Finding Interpretable Class-Specific Patterns through Efficient Neural Search

Nils Philipp Walter[1], **Jonas Fischer**[2], **Jilles Vreeken**[1]

[1]CISPA Helmholtz Center for Information Security,
[2]Harvard T.H. Chan School of Public Health, Department of Biostatistics
nils.walter@cispa.de, jfischer@hsph.harvard.edu, jv@cispa.de

Abstract

Discovering patterns in data that best describe the differences between classes allows to hypothesize and reason about class-specific mechanisms. In molecular biology, for example, this bears promise of advancing the understanding of cellular processes differing between tissues or diseases, which could lead to novel treatments. To be useful in practice, methods that tackle the problem of finding such *differential* patterns have to be readily *interpretable* by domain experts, and *scalable* to the extremely high-dimensional data.

In this work, we propose a novel, inherently interpretable binary neural network architecture DIFFNAPS that extracts differential patterns from data. DIFFNAPS is scalable to hundreds of thousands of features and robust to noise, thus overcoming the limitations of current state-of-the-art methods in large-scale applications such as in biology. We show on synthetic and real world data, including three biological applications, that, unlike its competitors, DIFFNAPS consistently yields accurate, succinct, and interpretable class descriptions.

1 Introduction

Machine learning can be broadly categorized into *predictive* and *discovery*-based approaches. *Predictive* tasks, such as object detection, protein folding (Jumper et al. 2021) and fusion reactor control (Degrave et al. 2022), are aimed at maximizing performance. Mastering such a given task often requires learning deep and intricate models from which it is hard up to impossible to understand how it arrived at a decision. In data-driven *discovery*, the goal is to find *interpretable* relations, called *patterns*, in the data that best describe observed classes. That is, the focus is on interpretability rather than maximizing performance. Discovery-based approaches are in especially high demand in biology, where the complex gene-regulatory dynamics and their differences between tissues or across diseases remain unclear, but, when elucidated, can offer new avenues for treatment and prevention. Here, *symbolic* explanations are essential for domain experts, for example, patterns of gene expression that are associated with cancer subtypes, to be able to directly understand and act on these patterns.

Although there exist massive amounts of high-dimensional data, such as genetic human variation or gene expression data, most existing approaches are not applicable as they either do not scale or are limited to pair-wise interactions. Here, we suggest a novel neural network learning approach that follows the paradigm of neuro-symbolic learning: leverage the predictive power of, and efficient frameworks for neural networks, while constraining the models such that learned patterns are fully interpretable. In particular, we learn a modified NN architecture that in the forward pass leverages binary weights and activations to achieve symbolically interpretable intermediate features, while leveraging efficient continuous optimization during backpropagation (Fischer and Vreeken 2021).

To learn patterns that differentiate classes, such as healthy and tumor tissue, we build an architecture that is comprised of both a binary autoencoder and a separate classification head, which we call DIFFNAPS. We propose a multi-task objective to jointly optimize reconstruction and classification, driving learned patterns to differentiate between classes through a bottleneck in the autoencoder (see Fig. 1). We additionally introduce regularizers that improve optimization and emphasize interpretability of learned patterns.

We empirically evaluate DIFFNAPS on synthetic and real-world data, comparing against baseline approaches such as classification trees, but also recent proposals such as rule lists, statistical and compression-based pattern mining, and neuro-symbolic learning. We show that DIFFNAPS faithfully reconstructs patterns relevant for distinguishing between classes, is robust to noise, and easily scales to hundreds of thousands of features, which makes it unique among existing work. We consider three high-dimensional biological applications, including breast cancer genomics, on which DIFFNAPS finds meaningful patterns that hold promise for giving domain experts insight in the drivers of these diseases. We make the code publicly available.[1]

2 Related Work

Finding class-specific descriptions is at the core of discovery-oriented approaches in machine learning and data mining. A text-book example—and still widely used in practice—is the decision tree, which yields an interpretable decision path leading to a classification.

[1]https://eda.rg.cispa.io/prj/diffnaps/

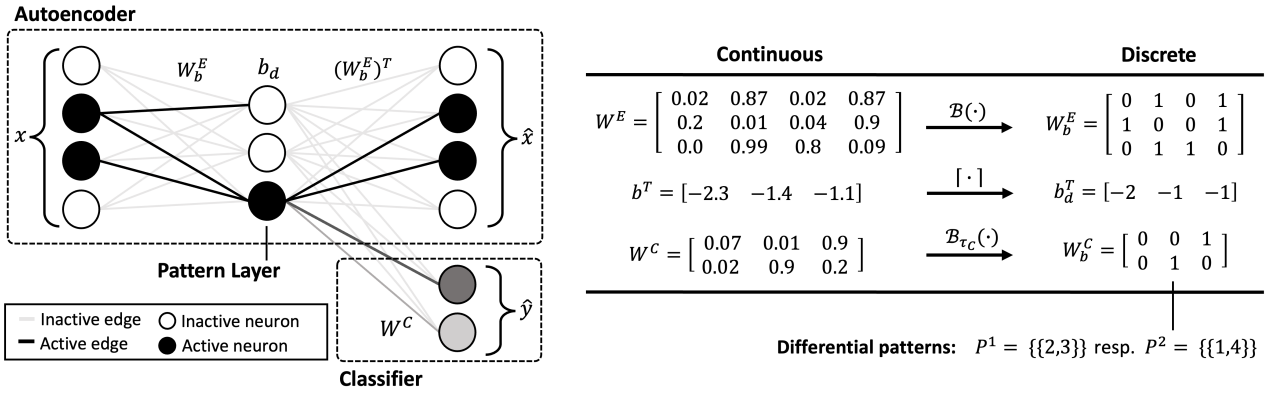


Figure 1: Left: The architecture of DIFFNAPS consists of a binarized autoencoder and a classifier attached to the hidden layer. The neurons in the hidden layer encode patterns and are active if the corresponding pattern is present in the data. Right: The table shows the parameters of DIFFNAPS. In the forward pass, the continuous weights W^E are stochastically binarized (W_b^E), while the classifier weights W^C are kept continuous. The bias b in the hidden layer is ceiled to b_d^T. To extract the differential patterns (bottom right) per class P^1 and P^2, both matrices, W^E and W^C, are deterministically binarized using the thresholds, τ_E and τ_C. A pattern, encoded by a neuron, is given by the index set of all 1 in the corresponding row of the weight matrix W^E. For the differential patterns, the binarized classifier weight matrix functions as a multiplexer to assign patterns to classes.

In data mining, *emerging pattern mining* (Dong and Li 1999; García-Vico et al. 2018) and *subgroup discovery* (Klösgen 1995; Atzmueller 2015) are classic methods that aim to discover the conditions under which the class labels assume an exceptional distribution. Emerging pattern mining seeks to find *every* such condition, which results in extremely many, highly redundant, and mostly spurious results. Subgroup discovery yields the top-k patterns with the strongest association with the target. While this circumvents the pattern explosion, the results are still redundant (Van Leeuwen and Knobbe 2012). In contrast, we are interested in succinct and non-redundant descriptions.

Statistically significant pattern mining (Llinares-López et al. 2015; Pellegrina, Riondato, and Vandin 2019) aims to discover patterns that have statistically significantly different distributions between classes. These methods tend to suffer from the pattern explosion. That is, even on small data they often find tens of thousands redundant patterns, partially due to lack of multiple hypothesis test correction.

Pattern set mining (Bringmann and Zimmermann 2007; Budhathoki and Vreeken 2015; Hedderich et al. 2022) solves this by asking for a non-redundant set of class-specific patterns that together describe the data well. These methods work well on small data, but as they are based on combinatorial-search heuristics that are (at least) quadratic in the number of features, they are mostly inapplicable to high-dimensional data.

Rule-based classification (Lakkaraju, Bach, and Leskovec 2016; Dash, Gunluk, and Wei 2018; Chen and Rudin 2018; Proença and van Leeuwen 2020; Hüllermeier, Fürnkranz, and Loza Mencia 2020; McTavish et al. 2022; Huynh, Fürnkranz, and Beck 2023; Lin et al. 2022) aims to find interpretable classification rules of the form if $X_1 = 1 \wedge X_5 = 1$ then $Y = 0$. While such results are interpretable, these methods primarily focus on *prediction* rather than *description* and, hence, miss out on important details. Additionally, most are based on combinatorial optimization which prevents them from scaling to high-dimensional datasets.

Neuro-symbolic classification (Wang et al. 2020, 2021; Kusters et al. 2022; Dierckx, Veroneze, and Nijssen 2023) has been proposed to overcome these computational limitations. These approaches design neural architectures from which, after training, symbolic classification rules can be extracted. Their optimization aside, in spirit these methods are similar to traditional rule-based classifiers as they focus on classification accuracy rather than complete rule discovery. In contrast, DIFFNAPS combines data reconstruction with classification to discover the human-interpretable explanations relevant for the classes present in the dataset.

3 Method

In this section, we introduce DIFFNAPS, a fully interpretable binary neural network-based approach for finding patterns that describe the differences between classes in (very) high-dimensional data. We start by giving the intuition.

3.1 DIFFNAPS in a Nutshell

Given a binary dataset and corresponding class labels, we seek to find interpretable patterns, which succinctly and differentially describe the partitioning of the dataset induced by the labels. That is, we want to find patterns that are more prevalent in a class than in the rest of the data and, hence, allow us to discriminate between classes.

To this end, we propose DIFFNAPS, a binary neural-network architecture designed to find exactly such interpretable patterns. The architecture consists of a two-layer binary autoencoder, combined with a classification head (see Figure 1). The classification head is a fully connected layer, attached to the hidden layer of the autoencoder.

During the forward pass, we interpret the continuous weights $w_{ij} \in [0, 1]$ in the autoencoder as Bernoulli vari-

ables distributed as $\mathcal{B}(w_{ij})$ and binarize them stochastically. Each neuron performs a dot product between the binary weights and input. The neuron is active if the 1s in the input align with the 1s in the weight vector. Thus the weight vector can be interpreted as a pattern and hence we refer to the hidden layer as the *pattern layer*. Intuitively, the weights are optimized such that the autoencoder—the set of encoded patterns—reconstructs the data well.

To find differential patterns, we need to reward those patterns that are specific for a class. We achieve this by adding a classification head, corresponding to a logistic regression on the pattern layer. That is, we seek to classify samples based on the presence and absence of patterns.

To find a good set of patterns, the network is trained using a multi-task loss. The autoencoder is trained to minimize the reconstruction error, while the classification head is trained to minimize the classification error. As such, the network is driven towards the learning of relevant patterns in the data that are at the same time differential between classes.

3.2 DIFFNAPS in Detail

Next, we discuss DIFFNAPS in detail. We first introduce notation, and then, in turn, discuss the architecture, how to extract differential patterns, how to carry out the forward pass, the multitask loss, and how to backpropagate errors through DIFFNAPS.

Notation We consider labeled binary datasets $(X, Y) \in \{0,1\}^{n \times m} \times \{1, ..., K\}^n$ of n samples, m features and K classes. We write $X_{i,j}$ to refer to the value of the j-th feature of the i-th sample. We denote the partition of the dataset for class k by $X^k = \{X_i \mid Y_i = k\}$.

A pattern p is a subset of feature indices $p \subseteq \{1...m\}$ and represents feature co-occurrences. A row X_i contains a pattern p iff $X_{ij} = 1, \forall j \in p$. The support $supp(p)$ of a pattern p is the number of rows that contain p, and analogue $supp_k(p)$ is the support where additionally $Y_i = k$. We have

$$\mathbb{P}(p \mid k) = \frac{supp_k(p)}{n_k} \quad \text{and} \quad \mathbb{P}(k \mid p) = \frac{supp_k(p)}{supp(p)} \ ,$$

where n_k is the number of samples where $Y_i = k$.

We say a pattern p_k is *differential* for class k if it both has a higher support in X^k than in $X \setminus X^k$, and the probability of class k is highest for records that contain p_k. Formally, iff

$$k = \underset{k' \in \{1...K\}}{\arg\max}\ \mathbb{P}(p_k \mid k') = \underset{k' \in \{1...K\}}{\arg\max}\ \mathbb{P}(k' \mid p_k) \ .$$

Our goal is to find a set P^k of such patterns per class k.

Architecture The architecture of DIFFNAPS consists of a binary autoencoder and a classification head attached to the hidden layer. We graphically depict it in Fig. 1.

The encoding and decoding layers of the autoencoder share a set of continuous weights W^E, which are learned during backpropagation. The forward pass uses a binarized version of this weight matrix W_b^E. A hidden neuron j represents a pattern, and a feature i is part of the pattern corresponding to neuron j, iff $W_b^E[i, j] = 1$. The activation function of the encoder λ_E is a binary step function centered at a learned bias term, which represents how many features need to be present for the neuron to "fire"—i.e. for the pattern to be considered present in the sample. We refer to the hidden layer as the pattern layer.

The decoding layer performs the transposed linear transformation of the encoding layer i.e. $W_b^d = (W_b^E)^T$. Hence, if a neuron is active, the pattern encoded in that neuron is used as a whole for the reconstruction. Consequentially, to achieve a low reconstruction loss, the patterns formed during optimization must succinctly describe the data.

To reward differential patterns, we connect a classifier to the pattern layer with continuous weights W^C that is tasked to predict the label of a sample based on the presence and absence of patterns. The classifier is linear, and, hence, highly interpretable. To extract differential patterns, we binarize weight matrices W^E and W^C by thresholding with τ_e and τ_c, respectively. As described above, the patterns in the pattern layer are given by the index set of all i's such that $W_b^E[i, j] = 1$. The discretized classifier weights allow us to assign patterns to their respective classes. For a formal description of the pattern extraction, we refer to App. A.2.

Forward Pass We denote the size of the hidden dimension of the autoencoder by h and the binary weights of the encoder as $W_b^E \in \{0,1\}^{h \times m}$. We define a linear layer without bias as $f_W(x) = Wx$. For a binary input $x \in \{0,1\}^m$, we compute the activations of the pattern layer as

$$z = f_E(x) = \lambda_E(f_{W_b^E}(x)) \ .$$

where $\lambda_E : \mathbb{R} \to \{0,1\}$ is the binary step function as defined by Fischer and Vreeken (2021). To steer the encoded patterns to be differential rather than merely descriptive, we attach a classifier to the pattern layer. This classifier has continuous weights $W^C \in [0,1]^{K \times h}$ and computes a linear transformation followed by a softmax of the binary hidden activations $\hat{y} = \mathrm{softmax}(f_{W^C}(z))$. That is, its output depends only on the presence or absence of patterns.

To ensure interpretability, we use the transposed encoder weights as weights of the decoder $W_b^D = (W_b^E)^T \in \{0,1\}^{m \times h}$. The reconstruction $\hat{x}$ of the input x is given by

$$\hat{x} = f_D(x) = \lambda_D(f_{W_b^D}(z)) \ ,$$

where λ_D is the activation of the decoder as defined by Fischer and Vreeken (2021), clamping the input to the interval $[0, 1]$ and rounding it to the closest integer.

Objective Function Our objective function consists of four terms: one for the autoencoder, one for the classification, and two regularization terms. To optimize the classifier, we use the cross-entropy loss between the predicted logits $\hat{y}$ and the one-hot encoding of the ground truth label y: $l_c(y, \hat{y}) = \sum_{k=1}^{K} y_k \log(\hat{y}_k)$. As binary tabular data tends to be sparse, i.e., the number of ones #1 and number of zeros #0 are highly unbalanced, we use a sparsity-aware reconstruction loss (Fischer and Vreeken 2021) that weighs the importance of reconstructing a 1 proportional to the sparsity of the data. For a sample $x \in \{0,1\}^m$ and reconstruction $\hat{x} \in \{0,1\}^m$, the reconstruction loss is

$$l_e(x, \hat{x}) = \sum_{j=1}^{m} ((1 - x_j)\alpha + x_j(1 - \alpha))|x_j - \hat{x}_j| \ ,$$

where $\alpha = \frac{\#1s}{\#1s+\#0s}$ is the sparsity of the data.

Our overall goal is to find a succinct description of the classes in terms of class-specific patterns encoded by the neurons in the hidden layer. To promote such patterns, we adapt the $\mathcal{L}_2$-regularizer to penalize long patterns i.e. rows with a lot of 1s. This adapted regularizer is given by

$$r_s(W) = \sum_{i=1}^{m} \left(\sum_{j=1}^{h} W_{i,j} \right)^2 .$$

Instead of considering each weight individually, we sum the rows before squaring them. This penalizes a pattern as a whole by imposing a quadratic cost on the length of the pattern. Hence, the regularizer tilts the optimization to prefer shorter patterns. To further push the weights to a binary solution we employ a W-shaped regularizer (Bai, Wang, and Liberty 2019; Dalleiger and Vreeken 2022), defined as

$$r_b(W) = \sum_{w \in W} \min\{r(w), r(w-1)\} ,$$

$$r(w) = \kappa \|w\|_1 + \lambda \|w\|_2^2 .$$

This regularizer is based on the elastic-net regularizer and the hyperparameters κ and λ specify the trade-off between the ridge and lasso penalty. For $\kappa = \lambda = 1$, the regularizer is depicted in Figure 3 in the Appendix. Compared to r_s, the W-shape regularizer is applied element-wise to push the individual weights towards zero or one.

In the forward pass, we apply stochastic quantization $W_b[i,j] \sim \mathcal{B}(W[i,j])$. If all $W[i,j]$, for $j = \{1...m\}$, have the same value, a sample of a row is binomially distributed with $p = W[i,j]$ and m trials. The expected value is then $mW[i,j]$. Considering a minimum of two features for a neuron to fire, this means that when all $W[i,j]$ drop below $1/m$ the neuron is on expectation 'dead'. To prevent regularizers from zeroing out a neuron by pushing W_{ij} below this threshold, we offset the weights by $-1/m$ before applying the regularizers. For the same reason, we set the gradients for r_s to zero if $\sum_{j=1}^{h} W[i,j] < 1$.

Given the parameters of the network $\theta = \{W^E, W^C\}$ the loss function for a dataset (X, Y) is given by

$$\mathcal{L}(X, Y; \theta) = \sum_{i=1}^{n} l_e(y_i, \hat{y}_i) + \lambda_c \, l_c(x_i, \hat{x}_i) + r_s(W^E) + r_b(\theta) ,$$

where λ_c is a parameter that weighs the classification loss.

Backward Pass We minimize this loss function using gradient descent. For this, we need to compute the partial derivatives with respect to the weights of the network. To be able to pass gradients through step-functions, we use the straight-through-estimator (STE), which is commonly employed in binary neural networks (Bengio, Léonard, and Courville 2013). For a particular layer, g_u denotes the upstream gradient. For the derivatives with respect to the autoencoder, we follow the approach of Fischer and Vreeken (2021). In particular, for encoding layer W^E and input x

$$\frac{\partial f_{W_b^E}}{\partial W^E} := g_u x^\top , \quad \frac{\partial f_{W_b^E}}{dx} := (W^E)^\top g_u .$$

The derivative through the activation function of the decoder λ_D is given by $\frac{\partial \lambda_D}{\partial x} := \mathbb{1} g_u$. For the activation function of the pattern layer, the STE above is inapplicable. In the case that features are wrongly reconstructed, the resulting loss would propagate negative gradients through the STE, even to inactive neurons. Hence, we adapt the *gated* STE, which gates the gradient depending on whether a neuron was active in the forward pass. The derivatives for bias b and input x are

$$\frac{\partial \lambda_E}{\partial b} := \begin{cases} g_u & \text{if } \lambda_E(x) = 1 \\ 0 & \text{if } \lambda_E(x) = 0 \end{cases} ,$$

$$\frac{\partial \lambda_E}{\partial x} := \begin{cases} g_u & \text{if } \lambda_E(x) = 1 \\ \max(0, g_u) & \text{if } \lambda_E(x) = 0 \end{cases} .$$

In quantized neural networks, it has been observed that quantizing the classification layer has a negative impact on performance (Choi et al. 2018; Liu et al. 2018; Hubara et al. 2017). Thus we do not quantize the weights of the classifier during training. Although the classifier is not quantized, the classifications are transparent and interpretable, since the classification head is similar to logistic regression and the weights are constrained to be in the interval $[0, 1]$.

Finally, after a round of backpropagation, all weights are clipped to the interval $[0, 1]$. This enables stochastic binarization for the autoencoder and the classifier for the next forward pass and to transparently interpret the contribution of a pattern to a certain class. We clamp the bias at a maximum of -1, such that at least two features have to be present for a neuron to become active.

This concludes the formal description of DIFFNAPS.

DIFFNAPS in Practice To use DIFFNAPS in practice, we need to choose the number h of hidden neurons and set λ_c.

For medium to high-dimensional data, setting the size of the hidden layer lower than the dimensionality of the data, m, creates an inductive bias towards differential patterns. Since to achieve both a low reconstruction loss and low classification loss, the patterns in the hidden layer have to be predictive, i.e., high $\mathbb{P}(k \mid p_k)$, and due to the bottleneck, the patterns must cover the partition well, i.e., high $\mathbb{P}(p_k \mid k)$.

For low dimensional data, choosing a small hidden layer results in an under-parameterized network that will underfit. Choosing a larger hidden layer, thus having more parameters, outweighs the benefits of the bottleneck.

Parameter λ_c weighs the effect of the reconstruction and classification losses. The magnitude of the reconstruction loss varies strongly among different datasets. In practice, we increase λ_c until the classification error saturates.

4 Experiments

We compare DIFFNAPS five state-of-the art methods on synthetic and real-world data. In particular, we compare to decision trees (CART, Breiman 1984), significant pattern mining (SPUMANTE Pellegrina, Riondato, and Vandin 2019), MDL-based label-descriptive (PREMISE, Hedderich et al. 2022) and classification rule learning (CLASSY, Proença and van Leeuwen 2020), and neuro-symbolic classification rule learning (RLL, Wang et al. 2021).

We additionally considered top-k subgroup discovery (Lemmerich and Becker 2018), difference description (Budhathoki and Vreeken 2015), falling rule lists (Chen and Rudin 2018; Lin et al. 2022), optimal sparse decision trees (McTavish et al. 2022), and class-specific BMF (Hess and Morik 2017), but found these do not scale to, or do not find patterns on non-trivial data.

PREMISE and SPUMANTE consider only binary classes. To allow fair comparison in a multiclass scenario, we run them in a one-versus-all for each class and merge the results.

The hyperparameters for the predictive approaches are tuned based on accuracy on a hold-out set. For SPUMANTE, we used the default parameters given by the authors. We fit the hyperparameters of DIFFNAPS based on our loss function. The experiments for the neural approaches, i.e. DIFFNAPS and RLL, are executed on GPUs. For more on the experimental setup, we refer to Appendix A.3.

4.1 Synthetic Data

To evaluate all methods on data with known ground we first consider synthetic data. We measure success in terms of soft F1 (Hedderich et al. 2022), by which we avoid overly penalizing methods that recover only parts rather than exact matches of ground truth patterns. The formal definition can be found in Appendix A.4. Informally, the soft F1 score does not require strict equality between a discovered pattern p_d and the corresponding ground truth pattern p_g but uses a soft equality, i.e., the Jaccard distance of p_d and p_g.

Data Generation In the experiments below we generate synthetic data as follows. We start with an empty data matrix of n rows and m features. We sample 10 patterns per class, uniformly at random (u.a.r.) across features, drawing their length from $\mathcal{U}(5, 15)$. We sample 20 common patterns u.a.r, but draw their length from $\mathcal{U}(0.01m, 0.025m)$ to maintain the density of the data. Per class, we generate equally many rows. Per row, we plant u.a.r. two common and three class-specific patterns. We then apply both additive noise by flipping ten 0s to 1s, as well as destructive noise by flipping 1s due to a pattern to 0s with a probability of 2.5%. Finally, we assign the class label such that $\mathbb{P}(k \mid p \in P_k) = 0.9$. Unless specified otherwise, we report the average results over five independently drawn datasets.

Scalability in m First, we consider how well DIFFNAPS scales to high dimensional data. We fix the number of classes K to 2, the number of rows n to 10 000, and vary $m \in \{100, 500, 1\text{k}, 5\text{k}, 10\text{k}, 15\text{k}, 20\text{k}, 25\text{k}, 50\text{k}, 100\text{k}\}$. To reduce the overlap across patterns in low-dim. data $m < 1000$, we sample 5 patterns per class and no shared patterns.

We run all methods and report their results in Fig. 2a,b. Except for PREMISE and SPUMANTE, all terminate within 24 hours. PREMISE runs out of time for $m > 20\text{k}$. SPUMANTE runs out of memory for $m < 1\text{k}$ and $m > 10\text{k}$. We see in Fig. 2a that CLASSY is one order of magnitude slower than DIFFNAPS, RLL, and CART all of which perform on par in terms of runtime.

Next, we inspect the average F1-scores, which we show in Fig. 2a. We see that CLASSY, RLL and CART perform poorly, as they recover only small parts of the ground truth patterns, and that SPUMANTE varies in performance due to having to sub-sample the data. PREMISE achieves scores of approx. 0.5 across all m. DIFFNAPS consistently recovers ground truth well across many orders of magnitudes of m. For very high dimensional data, performance slowly deteriorates but still outcompetes the state-of-the-art by a wide margin.

Multi-class Next, we examine how well DIFFNAPS scales to a large number of classes. To this end, we generate data as above, varying the number of classes $K \in \{2, 5, 10, 15, ..., 25, 50\}$, generating 1 000 rows per class, setting $m = 5\,000$. We give the results in Fig. 2c. RLL, CLASSY, CART, and SPUMANTE fail to recover more than a small subset of the ground truth. In addition SPUMANTE runs out of memory for more than 10 classes. PREMISE achieves scores of around 50% up to 20 classes, but fails to terminate for $K \geq 40$, as running in a one-versus-all setting incurs high computational costs. In contrast, DIFFNAPS stably performs best in this setting.

Robustness to Noise Finally, we evaluate how robust methods are to noise. Here, we set $K = 2$, $m = 5\,000$, and $n = 2\,000$. First, we consider additive noise by varying the number of randomly added 1s per row, from 0 to 100. In the interest of space, we postpone the Figure to App. Fig. 5. We find that SPUMANTE rapidly fails to discover meaningful results, and runs out of memory for a = 100. In contrast, DIFFNAPS and PREMISE are robust across varying a, with DIFFNAPS outperforming its competitors by a wide margin.

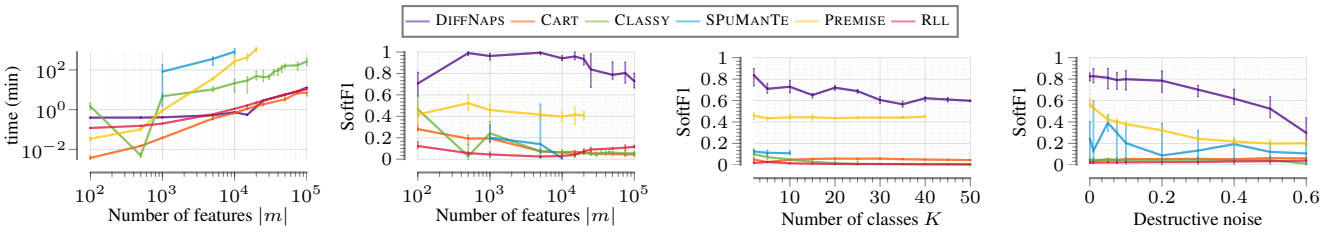
Second, we consider destructive noise by varying the probability of flipping a 1 to a 0, from 0% to 60%. We show the results in Fig. 2d. CART, CLASSY, and RLL all obtain F1 scores of near-zero, SPUMANTE performs slightly better on average but shows a large variance in the performance across repetitions. PREMISE is the best among competitors, but its performance declines rapidly even for small amounts of destructive noise. In contrast, DIFFNAPS is robust, its performance virtually unaffected up to 20% destructive noise, i.e., up to a signal-to-noise ratio of 6dB.

4.2 Real-World Data

Next, we evaluate DIFFNAPS on five biological datasets. We consider phenotypical *Cardio* data (Ulianova 2017), a *Disease* diagnosis (Patil and Rathod 2020) dataset, two high-dimensional binarized gene expression datasets for breast cancer, *BRCA-N* and *BRCA-S*, that we derived from The Cancer Genome Atlas (TCGA) (see App. A.5), and a human genetic variation data set (The 1000 Genomes Project Consortium 2015; Fischer and Vreeken 2020).

We consider the same competitors as before, except RLL as it returns no patterns for any data but *Cardio*. To obtain results with SPUMANTE we had to restrict it to 250 samples for *Cardio*, 4000 for *Disease*, 50 for both *BRCA* datasets. We could not find any setting to make it work on *Genomes*. We report running time for all methods in App. Tab. 2.

Quantitative Results As the ground truth is unknown, we report the number of discovered patterns, their average length, and the area under the curve of what percentage of



(a) Runtime (lower is better) (b) F1-score (higher is better) (c) F1-score (higher is better) (d) F1-score (higher is better)

Figure 2: Scalability. We show runtime (a) and F1-scores (b) when varying m, resp. F1-scores when varying the number of classes K (c) and varying the amount of destructive noise (d). Our method, DiffNaps can confidently handle a large number of features with a negligible increase in runtime, outperforms all state-of-the-art competitors significantly in terms of F1.

				DiffNaps (ours)			Cart			Classy			Premise			SpuMaNTe		
Dataset	n	m	K	$\#P$	$\overline{\lvert P\rvert}$	AUC	$\#P$	$\overline{\lvert P\rvert}$	AUC	$\#P$	$\overline{\lvert P\rvert}$	AUC	$\#P$	$\overline{\lvert P\rvert}$	AUC	$\#P$	$\overline{\lvert P\rvert}$	AUC
Cardio	68k	45	2	14	2	.56	7k	6	**.62**	10	2	.36	28	1	.51	346	4	.56
Disease	5k	131	41	838	2	**.84**	1	2	.00	25	2	.11	187	3	**.84**	2k	3	.39
BRCA-N	222	20k	2	146	9	.91	1	2	.00	3	1	.45	–	–	–	4k	3	**.95**
BRCA-S	187	20k	4	1k	2	**.86**	22	2	.31	2	1	.23	–	–	–	0	0	0
Genomes	2.5k	225k	6	732	7	**.77**	127	4	.46	7	2	.36	–	–	–	–	–	–

Table 1: Real world data. We give the number of samples (n), features (m), and classes (K), and per method the number of discovered patterns ($\#P$), average length of discovered patterns ($\overline{|P|}$) and area under the curve (AUC) of the sensitivity-specificity plot (see App. Fig. 6, Sec. A.5). We aborted experiments taking longer than 24h or running out of memory (–).

the data the patterns cover when we order them by probability of seeing a class given a pattern (see App. A.5). Intuitively, this corresponds to sensitivity (how much do we cover) versus specificity (how specific are patterns for that class). To filter spurious patterns we compute this measure over patterns for which at least $1/k + 0.1$ of their probability mass is assigned to one class. For example, for the binary setting only those patterns that occur at least 60% in the class, where 50% would correspond to independence (random coin flip).

We report basic statistics and results in Tab. 1. We observe that DiffNaps performs well across all datasets, obtaining AUC scores that are either best by a wide margin (*Disease*, *BRCA-S*, *Genomes*) or close second best (*Cardio*, *BRCA-N*). Consistent with our synthetic data study, our competitors yield mixed results; they do not scale to high dimensional data (Premise, SpuMaNTe), or return prohibitively many or unspecific patterns (SpuMaNTe, Cart, resp. Classy).

Regarding the length of discovered patterns, we observe that those by DiffNaps reflect the complexity of the datasets: on *Cardio* and *Disease*, which contain complex, information rich features, it finds smaller patterns, while for the other datasets, that consist of low-level molecular information as features, it finds longer patterns to capture complex relationships. In contrast, Classy generally discovers only few, medium-length patterns across datasets, while Cart recovers more complex relationships that are, however, less descriptive of the classes as measured by the AUC.

Qualitative Results Next, we analyse the results of DiffNaps in detail and show their relevance for biological research on the breast cancer datasets.[2]

Differentiating Breast Cancer and Healthy Tissue Breast cancer (BRCA) is the most common cancer and the leading cause of death from cancer among women in the world (Lukasiewicz et al. 2021). The exact underlying gene regulatory dynamics are actively researched.

We apply DiffNaps on *BRCA-N* and discover 146 differential patterns of gene co-expression for BRCA and adjacent normal tissue. To see if these capture relevant molecular differences, we run a statistical gene set over-representation analysis using KEGG (see App. A.5), a manually curated gold standard for molecular interactions, reactions, and relations (Kanehisa et al. 2017).

We first do a pooled analysis over all genes identified by any discovered pattern for a class, i.e., the union of features in the respective patterns. We find that enriched pathways for tumor tissue correspond to known cancer drivers, such as MAPK and WNT signalling, while for the healthy tissue we find pathways linked to the regulation of lipolysis in adipocytes as well as PPAR signalling, both of which are known to be dysregulated in BRCA (Yang et al. 2018; Zhao

[2]Human variation data, such as the *Genomes* dataset, is an ideal application for DiffNaps as it is a high-dimensional resource of binary data in which we can uncover potential genetic predispositions of individuals to diseases, thus allowing to advance early detection and treatment. However, in the available data, the target class is the population membership of the individual, which raises ethical concerns for detailed analysis. Sadly, no further meta-data is available to meaningfully split *Genomes* for differential analysis.

et al. 2022). In short, DIFFNAPS discovers patterns that together describe complex, cancer-related functions.

Investigating *individual* patterns, we find that while many identify general pathways like above, others are enriched for *specific* pathways, such as PPAR. This shows the discovered patterns reveal details that can potentially be used for discovering alternative treatment targets for these pathways.

Differentiating Cancer Subtypes It is well known that breast cancer is not one single disease, e.g. the Luminal A, Luminal B, HER2+, and the Triple Negative subtypes all show distinct molecular behaviour, response to treatment, and patient survival. To investigate whether DIFFNAPS can elucidate differences between these subtypes, we run DIFFNAPS on *BRCA-S*, a balanced dataset of primary BRCA tissue with subtype label, and again analyse the discovered patterns using a gene set over-representation analysis in KEGG.

Starting with a pooled analysis, we find significantly enriched pathways that capture specifics of classes. Luminal A, for example, is defined by a lack of HER2. For this subtype, DIFFNAPS discovers patterns that are enriched for (i.e. related to) dilated cardiomyopathy. This is a common side-effect in Trastuzumab treatment, a drug which targets and depletes HER2 in HER2 positive subtypes (Crone et al. 2002). Luminal B is Estrogen receptor positive, meaning it expresses this receptor. For this subtype, we find patterns that are significantly enriched for sphingolipid metabolism. This is an important component for cell survival, proliferation, and promotion of cell migration and invasion in Estrogen receptor-positive BRCAs (Corsetto et al. 2023). These metabolites are also targets of treatment, and the discovered patterns could reveal insights leading to potential new therapeutic targets.

Promising Novel Patterns On both *BRCA* datasets, we find highly class-specific patterns, with average log-odds of $\mathbb{P}(p \mid k)$ against $\mathbb{P}(p \mid \neg k)$ of ≈ 8 resp. ≈ 5. Encouragingly, the above analysis above showed that many of these patterns capture complex biological processes related with BRCA progression or tumorigenes. More exciting perhaps are those patterns for which the genes are not yet annotated in a pathway but are strongly associated with BRCA or its subtypes. We are looking forward to conducting an in-depth analysis with oncologists, relating these patterns with more fine-grained subtypes or treatment groups.

5 Discussion

Experiments show that DIFFNAPS finds succinct sets of differential patterns, scales to hundreds of thousands of features, large number of classes, and is robust to noise.

On synthetic data, we saw that existing methods fail to recover significant portions of the planted differential patterns. Rule-based methods only recovered small subsets of incomplete patterns. SPUMANTE suffers from memory problems, and returns overly large, redundant results. PREMISE does account for redundancy, which results in better performance, but its combinatorial search does not scale well. RLL and CART scale very well, but show poor performance on synthetic data. Surprisingly, none of the existing approaches are robust to destructive noise.

On real world data, we find DIFFNAPS is the only approach that scales well *and* retrieves high-quality patterns. While other approaches show good performance on individual datasets, e.g. CART on *Cardio* and PREMISE on *Disease* data, they fail to do so in general. We also note that CART and SPUMANTE tend to return thousands of patterns, which undermines the goal of human interpretation.

DIFFNAPS fulfills the goal we set for this work and presents itself as a suitable candidate to take on the challenge of high-dimensional pattern mining in applications like genomics. As encouraging its ability in retrieving class-descriptive patterns at scale is, there is of course no free lunch. For example, on low-dimensional data of up to a hundred features, DIFFNAPS has a harder time differentiating classes and individual patterns and performs 'only' on par with other approaches. For such low-dimensional regimes, employing methods with guarantees, that are usually infeasible for large-scale data is still preferential.

Similar to most existing work, DIFFNAPS considers only conjunctions of features as patterns. In many applications, relations can be more complex, such as mutually exclusive features. It would make for engaging future work to study extensions of DIFFNAPS to capture such relations. In a case study on breast cancer datasets, we show that DIFFNAPS discovers patterns that capture class-relevant biological processes. The results are not only encouraging, but also contain many patterns for which the genes are not yet annotated to a pathway or process, or the function of individual genes is still unknown. These results offer an exciting opportunity to investigate novel links between genes and diseases in follow-up studies with domain experts.

6 Conclusion

We studied the problem of discovering differential patterns, i.e., patterns that succinctly describe and differentiate between the classes present in the data. Existing methods are often limited to binary classes, do not scale to high-dimensional data, or retrieve uninformative pattern sets.

To tackle this problem, we proposed a novel neural network architecture DIFFNAPS consisting of a binary autoencoder and a classification head. With a flat, binary architecture, the learned intermediate layer captures symbolic patterns. For the optimization, we proposed a multi-task objective to jointly optimize the reconstruction and classification, thus driving learning of patterns that both reconstruct the data well and differentiate between classes.

On synthetic and real-world data, including biological case studies on breast cancer, we show that DIFFNAPS strikes a unique balance among existing work, scales to high-dimensional data, is robust to noise, and accurately retrieves differential patterns that are highly interpretable.

Acknowledgements

The *BRCA* datasets were derived from data made available by the TCGA Research Network.[3]

[3] https://www.cancer.gov/tcga

References

Atzmueller, M. 2015. Subgroup discovery. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 5(1): 35–49.

Bai, Y.; Wang, Y.-X.; and Liberty, E. 2019. Quantized Neural Networks via Proximal Operators. In *Proceedings of the International Conference on Learning Representations (ICLR)*. OpenReview.

Bengio, Y.; Léonard, N.; and Courville, A. 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*.

Breiman, L. 1984. *Classification and regression trees*. Routledge.

Bringmann, B.; and Zimmermann, A. 2007. The Chosen Few: On identifying valuable patterns. In *Proceedings of the 7th IEEE International Conference on Data Mining (ICDM), Omaha, NE*, 63–72.

Budhathoki, K.; and Vreeken, J. 2015. The Difference and the Norm – Characterising Similarities and Differences between Databases. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD), Porto, Portugal*. Springer.

Chen, C.; and Rudin, C. 2018. An optimization approach to learning falling rule lists. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 604–612. PMLR.

Choi, J.; Chuang, P. I.-J.; Wang, Z.; Venkataramani, S.; Srinivasan, V.; and Gopalakrishnan, K. 2018. Bridging the accuracy gap for 2-bit quantized neural networks (qnn). *arXiv preprint arXiv:1807.06964*.

Corsetto, P. A.; Zava, S.; Rizzo, A. M.; and Colombo, I. 2023. The Critical Impact of Sphingolipid Metabolism in Breast Cancer Progression and Drug Response. *Int J Mol Sci*, 24(3).

Crone, S. A.; Zhao, Y. Y.; Fan, L.; Gu, Y.; Minamisawa, S.; Liu, Y.; Peterson, K. L.; Chen, J.; Kahn, R.; Condorelli, G.; Ross, J.; Chien, K. R.; and Lee, K. F. 2002. ErbB2 is essential in the prevention of dilated cardiomyopathy. *Nat Med*, 8(5): 459–465.

Dalleiger, S.; and Vreeken, J. 2022. Efficiently factorizing boolean matrices using proximal gradient descent. *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, 35: 4736–4748.

Dash, S.; Gunluk, O.; and Wei, D. 2018. Boolean decision rules via column generation. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*.

Degrave, J.; Felici, F.; Buchli, J.; Neunert, M.; Tracey, B.; Carpanese, F.; Ewalds, T.; Hafner, R.; Abdolmaleki, A.; de Las Casas, D.; Donner, C.; Fritz, L.; Galperti, C.; Huber, A.; Keeling, J.; Tsimpoukelli, M.; Kay, J.; Merle, A.; Moret, J. M.; Noury, S.; Pesamosca, F.; Pfau, D.; Sauter, O.; Sommariva, C.; Coda, S.; Duval, B.; Fasoli, A.; Kohli, P.; Kavukcuoglu, K.; Hassabis, D.; and Riedmiller, M. 2022. Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature*, 602(7897): 414–419.

Dierckx, L.; Veroneze, R.; and Nijssen, S. 2023. RL-Net: Interpretable Rule Learning with Neural Networks. In *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 95–107. Springer.

Dong, G.; and Li, J. 1999. Efficient mining of emerging patterns: Discovering trends and differences. In *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*.

Fischer, J.; and Vreeken, J. 2020. Discovering Succinct Pattern Sets Expressing Co-Occurrence and Mutual Exclusivity. In *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, 813–823.

Fischer, J.; and Vreeken, J. 2021. Differentiable pattern set mining. In *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, 383–392.

García-Vico, A.; Carmona, C. J.; Martín, D.; García-Borroto, M.; and del Jesus, M. J. 2018. An overview of emerging pattern mining in supervised descriptive rule discovery: taxonomy, empirical study, trends, and prospects. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(1): e1231.

Hedderich, M. A.; Fischer, J.; Klakow, D.; and Vreeken, J. 2022. Label-descriptive patterns and their application to characterizing classification errors. In *Proceedings of the International Conference on Machine Learning (ICML)*, 8691–8707. PMLR.

Hess, S.; and Morik, K. 2017. C-salt: Mining class-specific alterations in boolean matrix factorization. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, 547–563. Springer.

Hubara, I.; Courbariaux, M.; Soudry, D.; El-Yaniv, R.; and Bengio, Y. 2017. Quantized neural networks: Training neural networks with low precision weights and activations. *Journal of Machine Learning Research*, 18(1): 6869–6898.

Hüllermeier, E.; Fürnkranz, J.; and Loza Mencia, E. 2020. Conformal rule-based multi-label classification. In *Proceedings of Advances in Artificial Intelligence (KI)*, 290–296. Springer.

Huynh, V. Q. P.; Fürnkranz, J.; and Beck, F. 2023. Efficient learning of large sets of locally optimal classification rules. *Machine Learning*, 112(2): 571–610.

Jumper, J.; Evans, R.; Pritzel, A.; Green, T.; Figurnov, M.; Ronneberger, O.; Tunyasuvunakool, K.; Bates, R.; Žídek, A.; Potapenko, A.; Bridgland, A.; Meyer, C.; Kohl, S. A. A.; Ballard, A. J.; Cowie, A.; Romera-Paredes, B.; Nikolov, S.; Jain, R.; Adler, J.; Back, T.; Petersen, S.; Reiman, D.; Clancy, E.; Zielinski, M.; Steinegger, M.; Pacholska, M.; Berghammer, T.; Bodenstein, S.; Silver, D.; Vinyals, O.; Senior, A. W.; Kavukcuoglu, K.; Kohli, P.; and Hassabis, D. 2021. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873): 583–589.

Kanehisa, M.; Furumichi, M.; Tanabe, M.; Sato, Y.; and Morishima, K. 2017. KEGG: new perspectives on genomes,

pathways, diseases and drugs. *Nucleic Acids Res*, 45(D1): D353–D361.

Klösgen, W. 1995. Explora: A Multipattern and Multistrategy Discovery Assistant. In *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*.

Kusters, R.; Kim, Y.; Collery, M.; Marie, C. d. S.; and Gupta, S. 2022. Differentiable Rule Induction with Learned Relational Features. In *Proceedings of the International Workshop on Neural-Symbolic Learning and Reasoning*.

Lakkaraju, H.; Bach, S. H.; and Leskovec, J. 2016. Interpretable decision sets: A joint framework for description and prediction. In *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, 1675–1684.

Lemmerich, F.; and Becker, M. 2018. pysubgroup: Easy-to-use subgroup discovery in python. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, 658–662.

Lin, J.; Zhong, C.; Hu, D.; Rudin, C.; and Seltzer, M. 2022. Generalized and Scalable Optimal Sparse Decision Trees. arXiv:2006.08690.

Liu, Z.; Wu, B.; Luo, W.; Yang, X.; Liu, W.; and Cheng, K.-T. 2018. Bi-real net: Enhancing the performance of 1-bit cnns with improved representational capability and advanced training algorithm. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 722–737.

Llinares-López, F.; Sugiyama, M.; Papaxanthos, L.; and Borgwardt, K. 2015. Fast and memory-efficient significant pattern mining via permutation testing. In *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, 725–734.

Lukasiewicz, S.; Czeczelewski, M.; Forma, A.; Baj, J.; Sitarz, R.; and Stanislawek, A. 2021. Breast Cancer-Epidemiology, Risk Factors, Classification, Prognostic Markers, and Current Treatment Strategies-An Updated Review. *Cancers (Basel)*, 13(17).

McTavish, H.; Zhong, C.; Achermann, R.; Karimalis, I.; Chen, J.; Rudin, C.; and Seltzer, M. I. 2022. Fast Sparse Decision Tree Optimization via Reference Ensembles. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.

Patil, P.; and Rathod, P. 2020. Disease Symptom Prediction. https://www.kaggle.com/datasets/itachi9604/disease-symptom-description-dataset. *Accessed:* November 7, 2022.

Pellegrina, L.; Riondato, M.; and Vandin, F. 2019. SPuManTE: Significant pattern mining with unconditional testing. In *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, 1528–1538.

Proença, H. M.; and van Leeuwen, M. 2020. Interpretable multiclass classification by MDL-based rule lists. *Information Sciences*, 512: 1372–1393.

The 1000 Genomes Project Consortium. 2015. A global reference for human genetic variation. *Nature*, 526(7571): 68–74.

Ulianova, S. 2017. Cardiovascular Disease dataset. https://www.kaggle.com/datasets/sulianova/cardiovascular-disease-dataset. *Accessed:* November 5, 2022.

Van Leeuwen, M.; and Knobbe, A. 2012. Diverse subgroup set discovery. *Data Mining and Knowledge Discovery*, 25: 208–242.

Wang, Z.; Zhang, W.; Liu, N.; and Wang, J. 2021. Scalable rule-based representation learning for interpretable classification. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, 30479–30491.

Wang, Z.; Zhang, W.; Ning, L.; and Wang, J. 2020. Transparent classification with multilayer logical perceptrons and random binarization. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 6331–6339.

Yang, D.; Li, Y.; Xing, L.; Tan, Y.; Sun, J.; Zeng, B.; Xiang, T.; Tan, J.; Ren, G.; and Wang, Y. 2018. Utilization of adipocyte-derived lipids and enhanced intracellular trafficking of fatty acids contribute to breast cancer progression. *Cell Commun Signal*, 16(1): 32.

Zhao, B.; Xin, Z.; Ren, P.; and Wu, H. 2022. The Role of PPARs in Breast Cancer. *Cells*, 12(1).