Learning influential genes on cancer gene expression data with stacked denoising autoencoders

Vítor Teixeira
MIEIC
Faculdade de Engenharia da Universidade do Porto
Portugal
Email: vitorf24@gmail.com

Rui Camacho
DEI
Faculdade de Engenharia da Universidade do Porto
Portugal
Email: rcamacho@fe.up.pt

Pedro G. Ferreira
Ipatimup - Institute of Molecular Pathology and Immunology of the University of Porto
I3S - Instituto de Investigação e Inovação em Saúde, Universidade do Porto
Portugal
Email: pferreira@ipatimup.com

***Abstract*—Cancer genome projects are characterizing the genome, epigenome and transcriptome of a large number of samples using the latest high-throughput sequencing assays. The generated data sets pose several challenges for traditional statistical and machine learning methods. In this work we are interested in the task of deriving the most informative genes from a cancer gene expression data set.**

For that goal we built denoising autoencoders (DAE) and stacked denoising autoencoders and we studied the influence of the input nodes on the final representation of the DAE. We have also compared these deep learning approaches with other existing approaches. Our study is divided into two main tasks. First, we built and compared the performance of several feature extraction methods as well as data sampling methods using classifiers that were able to distinguish the samples of thyroid cancer patients from samples of healthy persons. In the second task, we have investigated the possibility of building comprehensible descriptions of gene expression data by using Denoising Autoencoders and Stacked Denoising Autoencoders as feature extraction methods. After extracting information related to the description built by the network, namely the connection weights, we devised post-processing techniques to extract comprehensible and biologically meaningful descriptions out of the constructed models.

We have been able to build high accuracy models to discriminate thyroid cancer from healthy patients but the extraction of comprehensible models is still very limited.

Keywords: Deep Learning; Gene Expression Analysis; Knowledge Extraction; RNA-Seq; Cancer

I. Introduction

Cancer is the name given to a collection of related diseases, all of them being connected to the abnormal cell multiplication and spreading into surrounding tissues [1]. Given that it affects millions of people worldwide each year, cancer attracts a substantial amount of medical research due to the number of deaths and the impact on the quality of life of the patients. Cancer research focus on prevention, detection and treatment with the more prevalent and worst prognosis types of cancer receiving more research attention.

Morphological and clinical based prediction of cancer has some limitations, tumors with similar characteristics can follow significantly different clinical paths and show different responses to therapy. To overcome this problem and gain better insights, gene expression analysis has been frequently used in cancer research studies [2][3][4].

Gene expression is an intermediate phenotype that reflects the cellular state. Characterization of the patterns of gene expression provides a better understanding of the molecular basis of the disease with possible implications in diagnosis and treatment. Large scale cancer genomics projects such as The Cancer Genome Atlas (TCGA) [5] and the International Cancer Genome Consortium [6] and have collected and profiled through next generation sequencing thousands of transcriptomic samples across dozens of different types of cancers.

Molecular biomarkers provide important information in terms of diagnosis, prognosis and treatment of disease [7]. The task of discovering such predictive markers from either gene or protein abundance large-scale and genome-wide assays, has been of central relevance in cancer research. Statistics, data mining and machine learning methods have been widely used to tackle this task. Cancer genome projects are deriving data sets of genome-wide gene expression assays such as microarrays or high-throughput RNA-sequencing, characterized by having dozens of thousands (> 50K) features, as measured genes, and hundreds ($\sim$500) samples from tumor and normal tissue. These data sets pose different challenges at the time of applying traditional data analysis methods. Moreover, they typically suffer from highly imbalanced distribution of cancer sub-types or between tumor and normal sample size and underlying noise introduced be eventual technical and biological covariates. Here, we are interested in understanding the performance of Deep learning methods when applied to the analysis of gene expression data. Of particular interest is the performance on data sets that present an imbalance of the underlying classes. Deep learning is a machine learning approach that uses artificial neural networks with several hidden layers. It is a rather recent term and has been used to

978-1-5090-3050-7/17/$31.00 ©2017 IEEE

solve a great variety of problems. It is getting more and more interest, as it continues to outperform state-of-the-art classifiers in several tasks.[8].

II. Methodology

The main goal of this work was to develop a machine learning approach to capture the most relevant relations between the input genes towards a characterization of the gene expression data. For that purpose we have built DAE and stacked denoise autoencoders (SDAE) and studied the influence of the input nodes on the final representation of the DAE. Before feeding the data to the models, we have first evaluated the performance of various data sampling methods in order to improve the performance of the models. Then, we compared the performance of single and stacked denoising autoencoders with other feature reduction methods such as PCA [9] and KPCA.

In order to assess the quality of the generated features we used a shallow artificial neural network as a classifier. Every approach was tested using 5-fold cross-validation in order to guarantee the robustness of the results. For the stacked autoencoder approach, the final architecture was found by choosing the best performance model. The number of features for the final representations was chosen to be 400 for comparison with other methods as the number of final features should not be higher than the number of samples. Finally, after having a trained model, we studied ways to extract information about the resulting representation generated by the autoencoders. For that we started by studying the resulting weights from the training process to measure the individual contribution of each neuron to the final representation. We have also tried to assess if there was any relation between the neurons of the final representation and known human pathways in terms of involved genes.

A. Data pre-processing

We have used papillary thyroid carcinoma RNA-Seq data from TCGA[5] having 510 cancer samples and 58 healthy samples with gene expression for 60483 genes. We have used Gencode Annotation, which is a very comprehensive annotation that includes different gene biotypes. While the number of protein coding genes are ∼21K, we also considered pseudogenes accounting for another ∼14K, long non coding RNAs ∼13K, miRNA ∼3K and other less frequent biotypes, summing up to the ∼60k genes. We first started by removing genes with zero expression across all samples, resulting in 57490 genes. Then, we divided the data set in train and test (80%-20% proportion) and ensured that no fold contains samples from the major class only. Data was normalized after log transformation (adding pseudo-count of one) after a Z-score normalization (subtracting the mean and dividing by the standard deviation).

B. Data sampling and feature reduction

After processing the data we compared the performance of the combination between data sampling methods like Synthetic Minority Over-sampling TEchnique (SMOTE) [10] and Adaptive Synthetic Sampling (ADASYN) [11] and feature reduction methods like PCA and KPCA.

In this technique the minority class is over sampled based on the feature space similarities between its samples. This approach, however, has some drawbacks as when it is generating synthetic samples it doesn't consider neighboring examples, which can lead to over generalization and increases the overlapping between classes [12][13]. In order to deal with this drawback, adaptive sampling methods such as ADASYN have been proposed. ADASYN uses a method to create data according to their distributions, that is, more samples are generated for minority class samples that are harder to learn compared to samples that are easier to learn[14]. In addition to ADASYN, there are also other ways to deal with the issues of SMOTE. One such way is to use it with data cleaning techniques like SMOTE+Tomek or SMOTE+ENN[15] that clean the unwanted overlapping between classes after the oversampling, each one using different methods for the task.

For feature reduction methods we used PCA and KPCA. PCA stands for Principal Component Analysis, it is a linear technique whose goal is to encode high-dimensional data into a lower dimensional representation. It finds the most important information within the data and generates a set of orthogonal variables called principal components that best differentiate the data points[16]. KPCA is an extension of PCA, however, unlike PCA, KPCA tries to find a low-dimensional nonlinear space[17]. It uses kernel methods to compute the principal components in high-dimensional feature spaces.

These approaches were implemented using Python packages *imbalanced-learning*[18] and *scikit-learn*[19], respectively.

C. Stacked Denoising Autoencoders

An autoencoder is a feed-forward neural network that aims to reconstruct the given input by using a lower dimensional hidden layer [20]. It consists of an encoder and decoder. In the encoder part the input is mapped to a hidden representation by applying an affine transformation followed by a non-linear function. In this work Leaky ReLU (Rectifier Linear Unit)[21] was used. That transformation can be expressed as $f_\theta(x) = \sigma(Wx + b)$ with parameters $\theta = W, b$, where W is a $d' \times d$ weight matrix, and b the bias vector. To reconstruct the input, the hidden representation is mapped back by applying $z = g_{\theta'}(y)$ where $g_{\theta'}(y) = \sigma(W'x + b')$ with $\theta = W', b'$. The reconstruction error is then optimized using a loss function. We used Kullback-Leibler divergence to measure the difference between the two distributions. In DAE the input is intentionally corrupted with noise in order to avoid overfitting and extract more meaningful information[20].

Denoising autoencoders (DAE) can be stacked to form a deep network by using the hidden representation of one as input for the next autoencoder [20]. These architectures are called stacked denoising autoencoders (SDAE) and the training of these architectures is usually done by pre-training. We followed this approach where each autoencoder layer is trained independently, that is, the output of an autoencoder layer serves as input for the next one. This approach is proved to lead to a better generalization[20]. After the pre-training of all the layers, a supervised fine-tuning step was performed. The network parameters were updated to better model the training data. Autoencoders can be useful because they can extract both linear and nonlinear information from data. To avoid overfitting and to improve the model generalization, in addition to adding noise to the input we also used a regularization factor called dropout. This technique, applied during training, randomly drops-out nodes with a given probability at each weigh update cycle [8].

D. Knowledge extraction

After having trained the models we have extracted the weights of all autoencoder layers. In order to estimate the effect of each autoencoder in the model we calculated the sum of products of all layers as:

$$W = \prod_{i=1}^{n} W_i$$

The resulting matrix W has size 57490×400, which is the size of the feature space and the size of the final feature representation respectively.

Then, two different methods of calculating input node importance were used: Connection Weights method [22], and a method proposed by us that we called Top Percentage Contributions. In a comparison reported in [23], the connection weights method was shown to be one of the best performing methods. In this methods, the relative importance of each input neuron can be defined as the sum as the outgoing connections for each input neuron. In the method here proposed, the matrices product is also used, however, this time, for each outer node we iterate through its incoming connections and add the top X nodes to a set, with X being a carefully chosen percentage.

With the list of the most relevant genes we have used DAVID [24][25] to perform a functional annotation clustering analysis to group similar gene terms in clusters to find how they are connected and if the clusters made sense biologically.

Deep Learner's layered structure may be interpreted as a succession of abstractions. The input layer being the most concrete one and the output layer the most abstract one. In this work we have tried to "match" some internal nodes with biologically relevant concepts. We have verified the similarities between the most influential genes in each node of the hidden representations and the genes of known human pathways. For that, we downloaded all pathways and corresponding genes from KEGG[26] online database and, for each node in the final representation we compared the high weight genes with the genes involved in each pathway.

III. Results

A. Data oversampling experiment

In Table I we can see that the best overall approaches were SMOTE+TOMEK with KPCA or SMOTE+ENN with PCA, that yielded the same performance. We chose to work with SMOTE+TOMEK since it yielded the best performance using denoising autoencoders as a feature reduction method. In this experiment we did not assess the performance of stacked denoising autoencoders due to the extensive computation time and hardware requirements of training such deep networks.

TABLE I
Sampling methods performance results (%)

Sampling method	**Features**	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Raw	None	76.07	**65.04**	85.93	74.03
	PCA	98.95	96.77	97.87	97.29
	KPCA	98.95	96.77	97.87	97.29
	DAE	93.48	82.61	91.08	86.37
SMOTE + TOMEK	Raw	97.54	91.12	97.90	94.34
	PCA	99.12	98.04	97.23	97.63
	KPCA	**99.30**	**98.20**	**98.06**	**98.12**
	DAE	98.07	94.60	95.09	94.83
SMOTE + ENN	Raw	95.07	84.18	96.52	89.90
	PCA	**99.30**	**98.20**	**98.06**	**98.12**
	KPCA	98.95	97.87	96.39	97.11
	DAE	95.78	86.67	96.10	91.08
ADASYN	Raw	98.41	96.64	94.63	95.57
	PCA	98.95	97.87	96.39	97.11
	KPCA	98.77	97.77	95.48	96.61
	DAE	97.18	93.46	91.94	92.14

TABLE II
Feature extraction methods performance results (%)

Features	**Accuracy**	**Precision**	**Recall**	**F1 Score**
Raw	97.77	91.62	97.99	94.66
PCA	99.12	98.04	97.23	97.63
KPCA	**99.30**	**98.20**	**98.06**	**98.12**
DAE	98.07	94.60	95.09	94.83
SDAE	97.36	94.70	92.05	93.30

B. Feature reduction experiment

From Table II we concluded that the best overall performance of generated features was from KPCA. KPCA is a good choice to boost high dimensionality data classification performance. However, it does not give us any specific information about the individual relevant genes that can potentially play the role of biomarkers.

Overall, the denoising autoencoder approach yielded similar results. We believe that with better hardware we can eventually tune the networks to yield similar or better results. Nevertheless, it could also be the case that be that a less complex model is more adequate to fit the data.

C. Knowledge extraction experiments

In the first experiment, since we did not have the opportunity to validate the results with a thyroid cancer specialist, we performed a preliminary analysis on the resulting clusters from the best scoring method which was the combination of stacked denoising autoencoders using the Top Percent Contributions method. From the analysis of the functional annotation clustering filtered for p-values under 0.05 we got three different clusters presented in Table III.

TABLE III
FUNCTIONAL ANALYSIS CLUSTERING USING ALGORITHM TOP CONTRIBUTIONS METHOD ON STACKED DENOISING AUTOENCODER RESULTS WITH 378 DAVID IDS (0.0015% OF THE GENES WITH HIGH WEIGHTS IN EACH NODE) WITH A P-VALUE THRESHOLD OF 0.05

Annotation Cluster 1 **Enrichment Score:** 3.88

Category	**Term**	**Gene count**	**P-Value**	**Benjamini**
UP_SEQ_FEATURE	repeat:CSPG: 1	4	3.1^{-5}	2.8^{-2}
UP_SEQ_FEATURE	repeat:CSPG: 2	4	3.1^{-5}	2.8^{-2}
	...			
UP_SEQ_FEATURE	repeat:CSPG: 12	4	3.1^{-5}	2.8^{-2}
UP_SEQ_FEATURE	domain:Calx-beta 3	3	1.3^{-3}	3.2^{-1}
UP_SEQ_FEATURE	domain:Calx-beta 1	3	4.3^{-3}	6.3^{-1}
UP_SEQ_FEATURE	domain:Calx-beta 2	3	4.3^{-3}	6.3^{-1}
SMART	Calx_beta 3	3	6.4^{-3}	6.2^{-1}
INTERPRO	Na:Ca exchanger/integrin-beta4	3	7.7^{-3}	7.9^{-1}

Annotation Cluster 2 **Enrichment Score:** 1.75

Category	**Term**	**Gene count**	**P-Value**	**Benjamini**
UP_SEQ_FEATURE	domain:EF-hand 4	5	1.3^{-2}	8.7^{-1}
UP_SEQ_FEATURE	domain:EF-hand 3	6	1.4^{-2}	8.4^{-1}
GOTERM_MF_DIRECT	calcium ion binding	20	1.4^{-2}	9.5^{-1}
INTERPRO	EF-hand domain	9	2.3^{-2}	1.0^{0}
INTERPRO	EF-hand-like domain	10	2.7^{-2}	1.0^{0}

Annotation Cluster 3 **Enrichment Score:** 1.42

Category	**Term**	**Gene count**	**P-Value**	**Benjamini**
UP_KEYWORDS	Cell division	12	2.9^{-2}	7.3^{-1}
UP_KEYWORDS	Mitosis	9	4.0^{-2}	7.2^{-1}
GOTERM_BP_DIRECT	cell division	11	4.8^{-2}	9.9^{-1}

From this analysis we found that Chondroitin sulfate proteoglycans (CSPGs), a group of genes known to be structural components of a variety of human tissues and known to be involved in certain cell processes such as cell growth, cell migration, cell adhesion, receptor binding and interaction with other extra cellular matrix constituents[27] are represented in annotation cluster 1. It has been previously identified that CSPGs play a role in tumor progression, growth and metastasis [28]. Furthermore, in cluster 2, most of gene ontology terms are related to calcium-binding regions or to EF-hand domain, which is the most common calcium-binding motif found in proteins. EF-hand proteins are known to be related to the growth and metastasis of various types of cancer, including papillary thyroid carcinoma [29][30]. Finally, the last cluster is related to cell division and mitosis, which is directly related to abnormal cell proliferation.

For the pathway similarity experiment we did not get yet any relevant similarities between the computed representation and the extracted pathways.

IV. CONCLUSIONS

In this paper we have described a set of experiments on using Deep Learning to build a classifier to help diagnosing thyroid cancer. We had two main objectives: i) build accurate models; ii) extract biological/medical useful information to help domain experts to better understand thyroid cancer. We got very good performance from the Machine Learning point of view. As far as knowledge extract is concerned we have not achieved fully that goal. The analysis made has only been able to identify some genes that are already reported in the literature.

While there is still the need to validate the results with a specialist in thyroid cancer, we believe that stacked denoising autoencoders can extract good representations from this high dimensional data and enable the extraction of biologically meaningful information that can give cancer researchers a direction for further studies.

REFERENCES

[1] “What is cancer?” https://www.cancer.gov/about-cancer/understanding/what-is-cancer, accessed: 2017-07-31.

[2] L. J. Van't Veer, H. Dai, M. J. Van De Vijver, Y. D. He, A. A. Hart, M. Mao, H. L. Peterse, K. Van Der Kooy, M. J. Marton, A. T. Witteveen *et al.*, “Gene expression profiling predicts clinical outcome of breast cancer,” *nature*, vol. 415, no. 6871, pp. 530–536, 2002.

[3] S. A. Tomlins, D. R. Rhodes, S. Perner, S. M. Dhanasekaran, R. Mehra, X.-W. Sun, S. Varambally, X. Cao, J. Tchinda, R. Kuefer *et al.*, “Recurrent fusion of tmprss2 and ets transcription factor genes in prostate cancer,” *science*, vol. 310, no. 5748, pp. 644–648, 2005.

[4] R. Chen, P. Khatri, P. K. Mazur, M. Polin, Y. Zheng, D. Vaka, C. D. Hoang, J. Shrager, Y. Xu, S. Vicent *et al.*, “A meta-analysis of lung cancer gene expression identifies ptk7 as a survival gene in lung adenocarcinoma,” *Cancer research*, vol. 74, no. 10, pp. 2892–2902, 2014.

[5] C. D. Kyle Chang, Chad J Creighton, “The cancer genome atlas pan-cancer analysis project,” *Nature Genetics*, vol. 45, no. 10, pp. 1113–1120, sep 2013. [Online]. Available: https://doi.org/10.1038/ng.2764

[6] T. J. e. a. Hudson, “International network of cancer genome projects,” *Nature*, vol. 464, no. 7291, pp. 993–998, Apr 2010.

[7] M. J. Duffy, “Predictive markers in breast and other cancers: A review,” *Clinical Chemistry*, vol. 51, no. 3, pp. 494–503, mar 2005. [Online]. Available: https://doi.org/10.1373/clinchem.2004.046227

[8] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detectors,” 2012.

[9] K. Pearson, “On lines and planes of closest fit to systems of points in space,” *Philosophical Magazine, Series 6*, vol. 11, no. 2, pp. 559–572, 1901.

[10] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “Smote: synthetic minority over-sampling technique,” *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[11] H. He, Y. Bai, E. A. Garcia, and S. Li, “Adasyn: Adaptive synthetic sampling approach for imbalanced learning,” in *IN: IEEE INTERNATIONAL JOINT CONFERENCE ON NEURAL NETWORKS (IEEE WORLD CONGRESS ON COMPUTATIONAL INTELLIGENCE), IJCNN 2008*, 2008, pp. 1322–1328.

[12] H. He and E. Garcia, “Learning from imbalanced data,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, sep 2009.

[13] V. López, A. Fernández, S. García, V. Palade, and F. Herrera, “An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics,” *Information Sciences*, vol. 250, pp. 113–141, nov 2013.

[14] H. He, Y. Bai, E. A. Garcia, and S. Li, “Adasyn: Adaptive synthetic sampling approach for imbalanced learning,” in *Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*. IEEE, 2008, pp. 1322–1328.

[15] G. E. Batista, R. C. Prati, and M. C. Monard, “A study of the behavior of several methods for balancing machine learning training data,” *ACM Sigkdd Explorations Newsletter*, vol. 6, no. 1, pp. 20–29, 2004.

[16] H. Abdi and L. J. Williams, “Principal component analysis,” *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 4, pp. 433–459, 2010.

[17] B. Schölkopf, A. Smola, and K.-R. Müller, “Kernel principal component analysis,” in *International Conference on Artificial Neural Networks*. Springer, 1997, pp. 583–588.

[18] G. Lemaître, F. Nogueira, and C. K. Aridas, “Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning,” *Journal of Machine Learning Research*, vol. 18, no. 17, pp. 1–5, 2017. [Online]. Available: http://jmlr.org/papers/v18/16-365.html

[19] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[20] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning Research*, vol. 11, no. Dec, pp. 3371–3408, 2010.

[21] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. ICML*, vol. 30, no. 1, 2013.

[22] J. D. Olden and D. A. Jackson, "Illuminating the black box: a randomization approach for understanding variable contributions in artificial neural networks," *Ecological modelling*, vol. 154, no. 1, pp. 135–150, 2002.

[23] J. D. Olden, M. K. Joy, and R. G. Death, "An accurate comparison of methods for quantifying variable importance in artificial neural networks using simulated data," *Ecological Modelling*, vol. 178, no. 3, pp. 389–397, 2004.

[24] D. W. Huang, B. T. Sherman, and R. A. Lempicki, "Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists," *Nucleic Acids Research*, vol. 37, no. 1, pp. 1–13, nov 2008.

[25] ——, "Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources," *Nature Protocols*, vol. 4, no. 1, pp. 44–57, dec 2008.

[26] M. Kanehisa and S. Goto, "Kegg: kyoto encyclopedia of genes and genomes," *Nucleic acids research*, vol. 28, no. 1, pp. 27–30, 2000.

[27] K. E. Rhodes and J. W. Fawcett, "Chondroitin sulphate proteoglycans: preventing plasticity or protecting the CNS?" *Journal of Anatomy*, vol. 204, no. 1, pp. 33–48, jan 2004. [Online]. Available: https://doi.org/10.1111/j.1469-7580.2004.00261.x

[28] Y. Wegrowski and F.-X. Maquart, "Chondroitin sulfate proteoglycans in tumor progression," in *Chondroitin Sulfate: Structure, Role and Pharmacological Activity*. Elsevier, 2006, pp. 297–321. [Online]. Available: https://doi.org/10.1016/s1054-3589(05)53014-x

[29] J. Jendrzejewski, A. Thomas, S. Liyanarachchi, A. Eiterman, J. Tomsic, H. He, H. S. Radomska, W. Li, R. Nagy, K. Sworczak, and A. de la Chapelle, "PTCSC3 is involved in papillary thyroid carcinoma development by ModulatingS100a4gene expression," *The Journal of Clinical Endocrinology & Metabolism*, vol. 100, no. 10, pp. E1370–E1377, oct 2015. [Online]. Available: https://doi.org/10.1210/jc.2015-2247

[30] K. Zhang, M. Yu, F. Hao, A. Dong, and D. Chen, "Knockdown of s100a4 blocks growth and metastasis of anaplastic thyroid cancer cells in vitro and in vivo," *Cancer Biomarkers*, vol. 17, no. 3, p. 281291, Sep 2016. [Online]. Available: http://doi.org/10.3233/CBM-160640