dataset