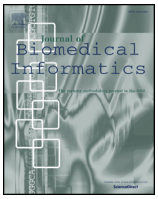
Journal of Biomedical Informatics 84 (2018) 159–163

Contents lists available at ScienceDirect

Journal of Biomedical Informatics

journal homepage: www.elsevier.com/locate/yjbin



Modeling association detection in order to discover compounds to inhibit oral cancer

Suhas Vittal*, Gokul Karthikeyan

BASIS Scottsdale, Scottsdale, AZ, United States

ARTICLE INFO

Keywords:
Machine learning
Graph analytics
Oral cancer
Compounds
Biological pathways

ABSTRACT

In the past, algorithms exploiting varying semantics in interactions between biological objects such as genes and diseases have been used in bioinformatics to uncover latent relationships within biological datasets.

In this paper, we consider the algorithm Medusa in parallel with binary classification in order to find potential compounds to inhibit oral cancer. Oral cancer affects the mouth and pharynx and has a high mortality rate due to its late discovery. Current methods of oral cancer treatment, such as chemoradiation and surgery, fail to provide better chances for survival, warranting an alternative approach. By running Medusa on a data fusion graph consisting of biological objects, we incorporate binary classification to model the algorithm's association detection to discover compounds with the potential to mitigate the effects of oral cancer.

1. Introduction

Current methods of treatment for oral cancer – chemoradiation and surgery – do not present any greatly increased chance of survival or life expectancy [1]. To date, concurrent chemoradiotherapy (CCRT) has been reported as effective and has become an acceptable treatment for advanced oral cancer, yet CCRT lacks sufficient data to demonstrate a good survival outcome; past studies have shown CCRT has no statistically different survival outcome from other treatment options [1]. The lack of a currently acceptable treatment that significantly combats oral cancer urges us to examine new ways to inhibit oral cancer; we choose to discover compounds that would inhibit oral cancer through machine learning. Research on using compounds to mitigate cancer has been conducted in the past: Li et al. conducted research on the use of compounds to prevent metastasis [2] and Kim and Roberts conducted research on using compounds to target EZH2[1] [3].

However, the application of machine learning to biological data presents a novel method to find inhibitory compounds to target cancers. Machine learning has been used in past research regarding inhibitory compounds for oral cancer and other diseases. In 2015, Bundela et al. discovered multiple therapeutic compounds to treat oral cancer patients using support vector machines [4]. In 2016, Hohman et al. discovered gene-gene interactions amongst multiple datasets concerning late-onset Alzheimer disease using Biofilter[2] [5]. Most recently, Agrawal et al. aimed to discover disease pathways by analyzing a protein–protein interaction (PPI) network, finding that there is detectable PPI network structure around disease proteins [6]. Similarly, we consider the Medusa algorithm in our interactive data analysis due to its higher cross-validated accuracy compared to other algorithms like a meta-path based approach and random-walk [7]. Concurrently, we incorporate binary classification in conjunction with Medusa to obtain possible compounds to inhibit oral cancer.

2. Methodology

As we aim to discover compounds that could potentially inhibit oral cancer with machine learning, we first must describe the Medusa algorithm. Given a data fusion graph $\mathscr{G} = (V, R, T)$ (see Fig. 1), where V are the nodes, R are the edges (relations), and T are the constraints, Medusa intends to tri-factorize the sets of matrices $R = \{\mathbf{R}^{IJ} \in \mathbb{R}^{n_I \times n_J} | I, J \in V, I \neq J\}$ and $T = \{\theta^I \in \mathbb{R}^{n_I \times n_I} | I \in V\}$ into two sets of factored matrices G and S through the algorithm Data Fusion by Matrix Factorization (DFMF), such that:

$$\widehat{\mathbf{R}}^{IJ} = \mathbf{G}^I \mathbf{S}^{IJ} (\mathbf{G}^J)^\top \tag{1}$$

where $\mathbf{G}^I, \mathbf{G}^J \in G$, $\mathbf{S}^{IJ} \in S$, $\hat{\mathbf{R}}$ is a latent relational matrix [7]. Using the output from DFMF, a semantic path $\mathscr{P}$ containing nodes between and inclusive of a start node S and end node T is used in creating a

* Corresponding author.
E-mail address: suhaskvittal@gmail.com (S. Vittal).

[1] EZH2 is a histone methyl transferase that has been observed to be mutated in several forms of cancers and highly expressed in many others.

[2] Biofilter integrates multiple public databases of gene groupings and sets of disease-related genes to produce multi-SNP models.

https://doi.org/10.1016/j.jbi.2018.07.005
Received 2 April 2018; Received in revised form 2 July 2018; Accepted 7 July 2018
Available online 09 July 2018

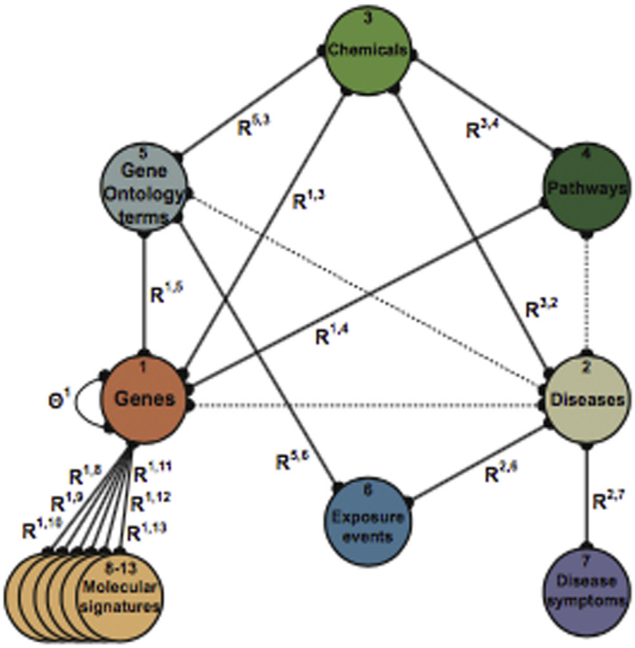


Fig. 1. An example of a data fusion graph. Image from [7].

materialized chain matrix $\mathbf{C}^{S,T}$ such that:

$$\mathbf{C}^{S,T} = \prod_{I,J \in \mathscr{P}} \widehat{\mathbf{R}}^{IJ} \tag{2}$$

Using this information, Medusa then calculates the most significant module, or set of objects, relative to a set of pivots S_0 that is a subset of the node S in the candidate-pivot-equivalence (CPE) regime, or a subset of the node T in the candidate-pivot-inequivalence (CPI) regime. The output of Medusa is a list of p-values assigned to a set of objects for a null hypothesis test. The null hypothesis is that there is no relationship between an object and the pivot set, and the alternate hypothesis is that there is some relationship between an object and the pivot set.

Nevertheless, due to the memory and time used in an execution of the Medusa algorithm, we incorporate binary classification to estimate association detection by extracting samples from the model yielded by Medusa; sampling requires less matrix operations and reduces the memory load and time taken, making it a plausible solution. We specifically use binary classifications because first, connections to the pivot set are binary, and second, our input is real-valued, not discretized. Note that in our data analysis, we only consider the CPE regime, and thus our sampling efforts will be directed towards the CPE regime.

Binary classification seeks to minimize the objective function $J(\theta) = (h_\theta(\mathbf{x}) - y)^2$, where $\mathbf{x}$ is a real-valued vector of features, y is a binary response variable, θ is a real-valued vector of coefficients, and h_θ is the hypothesis function we seek to have approach y. We define the hypothesis function as the sigmoid function (see Fig. 2). That is:

$$h_\theta(\mathbf{x}) = \frac{1}{1 + \exp(\theta^{\top}\mathbf{x})} \tag{3}$$

In creating a sampled training set, we first execute DFMF and obtained the appropriate sets of matrices G and S. Then we chose our respective start and end nodes to build $\mathscr{P}$; let our start node be A and our end node be B. We then conducted a systematic random sample of objects $x \in A$ and placed these objects in a set A'; note that the pivot set must be a subset of A', and that $A' \subseteq A$. Given A', define a new matrix:

$$\mathbf{G}^{(A')} = \begin{bmatrix} -\mathbf{g}_1- \\ -\mathbf{g}_2- \\ -\mathbf{g}_3- \\ \vdots \\ -\mathbf{g}_{n-1}- \\ -\mathbf{g}_n- \end{bmatrix} \tag{4}$$

where $\mathbf{g}_i$ is a row vector in $\mathbf{G}^A$, $i \in A'$. Finally, set $\mathbf{G}^A \coloneqq \mathbf{G}^{(A')}$ and $A \coloneqq A'$, and construct the chain matrix $\mathbf{C}$ by using Eqs. (1) and (2).[3] After constructing $\mathbf{C}$, we run Medusa and collect a sample of the results which returns a vector of p-values $\mathbf{p}$. From $\mathbf{p}$, we can construct a vector of discrete binary values, which we describe as $\mathbf{y}$, given by:

$$\mathbf{y} = \begin{bmatrix} \{1: p_1 < \alpha\} \\ \{1: p_2 < \alpha\} \\ \vdots \\ \{1: p_n < \alpha\} \end{bmatrix} \tag{5}$$

where α is the level of significance and p_i is the i-the component of $\mathbf{p}$. We write $\{1: P\}$ as the indicator function, which is equal to 1 if a predicate P is true and 0 otherwise.

As a result, we can construct dataset where the individual components of $\mathbf{c}_i$ (the i-th row vector of $\mathbf{C}$) are the explanatory variables and y_i (the i-th component of $\mathbf{y}$) is the response variable, resulting in training sets seen in Table 1, which we then use to train the binary classification model with the hypothesis function from (3).

The method of sampling in (4) enables fast computation because the samples taken with the initial matrix affects the runtime of the rest of the algorithm. Matrix multiplication is often implemented with complexity $\Theta(n^3)$. Thus, by simply decreasing the dimensions used in the calculation through sampling, the runtime significantly scales down. We specifically use systematic random sampling in (4) instead of other methods to avoid sampling selection bias, which can negatively affect discriminative algorithms [8]; that is, algorithms that model $P(y|x)$, which binary classification is since it is a Generalized Linear Model.

2.1. Measuring accuracy

Finally, before we analyze our data, we must show that our method of sampling is a valid method of modeling Medusa's output. We use sample data given by Zitnik and Zupan.[4]

In our measurement of model accuracy, we first constructed three materialized chain matrices obtained from three different runs of DFMF (note that DFMF has random initialization). For datasets, we sampled the first 400 objects from Medusa's output, randomly choosing 300 to be the training dataset for binary classification and leaving the other 100 as the validation dataset to confirm the accuracy of the model.

For the binary classification models created from each sample, we calculated the following measures of prediction accuracy:

1. **Confusion Matrix.** A confusion matrix demonstrates the frequency of false positives and negatives with the (0, 0) index being true positives, (1, 0) being false positives, (0, 1) being false negatives, and (1, 1) being true negatives.
2. **% Error: Validation.** This demonstrates the model's error in predicting a separate validation dataset.
3. **Root-Mean-Square Error (RMSE): Validation.** This is the model's RMS error differential in predicting the validation dataset.
4. **% Error: Entire.** This demonstrates the model's error in predicting the entire dataset used as input for Medusa. This metric is particularly important because we want to be able to predict Medusa's output correctly while avoiding a large error. Note that a sample of

[3] We specifically use ≔ instead of = because the former implies assignment of value, while the latter implies the two are equivalent.

[4] Can be obtained at: https://github.com/marinkaz/medusa.