Integrative learning of gene essentiality using data and knowledge with cluster specific models

Simone Rizzetto[1], Paurush Praveen[1*], Mario Lauria[1] Corrado Priami [1,2]
1 The Microsoft Research-University of Trento Centre for Computational and Systems Biology, Rovereto, Italy
2 Department of Information Engineering and Computer Science - University of Trento, Povo, Italy
[*] **Corresponding author: praveen@cosbi.eu**

Introduction

Essential genes are the genes, which are essential for survival of cell i.e. phenotypic effects of interfering with the function of the corresponding gene [1]. Predictive models to infer gene essentiality in cancer cell lines can aid the molecular characterization of the cancer cell lines, which can be ultimately used to identify biomarkers and tailored treatments as well as identify patients with higher treatment efficacy. The approaches to predict essentiality have been based on genome scale data [2] as well as on network of genes [3]. However, the essentiality should be seen as a context based term or measurement. For example, a gene essential in lung cancer cell line may not be essential to a breast cancer cell line and vice versa. Therefore, a generalized model for quantitative estimation of gene essentiality across heterogeneous cell lines is not suitable. Another limitation of current approaches, even with gene-specific models is that they tend to use data from heterogeneous or diverse cell lines rendering the overall model noisy and hence causing a reduction in their predictive power. Furthermore, an area that has been less exploited is the use of existing information or knowledge on the genes as well as cell lines (like, oncogene information, cell line information, etc.) that can boost the performance of quantitative prediction models. We aim to exploit these hypotheses while solving the issue of essentiality prediction in the challenge [4]. We propose two models, (1) One Gene-One Model (OGOM) with integrated knowledge features and (2) Cluster Specific-OGOM (CS-OGOM) addressing these issues.

Methods

A Support Vector Regression (e-SVR) [6] forms the underlying engine for prediction. The key aspects of the models have been highlighted below. The regression is aided with knowledge attributes (features) and clustering in OGOM and CS-OGOM methods respectively.

Support Vector Regression

The objective of a Support Vector regression is to estimate a real valued function. To illustrate, let us assume x is the multivariate input and y be the output. Starting with a training data $((x_1, y_1), \dots,$ $(x_n, y_n))$, where n is the number of training points. The SVR aims to optimize a function that models y_i based on x_i by mapping the data to a new hyperspace via a kernel function [5, 6]. This enables the model to identify a hyperplane for regression purpose.

Data

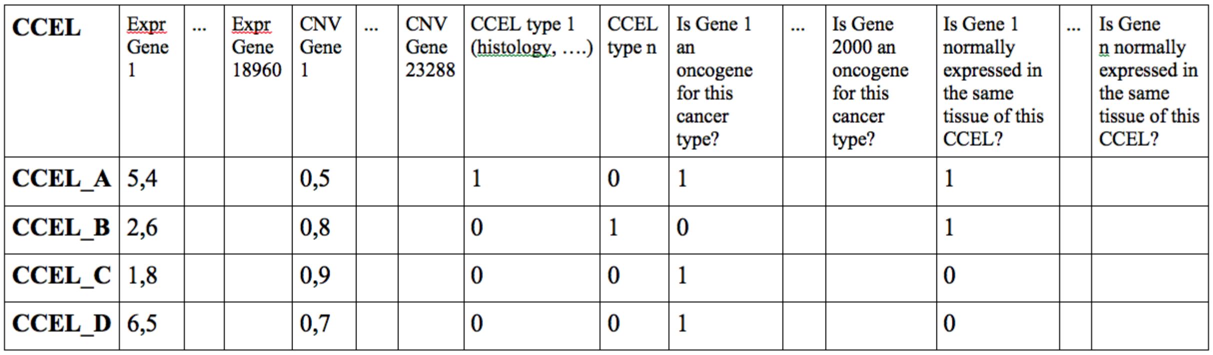
The data used as as the input to the models is the expression data for 105 cell lines and the corresponding essentiality measurement for 14760 genes.

Knowledge sources

The initial features (columns) are expression values of every gene in a specific cell line (18960 features) followed by CNVs (23288). Additionally, we included all the cell line information from the challenge data and from CCLE (Cancer Cell Line Encyclopedia) [1]. This included gender, site (primary), histology and histo-subtype. We also retrieved the related oncogenes from NCG database [2] for each cancer type. This was followed by looking for a correspondence between cancer type and cell line information. From TiGER database [3] we selected which genes are normally expressed in a specific tissue. Again, we used CCEL information to identify which genes are expressed in the same tissue of a given CCEL. However, for the CS-OGOM approach we used only the data (no knowledge) as the phenotypically related cell lines were already close in the clustering space.

Feature design

Each model has as many entries(rows) as the number of cell lines in the training data set. The categorical variables from knowledge sources were transformed to bit vectors of appropriate size depending on the feature (Figure 1).



CCEL	Expr Gene 1	...	Expr Gene 18960	CNV Gene 1	...	CNV Gene 23288	CCEL type 1 (histology, ….)	CCEL type n	Is Gene 1 an oncogene for this cancer type?	...	Is Gene 2000 an oncogene for this cancer type?	Is Gene 1 normally expressed in the same tissue of this CCEL?	...	Is Gene n normally expressed in the same tissue of this CCEL?
CCEL_A	5,4			0,5			1	0	1			1		
CCEL_B	2,6			0,8			0	1	0			1		
CCEL_C	1,8			0,9			0	0	1			0		
CCEL_D	6,5			0,7			0	0	1			0		

Figure 1. Feature design table.

Feature selection

We performed univariate statistical tests to select the best features to learn the model. After performing the univariate test we used the top 10-20 percentile features in different simulation studies. We also included the knowledge features for the univariate test as described in the feature design section above.

Algorithm

We used two different algorithms for each of our models.

OGOM approach

The One Gene One Model is based on training one model for each gene for all cell lines using support vector regression. The features used include the knowledge attributes as described in the sections above.

[1] http://www.broadinstitute.org/ccle
[2] http://ncg.kcl.ac.uk
[3] http://bioinfo.wilmer.jhu.edu/tiger/

CS-OGOM approach

The CS-OGOM approach is based on the hypothesis that closely related cell lines will follow one model for each gene in order to predict its essentiality. Thus, compared to the OGOM approach we have a gene specific model for every cluster. The approach first performs a hierarchical clustering on the expression data to identify closely related cell lines. For clustering we used the distance matrix computed using the rank based approach as described by Lauria et al. [7]. Now within each cluster we identify the training and test data. The training data within that cluster is used to learn a model for each gene and use this model to predict the essentiality of the corresponding gene in the test cell line present in that cluster only.

The back-end SVR engine was adapted from python based learning package scikit [8]. For each gene we trained a model. The training used the RBF (Radial Basis Function) kernel to learn the model.

Parameter optimization

As the quality of support vector based approach depends on proper setting of SVM hyper-parameters, this makes the selection of hyper-parameters a critical in learning a model from data. The two types of SVR that were used are ε-SVR and ν-SVR. The parameters used were the loss parameter, kernel width and the Cost parameter. In order to optimize the parameters were based on the work of Cherkassy et al. [9]. We optimized the epsilon (ε) and cost (C) function as in the equation 1 and 2.

$$\varepsilon = a\sigma_y\sqrt{\frac{ln(n)}{n}} \tag{1}$$

Where, σ_y is the stadard deviation of the output values and a is range specific constant to avoid SVM flattening [6].

$$C = max(|\bar{y} + a\sigma_y|, |\bar{y} - a\sigma_y|) \tag{2}$$

Where, $\bar{y}$ is the mean of the output values.

Discussion

Our observation on the essentiality values (y) from training data showed that very few genes show distinct behavior in terms of essentiality scores between two cell line (hence a low correlation) (Figure 2 A). This noisy behavior makes it difficult for any generalized model to predict the essentiality. The cluster computed with a rank based distance measure [7] depicted the close relation of phenotypically related cell lines in the clustering space (Figure 2 B), falling in line with our assumption and hypotheses.

We used leave one out cross validation to assess the performance of our methods. Correlation coefficients (both Pearson and Spearman) were used as performance measure for the methods. We randomly sampled 100 genes to observe the performance of the above-mentioned models. Our result showed that the CS-OGOM performed substantially better than the OMOM models in terms of Pearson and spearman coefficients (Figure 2 C and D) . The non specific models (one model for all genes and all cell line) yielded a considerably low score. The Pearson correlation for the CS-OGOM was found to be > 0.4 on average whereas on the spearman scale it oscillated around 0.3 depending on the gene under observation. The score for the OGOM were ∼0.2 along both the scale. The difference between the two scales in the CS-OGOM can be attributed to the mixing of predicting between different clusters which are not required in the OGOM approach.

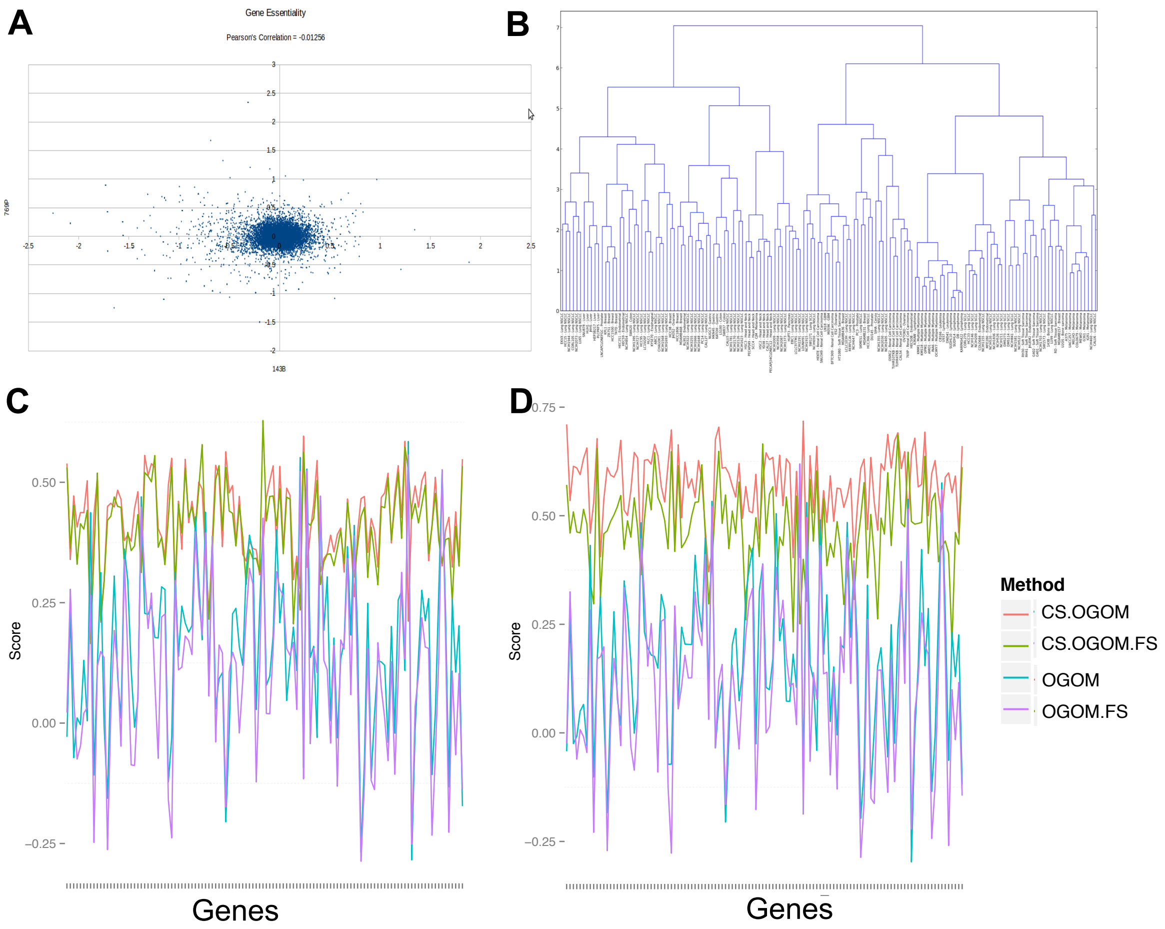


Figure 2. A. Unrelated nature of gene essentialities between two cell lines. B. Heirarchical clustering of the cell lines (training and test). C, D. Performance of the two approaches simulated with a leave one out cross-validation on 100 randomly selected genes. Performance measured with Spearman (C) and Pearson (D) correlation. The suffix '.FS' represents the use of feature selection in combination with the algorithms

Two critical aspects observed during our simulations were the choice of clusters and mixing of cluster prediction to get an overall prediction. Choosing appropriate number of clusters is critical, first in order to have enough representations from training and test set. For example, a cluster where the $n_{test} >> n_{train}$ can lead to inadequate learning samples. Second, the clustering should lead to a homogenous cell lines within clusters to improve the learning accuracy. Mixing predictions from multiple clusters to get overall prediction across all cell lines can lead to resorting of the prediction that can lead to lower Spearman correlation. Our observation indicated that the Spearman correlation within cluster was relatively higher than when joining prediction across all clusters. The decline in performance scores drops in case of CS-OGOM due to reordering of the essentialities, nevertheless the change in pearson correlation is minimal with all the cell lines. Therefore, feature and output scaling before applying these methods it is extremely important.

Contributions

PP and SR developed algorithms; PP, SR and ML worked on knowledge integration; SR and ML performed data analysis; ML developed distance metrics; PP, SR ML and CP designed experiments; PP, SR, ML and CP wrote the manuscript.

References

1. Acevedo-Rocha CG, Fang G, Schmidt M, Ussery DW, Danchin A (2013) From essential to persistent genes: a functional approach to constructing synthetic life. Trends in Genetics 29: 273 - 279.

2. Roberts SB, Mazurie AJ, Buck GA (2007) Integrating Genome-Scale Data for Gene Essentiality Prediction. Chemistry & Biodiversity 4: 2618–2630.

3. Kim J, Kim I, Han SK, Bowie JU, Kim S (2012) Network rewiring is an important mechanism of gene essentiality change. Sci Rep 2.

4. (2014). Broad-dream gene essentiality prediction challenge. URL `https://www.synapse.org/#!Synapse:syn2384331`.

5. Vapnik VN (1998) Statistical Learning Theory. New York, NY, USA: John Wiley & Sons, Inc., 1st edition.

6. Smola AJ, Schölkopf B (2004) A tutorial on support vector regression. Statistics and Computing 14: 199–222.

7. Lauria M (2013) Rank-based transcriptional signatures: A novel approach to diagnostic biomarker definition and analysis. Systems Biomedicine 1: 228–239.

8. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, et al. (2011) Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12: 2825–2830.

9. Cherkassky V, Ma Y (2004) Practical selection of svm parameters and noise estimation for svm regression. Neural Netw 17: 113–126.