Improving Learning Outcomes by Using Clustering Validity Analysis to Reduce Label Uncertainty

Department of Computer Science

July 11th, 2013

U, Man Chon (Kevin)

manchonu@uga.edu

www.cs.uga.edu/~manchonu

Committee Members

- Khaled Rasheed, Ph.D. (Major Advisor)
  - Department: Computer Science
  - Email: khaled@cs.uga.edu
- Natarajan Kannan, Ph.D.
  - Department: Biochemistry & Molecular Biology
  - Email: kannan@bmb.uga.edu
- Roberto Perdisci, Ph.D.
  - Department: Computer Science
  - Email: perdisci@cs.uga.edu
- Hamid R. Arabnia, Ph.D.
  - Department: Computer Science
  - Email: hra@cs.uga.edu

Outline

- Motivation
- Introduction
- GOAL
- ROMP (**R**are **O**ncogenic **M**utation **P**redictor)
- VAMO (**V**alidity **A**nalysis of **M**alware-clustering **O**utputs)
- Conclusions & Future Work

Motivation

- We generate a huge amount of data in our daily lives
  - searching, online shopping, phone calls, accessing social media
- Researchers have been trying various methods to extract useful information from different sources of data
  - Machine Learning has become one of the most active research areas
- Clean and well structured data seldom exists
  - Missing label, no label, labeled with certain degrees of uncertainty
- No best machine learning approach for all different domains
  - Combining multiple learning outcomes and/or multiple learning approaches
- Introduce ROMP and VAMO
  - Combine multiple learning outcomes
  - Improve the learning outcomes by using clustering validity analysis to reduce label uncertainty

Introduction

- Present ROMP and VAMO in two different domains
  - ROMP – Cancer Research
  - VAMO – Malware Clustering
- Both domains containing some degrees of uncertainty
  - Confirmed cancer-causing mutations (correct labels) are limited, and mutations labeled as cancer-causing in some public databases (i.e. somatic mutations observed in clinical samples) are under certain assumptions
  - Labels of a same malware sample that generated by different anti-virus scanners are greatly different, and the mapping between these labels are usually unavailable
- Can be apply to many other domains
  - Similar data structure, similar problem to solve

Goal

- ROMP
  - To identify rare oncogenic mutations as well as mutations with suspicious labels
- VAMO
  - To provide a fully automated quantitative analysis of the validity of malware clustering results

ROMP – Rare Oncogenic Mutation Predictor

Introduction

- Given a labeled dataset
  - Supervised learning is always a good choose to analyze the data and to produce an inferred function for mapping new samples
- Given a dataset labeled with certain degrees of uncertainty
  - Using only supervised learning schemes might not be sufficient
- ROMP
  - Construct a high performance ensemble classifier using the labeled dataset
  - Use clustering algorithm and our self-invented cluster validity metrics to improve the learning outcomes

Possible Application

- ROMP can be used in any problem domain with labeled dataset
- It will be a great fit to domains that satisfy (or partially satisfy) the following criteria:
  - Labels in the dataset contain some degrees of uncertainty
  - Prefers to consider the outputs from multiple experts (learning algorithms)
  - Requires an probability score rather than just a class label as prediction

Our application

- We use ROMP for identifying causative mutations in human protein kinases, a class of signaling proteins known to be frequently mutated in human cancers

Background

- Cancer is a genetic disease which develops through a series of somatic mutations, a subset of which drive cancer progression
  - Not all mutations have equal influence on the disease state of a cell
  - “Driver” (Causative) and “Pessenger” (Non-Causative)
- Mutated driver genes are worthwhile targets for drug discovery
  - Counteracting the mutation's effects can potentially slow or reverse cancer progression in individual patients
- Machine Learning approaches have been used extensively to predict/prioritize causative mutations, but…
  - Used standard features of mutated residues, didn’t consider gene- or family-specific features
  - Used one or two popular machine learning algorithms, didn’t consider the bias
  - Provided binary classification results, seldom provide ranking of mutations
  - Unsupervised learning approaches have never been utilized to tackle this problem

Evaluation

- To evaluate our framework, we conducted experiments on benchmark datasets from UCI Machine Learning Repository. We selected 3 datasets:
  - Tic-Tac-Toe Dataset
  - Wisconsin Breast Cancer (Original) Dataset
  - Wisconsin Breast Cancer (Diagnostic) Dataset

Performance of ROMP

Dataset	TP Rate	FP Rate	Accuracy (%)	F-Measure	S-P	S-N
Tic-Tac-Toe	1	0	100	1	0	0
WBC-Original	0.9959	0..0209	98.33	0.9677	2	19
WBC-Diagnostic	0.9764	0.0112	98.42	0.9787	8	11

Tic-Tac-Toe Dataset

Reference	Method	Accuracy (%)	F-Measure
[82]	IB3-CI	99.1	n/a
[83]	CI3	98.4	n/a
[84]	Cluster Based Classification	99.2	n/a
[85]	ROMP	**100**	**1**

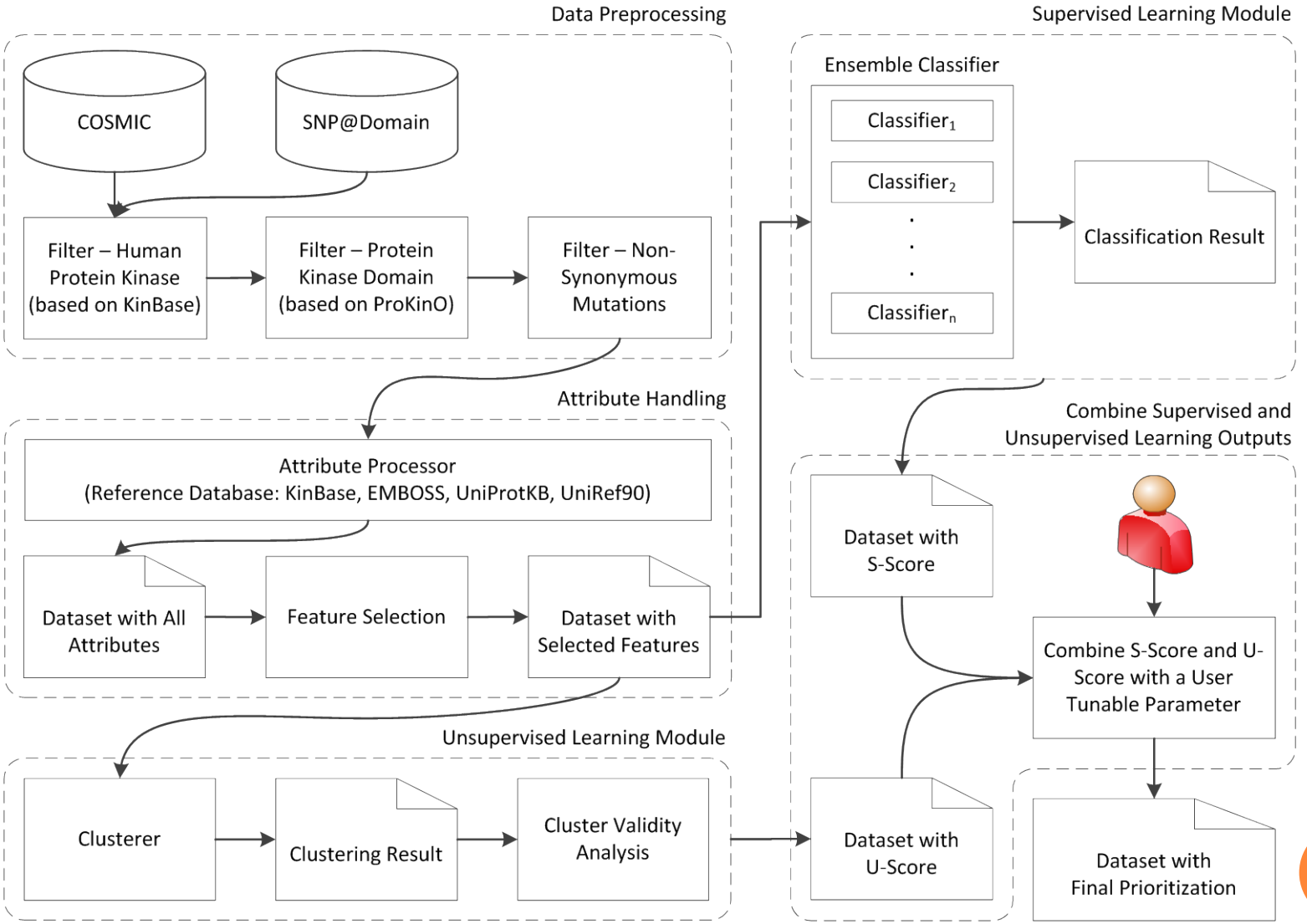
Wisconsin Breast Cancer (Original) Dataset

Reference	Method	Accuracy (%)	F-Measure
[86]	Fuzzy c-means clustering	91.42	0.8893
[85]	ROMP	**98.33**	**0.9677**

Wisconsin Breast Cancer (Diagnostic) Dataset

Reference	Method	Accuracy (%)	F-Measure
[87]	Fuzzy k-nearest neighbor	97.17	0.9534
[88]	Random Forest with feature selection	**99.82**	**0.9952**
[85]	ROMP	98.42	0.9787

System Overview



Data Sources

- Welcome Trust Sanger Institute's Cancer Genome Project (CGP) – version 50 and version 57
  - Catalogue of Somatic Mutations in Cancer (COSMIC)
  - Causative dataset
- SNP@Domain
  - http://variome.kobic.re.kr/SnpNavigator/
  - Non-causative dataset (non-synonymous)
- KinBase, EMBOSS, UniProtKB, and UniRef90
  - Extract Protein Features

Data Subset (Both v50 and v57)

- COSMIC-ALL
  - The positive set consist of all gene mutations (Kinase domain) from the COSMIC dataset. The non-causative set is the non-synonymous mutations obtained from SNP@Domain
- COSMIC-FG1 **(Main Focus – Training)**
  - The causative set of mutations in COSMIC that are observed in more than one distinct sample (Frequency Greater than 1). The non-causative set is the non-synonymous mutations obtained from SNP@Domain
- COSMIC-FE1 **(Main Focus – Prediction)**
  - The “unconfirmed”set of mutations in COSMIC that are observed only once. (Frequency Equal to 1)

Features

- Structural and Sequential features

- New: Multiple levels of evolutionary conservation
  - Among all protein kinases (superfamily)
  - Within the 7 recognized major groups
  - Within each family and subfamily

- 29 Attributes in total (Before Feature Selection)
  - 17 features (After Feature Selection)

Feature Selections

Selected Feature	Votes	Avg Rank
Protein Kinase Family	5	1.40
Protein Kinase Group	5	1.80
Amino Acid Type, WT	5	8.00
BLOSUME 62 pairwise score	5	8.20
Side-Chan polarity, Mutant	5	11.00
Conservation of wild type in all kinases	5	11.60
Conservation of consensus type in kinase group	5	11.60
Conservation of consensus type in all kinases	5	13.00
Conservation of consensus type in kinase family	4	5.75
Kinase subdomain	4	6.00
Average mass of amino acid, WT	4	7.50
Is a binding site?	4	8.25
Van der Waals volume, WT	4	8.75
Site modification type (if any)	4	9.25
Amino Acid Type, Mutant	4	10.75
Side-Chain polarity, WT	4	11.50
Is in protein kinase domain?	3	11.67

Selected Features

- 5 feature selection methods
  - OneR Algorithm, with a minimum bucket size of 14.
  - Relief-based selection , with 10 nearest neighbors for attribute estimation.
  - Chi-Square selection with ranker search
  - Filter approach, utilizes Gain Ratio Attribute Evaluator and Spread Subsample approach.
  - Correlation-based selection , with Greedy (forward) searching algorithm
- Use Majority Voting to select the features, then select the top 60% of features
- Finally we have 17 features

Classifiers

- We compare the performance of 11 established machine learning algorithms.
- Covers the major categories of classifier algorithms: Trees, Rules, Instance-based, Functions, and Bayes.
  - J48 (Tree)
  - Random Forest
  - NB Tree
  - Functional Tree
  - Decision Table
  - DTNB
  - LWL (J48+KNN)
  - Bayes Net
  - Naïve Bayes
  - SVM
  - Neural Network

Combining Multiple Classifiers

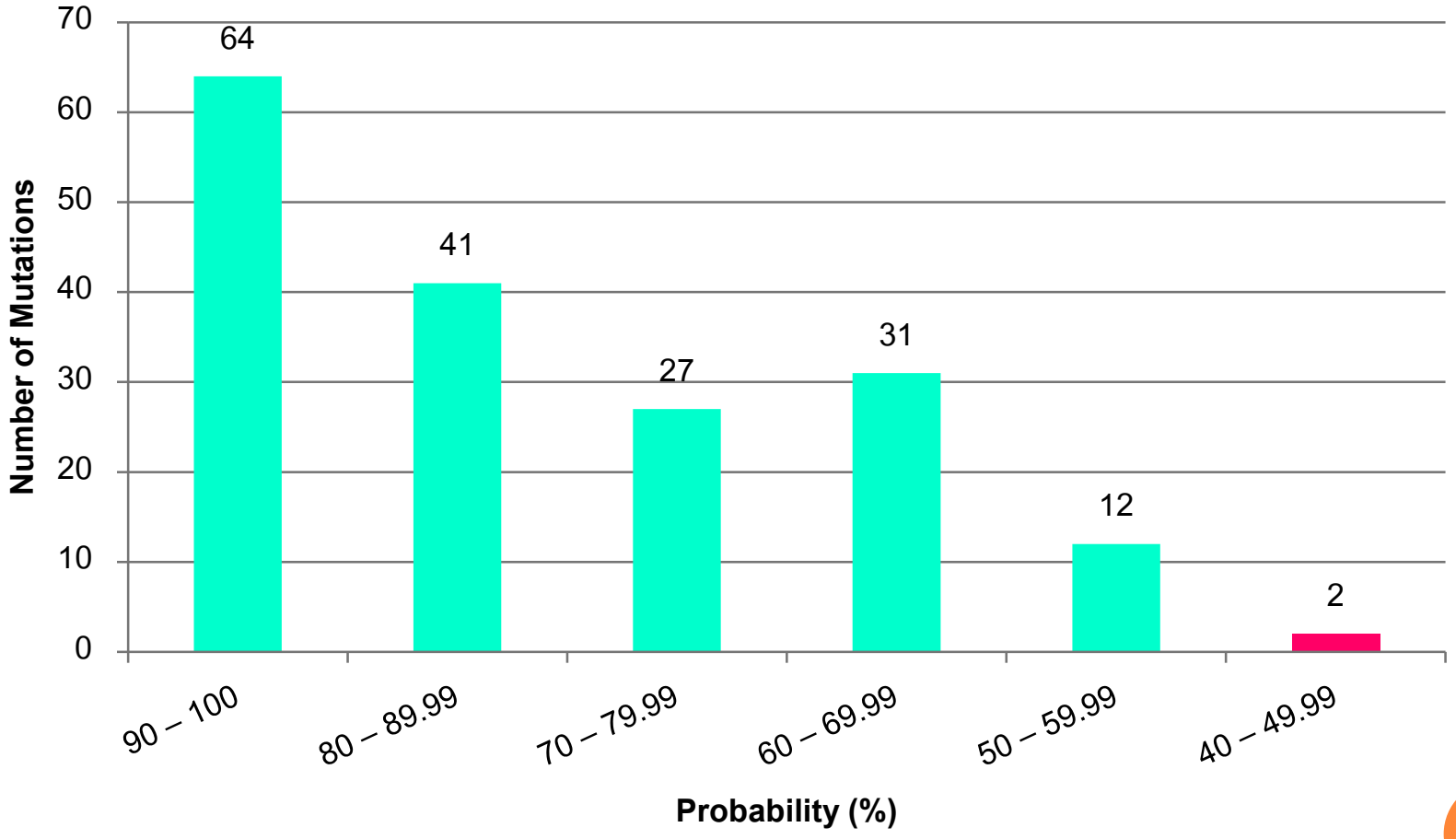
- Weighted Voting: $U - Score_i = \frac{\sum_i (Pr_i^{j,D} \times Accuracy^j)}{\sum_j Accuracy^j}$
- Stacking
- Grading

Performance of Ensemble Classifier Trained with Selected Features on COSMIC-FG1 v50 Dataset

Algorithms	TP Rate	FP Rate	Precision	Recall	F-Measure
Weighted Voting	**1**	**0.009**	**0.957**	**1**	**0.978**
Stacking	0.925	0.115	0.62	0.925	0.743
Grading	0.91	0.021	0.897	0.91	0.904

- COSMIC-FG1 v50: 67 causative mutations, 331 non-causative mutations
- Show the advantage of using the combined classifier by various measurement indexes
- F-Measure with 50% threshold, the combined classifier performs:
  - ~7.4% better (from 0.904 to 0.978) than the best single classifier (SVM) in COSMIC FG1 dataset

Application of Ensemble Classier to Predict Rare Variants in EGFR with COSMIC v50

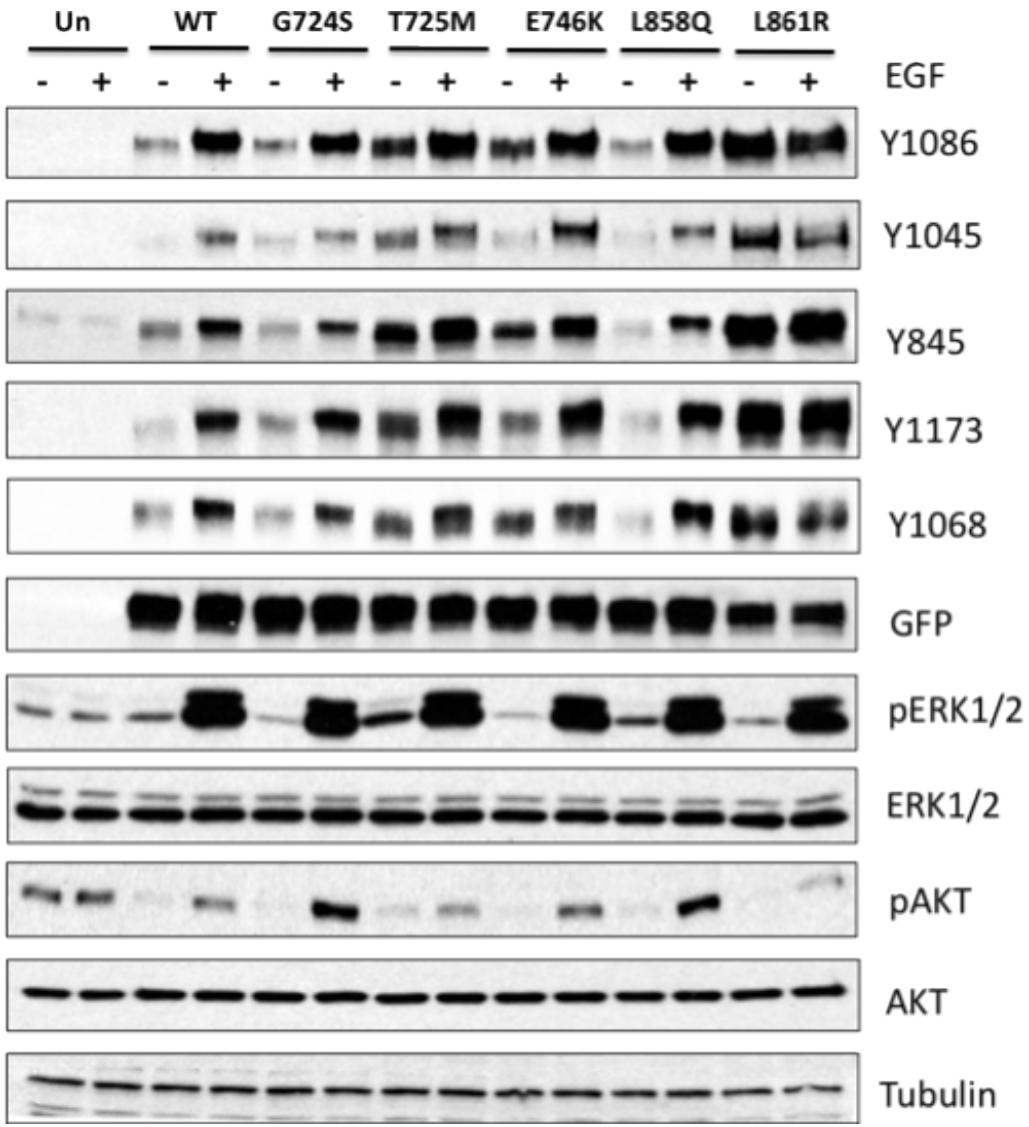


Select Mutations for In-Vitro Experiments

- Use the combined classifier to rank the EGFR mutations that appear only once in COSMIC v50

Mutation	Rank	U-Score	Experiment Result
L861R	1	0.97699	Activating
G724S	2	0.97649	--
T725M	21	0.96238	Activating
L858Q	25	0.95649	--
E746K	161	0.61788	Activating

Auto-Phosphorylation of Wild-Type and Mutant Type EGFR

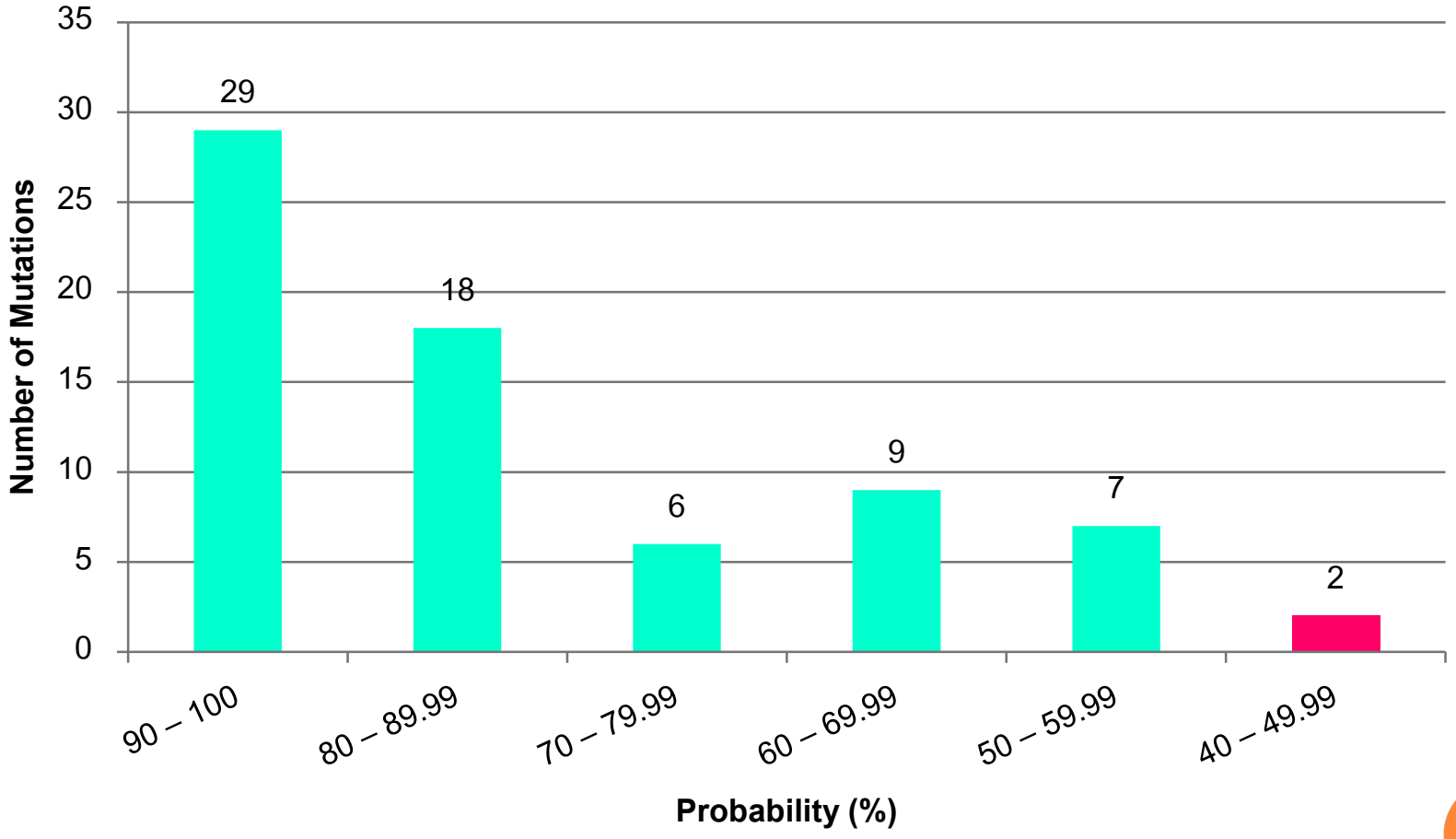


Comparison of COSMIC v50 and v57 : Justification of Using COSMIC-FG1 as Positive Set

- COSMIC-FG1 v57: 226 causative mutations, 331 non-causative mutations
  - COSMIC-FG1 v50: 67 causative mutations, 331 non-causative mutations
- 177 single-observation EGFR mutations in v50
- 165 single-observation EGFR mutations in v57
- 106 single-observation EGFR mutations shared between v50 and v57
- 71 EGFR mutations observed once in v50 but more than once in v57
- 59 EGFR mutations in v57 are new (not in v50)

- **Question:** How well can our previously trained model (with COSMIC v50 FG1 as positive set) predicted the 71 EGFR mutations that appear only once in COSMIC v50 but appear more than once in COSMIC v57

Application of Ensemble Classier to Predict Rare Variants in EGFR with COSMIC v50

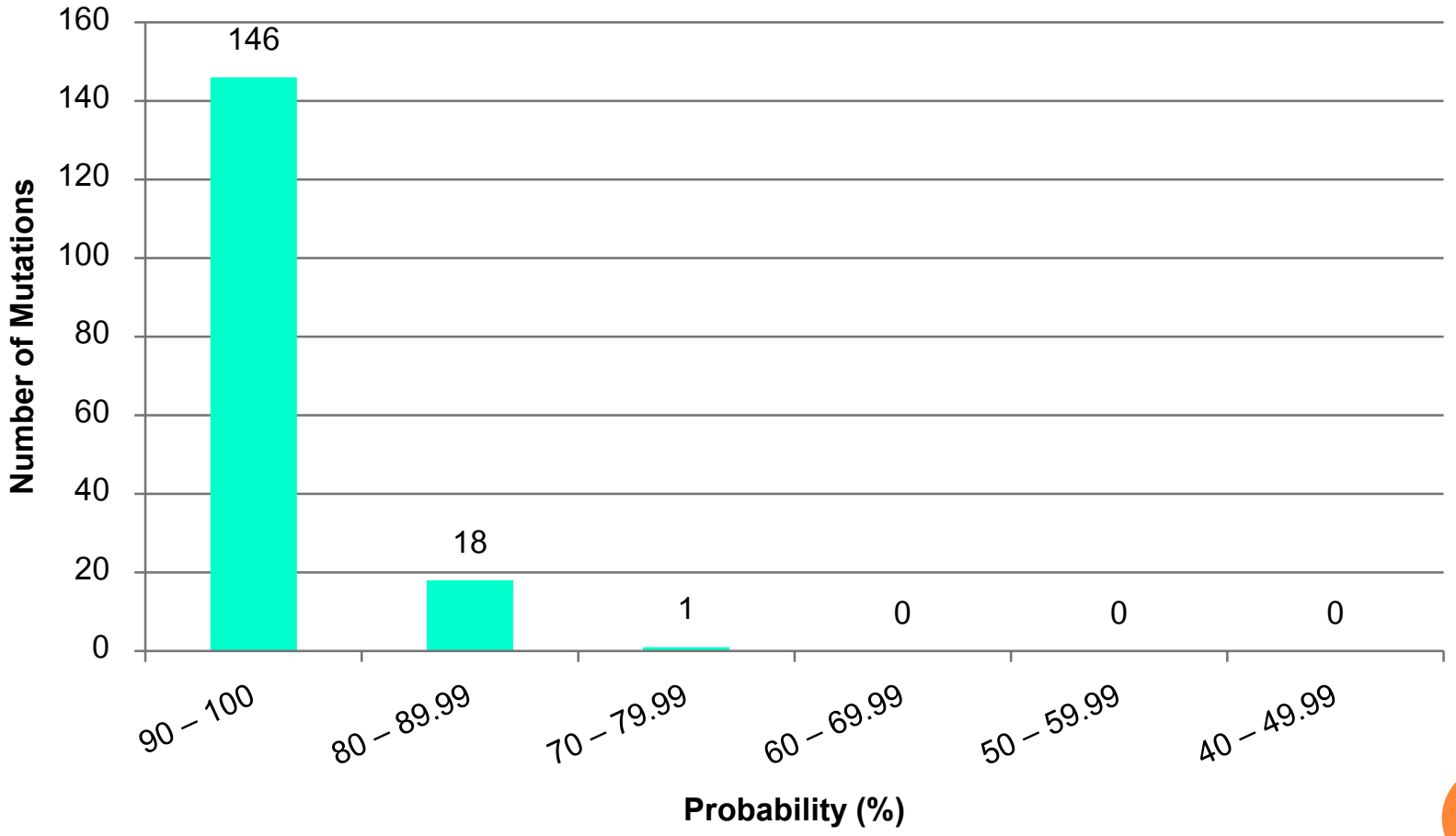


Performance of Ensemble Classifier Trained with Selected Features on COSMIC-FG1 v57 Dataset

Algorithms	TP Rate	FP Rate	Precision	Recall	F-Measure
Weighted Voting	**0.987**	**0.009**	**0.987**	**0.987**	**0.987**
Stacking	0.96	0.024	0.964	0.96	0.962
Grading	0.973	0.024	0.965	0.973	0.969

- F-Measure with 50% threshold, the combined classifier performs:
  - ~2% better (from 0.967 to 0.987) than the best single classifier (SVM) in COSMIC FG1 dataset.

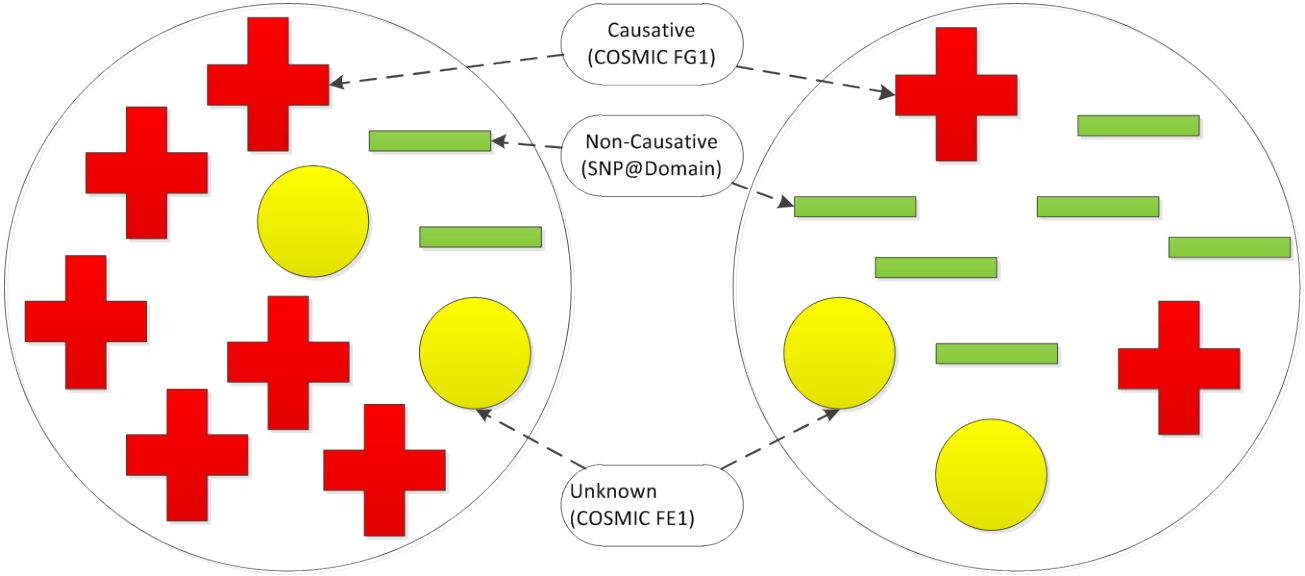
Application of Ensemble Classier to Predict Rare Variants in EGFR with COSMIC v57



Unsupervised Learning Module

- Expectation Maximization (EM) clustering algorithm
  - Finds clusters by determining a mixture of Gaussians that t a given dataset

- Our self-invented cluster validity metrics



Methodology

1. Filter the labeled (COSMIC-FG1 v57) and unlabeled (COSMIC-FE1) dataset with the selected features
2. Randomize the labeled dataset, followed by randomly splitting the labeled dataset into 90% and 10%
3. Combine the 90% split labeled data with the unlabeled dataset, apply EM algorithm onto this combined dataset to perform clustering
4. Use the trained EM model to cluster the split 10% labeled data
5. Integrate the unlabeled dataset, and the labeled dataset (both the 90% and 10%) together for validity analysis
6. For each instance in this combined dataset, we first find the cluster it belongs to, then we compute the metrics
7. Repeats step 1 to 6 for 10 times, then we generated the averaged result for every single instance in both the labeled and unlabeled dataset, namely “U-Score”

Measurement Metrics

- Class Ratio
- Intra Distance
- Raw Score
- Weighted U-Score
- Normalized U-Score

Measurement Metrics – Class Ratio

- Calculates the ratio of the causative instances and the ratio of the non-causative instances in the same cluster

$$R\_Pos_c^i = \frac{\sum_{j \in c_{pos}} j}{\sum_{j \in c_{pos}} j + \sum_{k \in c_{neg}} k}$$

$$R\_Neg_c^i = \frac{\sum_{k \in c_{neg}} k}{\sum_{j \in c_{pos}} j + \sum_{k \in c_{neg}} k}$$

Measurement Metrics – Intra Distance

- Calculates the Euclidean Distance of instances in a same cluster ***c***. ***distance(i, j)*** is the distance between instance ***i*** and instance ***j*** in cluster ***c***
- Distances between an instance ***i*** to other instances of a class (i.e. Positive or Negative) is infinity if there is no instance in ***c*** belongs to that particular class

$$D\_Pos\_Avg_c^i = \frac{\sum_{j \in c_{pos}} distance(i,j)}{\sum_{j \in c_{pos}} j}$$

$$D\_Pos\_Min_c^i = \arg\min_{j \in c_{pos}} \{distance(i,j)\}$$

$$D\_Neg\_Avg_c^i = \frac{\sum_{k \in c_{neg}} distance(i,k)}{\sum_{k \in c_{neg}} k}$$

$$D\_Neg\_Min_c^i = \arg\min_{k \in c_{neg}} \{distance(i,k)\}$$

$$D\_Pos\_Med_c^i = \arg\operatorname{med}_{j \in c_{pos}} \{distance(i,j)\}$$

$$D\_Neg\_Med_c^i = \arg\operatorname{med}_{k \in c_{neg}} \{distance(i,k)\}$$

Measurement Metrics – Raw Score

- Calculates the Purity Score of instance ***i*** in cluster ***c*** with the pre-computed distance parameters. Φ is an optional cost ratio that a user can set if the number of instances between different classes in the input dataset are highly imbalanced. ϵ is a very small value (10^{-32}) to avoid divided by zero error

$$Pos\_Score\_Avg_c^i = \sqrt{\Phi \times R\_Pos_c^i \times (1 - \frac{(D\_Pos\_Avg_c^i)^2}{(D\_Pos\_Avg_c^i)^2 + (D\_Neg\_Avg_c^i)^2 + \epsilon} + \frac{1}{\exp^{R\_Pos_c^i}})}$$

$$Neg\_Score\_Avg_c^i = \sqrt{\Phi \times R\_Neg_c^i \times (1 - \frac{(D\_Neg\_Avg_c^i)^2}{(D\_Pos\_Avg_c^i)^2 + (D\_Neg\_Avg_c^i)^2 + \epsilon} + \frac{1}{\exp^{R\_Neg_c^i}})}$$

Measurement Metrics – Weighted U-Score

- Calculates the level of oncogenicity of a instance ***i*** in a specific cluster ***c***, based on the average, median, and min pre-calculated Raw Scores. α=0.5, β=0.4, γ=0.1

$$Pos\_Score\_Weighted_c^i = \alpha \times Pos\_Score\_Min_c^i + \beta \times Pos\_Score\_Med_c^i + \gamma \times Pos\_Score\_Avg_c^i$$

$$Neg\_Score\_Weighted_c^i = \alpha \times Neg\_Score\_Min_c^i + \beta \times Neg\_Score\_Med_c^i + \gamma \times Neg\_Score\_Avg_c^i$$

Measurement Metrics – Normalized U-Score

- Calculates the level of oncogenicity of a instance ***i*** in a specific cluster c, with normalization

$$Pos\_Score\_Norm_c^i = \frac{Pos\_Score\_Weighted_c^i}{Pos\_Score\_Weighted_c^i + Neg\_Score\_Weighted_c^i}$$

$$Neg\_Score\_Norm_c^i = \frac{Neg\_Score\_Weighted_c^i}{Pos\_Score\_Weighted_c^i + Neg\_Score\_Weighted_c^i}$$

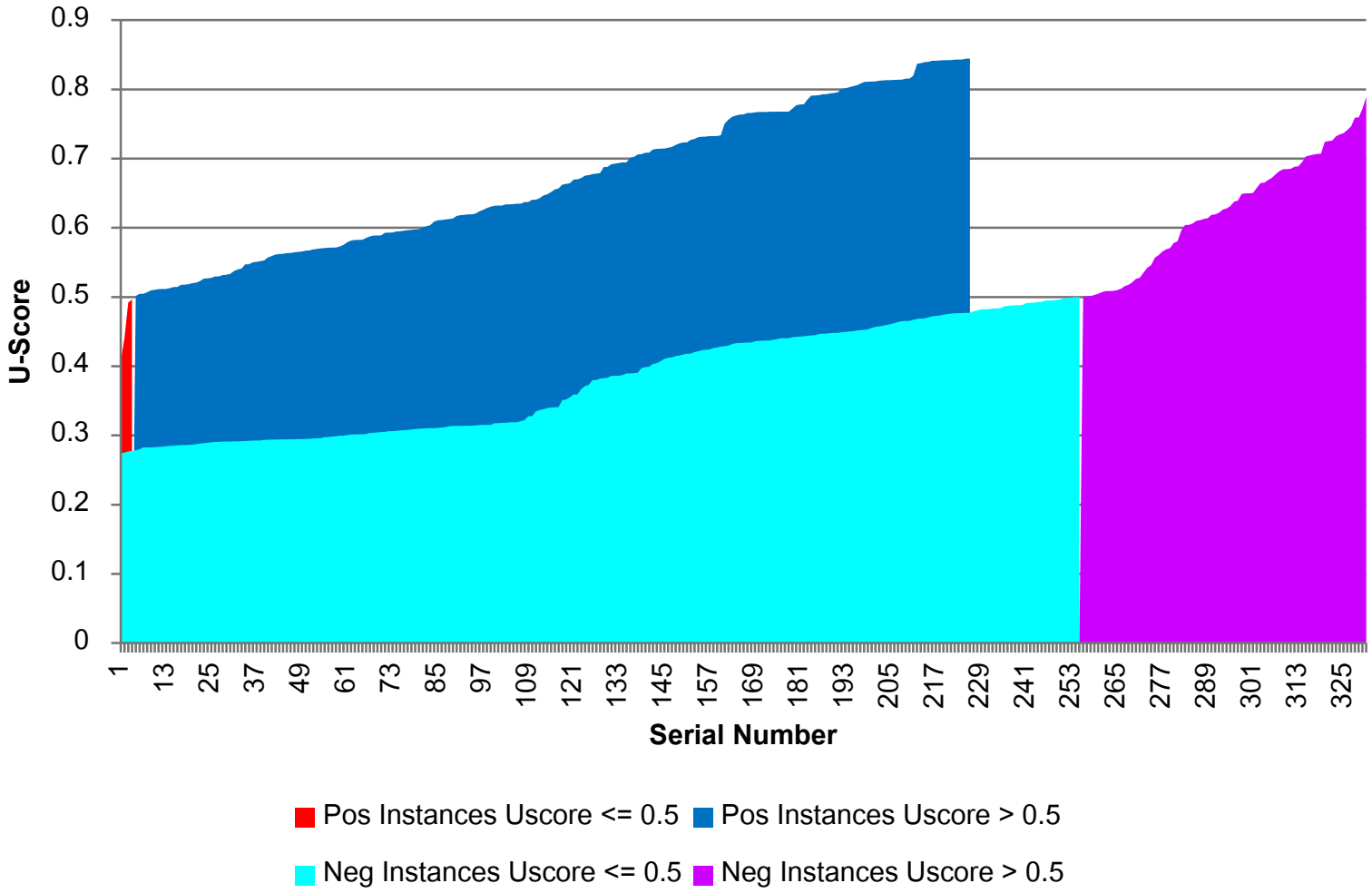
Evaluation of Module Using COSMIC-FG1 v57

- COSMIC-FG1 v57: 226 Causative, 331 Non-Causative

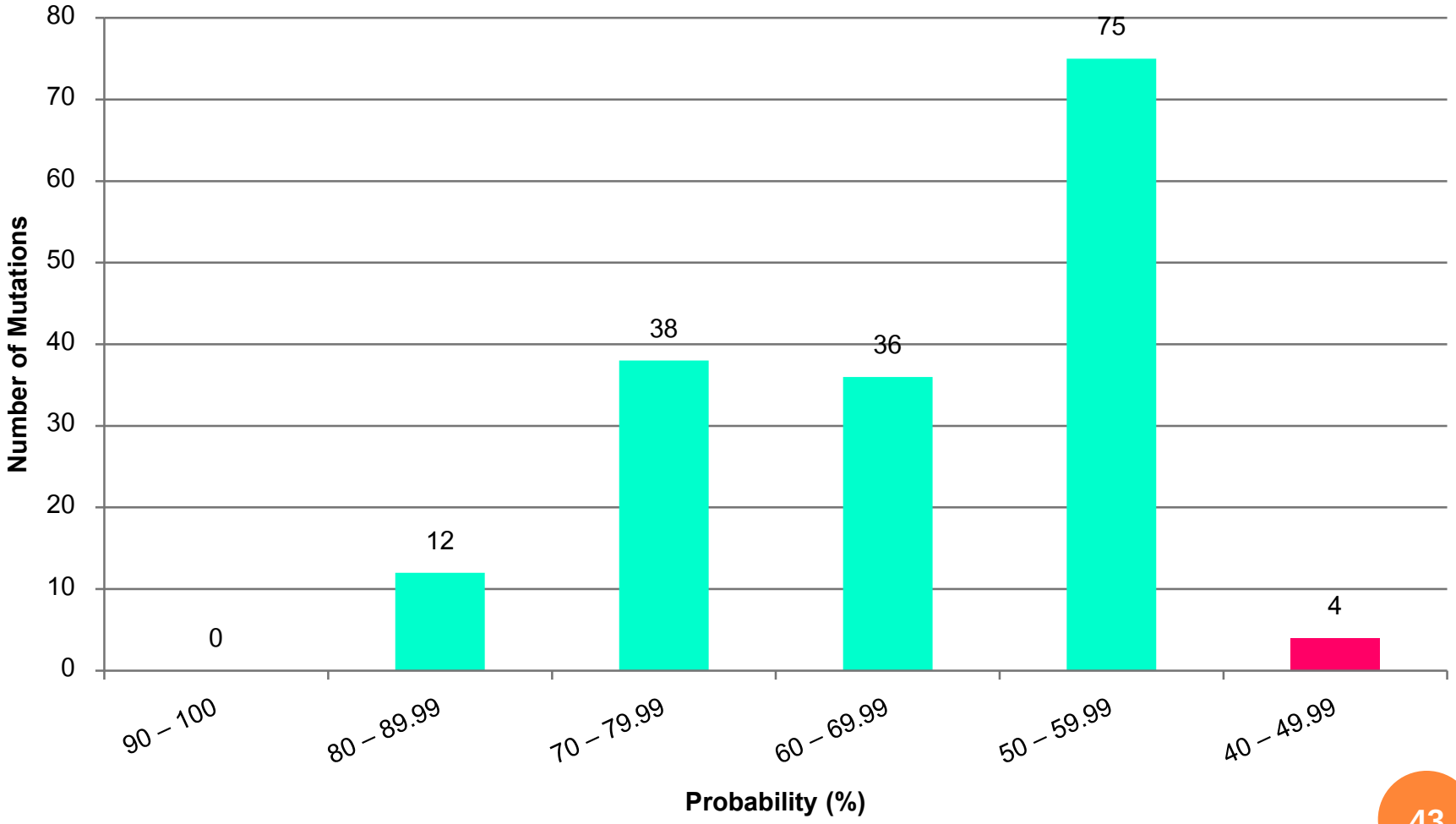
TP	FN	TN	FP
222	4	255	76

TP Rate	FP Rate	Recall	Precision	F-Measure	Accuracy
0.98230	0.22961	0.98230	0.74497	0.84733	0.85637

Evaluation of Module Using COSMIC-FG1 v57



Application of Unsupervised Learning Module to Predict Rare Variants in EGFR with COSMIC v57



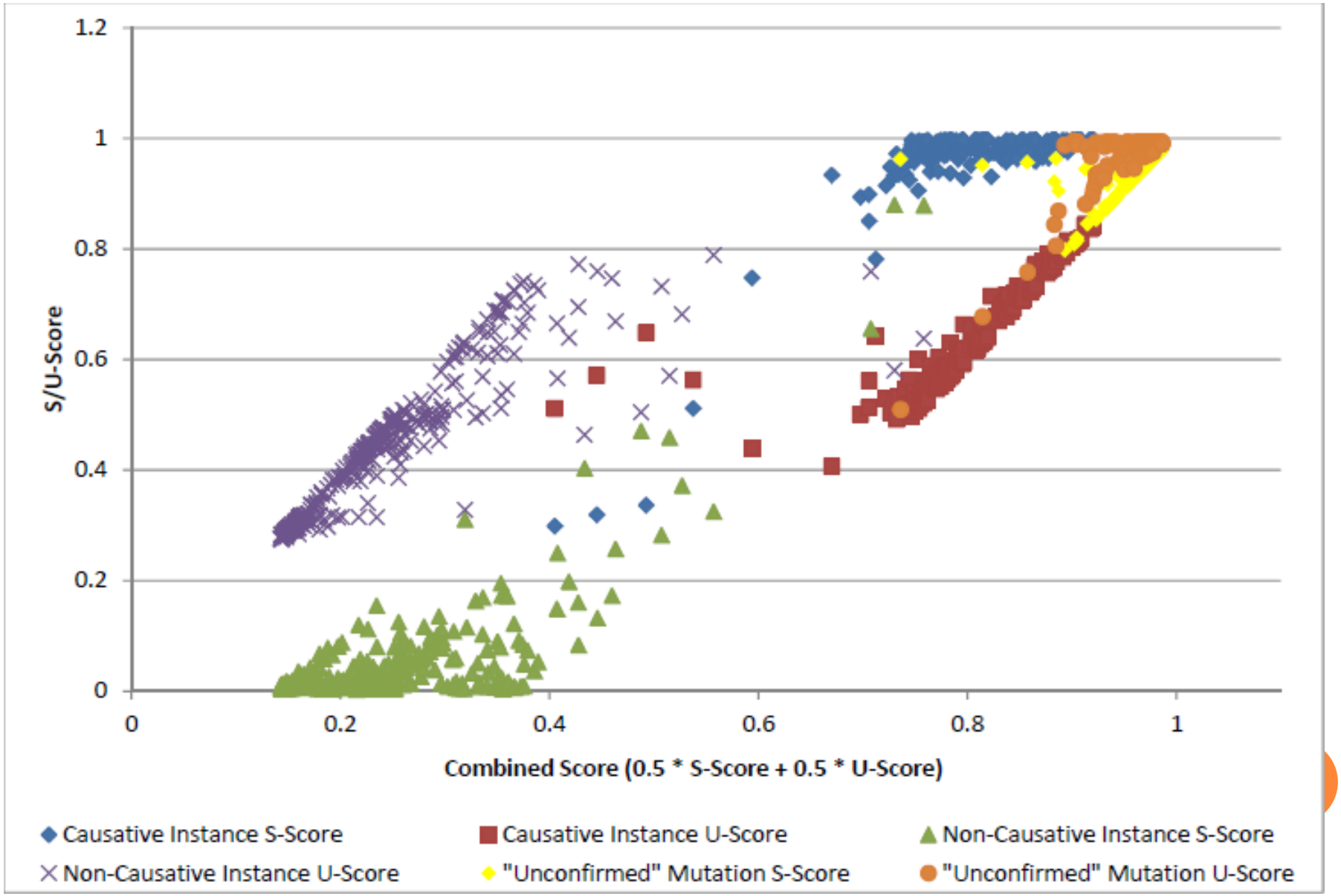
Combining Supervised and Unsupervised Learning Outputs to Predict Rare Variants in EGFR and to Identify Suspicious Mutations with COSMIC v57

	1S/9U	2S/8U	3S/7U	4S/6U	5S/5U	6S/4U	7S/3U	8S/2U	9S/1U
TP	223	225	224	224	223	223	223	223	223
FN	3	1	2	2	3	3	3	3	3
TN	272	284	309	321	324	326	328	328	328
FP	59	47	22	10	7	5	3	3	3

	1S/9U	2S/8U	3S/7U	4S/6U	5S/5U	6S/4U	7S/3U	8S/2U	9S/1U
TP Rate	0.9867	0.9956	0.9912	0.9912	0.9867	0.9867	0.9867	0.9867	0.9867
FN Rate	0.1782	0.142	0.0665	0.0302	0.0211	0.0151	0.0091	0.0091	0.0091
Recall	0.9867	0.9956	0.9912	0.9912	0.9867	0.9867	0.9867	0.9867	0.9867
Precision	0.7908	0.8272	0.9106	0.9573	0.9696	0.9781	0.9867	0.9867	0.9867
FMeasure	0.878	0.9036	0.9492	0.9739	0.9781	0.9824	0.9867	0.9867	0.9867

- TP Rates at least 98.67% and FP rates at most 17.82%
- The “5S/5U” split reaches 98.67% TP rate and 2.11% FP rate

Visualization of the Combined Score of mutations in COSMIC-FG1 v57 and COSMIC-FE1 v57 dataset.

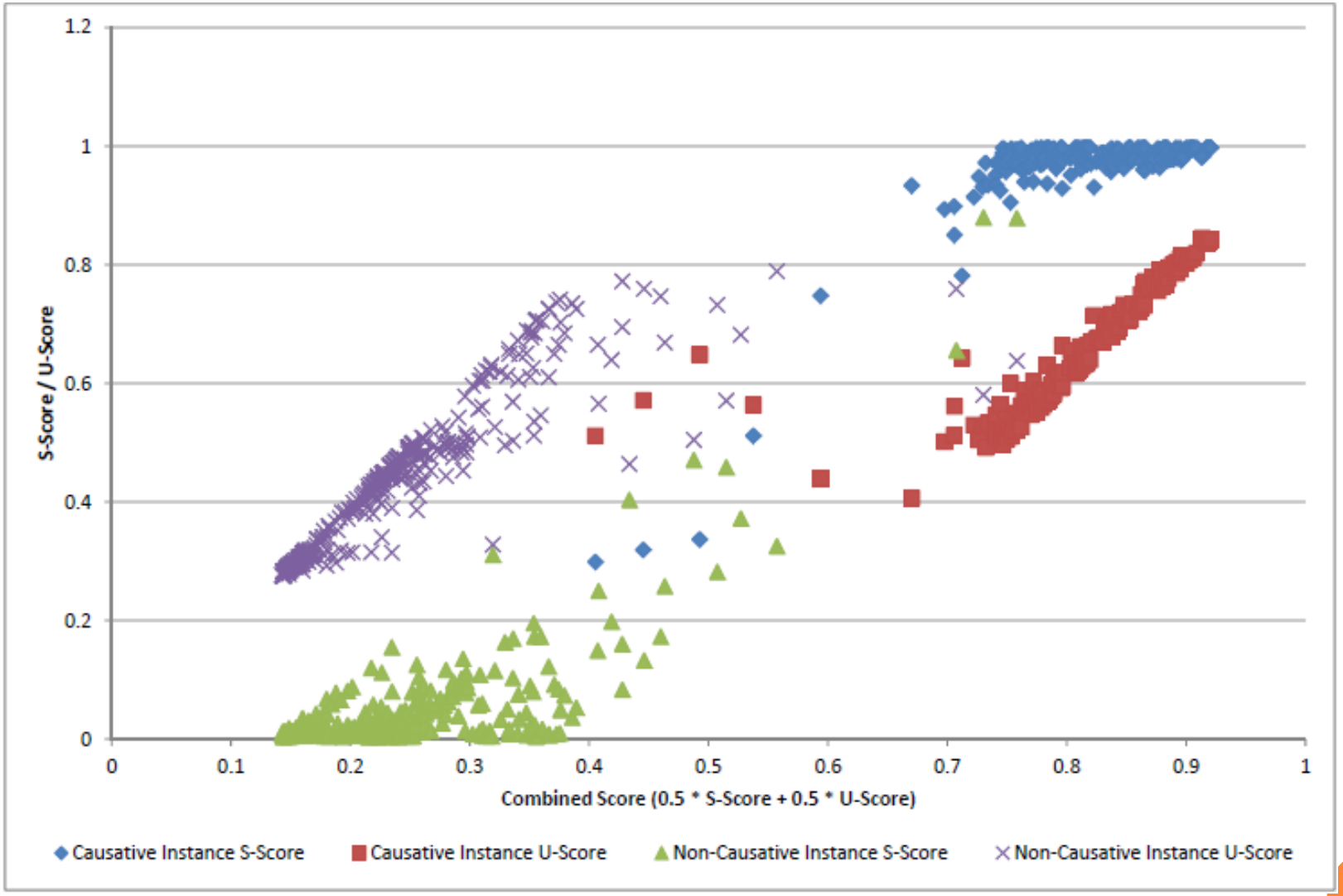


Application of ROMP to Identify Suspicious Mutations in COSMIC-FG1 v57

	Causative Labeled Instances		Non-Causative Labeled Instances	
	U-Score > 0.5	U-Score <= 0.5	U-Score > 0.5	U-Score <= 0.5
U-Score > 0.5	**E**	S	S	S
U-Score <= 0.5	S	S	S	**E**

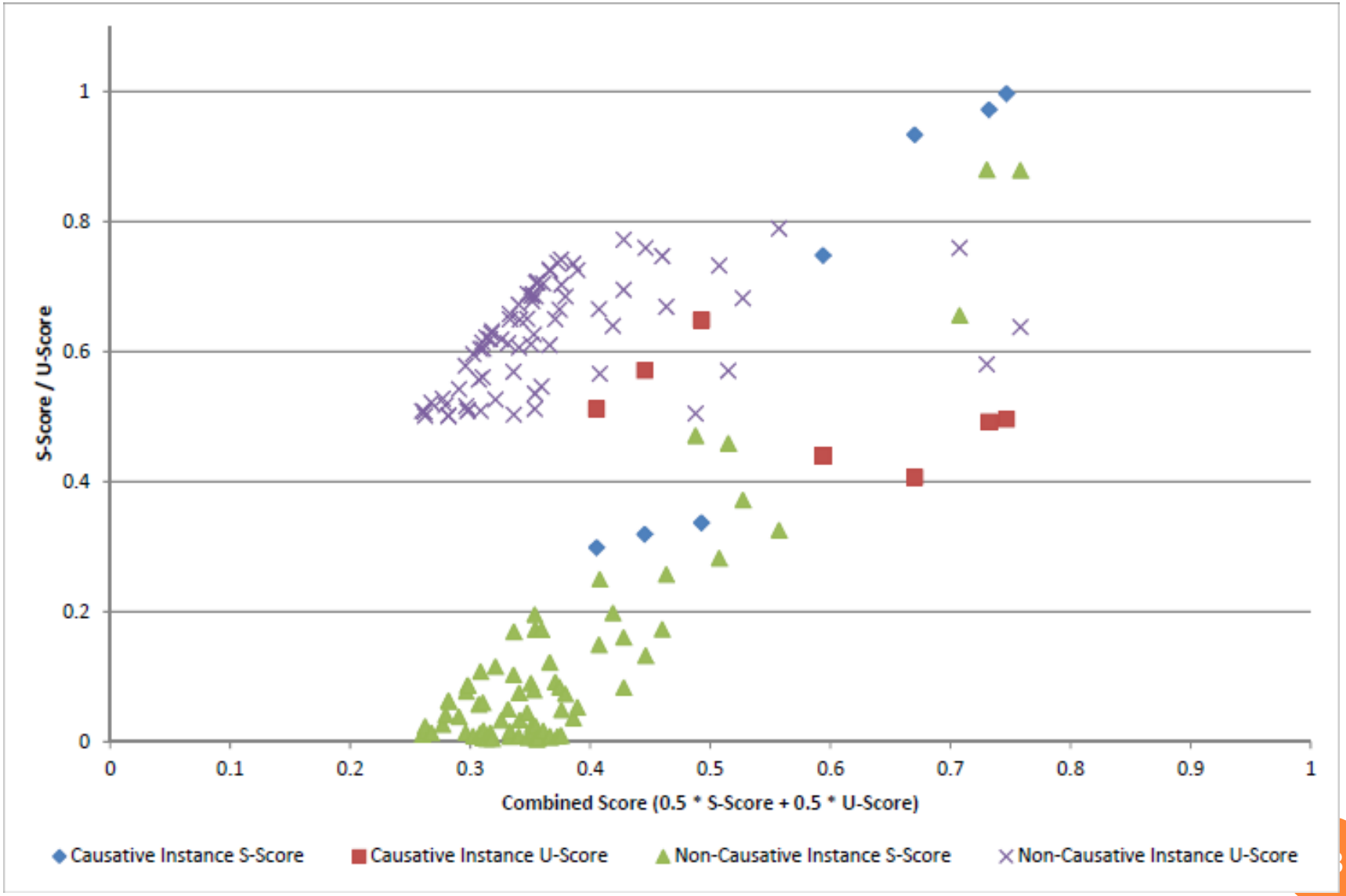
- E = Expected, S = Suspicious

Application of ROMP to Identify Suspicious Mutations in COSMIC-FG1 v57



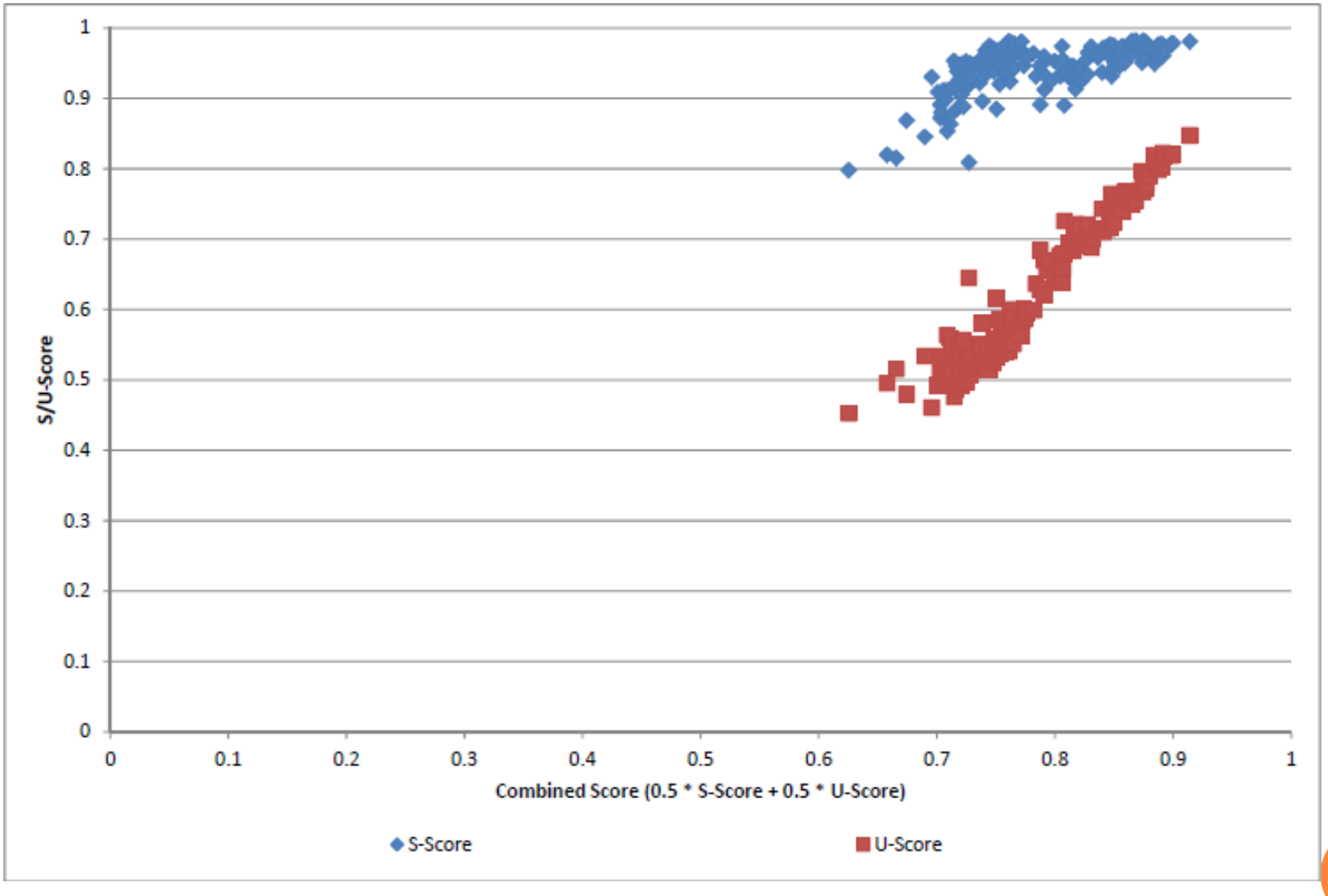
- 219 out of 226 instances (≈ 97%) that labeled as causative fall into the ``Expected'' category, and 255 out of 331 instances (≈ 77%) that labeled as non-causative fall into the ``Expected'' category

Details of the Suspicious Mutations in COSMIC-FG1 v57



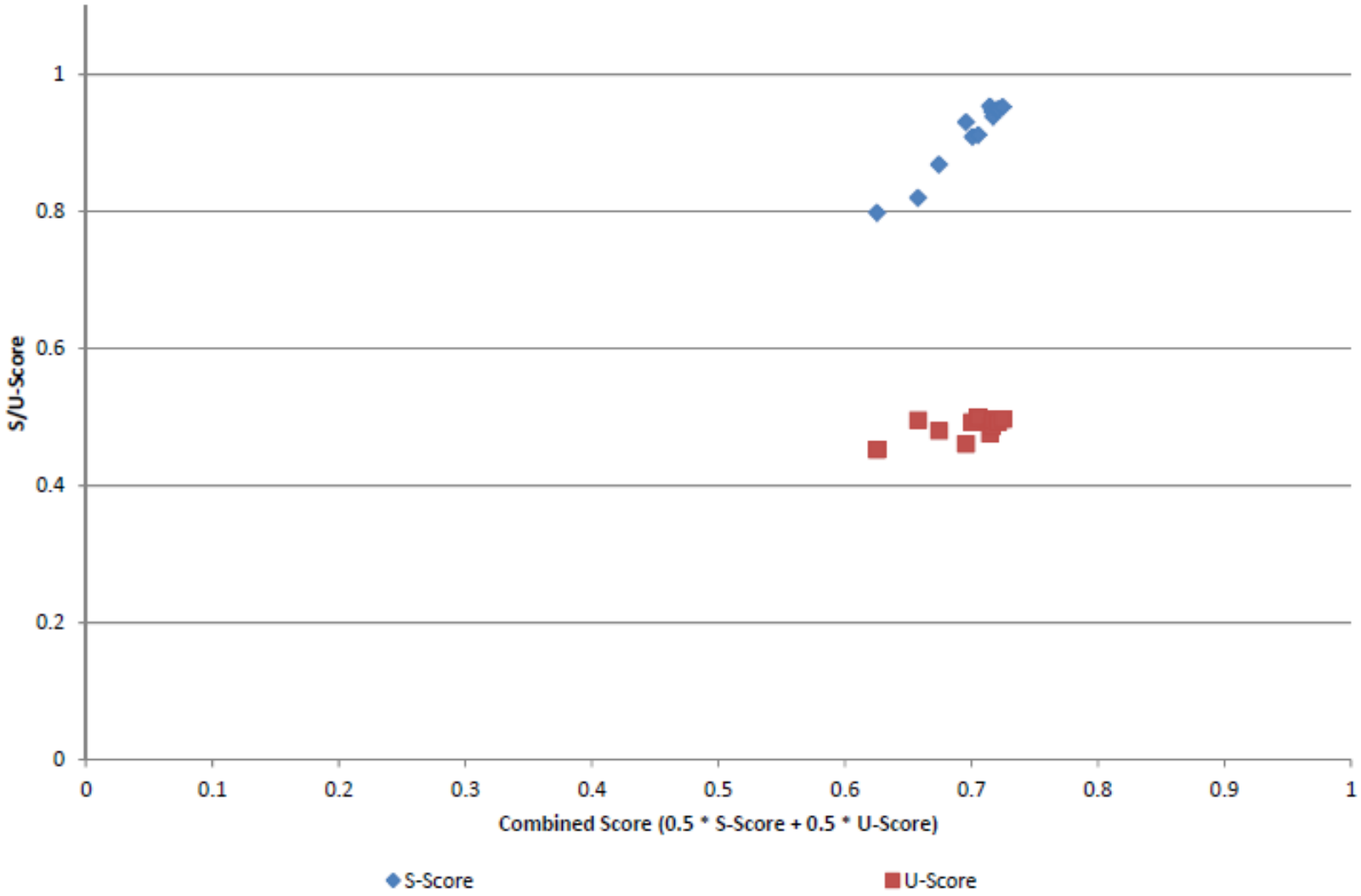
- 7 with causative label and 76 with non-causative label. (List on page 101)

Application of ROMP to Predict Rare Variants in EGFR with COSMIC-FE1 v57



- 1 mutations ranked between 90% and 100%, 67 ranked 80% to 89.99%, 91 ranked 70% to 79.99%, 6 ranked 60% to 69.99%

Application of ROMP to Identify Suspicious Rare Variantsin EGFR with COSMIC-FE1 v57



- 12 out of 165 (7.2%) suspicious instances

Contributions

- We introduce new kinase-specific features, beyond those used in previous methods to improve prediction accuracy
- We combine supervised and unsupervised learning to provide a reliable schema for the prioritization of a set of "unconfirmed" mutations
  - The supervised component combines multiple supervised learning algorithms (individual classifiers), overcomes the biases introduced by each method
  - The unsupervised component strengthens the confidence of our prioritization
- We use our machine learning schema to produce a numerical ranking of causative mutations in EGFR and test the impact of predicted mutations on EGFR kinase activity using cell-based assays
- Our studies identify T725M as a rare causative mutation inasmuch as the T725M mutation increases EGFR autophosphorylation and displays catalytic activity in the absence of the activating EGF ligand

Discussion

- In ROMP
  - No missing labels in the COSMIC dataset
  - Amount of different labels in the dataset are easy to manage
    - causative and non-causative
  - Labels that generated by different experts (individual learning algorithms) are commonly understood by each other
  - Have good understanding of how each expert performs on the given dataset
    - 10-fold cross-validation accuracy
- However, in some situations…
  - Experts do not have enough knowledge to label all instances
  - Labels that generated by different experts are partially understood (or even can't be understood) by each other
    - M=W32/Viut.gen
    - A=WORM/Korgo.U
    - T=PE_VIRUT.D-4
  - No prior information available about how accurate each expert performs

VAMO – Validity Analysis of Malware-clustering Outputs

- Generates reliable outputs by automatically reducing the label uncertainty
- Can be used by other experts to assess the quality of their learning results

Introduction

- Given a labeled or unlabeled dataset
  - Experts (automated algorithms, humans, etc.) analysis the data and generalize the dataset based on some rules
- However…
  - Experts might not have enough knowledge to label all instances
  - Labels that generated by different experts are partially understood (or even can't be understood) by each other
  - No prior information available about how accurate each expert performs

Introduction (Cont'd)

- VAMO
  - Build the Label Graph
    - Automatically learn the mapping between different labels assigned by multiple experts to the same object, thus avoiding the need to manually build or adjust such mappings
    - Identify cases in which one (or more) expert(s) tend to inconsistently use several labels to label samples that belong to the same group according to other experts
    - Learn the level of similarity between labels assigned by different experts, by looking at the number of times that certain labels are jointly assigned to the same samples.
  - Apply Average-Linkage Hierarchical Clustering Algorithm
    - Generate a dendrogram that expresses the "relationship" between the samples according to their labels
  - Generate robust reference clustering

Possible Application

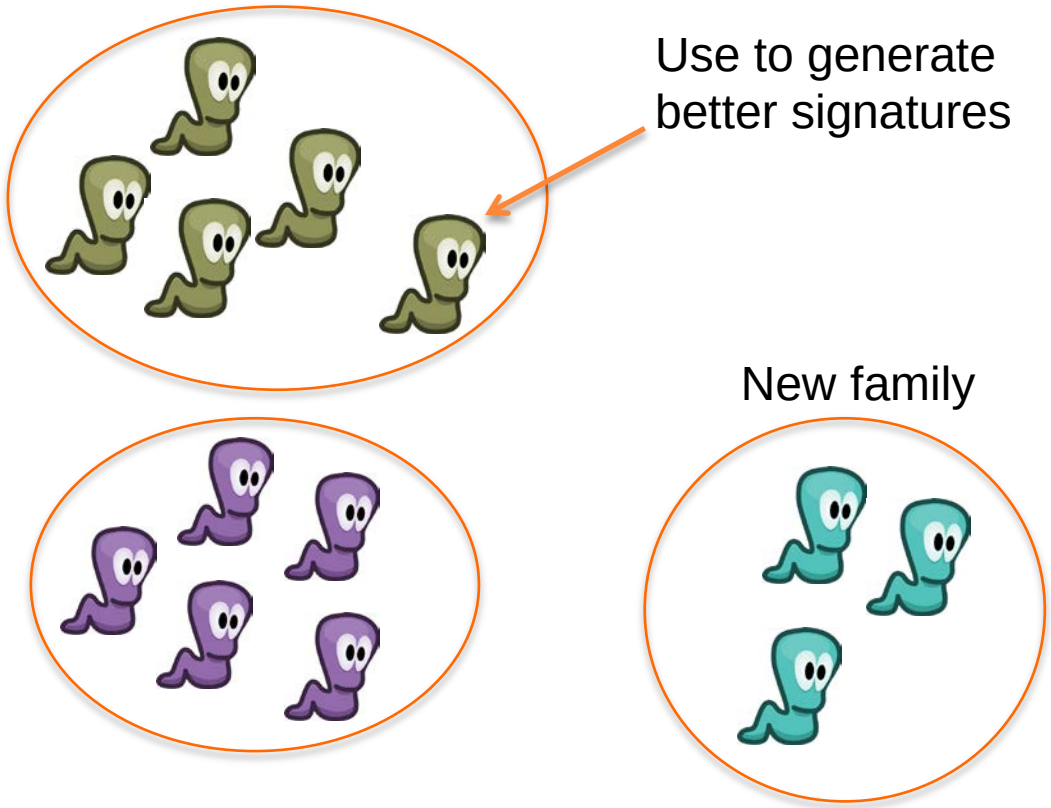
- VAMO can be used in any problem domain with labeled or unlabeled dataset
  - Missing label is allowed
- It will be a great fit to domains that satisfy (or partially satisfy) the following criteria:
  - Labels (outputs of an expert) in the dataset contain some degrees of uncertainty
  - Prefers to consider the outputs from multiple experts (i.e. AV Scanners)
  - Requires a fully automated quantitative analysis of the validity of clustering results

Our application

- We use VAMO for clustering malware samples according to the labels that assigned to these samples by multiple anti-virus scanners.

Malware Clustering

- Clustering malware into *families* is useful



Malware Triage

Malware Clustering Research

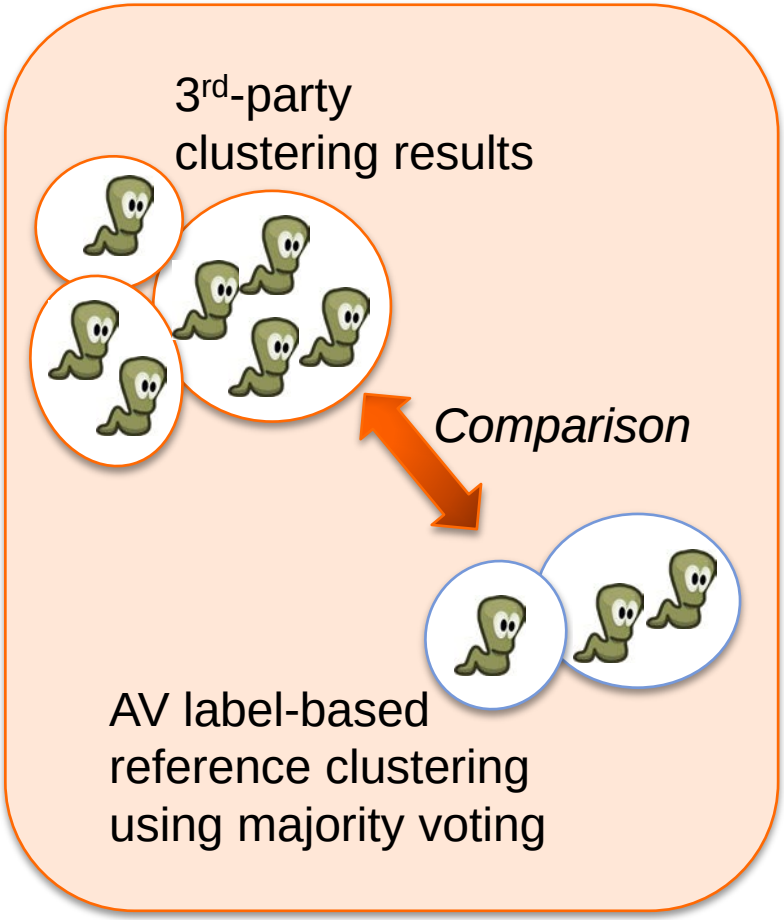
- Bailey et al. *Automated classification and analysis of internet malware* (RAID'07)
- Bayer et al. *Scalable, behavior-based malware clustering* (NDSS'09)
- Hu et al. *Large-scale malware indexing using function-call graphs* (CCS'09)
- Perdisci et al. *Behavioral clustering of http-based malware and signature generation using malicious network traces* (NSDI'10)
- Jang et al. *Bitshred: feature hashing malware for scalable triage and semantic analysis* (CCS'11)

Validating Clustering Results

- How do we know clustering output is good?
  - Need a **reference clustering** to compare
  - Challenge: **unsupervised learning**
  - Limited or no *ground truth*

- Reference clustering (previous work)
  - Use multiple AV scanners
  - Extract family names from AV labels
  - Samples that are assigned the same label by **majority of AVs** are considered in same family

Drawbacks of Majority Voting



- Different AV vendors use different notation
  - Different family names
  - One-to-many mapping
  - Missing and inconsistent labels
- Difficult to find majority consensus
  - Samples with no consensus are excluded

MD5	McAfee	Avira	TrendMicro
ec34ca31			
c2276216			
089ae4f5	M=W32/Virut.gen	A=W32/Virut.AX	T=PE_VIRUT.D-1
8ba552c9	M=W32/Virut.gen	A=TR/Drop.VB.DU.1	T=PE_VIRUT.XO-1
8cb0ab6c	M=W32/Virut.gen	A=WORM/Korgo.U	T=PE_VIRUT.D-4
b0b75f70	M=W32/Virut.gen	A=W32/Virut.X	T=PE_VIRUT.XO-1
a306b4e7	M=W32/Virut.gen	A=W32/Virut.Gen	T=PE_VIRUT.D-1
337a2cf4	M=W32/Virut.gen	A=W32/Virut.Gen	T=PE_VIRUT.D-1
62d18c7e	M=W32/Virut.gen	A=W32/Virut.Gen	T=PE_VIRUT.D-1
8dbca633	M=W32/Virut.gen	A=TR/Drop.VB.DU.1	T=PE_VIRUT.XO-1
ac433383	M=W32/Virut.n	A=W32/Virut.Gen	T=PE_VIRUX.A-3
cae61d9e	M=W32/Virut.gen	A=W32/Virut.X	T=PE_VIRUT.XO-2
7cc795f1	M=W32/Virut.gen	A=W32/Virut.Gen	T=PE_VIRUT.D-1
8de5214b	M=W32/Virut.gen.a	A=W32/Virut.AM	T=PE_VIRUT.XY
4d26cb0a	M=W32/Virut.gen	A=W32/Virut.Gen	T=PE_VIRUT.D-1
9fb75631	M=W32/Virut.n	A=W32/Virut.Gen	T=PE_VIRUX.A-3
229004b9	M=W32/Virut.gen	A=W32/Virut.X	T=PE_VIRUT.XO-1
28a85d8a		A=TR/Drop.VB.DU.1	T=PE_VIRUT.XO-1
663c5f6c		A=W32/Virut.Z	
de6f1e00	M=	A=W32/Virut.Gen	
1ff43bca	M=	A=W32/Virut.X	T=PE_VIRUT.XO-4
ea580f6d	M=W32/Virut.n	A=W32/Virut.Gen	T=PE_VIRUX.A-3
a844eeff	M=W32/Virut.gen	A=TR/Drop.VB.DU.1	T=PE_VIRUT.XO-1
4f8613fd	M=W32/Virut.gen	A=TR/Drop.VB.DU.1	T=PE_VIRUT.XO-1

One-to-many mapping of family names

Missing Labels

Inconsistent Labels

Drawbacks of Majority Voting

Majority consensus found only for a fraction of dataset!

Bayer et al. (NDSS'09)

$$\frac{2{,}658}{14{,}212} = 18.7\ \%$$

Our malware dataset

5.6% of 1.1M samples

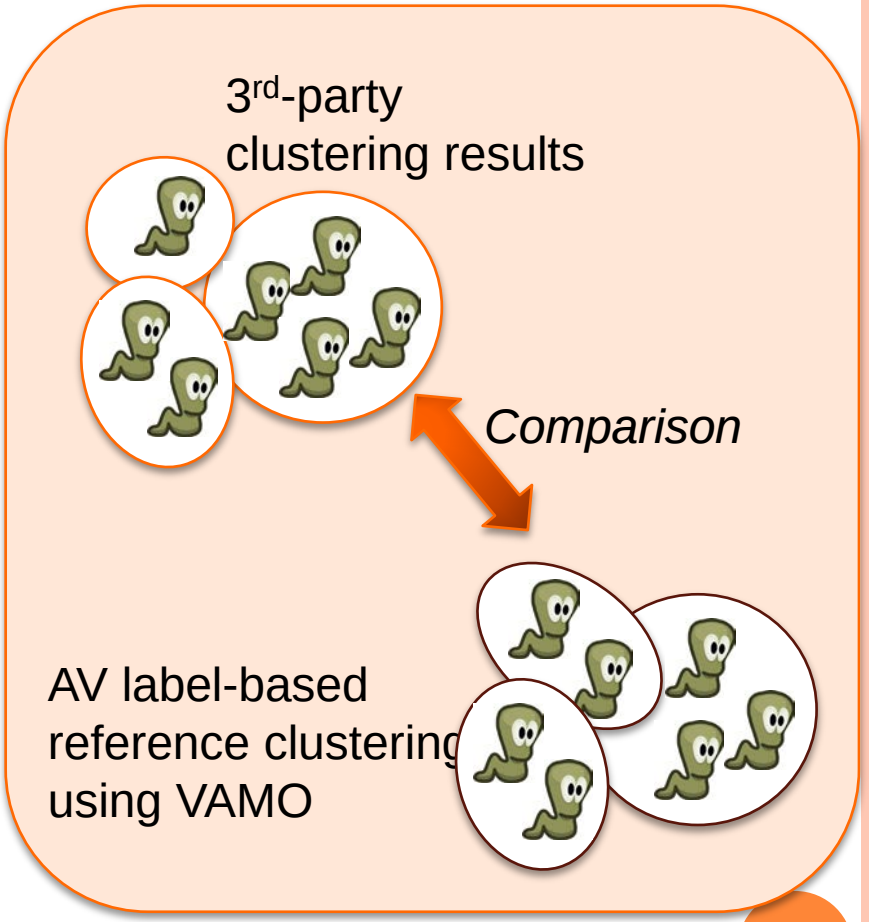
Reference clustering built using majority voting not representative of dataset

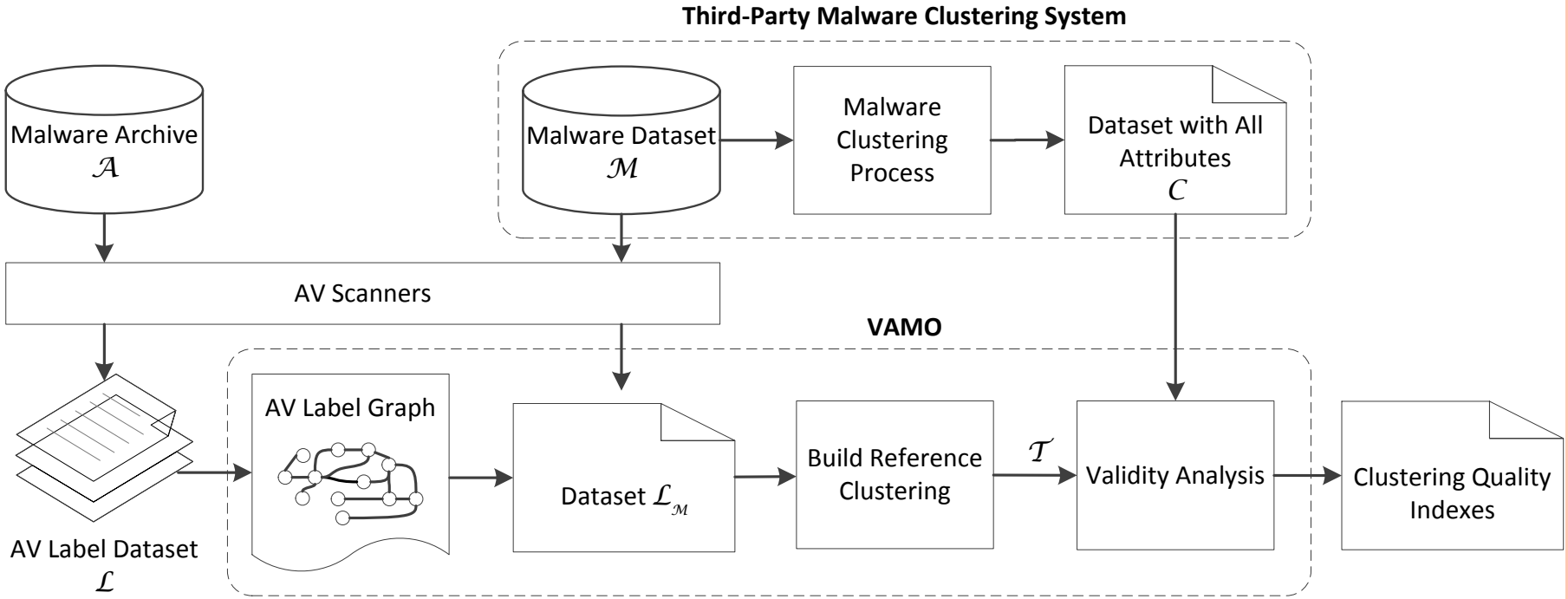
Li et al. *On challenges in evaluating malware clustering* (RAID'10)

"existing approaches to obtaining ground-truth data for malware clustering evaluation biases results by isolating those instances that are simple to cluster"

VAMO – *Validity Analysis of Malware-clustering Outputs*

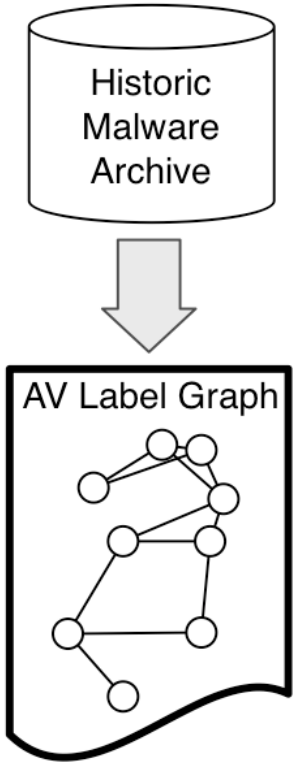
- Research Goals
  - Consider entire malware dataset for validation
  - No manual mappings between AV labels
  - Deal with AV naming inconsistencies
  - Fully automated





- Enables tuning clustering parameters
- Allows comparison of different systems

AV Label Graph

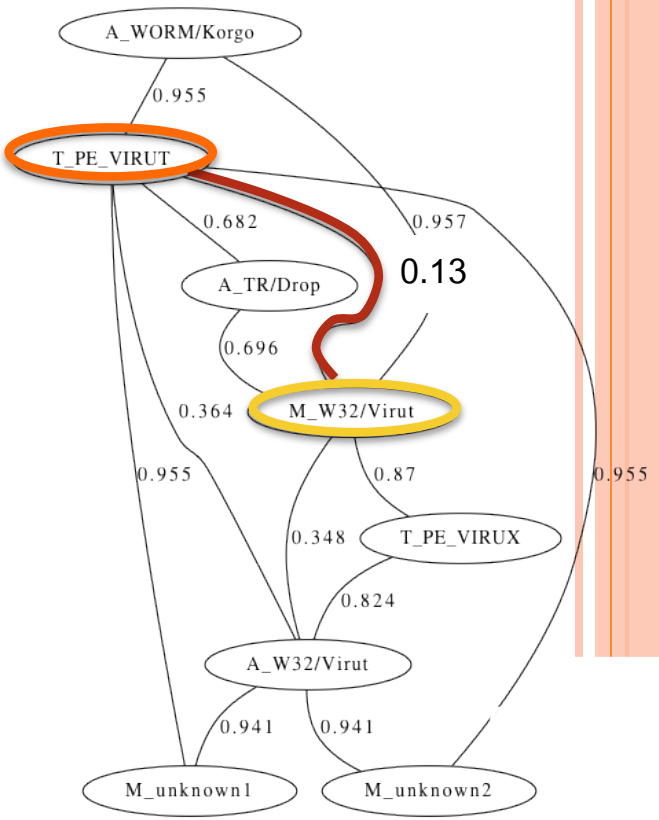


- Learns mappings between AV labels
- Labels that often appear “together” are considered *similar*

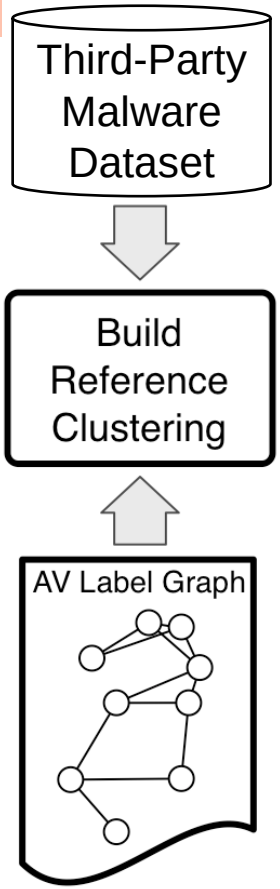
Node = <AV>_<Family>

Edge Weight = Label distance

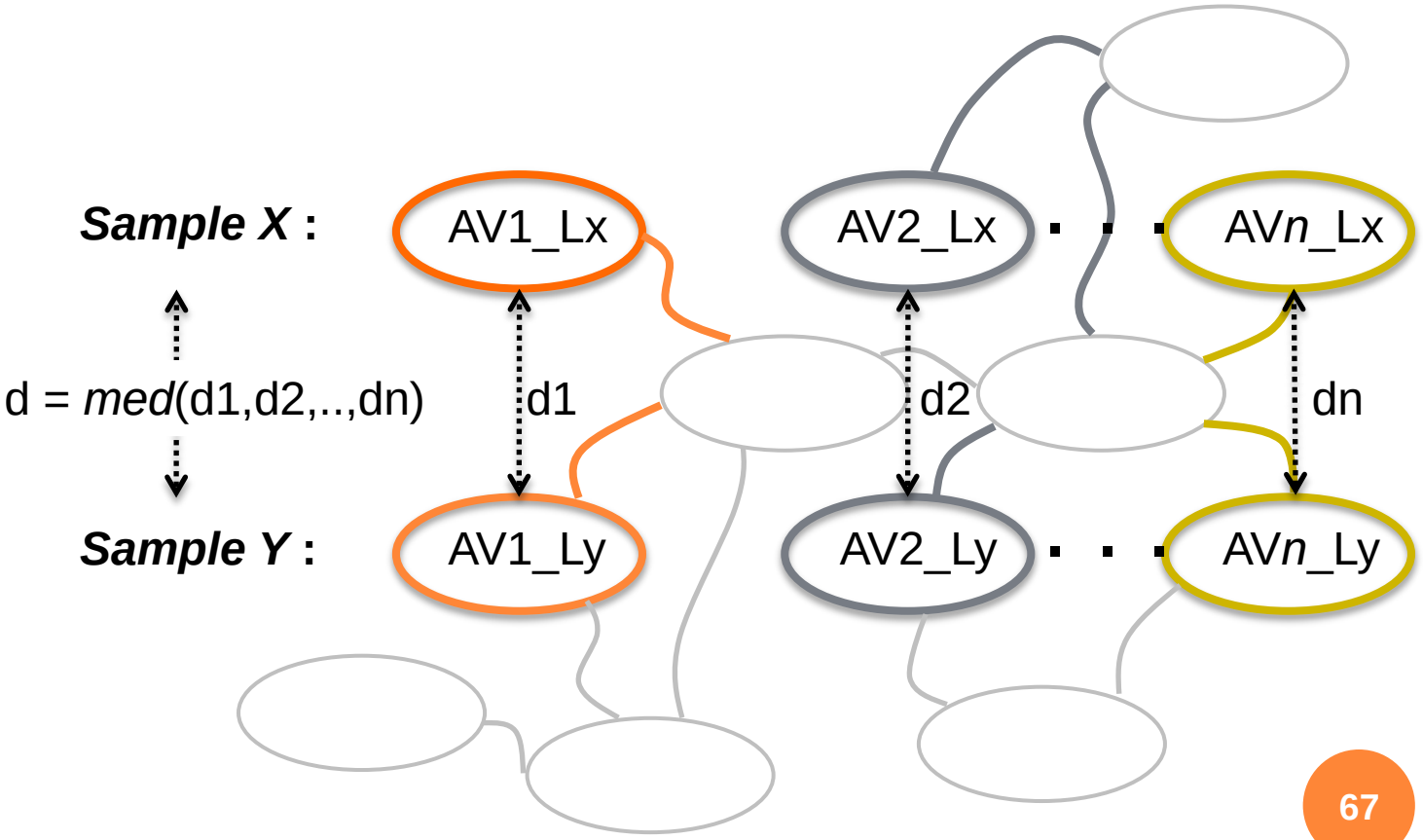
	McAfee	*Avira*	*Trend*
Sample X :	W32/Virut	TR/Drop	PE_VIRUT



Building Reference Clustering

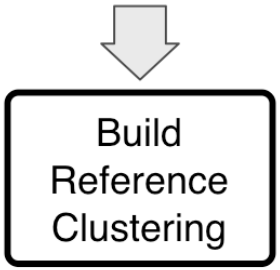


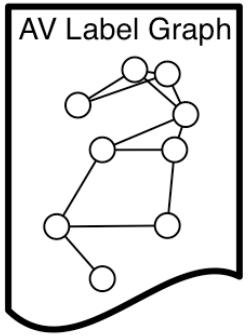
- Measure distance between each pair of malware samples in dataset

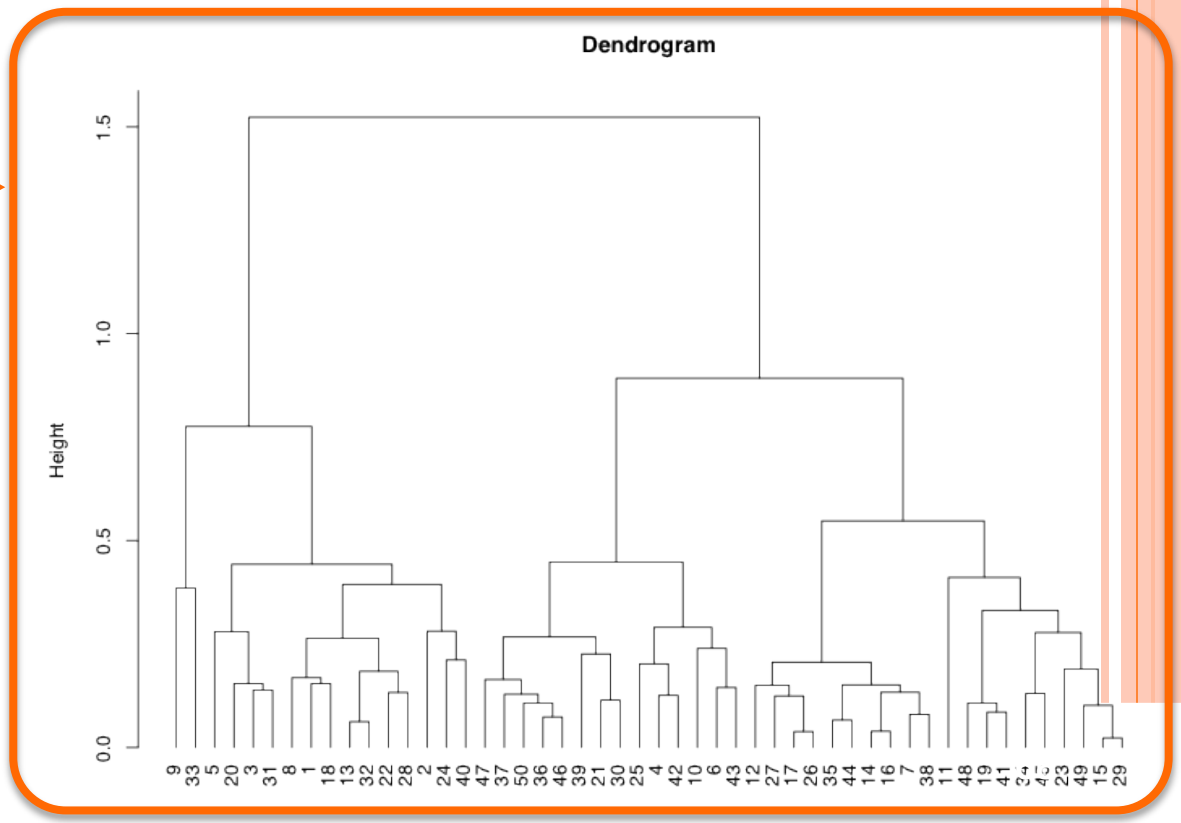


Building Reference Clustering

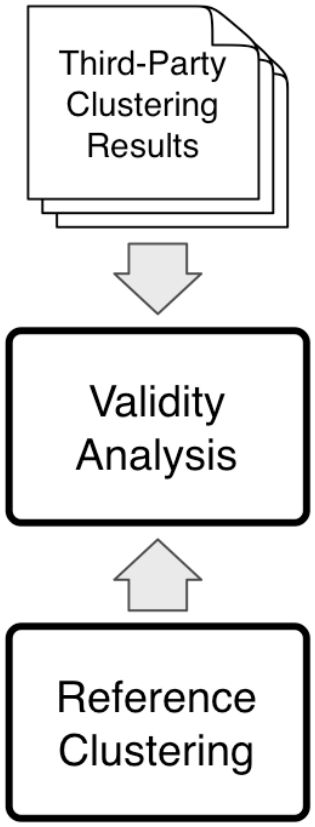
- Apply average-linkage hierarchical clustering on distance matrix



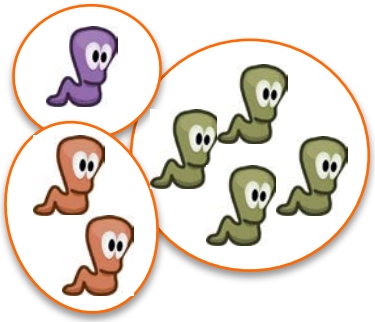




Computing Validity Indices

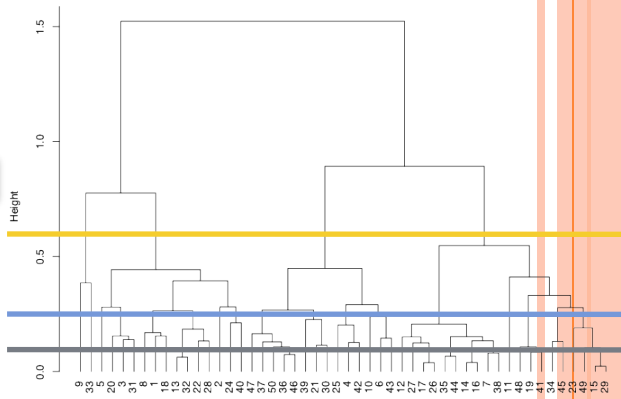


3rd-party Clustering Results



External Validity Index (e.g., Jaccard)

AV label-based Reference Clustering



Quality Index = *max*(validity index)

VAMO v.s. Majority Voting

- Which one can better tolerate AV label inconsistencies?

- Experimental Setup
  - Synthetic Dataset
    - complete *ground truth*
  - 3k samples in historic archive
  - 15 families, 200 samples each
  - 3 AVs (assume identical notation)
  - 300 samples in 3rd-party dataset

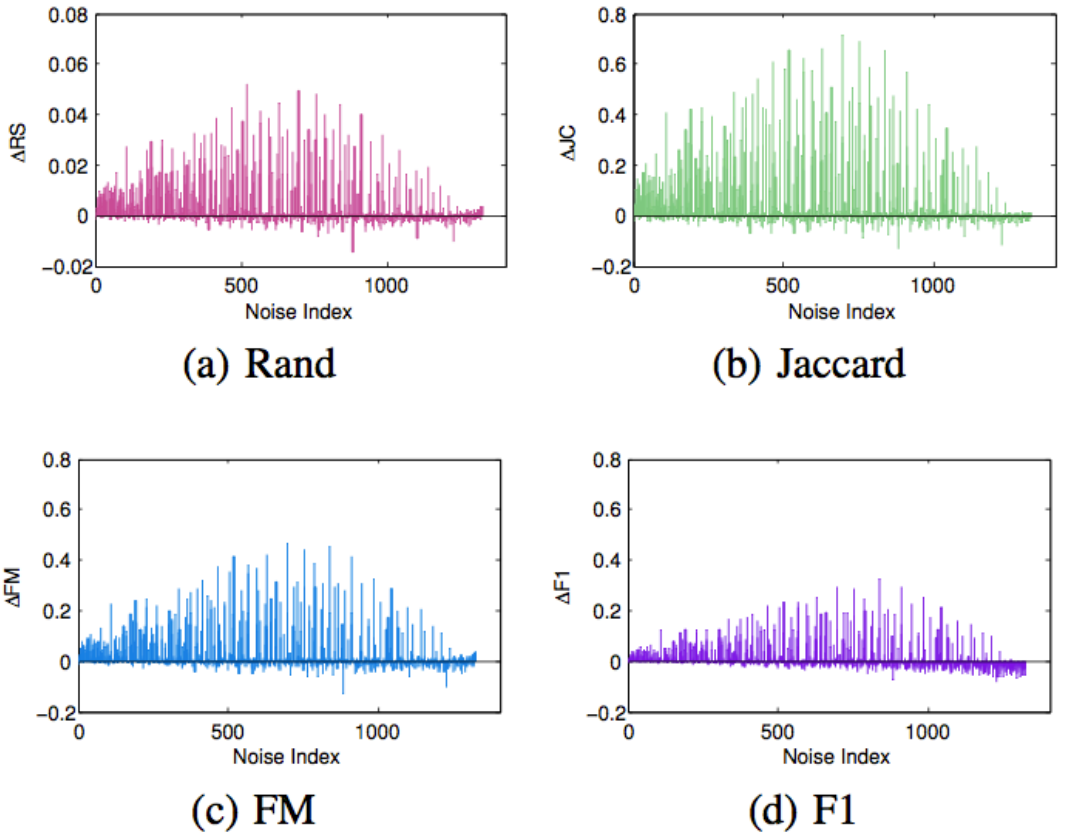
- Simulating AV Label Inconsistencies
  - Missing Labels
  - Label "Flips"

	AV1	AV2	AV3
M1	**Family5**	Family1	Family1
...		...	
M20	Family1	Family1	
M21	Family2		Family2
...		...	
M30	Family2	**Family11**	Family2
M31		Family3	**Family1**
⋮	⋮	⋮	⋮
M291	Family15	Family15	Family15
...		...	
M300	**Family8**	Family15	

VAMO v.s. Majority Voting

- VAMO's reference clustering *agrees more closely with ground truth*

- External validity indices
  - Rand Index
  - Jaccard Coefficient
  - Folkes-Mallows
  - F1 Index
    - Precision-Recall



VAMO IN PRACTICE

- Real-world malware dataset: 2,026 samples
- 3rd-party clustering algorithm: Bayer et al. (NDSS 2009)
  - Distance matrix based on system events
  - Hierarchical clustering
    - L = cut height
- VAMO's configuration
  - ~1M samples AV labels
  - 4 validity indices

l	*clusters*	**Rand**	**Jaccard**	**Folkes-Mallows**	**F1**
0.10	674	0.8767	0.2086	0.4494	0.7100
0.20	451	0.9172	0.5438	0.7308	0.7918
0.30	313	0.9205	0.5777	0.7482	0.7948
0.31	**301**	**0.9792**	**0.8924**	**0.9434**	*0.8436*
0.32	291	0.9790	0.8916	0.9430	0.8431
0.33	288	0.9759	0.8782	0.9357	**0.8501**
0.34	286	0.9759	0.8782	0.9357	0.8496
0.35	280	0.9758	0.8775	0.9353	0.8479
0.36	274	0.9757	0.8772	0.9352	0.8467
0.37	261	0.9721	0.8614	0.9265	0.8433
0.38	255	0.9721	0.8613	0.9265	0.8424
0.39	248	0.9722	0.8623	0.9270	0.8421
0.40	241	0.9721	0.8617	0.9268	0.8401
0.50	187	0.9585	0.8081	0.8971	0.7937
0.60	142	0.9260	0.7070	0.8366	0.7489
0.70	113	0.8527	0.5614	0.7354	0.7260
0.80	85	0.7789	0.4659	0.6656	0.7124

Limitations

- Beware of feature mismatch
  - AVs categorize malware based on reversing
  - Malware clustering systems use different features (e.g., behavioral)
- AV labels “evolve” in time
  - Samples detected using heuristics labeled as *generic*
  - Later, AVs may re-assign samples to a more specific family
- Heuristics-based detection more and more common
  - Will most samples be labeled as *generic* in the future?
  - Do AV customers care about reliable malware naming?

Contributions

- We proposed a novel system – VAMO, that enables a fully automated malware clustering validity analysis
- We performed an extensive evaluation of VAMO
  - How different types of AV label inconsistencies may negatively impact analysis performed via majority voting
  - The advantages that VAMO brings over previous work
- Demonstrated a practical application of VAMO over a real-world malware dataset

Conclusions and Future Work

- Present two effective machine learning frameworks to tackle challenging problems in the domain of Cancer Prediction and Malware Clustering
- The frameworks successfully improve the learning outcomes by using cluster validity analysis to reduce the label uncertainty
- ROMP – Ranked “unconfirmed” EGFR mutations, and identified mutations with suspicious labeled
- VAMO – Provided a fully automated assessment of the quality of malware clustering results
- Problem formulation is most important

Question?