SURVIVAL ANALYSIS FOR ACTIVE MODULES USING LEARNED REPRESENTATIONS

Daniel Beck
beck.daniel@indivumed.com

Jobst Landgrebe
landgrebe.jobst@indivumed.com

Indivumed Therapeutics

INTRODUCTION

In the multivariate multiomic biomolecule analysis based detection of novel pharmacological cancer treatment targets in cancer and matching normal tissues, the identification of active modules has been an evolving trend since the mid 2010s. Such modules are models of sets of proteins which physically interact to perform molecular functions and participate in cellular processes driving cancer phenotypes. They can be detected from tumour (T) and matching normal (N) tissues by partitioning PPIs according to the degree of differential T/N biomolecule expression. Usually many such modules are identified, and this begs the question which modules are the most relevant for down-stream validation and targeting steps. Here we show an approach to utilise the progression free survival (PFS) data of the patients from whom the tissues were obtained to rank the modules. Active modules are of varying size, but contain up to 60 biomolecules of three to four omics-analysis modalities which prevents a classical Cox survival regression given that even in larger tissue collections, the number of available survival events is in the two digit or lower three digit range. We therefore use a three-layer Graph Convolutional Network (GCN) with a linear readout layer on top to embed the molecule variables of the modules into fixed-size, low-dimensional representations. For supervised learning purposes, the survival problem is reformulated as a logistic-regression framed binary problem (survival less or more than a data-determined threshold) to drive the procedure. The aim of this training is to enforce the relevant features to be represented in the embedding via the back propagation of the classification error. The output of the GCN is a module specific embedding of the tissue data. The resulting low-dimensional embedding is used as input for Cox proportional-hazards model.

CONSTRUCTING THE GRAPH CONVOLUTIONAL NETWORKS

From the module identification procedure we obtain a set of modules $\mathcal{M}_1, \ldots, \mathcal{M}_m$. Given the protein-protein interaction network (PPI) that was used for identifying the modules, the modules define subgraphs of the PPI. The PPI $\mathcal{G}$ can be interpreted as a graph with $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ the set of edges. Then the subgraph for module i is defined as $\mathcal{G}_i = (\mathcal{M}_i, \{(e_1, e_2) \,|e_1, e_2 \in (\mathcal{M}_i \times \mathcal{M}_i) \cap \mathcal{V}\})$. Each of these subgraphs $\mathcal{G}_i$ defines a module-specific graph convolutional network (GCN). A GCN consists of multiple layers and realizes a function that takes a feature vector for every node as input which is presented as the feature matrix X and produces node level vectors as output combined in the matrix Z. A simplified version of the layer-wise propagation rule for GCNs is shown below:

$$H^{(l+1)} = \sigma\left(AH^{(l)}W^{(l)}\right)$$

with $H^{(0)} = X$ and $H^{(L)} = Z$ where L is the number of layers. The matrix A is the adjacency matrix representing the subgraph $\mathcal{G}_i$, $W^{(l)}$ is the weight matrix of the l-th layer, and $\sigma(\cdot)$ is a non-linear activation function. The full version of the propation rule can be found in [1].

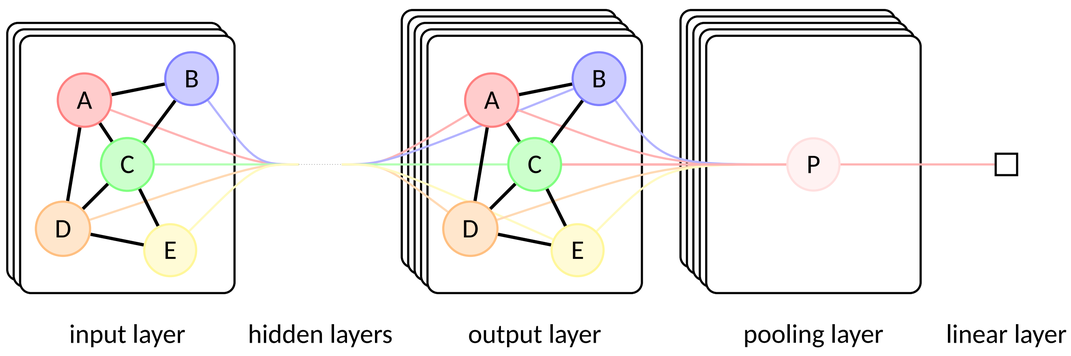

Figure 1: The network architecture is made up of three convolutional layers, a global mean pooling layer and a linear layer for the classification. By means of back-propagation of the classification errors we obtain a representation of the input at the pooling layer that exhibits features which are relant to the classification problem. In our case we predict whether a relapse occurs within the first 4.2 years or not and thus expose features which are relevant for progression-free survival.

TRAINING THE NETWORK

The training data for the GCN for module i is made up of tuples consisting of the input matrix X_k^i and the boolean classification result y_k where k ranges over the patients. The input matrix is constructed from values derived from the expression values for transcriptomics, proteomics, and phosphoproteomics for the patients' normal and tumor samples:

$$X_k^i = \begin{bmatrix} trx_{k,1}^t & \cdots & trx_{k,m_i}^t \\ trx_{k,1}^n & \cdots & trx_{k,m_i}^n \\ ptx_{k,1}^t & \cdots & ptx_{k,m_i}^t \\ ptx_{k,1}^n & \cdots & ptx_{k,m_i}^n \\ ppx_{k,1}^t & \cdots & ppx_{k,m_i}^t \\ ppx_{k,1}^n & \cdots & ppx_{k,m_i}^n \end{bmatrix}$$

where the columns corresond to the nodes in the network and thus to the genes in the modules. $trx_{k,m}^t$ is the TRX value for the tumor sample of patient k and gene m.

TRAINING THE NETWORK (CONTD.)

The intended outcome y_k is derived from the progression-free survival times recorded in the clinical data associated with multiomics data and is defined as:

$$y_k = \begin{cases} 1 & \text{if relapse occurs within first 4.2 years} \\ 0 & \text{otherwise} \end{cases}$$

The time threshold is chosen to yield a balanced training set, that is, for half of the patients a relapse occurs before and for the other half after the threshold or not at all. In total the training set contains 210 patients.
Our objective is to find reasonable, low-dimensional representations of the input data and we use the classification only as a way to incentivize the network to expose relevant features for survival. Consequently, we do not care about over-fitting during the training phase and thus use the whole dataset for training. The networks are trained for 300 epochs and the best model according to accuracy is chosen for generating the embeddings.

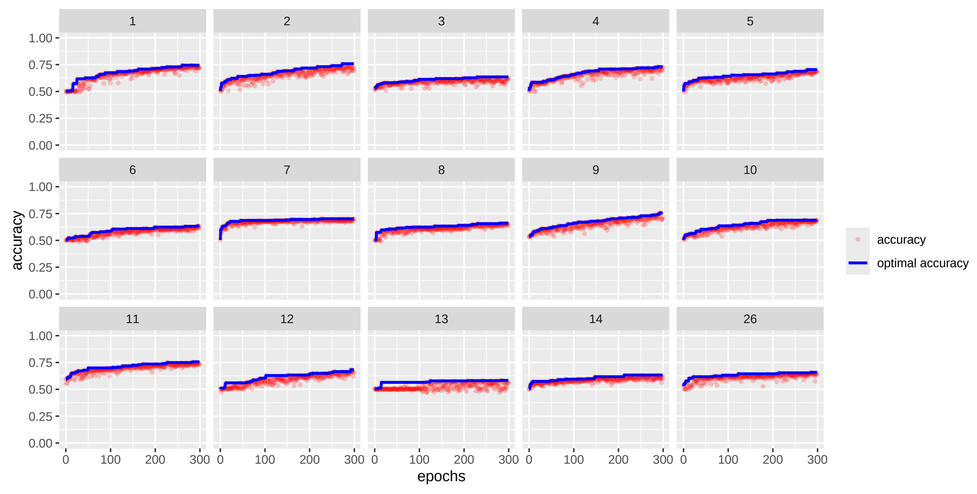

Figure 2: The graphs show the accuracy of the classification during training for a set of modules. The red dots indicate the accuracy achieved in the corresponding epoch; the best accuracy achieved during the training run is shown by the blue line. The classification results are mediocre at best but in a realistic range given the classification problem and the number of genes which are considered for each module.

SURVIVAL ANALYSIS

The components of the embedding vectors for each patient and module together with stage are used as covariates for the survival analysis. Stage shows to have a very high relevance for progression-free survival and thus we looked for embedding-covariates that are similiarly significant as stage, counted these and ranked the modules accordingly. With this heuristic we were generally able to identify 5-10% of the modules as highly relevant wrt. progression-free survival.

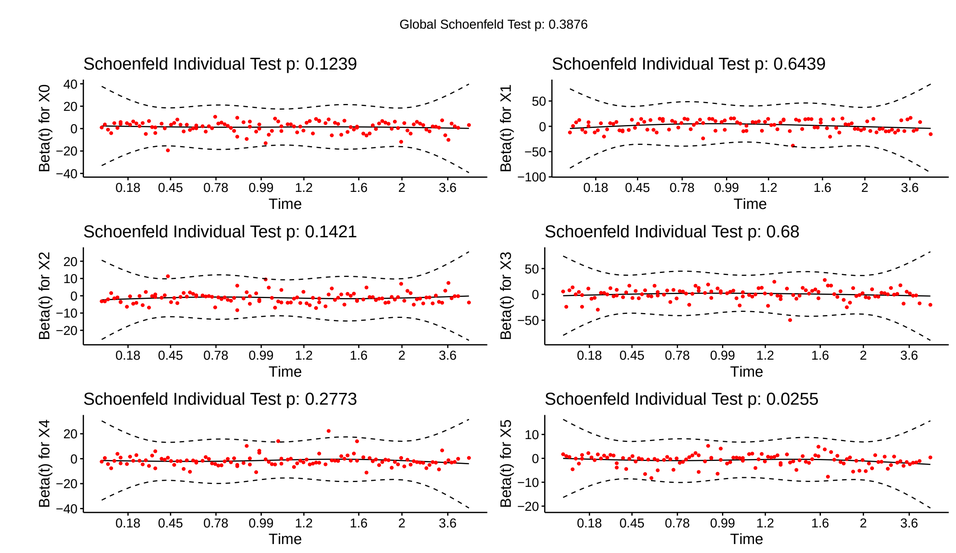

Figure 3: Analysis of the Schoenfeld residuals for the covariates of the Cox regression which correspond to the components of the embedding vectors. Mostly the proportionality assumptions are satisfied as shown by the non-significant p-values. If a covariant/embedding dimension proves to be time-dependent it is not considered for the heuristic used for the ranking of modules.

CONCLUSIONS

We present an approach that allows to find suitable low-dimensional embeddings representing the combined expression data for a set of genes. Since the dimension of the embedding vector is a independent of the number of genes in the module this allows to apply common analysis methods (e.g. Cox regression) in a multivariate setting by using the components of the embedding vectors as covariates. In our experiments we could find a large number of genes that are known cancer targets or are currently in various stages of clinical trials amoung the top ranked modules.

[1] Thomas N. Kipf and Max Welling. *Semi-Supervised Classification with Graph Convolutional Networks*. 2017. arXiv: 1609.02907. URL: https://arxiv.org/abs/1609.02907.