Article

Analyzing Biomedical Datasets with Symbolic Tree Adaptive Resonance Theory

Sasha Petrenko [1,*], Daniel B. Hier [1,2], Mary Bone [3], Tayo Obafemi-Ajayi [4], Erik J. Timpson [5], William E. Marsh [5], Michael Speight [5], and Donald C. Wunsch II [1]

[1] Department of Electrical and Computer Engineering, Missouri University of Science and Technology, Rolla, MO USA
[2] Department of Neurology and Rehabilitation, University of Illinois at Chicago, Chicago IL USA
[3] University of Southeastern Norway, Kongsberg, Norway
[4] Engineering Program, Missouri State University, Springfield, MO, USA
[5] Honeywell Federal Manufacturing & Technologies, Kansas City, MO, USA
* Correspondence: petrenkos@mst.edu

Abstract: Biomedical datasets distill many mechanisms of human diseases, linking diseases to genes and phenotypes (signs and symptoms of disease), genetic mutations to altered protein structures, and altered proteins to changes in molecular functions and biological processes. It is desirable to gain new insights from these data, especially with regard to the uncovering of hierarchical structures relating disease variants. However, analysis to this end has proven difficult due to the complexity of the connections between multicategorial symbolic data. This article proposes Symbolic Tree Adaptive Resonance Theory (START), with additional supervised, Dual-Vigilance (DV-START), and Distributed Dual-Vigilance (DDV-START) formulations, for the clustering of multicategorical symbolic data from biomedical datasets by demonstrating its utility in clustering variants of Charcot-Marie-Tooth disease using genomic, phenotypic, and proteomic data.

Keywords: adaptive resonance theory; biomedical data; categorical data; ontologies; knowledge graphs

Citation: Petrenko, S.; Hier, D.; Bone, M.; Obafemi-Ajayi, T.; Wunsch, D. Analyzing Biomedical Datasets with Symbolic Tree Adaptive Resonance Theory. *Information* **2023**, *1*, 0. https://doi.org/

Received:
Revised:
Accepted:
Published:

1. Introduction

Precision medicine depends upon a detailed unraveling of the relationships between diseases, phenotypes, genes, and the underlying proteins and biological pathways [1–7]. The ready availability of protein, disease, gene, phenotype, and biological pathway ontologies makes it possible to construct purpose-specific datasets for studying human disease. These can take the form of symbolic relationships that can be organized into formal ontologies that are instantiated as knowledge graphs defining the permissible relationships between classes and the instances within them.

However, many elements in these disease-gene-protein datasets are formatted as categorical rather than numerical variables, bringing a unique challenge to machine learning algorithms. Although tools exist to analyze and visualize categorical data, the tools for clustering these datasets depend heavily on recasting categories into real-valued spaces, which is largely unavoidable due to the definition of the problem statement; all modalities of machine learning assume distance metrics or similarity measures of their feature spaces, whereas categorical data contains symbols that do not belong to ordered sets and thus do not inhabit metric spaces. An important design choice then when working with mixed or fully categorical data is how to recast categorical features into spaces with similarity measures [8]. This recasting, whether by one-hot encoding, ordinal encoding, or other encoding scheme, can bring its own deleterious consequences; one-hot encoding of categories can generate large sparse feature vectors due to many different categories, while ordinal encoding can introduce measures of proximity between categories that do not intrinsically exist. Meta-analyses of symbolic datasets may yield similarity meta-metrics that useful for clustering

[9,10], but these meta-metrics require domain knowledge of the categories in the dataset, limiting both their transferability to other datasets and applicability to streaming learning. While statistical machine learning algorithms can compensate for some of these input feature space shortcomings through sophisticated machinery that relies on large dataset size and high degrees of feature cardinality, these methods naturally suffer in regimes with small categorical datasets. Furthermore, these encoding schemes and the machine learning algorithms do not gracefully extend to instances of hierarchical or nested attributes such as occurs with the variably-sized association of diseases with phenotypes, genes, and proteins. Adaptive Resonance Theory (ART) algorithms principally belong to the class of incremental neurogenesis clustering algorithms with many variants for use in supervised and reinforcement learning applications. The design of these algorithms allows them to update existing categories or create new ones from the data alone in a stable, incremental, and lifelong manner. With the notable exception of the binary-valued ART1 algorithm, most of these algorithms work upon real-valued preprocessed feature datasets [11]. The Gram-ART algorithm was designed for the meta-optimization of genetic algorithms and thus is designed to work with variable-length symbolic datasets [12], but it too has its shortcomings when tackling the large numbers of terminal symbols encountered in medical disease datasets.

With these myriad design challenges in mind, this article describes the design of a new ART algorithm named Symbolic Tree Adaptive Resonance Theory (START) for the clustering of variable-length symbolic statements. This formulation of START also includes both dual-vigilance (DV-START) and distributed dual-vigilance (DDV-START) variants [13,14]. This article also outlines methods for casting categorical disease-gene biomedical datasets into symbolic datasets for both unsupervised clustering and supervised training where labels are available.

The changes of START to the Gram-ART algorithm summarize the novel contributions of this article, in addition to the use of this algorithm to the study of biomedical disease variant data. START extends Gram-ART as a novel approach to analyzing biomedical disease variant data in the following ways:

1. Both a match and activation function for the Gram-ART match rule.
2. Optimizations to the prototype encoding scheme to mitigate memory complexity in grammars with large sets of terminal symbols.
3. A mechanism to grow prototype tree structures when novel production rule sets are encountered.
4. Both Dual-Vigilance and Distributed Dual-Vigilance START variants [13,14].
5. A supervised modification for each unsupervised START variant.

This article is organized into the following sections: Section 2 provides a background of literature necessary to the formulation of START, while Section 3 describes the derivation and structure of START and its dual-vigilance variants. Section 4 outlines the datasets and experimental methodology utilized in the evaluation of START, including benchmark machine learning datasets and the target biomedical disease variant datasets of the article, and Section 5 contains the results of these experiments. Section 6 discusses the experimental results and their biological plausibility, with Section 7 providing final conclusions on both START and biomedical dataset analysis of the previous sections.

2. Background

2.1. *Adaptive Resonance Theory*

Adaptive Resonance Theory (ART) is a neurocognitive theory of how biological neural networks for self-stable representations and learn without catastrophic forgetting, online and without supervision, through feedback and competitive dynamics [15,16]. Since its inception, a variety of machine learning models have been implemented using the theory as a basis. Though these algorithms in large part belong to the class of incremental neurogenesis clustering algorithms, they have been adapted for applications in supervised, reinforcement, and even multimodal learning [11,17], tackling clustering issues from sample granularity [13,14] to distributed representations [18] and context recognition [19,20]. Some algorithms based upon ART have even been combined with Incremental Cluster Validity Indices (ICVIs), metrics of clustering performance in the absence of supervised labels, to enable a variety of incremental, online, and multimodal clustering and biclustering applications [21–25] ART algorithms are additionally well suited for Lifelong Learning (L2) applications because they are derived from theories on how biological neural networks address the stability-plasticity dilemma to mitigate catastrophic forgetting [26,27].

Nearly all ART formulations trade the explicit coarseness parameters of other clustering algorithms for a vigilance parameter ($\rho \in (0, 1)$) which behaves as a threshold of agreement between a sample and expectations to determine whether to update existing knowledge or to create new categories altogether, a process known as the *ART match rule* [28]. Samples are provided in a feature representation layer $F1$, which is compared with a category representation layer $F2$ through ART competitive dynamics that include a check against this vigilance parameter.

2.2. Gram-ART

Gram-ART is a clustering algorithm, based on ART learning dynamics, that defines its prototypes and input features as trees of parsed statements adhering to a formal grammar [12]. Originally designed to tackle the problem of comparing similarity between symbolic expressions for the meta-optimization of genetic algorithms, it is capable of accepting statements of an arbitrary length according to a user-defined context-free grammar (CFG) expressed in the Backus-Naur form (BNF). In the original formulation, Gram-ART samples are statements adhering to a CFG that are parsed into rooted syntax trees. These parsed samples are then compared according to ART learning rules to Gram-ART prototypes that are themselves rooted trees containing distributions of encountered terminal symbols at each node. Gram-ART answers the questions of how to formulate prototype trees of varied shape, compute similarities of sample statements to prototypes of differing shapes, and update the terminal symbol distributions at each node during learning.

Gram-ART is the first ART algorithm capable of clustering inputs samples of arbitrary length, but it also inherits some problems from working with symbolic data. Terminal symbols under a grammar have no fuzzy membership or relation without an additional embedding scheme. Gram-ART tackles this by updating distributions of terminal symbols at each position along the rooted prototype trees during learning. However, this technique quickly grows in space and subsequent time complexity in grammars with sets of terminal symbols larger than the algebraic expressions that it was originally designed for.

3. Method

3.1. START: Symbolic Tree Adaptive Resonance Theory

This paper introduces a new formulation of the Gram-ART algorithm called START for the clustering of symbolic datasets. START is a prototype-based unsupervised clustering algorithm that when presented with a new sample utilizes ART dynamics to determine whether to update an existing template or to instantiate a new one. START targets symbolic expressions adhering to a context-free grammar $CFG(T, N, P, S)$ with a complete set of terminal symbols T, non-terminal symbols N, production rules P, and statement entry point S. The prototypes of START are rooted trees containing learned distributions of encountered terminal symbols at each node representing a non-terminal position, and symbolic statements are parsed into rooted constituency parse trees that are subsequently processed against these prototypes using ART learning dynamics. With such a formulation, the method is naturally extended to the clustering of purely categorical datasets of variable length sequences, such as in the myriad categorical fields of disease-gene-protein data.

3.1.1. Motivation

Given that START shares the objective of Gram-ART to cluster variable-length symbolic expressions, the key design challenges of START's design are in how to formulate metrics of similarity between these symbolic expressions. In such a formulation, statements are collections of symbols sampled from unordered sets; individual symbols share no fuzzy membership, so similarity between symbols is dictated by strict equivalence in a set theoretic sense. Furthermore, though statements of equal length introduce a step-wise fuzziness when symbols in the same relative positions are identical, many datasets do not satisfy the assumption of equivalent non-terminal structure across all statements. In the pursuit of creating a clustering algorithm for variable length symbolic datasets, START utilizes a prototype method as a proxy for direct comparisons between statements, using ART-based competitive learning dynamics for determining when to update templates and when to instantiate new ones. As with all ART algorithms, START therefore inherits both the theoretically unlimited learning capacity of neurogenesis algorithms and the problems of category proliferation that they bring; though new prototypes can be instantiated for an arbitrary number of categories, this growing knowledge base incurs its own search time complexity [11,29].

3.1.2. START Algorithm

START shares the nomenclature of Gram-ART and other ART algorithms from its structure to its learning dynamics, so existing terminology is preferred where available. START also follows the procedure of most ART unsupervised clustering algorithms with additional considerations for handling symbolic data. As in Gram-ART, START handles this symbolic data by working in the space of the syntactic trees representing the symbolic data as statements under a formal grammar.

Algorithm 1: Shared START notation. The learning dynamics of START and its variants follow the activation, competition, match, update, and initialization rules of unsupervised ART algorithms, so the notation here largely adheres the elementary ART algorithm notation outlined in [11]. Dual-vigilance lower-bound ρ_{lb} and upper-bound ρ_{ub} follow the notation in DVFA [13] and DDVFA [14].

```
/* Notation                                                              */
R: set of prototype nodes.
$R$: a single prototype node.
C: set of prototype nodes indices.
Λ: subset of active ART module nodes indices ($\Lambda \subset C$).
$\rho$: START vigilance threshold, $\rho \in (0, 1)$.
$\rho_{lb}$: dual-vigilance lower-bound vigilance threshold, ($\rho_{ub} > \rho_{lb} > 0$).
$\rho_{ub}$: dual-vigilance upper-bound vigilance threshold, ($1 > \rho_{ub} > \rho_{lb}$).
$n$: number of input dataset statements.
$\mathbf{X}$: statements parsed as syntax trees with terminal metadata.
Parser(·): syntactic parsing algorithm taking a set of statements and a grammar
  and producing rooted trees.
$f_T(\cdot)$: activation function.
$f_M(\cdot)$: match function.
$f_N(\cdot)$: node initialization function.
$f_L(\cdot)$: node weight update function.
U: internal supervised category indices.
L: set of cluster indices.
```

A START module is initialized to contain the $CFG(T, N, P, S)$ rules of the target symbolic dataset statements. This grammar can be inferred from an existing dataset of statements if all relevant symbols and production rules are represented in the dataset. Statements from the dataset are parsed according to the production rules of the grammar into rooted contituency parse trees, the basic unit of which is known in Gram-ART and START as a TreeNode. Each parsed statement tree is presented incrementally to the START module, and each sample either mutates an existing prototype or is used to instantiate an entirely

TreeNode
Symbol : GrammarSymbol Children : Vector{TreeNode}

Figure 1. A simple UML diagram of the stateful information of one START TreeNode [12]. A symbol in a TreeNode in START is realized by either a terminal or non-terminal symbol at the syntax tree position of the node. A rooted tree of TreeNode in this regard contains the minimum information necessary to describe the syntax tree of a statement parsed with a prescribed grammar.

ProtoNode

Symbol : NonTerminalGrammarSymbol
Distribution : Dictionary{TerminalGrammarSymbol, Float}
InstanceCount : Dictionary{TerminalGrammarSymbol, Integer}
Children : Vector{ProtoNode}

Figure 2. A simple UML diagram of the stateful information of one START ProtoNode, which is the basic element of the rooted trees constituting the prototypes of START [12]. A rooted tree of START ProtoNode encodes only through the non-terminal positions of the syntax tree of a TreeNode tree. Each ProtoNode encodes a PMF of terminal symbols encountered at and below the non-terminal position of the ProtoNode itself, with instance counts of each terminal encoded for the renormalization of the PMF when learning occurs at the node itself.

new prototype [11]. Prototypes in START are themselves rooted trees with a modified structure from the statement trees, the basic unit of which is known in Gram-ART as a ProtoNode. The stateful information of START ProtoNode and TreeNode can be seen in Figures 2 and 1, respectively.

Here, START and Gram-ART differ on an important point in formulation; Gram-ART treats ProtoNode and TreeNode as modified dependency relation syntax trees where each node representes a terminal symbol, the children of which are the dependents of that symbol. This formulation is most apparent in the case of operators, such as in the algebraic statement $x + y$, where the operator terminal $+$ would have branch dependents x and y. In START, however, ProtoNode and TreeNode are defined as relation parse trees with non-terminal symbols representing non-terminal positions and terminal symbols at the leaves of the rooted tree. The same algebraic statement $x + y$ is then treated as

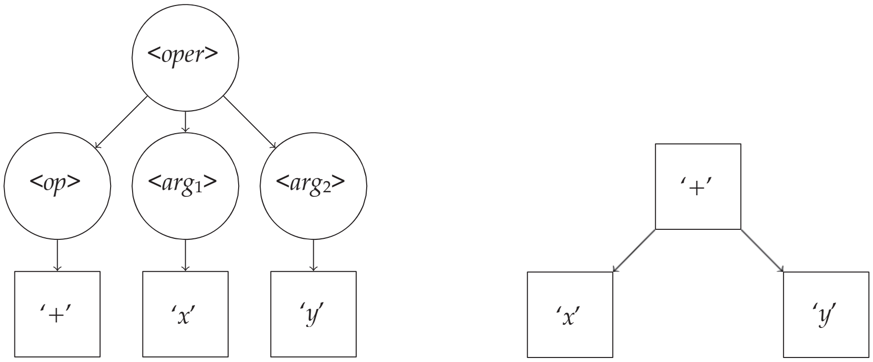
In START, sample symbolic statements are preprocessed into parse trees via a syntactic parser such as an Earley parser according to the production rules P of the grammar written most generally in an Extended Backus-Naur form (EBNF) [30]. These syntax trees can be interpreted as concrete constituency-relation parse trees belonging to constituency grammars, also known as phrase structure grammars, where branches of a parse tree are all non-terminal symbols in the grammar, including the statement entry point, and leaf nodes are terminal symbols [31]. These parse trees are then converted to statement trees via an inclusion of metadata at each node indicating the symbol to be terminal or non-terminal. Prototypes in START are rooted trees containing probability mass functions (PMF) of terminal symbols encountered at and below the position of each ProtoNode on the tree. In contrast with Gram-ART, these START prototypes do not contain terminal symbol leaves; instead, the nodes of the prototypes represent the non-terminal positions of the grammar production rules applied to the node's position on the tree, which reduces the effective size of each prototype tree while still encoding the occurrence of terminal symbols at and below those positions via their PMFs.

3.1.3. Derivation of the START Match Rule

A fundamental characteristic of ART algorithms is the use of a *match rule*, whereby a process of bottom-up *activations* drive the evaluation of how much the input sample *matches* existing top-down categories [28]. Because of the origins of these algorithms in the analysis of the competitive dynamics of biological neural networks, these *activation*

Algorithm 2: START algorithm. A set of symbolic statements under a formal context-free grammar are parsed into their syntax trees. Prototypes are defined as Learning dynamics otherwise follow the activation, competition, match, update, and initialization rules of unsupervised ART algorithms [11]. ART dynamics notation here largely follow the elementary ART algorithm outlined in [11]. Inference during classification follows the same match rule dynamics without the instantiation of new categories; in the case of complete mismatch, either an “unknown" label or the best matching unit (the category that maximizes the match criterion) may be returned.

```
    Data: Symbolic statements $\mathbf{S}$; CFG grammar $\mathbf{G}$ with terminal symbols $\mathbf{T}$,
          non-terminal symbols $\mathbf{N}$, production rules $\mathbf{P}$, and statement definition $\mathbf{S}$.
    Result: Cluster labels $\mathbf{Y} \in \mathrm{N}^n$
    /* Parse statements into contsituency parse trees                    */
 1  $\mathbf{X} \leftarrow \mathrm{Parser}(\mathbf{S}, \mathbf{G})$
    /* Iteration over parsed statement trees                             */
 2  foreach $\mathbf{x} \in \mathbf{X}$ do
        /* Compute activations for all nodes                             */
 3      $T_j \leftarrow f_T\ \mathbf{x}, \mathrm{R}_j\ , \forall j \in \mathrm{C}$
        /* Perform WTA competition for active nodes                      */
 4      $J \leftarrow \arg\max_{j \in \Lambda} T_j$
        /* Compute match for the winning category                        */
 5      $M \leftarrow f_M\ \mathbf{x}, \mathrm{R}_J$
        /* Vigilance test                                                */
 6      if $M > \rho$ then
            /* Update category                                           */
 7          $\mathrm{R}_J \leftarrow f_L\ \mathbf{x}, \mathrm{R}_J$
 8      else
            /* Deactivate category                                       */
 9          $\Lambda \leftarrow \Lambda - \{J\}$
10          if $\Lambda /= \emptyset$ then
                /* Continue match search                                 */
11              Goto Line 4
12          else
                /* Create and initialize new category                    */
13              $K \leftarrow \|\mathrm{C}\|_1 + 1$
14              $\mathrm{R}_K \leftarrow f_N(\mathbf{x}, \mathbf{G})$
```



(a) START relation parse tree TreeNode. **(b)** Gram-ART syntax tree TreeNode.

Figure 3. Comparison of the constituency relation parse trees of START to the syntax trees of Gram-ART to the simple algebraic statement $x + y$. START TreeNode are full constituency relation parse trees containining terminal symbols at the leaves of the tree, while START ProtoNode contain only non-terminal symbols at non-terminal positions on the parse tree, are full relation grammar parse trees. As in the grammar Listing 1, non-terminal symbols are surrounded by arrows <·> and terminal symbols are in single quotations. Here, <*oper*> is "operation," <*op*> is "operator," and <arg_1> and <arg_2> are "arguments" of the operator.

and *match* functions are frequently analogized with bottom-up *prediction* and top-down *expectation*, respectively.

Gram-ART utilizes an activation function, while START introduces separate activation and match functions. The distinction between the two lies in the normalization scheme of the activation and match functions; for example, in ART1 the match function (Equation 2) is the activation function (Equation 1) normalized by the size of the input [11].

$$T_j = \|\boldsymbol{x} \cap \boldsymbol{w}_j\|_1 \tag{1}$$

$$M_j = \frac{\|\boldsymbol{y}^{(F_1)}\|_1}{\|\boldsymbol{x}\|_1} = \frac{\|\boldsymbol{x} \cap \boldsymbol{w}_j\|_1}{\|\boldsymbol{x}\|_1} \tag{2}$$

FuzzyART replaces the binary intersection with the fuzzy intersection in both equations and normalizes the activation by the magnitude of the weight vector [11]. When evaluated at a single node, an input terminal symbol can be interpreted as a one-hot binary vector encoding at the terminal symbol position, so the magnitude of the membership of sample $\boldsymbol{x}$ in weight $\boldsymbol{w}_j$ is indeed the fuzzy intersection $\|\boldsymbol{x} \wedge \boldsymbol{w}_j\|_1$. This is computed in START for the terminal distribution of each ProtoNode climbing up from the aligned leaf representing the terminal symbol. In statements with many branches arising from non-trivial production rules, this means the evaluation of the activation at each protonode for potentially multiple terminal descendants.

The activation is then normalized by the size of the input pattern, which can be realized in multiple manners requiring a design decision; with the rooted tree definition of parsed input statements, the size of the input pattern could be interpreted as the number of nodes in the parsed statement, the number of terminal symbols in the unparsed statement, or a more complex function of the number of terminals that could be realized beneath the non-terminal position of the node in question according to the production rules of the grammar of the sample. For simplicity, the remainder of this study utilizes the length of the unparsed statement itself as a normalizing factor, having the effect of discounting the disproportional contributions to the match value of increasingly longer statements. In grammars where statements are of equal length such as in the processing of tables with single-category data, each decision trivially scales the required vigilance values to satisfy the vigilance criterion.

The remainder of the match rule follows the activation, competition, match, and vigilance test of unsupervised ART algorithms as can be seen in Algorithm 2, with the exception of the dual-vigilance variants of START which can be seen in Section 3.1.5 and Algorithm 3.

3.1.4. Derivation of the Weight Update

When a prototype is selected for learning according to the START match rule, the input TreeNode and selected ProtoNode are root-aligned and compared similar to in the activation and match processes. The terminal symbols contributing to the activation and match functions of the winning prototype are used for updating the PMF at each non-terminal symbol position at each ProtoNode up the prototype tree. The instance count of the observed terminal symbol is incremented, and the PMF update is weighted by the instance count of each terminal of the distribution to renormalize. In Equation 3, the weight value w of the PMF indexed at terminal T in node i is updated with instance count N and a Kronecker delta δ_T that is satisfied if the terminal symbol x being evaluated is equivalent to the PMF index T.

$$w_i^T = \frac{w_i^T * N + \delta_{Tx}}{N+1} \tag{3}$$

$$\delta_{Tx} = \begin{cases} 1 & \text{if } T = x \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

If no prototype satisfies the vigilance criterion, a new one is instantiated. START prototypes do not encode all combinations of non-terminal production evaluations during instantiation, as this would quickly combinatorially explode towards the Catalan number of the non-terminal production rules, and it could be infinite in some recursive grammars. Instead, prototypes are instantiated as structural clones of the input TreeNode without the inclusion of the terminal symbols at their leaves. This design decision is made to mitigate the time and memory complexity of the ProtoNode evaluation given that the non-terminal node preceding a terminal leaf already encodes all of the instances that the terminal symbol is encountered. The new structural clone prototype is then trained upon the input sample, updating the PMFs of each ProtoNode for the first time. In the case that an existing winning prototype does not contain a the input TreeNode as a structural subset (i.e., it is missing a non-terminal production rule path describing the parsed TreeNode), these new non-terminal paths are instantiated on the winning prototype and updated as usual.

3.1.5. Dual-Vigilance and Distributed Dual-Vigilance START

The FuzzyART algorithm provides a foundation of how to adapt ART learning rules to real-valued datasets [11]. Like most ART modules, FuzzyART utilizes the ART match rule evaluated at a single threshold value that is either the vigilance hyperparameter ρ or a function thereof. Dual-Vigilance FuzzyART (DVFA) utilizes instead two vigilance parameters for the match rule evaluation, a lower-bound ρ_{lb} and upper-bound ρ_{ub}, which separates prototypes in a many-to-one mapping from categories to clusters and introduces the ability to compensate for differing granularity both within and between clusters [13]. Distributed Dual-Vigilance FuzzyART (DDVFA) advances this idea by representing entire clusters with FuzzyART modules governed by a global FuzzyART module, compensating for even varying granularity within different clusters and enabling the ability learn arbitrary cluser shapes [14]. Each node in the global F2 layer competes for assignment of a provided sample through modified activation and match linkage methods defining the relevant proximity measures of the sample to an entire F2 FuzzyART module node.

The principles of dual-vigilance and distributed dual-vigilance are extended here for START. In the Dual-Vigilance formulation (DV-START), the same cascading technique as in DVFA is used for determining category-cluster assignments through upper- and lower-bound vigilance hyperparameters during the ART match evaluation:

1. **Case 1:** $M_J > \rho_{ub}$: if the current match candidate satisfies the upper vigilance threshold, then the winning category is updated according to the START weight update rules.
2. **Case 2:** $\rho_{ub} > M_J > \rho_{lb}$: if the current match candidate only satisfies the lower vigilance threshold but not the upper, then a new category prototype is instantiated that belongs to the same cluster as the winning node.
3. **Case 3:** $\rho_{lb} > M_J$: if the current match candidate does not satisfy even the lower-bound vigilance threshold, then the normal mismatch procedure is followed where a new category is instantiated belonging to an entirely new cluster.

In the Distributed Dual-Vigilance formulation (DDV-START), additional modifications are made to accommodate the rooted tree structures of the prototypes. DDVFA utilizes a global FuzzyART module that represents nodes themselves as FuzzyART modules [14]. The basic unit of DDV-START is the rooted ProtoNode trees, but global module dynamics are not restricted to their use; because the global module of DDV-START is largely agnostic to the formulation of the input samples, the global module may be approximated as a FuzzyART module coordinating the learning of its START F2 nodes. With the exception of the centroid linkage method, which in DDVFA is defined as a function of local FuzzyART weights, all other linkage methods from DDVFA can be utilized in DDV-START; by independently defining the activation and match values for each ProtoNode within an F2 START module, the global values can be compared using the Hierarchical Agglomerative Clustering (HAC) methods of DDVFA as can be seen in Table 1.

Table 1. Distributed Dual-Vigilance START activation and match linkage methods where hierarchical agglomerative clustering (HAC) functions and distributed dual-vigilance notation are shared with DDVFA [14]. Global activation T_i^g and match M_i^g functions are defined via the generic function h_i^g for global F2 node index i as a function of inner node indices $j = 1 \dots k$ where k is the number of F_2 nodes in the local START module i. Each HAC method then is a "function of functions" evaluated at each F2 node in the global module to determine either the match or activation value in the global module match rule dynamics.

HAC method	h_i^g
Single	$\max_j \{ f_j^i \}$
Complete	$\min_j \{ f_j^i \}$
Median	$\operatorname{median}_j \{ f_j^i \}$
Average	$\frac{1}{k_i} \sum_{j=1}^{k_i} f_j^i$
Weighted[1]	$\sum_{j=1}^{k_i} p_j f_j^i$

[1] $p_j = \frac{n_j^i}{n_i^g}$ where n_j^i is the number of samples (i.e., instance count) encoded by j of the local START module at global F2 index i and $n_i^g = \sum_j n_j^i$.

3.1.6. Supervised Variants

Most ART algorithms are designed as unsupervised clustering algorithms with variants and compositions of the elementary ART module motif providing supervised and reinforcement learning variants [11]. ARTMAP is a formulation of ART, comprised of two elementary ART modules and an inter-ART map field, that enables multidimensional mapping between two feature fields [32]. A simplified version of FuzzyARTMAP, where the second module ART_B is replaced with vectors representing class labels, provides a basic

Algorithm 3: Dual-Vigilance START algorithm. This algorithm combines Algorithm 2 with the dual-vigilance procedure of DVFA [13]. The vigilance test is split into a cascade of two vigilance checks for the current match candidate node. Passing the upper vigilance check updates the current category node, while passing only the lower vigilance check creates a new category node belonging to the same cluster label. Failing to pass both vigilance checks results in the instantiation of a new category node belonging to an incrementally new cluster label.

```
   Data: Symbolic statements $\mathbf{S}$; CFG grammar $\mathbf{G}$ with terminal symbols $\mathbf{T}$,
         non-terminal symbols $\mathbf{N}$, production rules $\mathbf{P}$, and statement definition $\mathbf{S}$.
   Result: Cluster labels $\mathbf{Y} \in \mathbb{N}^n$
   /* Parse statements into contsituency parse trees */
1  $\mathbf{X} \leftarrow \text{Parser}(\mathbf{S}, \mathbf{G})$
   /* Iteration over parsed statement trees */
2  foreach $\mathbf{x} \in \mathbf{X}$ do
       /* Compute activations for all nodes */
3      $T_j \leftarrow f_T\ \mathbf{x}, \mathrm{R}_j\ , \forall j \in \mathrm{C}$
       /* Perform WTA competition for active nodes */
4      $J \leftarrow \arg\max_{j \in \Lambda}\ T_j$
       /* Compute match for the winning category */
5      $M \leftarrow f_M\ \mathbf{x}, \mathrm{R}_J$
       /* Dual vigilance tests */
6      if $M > \rho_{ub}$ then
           /* Update current category */
7          $\mathrm{R}_J \leftarrow f_L\ \mathbf{x}, \mathrm{R}_J$
8      else if $M > \rho_{lb}$ then
           /* Create a new category within the same cluster */
9          $K \leftarrow \|\mathrm{C}\|_1 + 1$
10         $\mathrm{L}_K \leftarrow \mathrm{L}_J$
11         $\mathrm{R}_K \leftarrow f_N(\mathbf{x}, \mathbf{G})$
12     else
           /* Deactivate category */
13         $\Lambda \leftarrow \Lambda - \{J\}$
14         if $\Lambda /= \emptyset$ then
               /* Continue match search */
15             Goto Line 4
16         else
               /* Create and initialize new category and cluster */
17             $K \leftarrow \|\mathrm{C}\|_1 + 1$
18             $\mathrm{L}_K \leftarrow \max(\mathrm{L}) + 1$
19             $\mathrm{R}_K \leftarrow f_N(\mathbf{x}, \mathbf{G})$
```

procedure for adapting unsupervised ART modules to simple supervised ARTMAP variants [33]. Though START is designed as an unsupervised clustering algorithm, it utilizes these supervised modifications for evaluation on benchmark datasets in Section 4.1.

Algorithm 4: Simplified supervised modification for all START variants. The variation between START variants is captured in the evaluation of vigilance test as a function f_V; if some node satisfies the match rule of the START variant, the sample is said to fall within the vigilance region of the prototype [11]. Complete mismatch instead occurs when no vigilance test is satisfied, and the prototype initialization procedure of the START variant is triggered.

```
Data: Symbolic statements $\mathbf{S}$; supervisory labels $\Omega$; CFG grammar $\mathbf{G}$ with
      terminal symbols $\mathbf{T}$, non-terminal symbols $\mathbf{N}$, production rules $\mathbf{P}$, and
      statement definition $\mathbf{S}$.
Result: Cluster labels $\mathbf{Y} \in \mathrm{N}^n$
/* New prototype initialization procedure */
Function initialization($\omega$):
    /* Create and initialize new category */
    $K \leftarrow \|\mathrm{C}\|_1 + 1$
    $\mathrm{R}_K \leftarrow f_N(\mathbf{x}, \mathbf{G})$
    /* Add the label to the label map */
    $\mathrm{U}_K \leftarrow \omega$
/* Parse statements into syntax trees */
$\mathbf{X} \leftarrow$ Parser($\mathbf{S}, \mathbf{G}$)
/* Iteration over parsed statement trees */
foreach $\mathbf{x}, \omega \in \mathbf{X}, \Omega$ do
    /* Instantiate a new prototype with the supervised label if the
       label is completely novel */
    if $\omega \notin \mathrm{U}$ then
        initialization($\omega$)
    else
        /* Parse statements into contsituency parse trees */
        $V_J = f_V(\mathrm{R}, \mathbf{x})$
        /* Update winning node $J$ if it correctly predicts label $\omega$ */
        if $V_J \wedge (\omega \in \mathrm{U})$ then
            /* Run START update procedure */
            $f_L$  $\mathrm{R}_J, \mathbf{x}$
        else
            initialization($\omega$)
```

3.1.7. Comparison with Existing Methods

START is most directly comparable with Gram-ART for two important reasons: Gram-ART is the first and, prior to START, only ART-based symbolic data clustering algorithm, and the design of START uses Gram-ART as a basis with some important modifications. Details on the design differences between START and Gram-ART can be seen in the Supplementary Materials section of this paper.

4. Evaluation

START is evaluated here both on existing benchmark machine learning datasets with known labels (outlined in Section 4.1) and on a custom biomedical dataset (outlined in Section A).

4.1. Benchmark Datasets

Purely categorical machine learning benchmark datasets are not as widespread and well-studied as real-valued benchmark datasets, and the START algorithm and its variants are not designed to handle real-valued data without modification. Therefore, START is evaluated on a combination of both real-valued clustering datasets and purely categorical datasets with caveats.

Gram-ART is originally verified upon a discretized version of the UCI Iris dataset, the UCI Mushroom dataset, and the UCI Unix User dataset [12,34–36]. For comparison, START is evaluated upon the following open-source machine learning benchmark datasets with existing labels: a set of real-valued clustering benchmark datasets [37,38], the categorical UCI mushroom dataset [35], and a categorical lung cancer patient dataset [39]. Because benchmark datasets such as the Iris flower dataset elements are real-valued, each feature is range-normalized and binned into a set of terminal symbols representing each bin.

Both the written procedures for accommodating real-valued benchmark datasets for evaluation and the results of all real-valued and categorical benchmark evaluations can be viewed in the Supplementary Material of this paper.

4.2. Charcot-Marie-Tooth Disease Data Set

To test the ability of START to cluster rows in a complex data set with various multi-category fields of varying length, we created a test data set based on Charcot-Marie-Tooth disease (CMT). CMT, also known as hereditary motor and sensory neuropathy, is one of the most common neurogenetic diseases with a population prevalence of 1 in 2,500 [40]. As a starting point, we began with 81 variants of CMT in the Online Mendelian Inheritance of Man (OMIM) phylogenetic series. A known genetic mutation characterizes each variant. The protein associated with the mutation is known in all but three variants. For each CMT variant, we added a row to a flat file with the following columns: variant name, OMIM number, gene, gene location, chromosome, mode of inheritance, phenotype, protein name, UniProtKB number, protein location, biological process in which the protein participates, protein molecular function, protein length, and protein weight. External data sources were identified to populate the data set (Table 2), including the Online Inheritance in Man (OMIM), the Human Phenotype Ontology (HPO), UniProtKB and the Human Protein Reference Database (HPRD) [41–44]. The final data set had 81 rows and 17 columns, as shown in Table 2). Seven columns were multicategorical. Gene number (OMIM), phenotype number (HPO), protein number (UniProtKB) and variant number (OMIM) were not used in the clustering.

Example production rules resulting from the interpretation of this dataset as statements sampled from a grammar can be found in Appendix A.

4.3. Cluster Feature Means and Heat Maps

After clustering by START, a cluster membership (between 1 and 9) was assigned to each row. Multicategorical features (see Table 2) were flattened into individual features by one-hot encoding. Feature means for each cluster were calculated using the *Aggregrate* procedure from SPSS (Version 29.0, IBM). The features were visualized using heat maps from Orange 3.35 [45]. For the heat maps, raw feature means were used for the categorical variables, and normalized feature means (in the interval $[0, 1]$ were used for the numerical variables (see Table 2).

4.4. SHAP Values

SHAP summary values were calculated by the method of Lundberg et al. [46]. START cluster membership was added to the flattened feature array (see above). The cluster configuration was fitted to the *HistGradientBoostingClassifier* (scikit-learn). The *shap.TreeExplainer* and the *shap.summary_plot* procedures were used to compute SHAP values and create the SHAP summary plot (Figure 13).

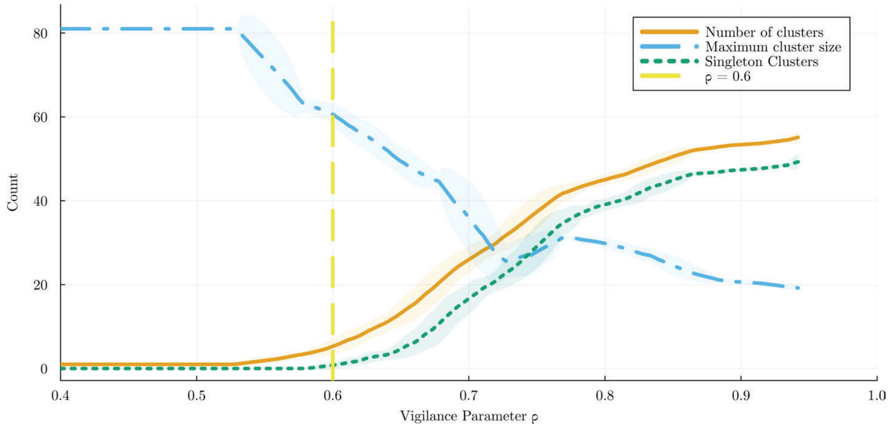


Figure 4. Effect of vigilance parameter ρ on number of clusters. As ρ was increased from 0.0 to 1.0, the maximum cluster size decreased, the number of clusters increased, and the number of singleton clusters increased. A value of $\rho = 0.6$ (yellow dashed line) was selected to yield 9 clusters with only two singleton clusters. Larger ρ values gave too many singleton clusters, and smaller ones put too many cases into one cluster.

5. Results

5.1. Selection of Cluster Configuration for the CMT Dataset

The vigilance factor ρ was varied between 0.0 and 1.0 (Fig: 4). To minimize the size of the largest cluster and minimize the number of clusters with one member, a $\rho = 0.6$ was selected, yielding 9 clusters (Figure 5).

Feature	Type	Format	Length	Multi-Category
variant name	categorical	string	variable	no
variant number	categorical	string	fixed	no
gene name	categorical	string	variable	no
gene number	categorical	integer	fixed	no
protein name	categorical	string	variable	no
protein number	categorical	string	fixed	no
protein length	numerical	integer	variable	no
protein weight	numerical	integer	variable	no
protein location	categorical	string	variable	yes
protein molecular function	categorical	string	variable	yes
protein biological process	categorical	string	variable	yes
protein class	categorical	string	variable	yes
mode of inheritance	categorical	string	variable	yes
Phenotype	categorical	string	variable	yes
phenotype number	categorical	string	variable	yes
Chromosome	categorical	string	variable	no
chromosome location	categorical	string	variable	no
chromosome location	categorical	string	variable	no

Table 2. Table of features and their characteristics in CMT flat file. Protein numbers were from UniProtKB [43]. Variant and gene numbers were from OMIM [41]. The phenotype numbers were HPO [1,47]. Since genes, proteins, and diseases have multiple names, the names were normalized to the standard form. Most of the features were categorical, and some were multicategorial. The features were formatted as integers or strings of variable or fixed length.

5.2. Cluster Characterization by Feature Composition

We used heat maps to visualize the features that characterized each cluster. The clusters differed in mode of inheritance, protein localization within the cell, protein participation in biological processes, protein length, molecular weight, motifs and domains in amino acid chains, phenotype, and protein molecular function (Figures 6, 7, 8, 9, 10, 11, 12, and 13).

The heat maps were used to create a narrative summary of each cluster's most important feature characteristics (Table 3).

5.3. Identifying features that contributed the most to cluster configuration

We used SHAP [46] to find the features that drove the cluster configuration. The SHAP summary plot (Figure: 13) showed that protein length, chromosome number (autosomes 1 – 22 and X and Y), mode of inheritance (autosomal recessive and autosomal dominant), protein localization in the cell (cytoplasm and plasma membrane) and phenotype (hypertonia, auditory and cognitive) contributed the most to cluster formation.

k	N	Process	Function	Location	Domain	Inherit	Phenotype Plus
1	6	apoptosis	Hydrolase			AD	auditory, visual
2	3			cytoplasm		AD	hypertonia
3	7	protein synthesis	Transferase			AD, AR	
4	53			plasma membrane	TM	AD,AR	
5	4			plasma membrane	TM	AD	cognitive, auditory
6	1	immunity transcription	Transferase	plasma membrane		AD	cognitive, ataxia seizure, hypertonia speech, hyperreflexia
7	4	transcription	DNA binding	plasma membrane		AD, AR	cognitive, hypotonia
			Transferase				
8	2	autophagy apoptosis	hydrolase GNRF	nucleus		AR	cognitive, auditory hypertonia
9	1		Transferase	mitochondrion	TM	XLR	cognitive, auditory

Table 3. Summary of features that characterize CMT clusters. **k** is the cluster number and **N** is count of members in each cluster. Phenotype Plus lists signs and symptoms in addition to weakness, atrophy, deformities, sensory loss, and hyporeflexia that characterize most cases of CMT. AD is autosomal dominant inheritance; AR is autosomal recessive; XLR is X-linked recessive. TM is the transmembrane protein domain. GNRF is the guanine nucleotide-releasing factor. Note that some of the characteristics identified by the SHAP analysis, including cognitive, hypertonia, auditory, plasma membrane, autosomal recessive, and autosomal dominant (Figure 13), recur in this summary table.

6. Discussion

6.1. Feasibility of Clustering Multi-Categorical Biomedical Data with START

START demonstrates several important capabilities that make it particularly useful for the clustering of multi-categorical data. Firstly, it directly represents the categorical data without an intermediate encoding representation and all the problems introduced therein; categorical data by definition does not define distance metrics or fuzzy membership between categories and features dimensions. The problem is circumvented here by the definition of prototype parse trees tracking the distributions of symbols from learned statements using the ART match and learning rules.

Secondly, it naturally compensates for data points with missing elements entries in its fields; rather than requiring a special encoding scheme for missing fields or removing data points altogether, START can represent missing fields as ununsed non-terminal positions when representing multi-categorical datasets as statements containing one or more attributes, which has the effect of penalizing the degree to which samples with missing features match existing prototypes while still accommodating prototypes of varying sizes.

Thirdly, and as a consequence of the previous point, START can handle symbolic data of varying length when interpreted as statements under a grammar; in fact, this paper demonstrates an analysis of multi-categorical datasets of depth 2 due to the nature of the CMT data available, but categorical datasets of arbitrary depth can be analyzed with START when treating categories as themselves non-terminal symbols with production rules mapping to other sets of categories. This can be interpreted as processing hierarchical symbolic databases where individual fields can themselves link to other symbolic database tables.

6.2. *Biological Interest and Plausibility of Derived Clusters*

When the START vigilance parameter was set to $\rho = 0.6$ (Figure 4), we obtained nine clusters (Table 3). Cluster 4, the largest, had 53 members, and clusters 6 and 9 were singleton clusters. The fact that cluster 4 is large is not surprising since most cases of CMT are similar and have similar core symptoms of weakness, sensory loss, hyporeflexia, orthopedic abnormalities, atrophy, and gait abnormalities in common [40]. Although it is usual to differentiate clinically between axonal forms (involving the neuron axon) and demyelinating forms (involving the myelin sheath of the axon) of CMT, it is not surprising that we did not find axonal and demyelinating clusters of CMT since we did not input electromyographic data into the clustering algorithm. The finding of small clusters of CMT variants with auditory, hypertonic, or cognitive phenotypes is interesting and plausible biologically and is consistent with clinical observations.

The clusters differed in inheritance (Table 3) in biologically plausible ways and consistent with clinical practice. Since each variant of CMT was due to a gene mutation and since each gene coded for a unique protein, protein weight, protein length, protein configuration (motifs and domains), protein involvement in biological processes, protein molecular function, and protein locations could be examined for each CMT cluster and compared to the observed phenotype (Figures 6, 7, 8, 9, 10, 11, 12, and 13, and Table 3). Although these observations are intriguing, they do not offer a precise path to connect protein function, location, and process to the neurological phenotype in CMT. As an example of explainable AI [48], the SHAP plots in Figure 13 provide biologically plausible explanations for how START depended on certain features to form clusters.

6.3. *Limitations*

One limitation of this work is that START is used to cluster a small biomedical dataset without ground truth labeling. Although the diagnosis of each row (CMT disease variant) is known, cluster memberships for the dataset as a whole are unknown. As a result, this work cannot contain an analysis of either truth in cluster membership and structure or performance of START with respect to such a ground truth.

Another limitation of this work is that all available features are used as inputs to the START clustering algorithm. A separate work is warranted to study how withholding some of the features as meta-features would allow potentially interesting cluster composition analyses.

6.4. *Future Work*

This paper demonstrates that START can work with data from a knowledge graph or ontology when flattened into a rectangular file. Alternatively, knowledge graphs and ontologies can be converted into triplets as subject-object-predicate triplets, which retains the underlying graph architecture. In the future, we plan to determine whether START can successfully cluster these triplets derived from knowledge graphs into meaningful clusters.

Additional future work includes a comparison of START cluster results with other standard clustering algorithms, evaluation of START clustering on large multi-categorical data sets with a known ground truth cluster membership, and further experiments on data sets in which some features are withheld from input and retained as meta-features for post-clustering analysis.

7. Conclusion

This work introduces the START algorithm and for the clustering of symbolic data with arbitrary length statements. This work also introduces dual-vigilance and distributed dual-vigilance variants of START along with a supervised modification for each. Because START is designed for symbolic datasets, it is naturally suited for the clustering of both categorical and multi-categorical datasets where each sample feature may realize multiple values. This multi-categorical clustering capability is demonstrated on a curated biomedical dataset of Charcot-Marie-Tooth disease variants and their disease-gene attributes, such as disease phenotypes and protein molecular functions.

Author Contributions:

Conceptualization, Daniel Hier, Sasha Petrenko, and Donald Wunsch II; methodology, Sasha Petrenko and Daniel Hier; software, Sasha Petrenko and Daniel Hier; validation, Sasha Petrenko; formal analysis, Sasha Petrenko and Daniel Hier; investigation, Sasha Petrenko and Daniel Hier; resources, Daniel Hier and Donald Wunsch II; data curation, Daniel Hier; writing—original draft preparation, Sasha Petrenko and Daniel Hier; writing—review and editing, Sasha Petrenko, Daniel Hier, Tayo Obafemi-Ajayi, Mary Bone, Erik J. Timpson, William E. Marsh, Michael Speight, and Donald Wunsch II; visualization, Sasha Petrenko and Daniel Hier; supervision, Donald Wunsch II; project administration, Donald Wunsch II and Erik J. Timpson; funding acquisition, Sasha Petrenko and Donald Wunsch II. All authors have read and agreed to the published version of the manuscript.

Funding:

This work is funded by the Department of Energy's Kansas City National Security Campus, operated by Honeywell Federal Manufacturing & Technologies, LLC, under contract number DE-NA0002839.

Data Availability Statement:

Disease-protein datasets are gathered from the openly available Online Mendelian Inheritance in Man (OMIM) knowledge base [49]. All data, code, and documentation related to the experiments outlined in this paper are contained in a version-archived repository [50].

Conflicts of Interest:

The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ART	Adaptive Resonance Theory
BNF	Bachus-Naur Form
CFG	Context-Free Grammar
CMT	Charcot-Marie-Tooth disease
DDVFA	Distributed Dual-Vigilance FuzzyART
DDV-START	Distributed Dual-Vigilance Symbolic Tree Adaptive Resonance Theory
DVFA	Dual-Vigilance FuzzyART
DV-START	Dual-Vigilance Symbolic Tree Adaptive Resonance Theory
EBNF	Extended Bachus-Naur Form
F1	ART Feature Input Layer (Field 1)
F2	ART Category Representation Layer (Field 2)
HAC	Hierarchical Agglomerative Clustering
L2	Lifelong Learning
ML	Machine Learning
PMF	Probability Mass Function
START	Symbolic Tree ART
WTA	Winner-Take-All

Appendix A Charcot-Marie-Tooth Dataset Grammar

An analysis of the Charcot-Marie-Tooth (CMT) dataset *a posteriori* demonstrates the process used in this article for interpreting tabular multi-categorical data as statements sampled from a context-free grammar that can be expressed as set of EBNF production rules, which can be seen in Grammar Listing 1. Gene-protein disease data is gathered for 81 variants of CMT with categorical attributes (Table 2). Categories such as phenotype are subsumed where hierarchically relevant to reduce attribute feature dimensionality (e.g., variants of "pain" symptomology are subsumed to one feature belonging to the "phenotype" attribute). This process results in a 81-row flat file dataset of features with multi-categorical attributes represented as piped entries for each disease variant, including attributes with missing entries.

Listing 1: Formal grammar for parsing Charcot-Marie-Tooth disease-protein flat-file data. EBNF syntax is used for production rules with the exception of the regular expression symbol '+', which is used to denote one or more occurrences of the preceding symbol. Statements are composed of a series of one or more categorical attributes, all of which are listed in the non-terminal symbol <attribute>. When an attribute is missing or otherwise unknown for a CMT variant, then it is not included in the parsed syntax tree and handled accordingly by START. The production rules for two notable multi-category attributes, <*phenotype*> and <*biologic_process*>, are listed to demonstrate how statements formulated from CMT disease variant entries illustrate how a gene can be associated with multiple phenotypes and biologic processes. Other multi-category attributes are not listed for brevity.

```
⟨S⟩ ::= ⟨attribute⟩+ ;

⟨attribute⟩ ::= ⟨num⟩ | ⟨gene_location⟩ | ⟨disease⟩ | ⟨disease_MIM⟩ | ⟨gene⟩ | ⟨gene_MIM⟩
    | ⟨inheritance⟩+ | ⟨protein⟩ | ⟨uniprot⟩ | ⟨chromosome⟩ | ⟨chromosome_location⟩ |
    ⟨protein_class⟩+ | ⟨biologic_process⟩+ | ⟨molecular_function⟩+ | ⟨disease_involvement⟩+
    | ⟨MW⟩ | ⟨domain⟩+ | ⟨motif⟩+ | ⟨protein_location⟩+ | ⟨length⟩ | ⟨disease_MIM2⟩ |
    ⟨phenotype⟩+ | ⟨weight_tag⟩ | ⟨length_tag⟩ ;

⟨phenotype⟩ ::= 'ataxia' | 'atrophy' | 'auditory' | 'autonomic' | 'behavior' |
    'cognitive' | 'cranial_nerve' | 'deformity' | 'dystonia' | 'gait' | 'hyperkinesia'
    | 'hyperreflexia' | 'hypertonia' | 'hypertrophy' | 'hyporeflexia' | 'hypotonia' |
    'muscle' | 'pain' | 'seizure' | 'sensory' | 'sleep' | 'speech' | 'tremor' | 'visual' |
    'weakness' ;

⟨biologic_process⟩ ::= 'Apoptosis' | 'Mitosis' | 'Lipid_metabolism' | 'Symport'
    | 'Ubl_conjugation_pathway' | 'Glycolysis' | 'Glucose_metabolism'
    | 'Ion_transport' | 'Unfolded_protein_response' | 'Cell_division'
    | 'DNA_repair' | 'Cell_adhesion' | 'Notch_signaling_pathway' |
    'Protein_biosynthesis' | 'Stress_response' | 'Endocytosis' | 'Transcription'
    | 'Sodium_potassium_transport' | 'Transcription_regulation' |
    'Fatty_acid_metabolism' | 'Host_virus_interaction' | 'Antiviral_defense'
    | 'Lipid_degradation' | 'Autophagy' | 'Sodium_transport' | 'Immunity' | 'none'
    | 'Protein_transport' | 'Nucleotide_biosynthesis' | 'Calcium_transport'
    | 'Transport' | 'Phagocytosis' | 'Inflammatory_response' | 'DNA_damage'
    | 'Potassium_transport' | 'Carbohydrate_metabolism' | 'Cell_cycle' |
    'Innate_immunity' ;
```

References

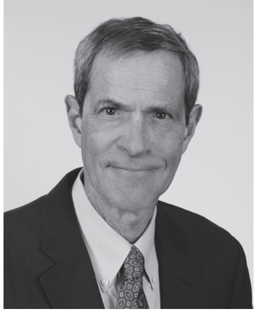
1. Robinson, P.N. Deep phenotyping for precision medicine. *Human mutation* **2012**, *33*, 777–780.
2. Sonawane, A.R.; Weiss, S.T.; Glass, K.; Sharma, A. Network medicine in the age of biomedical big data. *Frontiers in Genetics* **2019**, *10*, 294.
3. Collins, F.S.; Varmus, H. A new initiative on precision medicine. *New England journal of medicine* **2015**, *372*, 793–795.
4. Carrasco-Ramiro, F.; Peiró-Pastor, R.; Aguado, B. Human genomics projects and precision medicine. *Gene therapy* **2017**, *24*, 551–561.
5. Phillips, C.J. Precision medicine and its imprecise history. *Harvard Data Science Review* **2020**, *2*.
6. Ginsburg, G.S.; Phillips, K.A. Precision medicine: from science to value. *Health affairs* **2018**, *37*, 694–701.
7. Polster, A.; Cvijovic, M. Network medicine: Facilitating a new view on Complex Diseases. *Frontiers in Bioinformatics* **2023**, *3*, 47.
8. Xu, R.; Wunsch, D.C. *Clustering*; John Wiley & Sons, Inc.: Hoboken, New Jersey, 2009; pp. 1–21.
9. Gowda, K.; Diday, E. Symbolic clustering using a new similarity measure. *IEEE Transactions on Systems, Man, and Cybernetics* **1992**, *22*, 368–378. https://doi.org/10.1109/21.148412.
10. Chidananda Gowda, K.; Diday, E. Symbolic clustering using a new dissimilarity measure. *Pattern Recognition* **1991**, *24*, 567–578. https://doi.org/https://doi.org/10.1016/0031-3203(91)90022-W.
11. Brito da Silva, L.E.; Elnabarawy, I.; Wunsch, D.C. A Survey of Adaptive Resonance Theory Neural Network Models for Engineering Applications. *Neural Networks* **2019**, *120*, 167–203. https://doi.org/10.1016/j.neunet.2019.09.012.

12. Meuth, R.J. Adaptive multi-vehicle mission planning for search area coverage. PhD thesis, Missouri University of Science and Technology, 2007.
13. Brito da Silva, L.E.; Elnabarawy, I.; Wunsch, D.C.; Enzo, L.; Elnabarawy, I.; Wunsch, D.C.; Brito da Silva, L.E.; Elnabarawy, I.; Wunsch, D.C.; Enzo, L.; et al. Dual vigilance fuzzy adaptive resonance theory. *Neural Networks* **2019**, *109*, 1–5. https://doi.org/10.1016/j.neunet.2018.09.015.
14. Brito da Silva, L.E.; Elnabarawy, I.; Wunsch, D.C. Distributed dual vigilance fuzzy adaptive resonance theory learns online, retrieves arbitrarily-shaped clusters, and mitigates order dependence. *Neural Networks* **2020**, *121*, 208–228. https://doi.org/10.1016/j.neunet.2019.08.033.
15. Grossberg, S. How Does a Brain Build a Cognitive Code? *Psychological Review* **1980**, *87*, 1–51. https://doi.org/10.1037/0033-295X.87.1.1.
16. Grossberg, S. Adaptive Resonance Theory: How a brain learns to consciously attend, learn, and recognize a changing world. *Neural Networks* **2013**, *37*, 1–47. https://doi.org/10.1016/j.neunet.2012.09.017.
17. Petrenko, S.; Wunsch, D.C. AdaptiveResonance.jl: A Julia Implementation of Adaptive Resonance Theory (ART) Algorithms. *Journal of Open Source Software* **2022**, *7*, 3671. https://doi.org/10.21105/joss.03671.
18. Park, G.M.; Kim, J.H. Deep Adaptive Resonance Theory for learning biologically inspired episodic memory. In Proceedings of the 2016 International Joint Conference on Neural Networks (IJCNN), 2016, pp. 5174–5180. https://doi.org/10.1109/IJCNN.2016.7727883.
19. Grossberg, S.; Huang, T.R. ARTSCENE: A neural system for natural scene classification. *Journal of vision* **2009**, *9*, 6–6. https://doi.org/https://doi.org/10.1167/9.4.6.
20. Petrenko, S.; Brna, A.; Aguilar-Simon, M.; Wunsch, D. Lifelong Context Recognition via Online Deep Feature Clustering **2023**. https://doi.org/10.36227/techrxiv.23653737.v1.
21. Petrenko, S.; Wunsch, D.C. ClusterValidityIndices.jl: Batch and Incremental Metrics for Unsupervised Learning. *Journal of Open Source Software* **2022**, *7*, 3527. https://doi.org/10.21105/joss.03527.
22. da Silva, L.E.B.; Rayapati, N.; Wunsch, D.C. Incremental Cluster Validity Index-Guided Online Learning for Performance and Robustness to Presentation Order. *IEEE Transactions on Neural Networks and Learning Systems* **2022**, pp. 1–0. https://doi.org/10.1109/TNNLS.2022.3212345.
23. da Silva, L.E.B.; Rayapati, N.; Wunsch, D.C. iCVI-ARTMAP: Using Incremental Cluster Validity Indices and Adaptive Resonance Theory Reset Mechanism to Accelerate Validation and Achieve Multiprototype Unsupervised Representations. *IEEE Transactions on Neural Networks and Learning Systems* **2022**, pp. 1–14. https://doi.org/10.1109/TNNLS.2022.3160381.
24. Yelugam, R.; Brito da Silva, L.E.; Wunsch, D.C. TopoBARTMAP: Biclustering ARTMAP with or without Topological Methods in a Blood Cancer Case Study. In Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN), 2020, pp. 1–8. https://doi.org/10.1109/IJCNN48605.2020.9206684.
25. Yelugam, R.; Brito da Silva, L.E.; Wunsch II, D.C. Topological biclustering ARTMAP for identifying within bicluster relationships. *Neural Networks* **2023**, *160*, 34–49. https://doi.org/https://doi.org/10.1016/j.neunet.2022.12.010.
26. Chen, Z.; Liu, B. *Lifelong machine learning*; Morgan & Claypool Publishers, 2018; pp. 1–207.
27. Kudithipudi, D.; Aguilar-Simon, M.; Babb, J.; Bazhenov, M.; Blackiston, D.; Bongard, J.; Brna, A.P.; Chakravarthi Raja, S.; Cheney, N.; Clune, J.; et al. Biological underpinnings for lifelong learning machines. *Nature Machine Intelligence* **2022**, *4*, 196–210. https://doi.org/10.1038/s42256-022-00452-0.
28. Carpenter, G.A. Distributed Learning, Recognition, and Prediction by ART and ARTMAP Neural Networks. *Neural Networks* **1997**, *10*, 1473–1494. https://doi.org/https://doi.org/10.1016/S0893-6080(97)00004-X.
29. Wolpert, D.; Macready, W. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation* **1997**, *1*, 67–82. https://doi.org/10.1109/4235.585893.
30. Hester, J.R.; Shinan, E. Lerche: Generating data file processors in Julia from EBNF grammars. *Journal of Open Source Software* **2021**, *6*, 3497. https://doi.org/10.21105/joss.03497.
31. Chomsky, N. *Syntactic structures*; Syntactic structures., Mouton: Oxford, England, 1957.
32. Carpenter, G.A.; Grossberg, S.; Reynolds, J.H. ARTMAP: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network. In Proceedings of the IEEE Conference on Neural Networks for Ocean Engineering, 1991, pp. 341–342. https://doi.org/10.1016/0893-6080(91)90012-T.
33. Kasuba, T. Simplified Fuzzy ARTMAP, AI Expert, 1993.
34. Fisher, R.A. Iris. UCI Machine Learning Repository, 1988. DOI: https://doi.org/10.24432/C56C76.
35. Mushroom. UCI Machine Learning Repository, 1987. DOI: https://doi.org/10.24432/C5959T.
36. Lane, T. UNIX User Data. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5302K.
37. Ilc, N. Datasets package, 2013.
38. Fränti, P.; Sieranoja, S. K-means properties on six clustering benchmark datasets, 2018.
39. Ahmad, A.S.; Mayya, A.M. A new tool to predict lung cancer based on risk factors. *Heliyon* **2020**, *6*, e03402.
40. Rossor, A.M.; Polke, J.M.; Houlden, H.; Reilly, M.M. Clinical implications of genetic advances in Charcot–Marie–Tooth disease. *Nature Reviews Neurology* **2013**, *9*, 562–571.
41. Amberger, J.S.; Bocchini, C.A.; Scott, A.F.; Hamosh, A. OMIM.org: leveraging knowledge across phenotype–gene relationships. *Nucleic acids research* **2019**, *47*, D1038–D1043.

42. Köhler, S.; Gargano, M.; Matentzoglu, N.; Carmody, L.C.; Lewis-Smith, D.; Vasilevsky, N.A.; Danis, D.; Balagura, G.; Baynam, G.; Brower, A.M.; et al. The human phenotype ontology in 2021. *Nucleic acids research* **2021**, *49*, D1207–D1217.
43. The UniProt Consortium. UniProt: the Universal Protein Knowledgebase in 2023. *Nucleic Acids Research* **2023**, *51*, D523–D531.
44. Keshava Prasad, T.; Goel, R.; Kandasamy, K.; Keerthikumar, S.; Kumar, S.; Mathivanan, S.; Telikicherla, D.; Raju, R.; Shafreen, B.; Venugopal, A.; et al. Human protein reference database—2009 update. *Nucleic acids research* **2009**, *37*, D767–D772.
45. Demšar, J.; Curk, T.; Erjavec, A.; Cˇ Gorup.; Hocˇevar, T.; Milutinovicˇ, M.; Možina, M.; Polajnar, M.; Toplak, M.; Staricˇ, A.; et al. Orange: Data Mining Toolbox in Python. *Journal of Machine Learning Research* **2013**, *14*, 2349–2353. https://doi.org/10.5555/2567709.2567736.
46. Lundberg, S.M.; Erion, G.; Chen, H.; DeGrave, A.; Prutkin, J.M.; Nair, B.; Katz, R.; Himmelfarb, J.; Bansal, N.; Lee, S.I. From local explanations to global understanding with explainable AI for trees. *Nature machine intelligence* **2020**, *2*, 56–67.
47. Robinson, P.N.; Mungall, C.J.; Haendel, M. Capturing phenotypes for precision medicine. *Molecular Case Studies* **2015**, *1*, a000372.
48. Gunning, D.; Stefik, M.; Choi, J.; Miller, T.; Stumpf, S.; Yang, G.Z. XAI—Explainable artificial intelligence. *Science robotics* **2019**, *4*, eaay7120.
49. Hamosh, A.; Scott, A.F.; Amberger, J.S.; Bocchini, C.A.; McKusick, V.A. Online Mendelian Inheritance in Man (OMIM), a knowledgebase of human genes and genetic disorders. *Nucleic Acids Research* **2005**, *33*, D514–D517, [https://academic.oup.com/nar/article-pdf/33/suppl_1/D514/7621651/gki033.pdf]. https://doi.org/10.1093/nar/gki033.
50. Petrenko, S. AP6YC/OAR, 2023. https://doi.org/10.5281/zenodo.8117081.

Short Biography of Authors

Sasha Petrenko is a Ph.D. candidate of Computer Engineering in the Applied Computational Intelligence Laboratory of Missouri S&T. His research focus is in lifelong machine learning, clustering, cluster validation, and adaptive resonance theory neural networks. He holds both a B.S. and M.S. in Aerospace Engineering from Missouri S&T, having conducted research in spacecraft navigation and multitarget tracking with the Missouri S&T Mechanical and Aerospace Engineering (MAE) Applied Research in the Estimation of Uncertain Systems (AREUS) laboratory.
He has worked for Sandia National Laboratories and the Ball Aerospace & Technologies Corp. in Albuquerque, NM.
Sasha is a natural born U.S. citizen from Columbia, MO. He is a triathlete, performs jazz guitar and piano, and is fluent in both French and Russian.



Daniel Hier is Professor Emeritus of Neurology at the University of Illinois in Chicago and Adjunct Professor at the Kummer Institute of Artificial Intelligence and Autonomous Systems at Missouri University of Science and Technology. He holds his MD from Harvard Medical School and did his neurology training at Massachusetts General Hospital. His research interests are at the intersection of neurology and computer sciences. He is currently on the editorial boards of *Frontiers in Digital Health* and *BMC Biomedical Informatics and Medical Decision Making*.

Tayo Obafemi-Ajayi is an Associate Professor of Electrical Engineering in the Engineering Program at MSU. She received her B.S. in Electrical Engineering, M.S. in Electrical Engineering, and a Ph.D. in Computer Science from the Illinois Institute of Technology. She is the director of the Computational Learning Systems lab at MSU. Her research interests include machine learning, data mining, and biomedical informatics. She currently serves as the chair of IEEE Computational Intelligence Society (CIS) Bioengineering and Bioinformatics Technical Committee (BBTC) as well as a Technical Representative for IEEE Engineers in Medicine and Biology Society (EMBS) Administrative Committee.

Mary Bone has spent a majority of her over 20-year career working with the US Department of Defense; she is currently a consultant for NASA and Associate Professor II at University of Southeastern Norway. She received her doctorate in Systems Engineering from Stevens Institute of Technology and has a bachelors in Aerospace Engineering from Missouri University of Science and Technology. Her research interests include advancing Systems Engineering through Digital Engineering, System Architectures, Ontologies, and Artificial Intelligence specifically utilizing Semantic Web Technology to transform systems engineering. 616

Erik Timpson is a Principal Electrical Engineer in Quality Assurance Engineering (QAE) at Honeywell Federal Manufacturing & Technologies (FM&T), which manages and operates the Department of Energy's Kansas City National Security Campus (KCNSC). The Kansas City National Security Campus provides diverse engineering, manufacturing and secure sourcing services for national security. His role in QAE is leading Quality Technology Planning, Data Science to ensure the ability to utilize Machine learning and Artificial Intelligence for Quality Improvements, and Staff Development. Prior to this role, he served as a Product Development Engineer in the telecommunications industry. His education includes: a B.S. Electrical Engineering with honors and minors in Math, Physics, and Biology from the University of Missouri, Rolla; M.S. Electrical and Computer Engineering from the University of Missouri, Kansas City; Ph.D. in Electrical and Computer Engineering from University of Missouri, Columbia. He is Six Sigma Black Belt trained. He has authored/coauthored more than 20 peer reviewed journal articles, conference proceedings, or technical magazine column articles. He has 12 patents granted.
He is a member of Institute of Electrical and Electronic Engineers (IEEE), Eta Kappa Nu, Kappa Nu Epsilon, and NCSL International. He is a Professional Engineer licensed in the state of Missouri. He won the Henry Pusey Award for best paper in shock and vibration in 2015. 617

Will Marsh is a Quality Assurance Engineer at Honeywell Federal Manufacturing and Technologies. His work includes root cause analysis, Six Sigma implementation, and process characterization and control. His previous work includes publication of peer reviewed articles while researching stroke rehabilitation at Moss Rehab Research Institute in Elkins Park, Pennsylvania. Will Marsh earned a bachelor's degree in Exercise Science from the University of Kansas, and a bachelor's degree in Biomedical Engineering with honors from Wichita State University. Will has earned a Lean Six Sigma Green Belt. 618

Michael Speight is a Lead Chemical Engineer in Quality Assurance Engineering (QAE) at Honeywell Federal Manufacturing & Technologies (FM&T), which manages and operates the Department of Energy's Kansas City National Security Campus (KCNSC). His role in QAE is leading the development of optimal production strategies through the implementation of Lean and Six Sigma methodologies during product development. He also works to leverage advanced data analytics and machine learning capabilities to drive better decision-making and maximize production efficiency.
Prior to this role, Michael served as a Launch Project Engineer in the automotive rubber and plastics industry where he researched and developed new production processes and led scale up efforts for full production implementation. Process Improvement and heightening performance capabilities through creative thinking have always been a passion for Michael. While earning his bachelor's degree in chemical engineering from The University of Michigan – Ann Arbor, he studied engineering statistics and problem solving. He now holds a Lean Six Sigma Black Belt certification from The International Association for Six Sigma Certification (IASSC). 619

Donald C. Wunsch II is the Mary Finley Missouri Distinguished Professor and the inaugural Director, Kummer Institute Center for Artificial Intelligence and Autonomous Systems at Missouri University of Science and Technology (Missouri S&T). He is also Director of the Applied Computational Intelligence Laboratory. Earlier employers were: Texas Tech University, Boeing, Rockwell International and International Laser Systems. His education includes: Ph.D., Electrical Engineering - University of Washington (Seattle), Executive MBA - Washington University in St. Louis, M.S., Applied Mathematics - University of Washington (Seattle), B.S., Applied Mathematics - University of New Mexico (Albuquerque), and Jesuit Core Honors Program, Seattle University. He also completed the Kellogg Executive Scholar graduate certificate program in Nonprofit Management at Northwestern University. Key research contributions are real-time design of Unsupervised and Reinforcement Learning and their applications. He is an IEEE Fellow and previous International Neural Networks Society (INNS) President, INNS Fellow, NSF CAREER Awardee, and recipient of the 2015 INNS Gabor Award, 2019 INNS Ada Lovelace Service Award, and 2023 IEEE Neural Networks Pioneer Award. He served as IJCNN General Chair, and on several Boards, including the St. Patrick's School Board, Missouri S&T Newman Center Board, IEEE Neural Networks Council, INNS, and the University of Missouri Bioinformatics Consortium, chaired the Missouri S&T Information Technology and Computing Committee as well as the Student Design and Experiential Learning Center Board. He served as Interim Director of the Missouri S&T Intelligent Systems Center and as a Program Director at the National Science Foundation. He has produced 23 Ph.D. recipients in Computer Engineering, Electrical Engineering, Systems Engineering and Computer Science.

Notice: This manuscript has been authored by Honeywell Federal Manufacturing & Technologies, LLC under Contract No. DE-NA-0002839 with the U.S. Department of Energy Honeywell FM&T Proprietary National Nuclear Security Administration. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a nonexclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes.

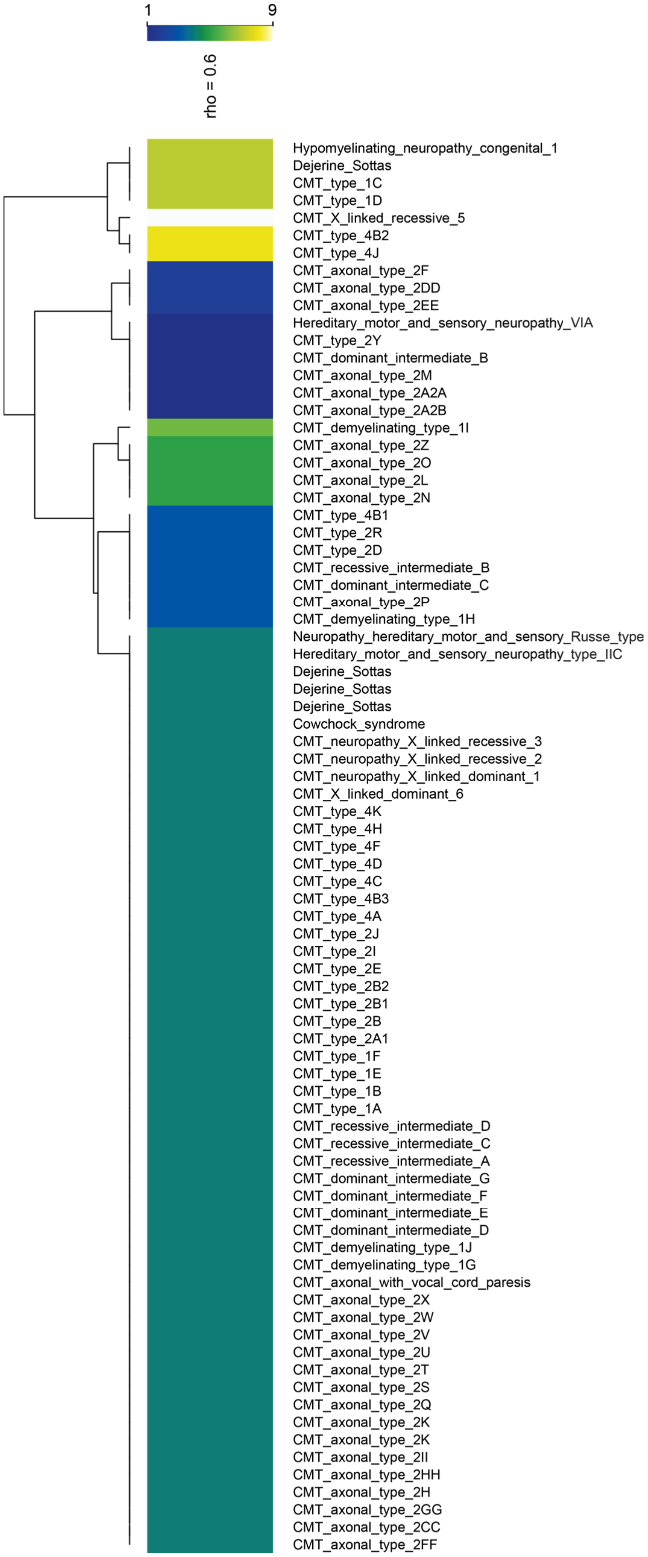


Figure 5. With $\rho = 0.6$, clustering by START yielded nine clusters from 81 variants of CMT. Each cluster is a different color on the heat map. Order of clusters on heat map is **7**, **9**, **8**, **2**, **1**, **6**, **5**, **3**, **4** with ordering by Euclidean distance between cluster centroids [45]. The largest cluster is **4** (dark green), with 53 members. Singleton clusters are **9** (white) and **6** (pea green). A shortened variant name is shown in the right margin. Dejerine Sottas disease appears four times in the heat map because it is caused by four distinct mutations in the MPZ, PMP22, PRX, and EGR2 genes.

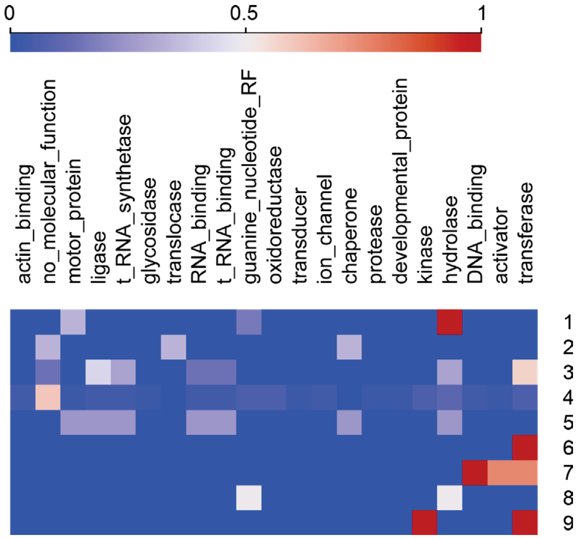


Figure 6. Heat map of molecular function for proteins in CMT clusters. Kinase function is associated with cluster 9, hydrolase function with clusters 1 and 8, DNA binding with cluster 7, activator function with cluster 7, and transferase function with cluster 9.

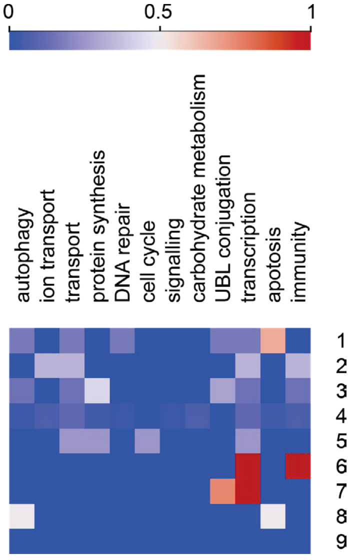


Figure 7. Heat map of biological process for proteins by CMT cluster. Cluster 1 is apoptosis, Cluster 8 is autophagy and apoptosis, Cluster 3 is protein synthesis, Cluster 6 is transcription and immunity, and Cluster 7 is UBL protein conjugation and transcription.

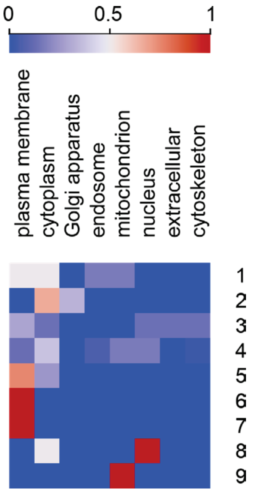


Figure 8. Heat map of protein locations by CMT cluster. Cluster 2 is cytoplasm, clusters 5, 6, and 7 are plasma membrane, cluster 8 is nucleus, and cluster 9 is mitochondrion.

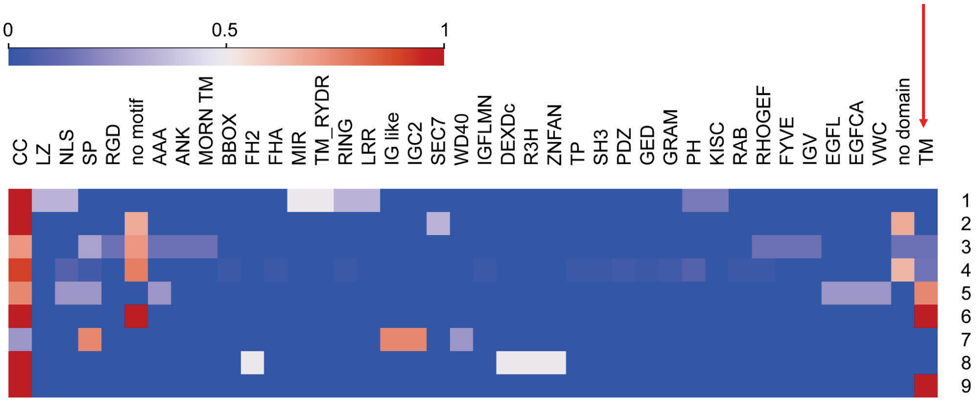


Figure 9. Heat map showing protein motifs and domains by CMT cluster. Motifs and domains are characteristics of configurations of the amino acid chains that make up proteins and are often associated with a specific function. Note the over-representation of the transmembrane (TM) domains in clusters 5, 6, and 9 (red arrow). The CC motif is found in most proteins except for cluster 7.

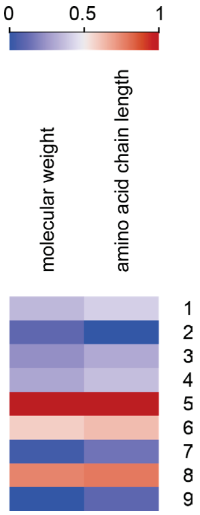
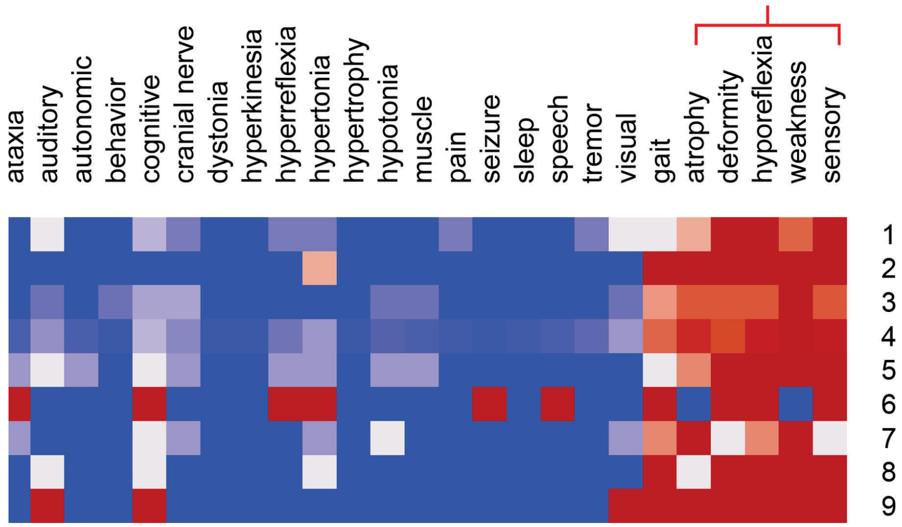


Figure 10. Heat map of molecular weights and amino acid chain lengths for proteins for CMT clusters.



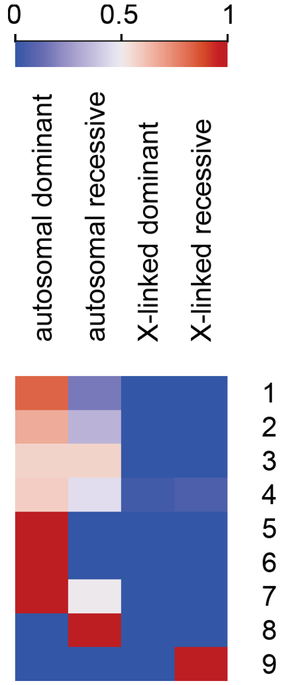


Figure 11. Phenotype scores for each of the nine clusters for the 81 variants of CMT. Scores have been normalized to the interval [0, 1] where 1 indicates 100% and 0 indicates 0%. Note, as expected, that gait, atrophy, deformity, hyporeflexia, weakness, and sensory loss are common features in most cases (red bracket). Cluster 6 with one case and Cluster 9 with one case are different because they manifest auditory and cognitive symptoms (Cluster 9) or ataxia, cognitive, hyperreflexia, hypertonia, seizures, and speech symptoms (Cluster 6). Cluster 6 is also of interest because it lacks weakness and atrophy, two of the core symptoms of CMT. Cluster 2 (3 cases) is also interesting because subjects have hypertonia. Cluster 4, with 53 cases, is the most common pattern and shows a typical phenotype of gait, atrophy, deformity, hyporeflexia, weakness, and sensory symptoms, which is characteristic of CMT.

Figure 12. Modes of inheritance for the nine CMT clusters. Cluster 8 is largely autosomal recessive. Cluster 9 is X-linked recessive. Clusters 5, 6, and 7 are autosomal dominant inheritance.

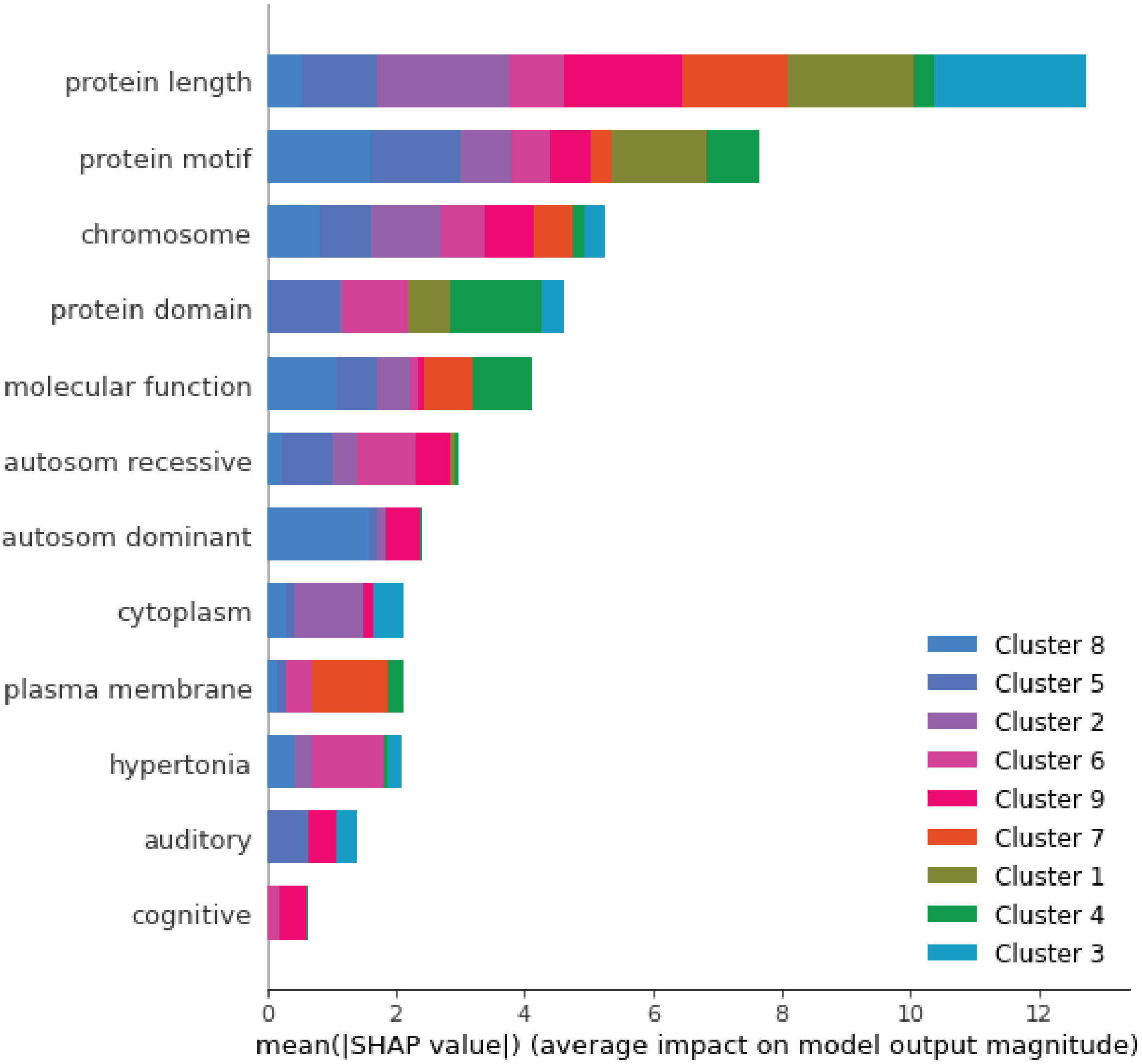


Figure 13. SHAP cluster summary plot for the 9 clusters derived from CMT dataset with $\rho = 0.6$. The SHAP plot shows which features contributed the most to the cluster configuration by cluster. Important features were protein length, chromosome, mode of inheritance (autosomal dominant and recessive), protein location (cytoplasm and plasma membrane), and certain phenotypes (auditory, cognitive, and hypertonia). The domain expert rated these features as highly biologically plausible. SHAP plots were created using the method of Lundberg et al. [46].