DNetPRO: A network approach for low-dimensional signatures from high-throughput data

Nico Curti
Department of Experimental, Diagnostic and Specialty Medicine, University of Bologna

Giuseppe Levi
Department of Physics and Astronomy, University of Bologna

Enrico Giampieri (enrico.giampieri@unibo.it)
Department of Experimental, Diagnostic and Specialty Medicine, University of Bologna

Gastone Castellani
Department of Experimental, Diagnostic and Specialty Medicine, University of Bologna

Daniel Remondini
Department of Physics and Astronomy, University of Bologna

Article

Keywords:

Posted Date: June 29th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1757975/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License
DNetPRO: A network approach for low-dimensional signatures from high-throughput data

Nico Curti\textsuperscript{a, †}, Giuseppe Levi\textsuperscript{b, c, †}, Enrico Giampieri\textsuperscript{a, *}, Gastone Castellani\textsuperscript{a}, and Daniel Remondini\textsuperscript{b, c}

\textsuperscript{a}Department of Experimental, Diagnostic and Specialty Medicine, University of Bologna
\textsuperscript{b}Department of Physics and Astronomy, University of Bologna
\textsuperscript{c}INFN Bologna
\textsuperscript{†}Both authors contribute equally to this work
\textsuperscript{*}Corresponding author

ABSTRACT

One of the main objectives of high-throughput genomics studies is to obtain a low-dimensional set of observables - a signature - for sample classification purposes (diagnosis, prognosis, stratification). We propose DNetPRO, Discriminant Analysis with Network PROcessing, a supervised network-based signature identification method. The algorithm is easily scalable, allowing efficient computing for high number of observables ($10^3$–$10^5$). We show applications on real high-throughput genomic datasets in which our method outperforms existing results or is compatible with them, but with a smaller number of selected features. Moreover, the geometrical simplicity of the resulting class-separation surfaces allows a clearer interpretation of the obtained signatures in comparison to nonlinear classification models.

Introduction

The huge dimensionality of omics data (e.g. microarray or NGS transcriptomics, epigenomics, SNP profiling, proteomics, metabolomics, metagenomics of gut microbiota) poses enormous challenges as how to extract useful information from them. One of the prominent problems is the identification of a low-dimensional set of features, called a “signature”, for classification and diagnostic purposes, for example to better stratify patients for personalized intervention strategies based on their molecular profile\textsuperscript{1–4}.

Many approaches are used for these classification purposes\textsuperscript{5}, such as Support Vector Machine, K-nearest Neighbor, Neural networks, Penalized regression (ridge, LASSO and Elastic Net)\textsuperscript{6} and Random Forest\textsuperscript{7}. Some methods build signatures by means of single-feature scoring methods\textsuperscript{8,9} (e.g. inferential testing for two-class comparison) but these approaches could fail even in simple 2-dimensional situations. An example is shown in Fig. 1a, in which both features perform poorly when taken individually, but their performance becomes optimal in a 2-dimensional combination, through a simple linear separation of the two classes. Other methods search for projections in a latent space, and then perform a dimensionality reduction by thresholding the projection weights, but these approaches can quickly completely negate the explainability of the results, and require an exceeding amount of the original features to be included.

It is known that complex separation surfaces characterize classification tasks associated to image and speech recognition, for which Deep Networks have been successfully applied in recent times. On the contrary, many biological data, such as gene or protein expression, are more likely characterized by an up/down regulation behavior (as shown in Fig. 1b top), while more complex patterns (e.g. a “windowed” optimal range of activity, Fig. 1b bottom) are much less common. Thus, discriminant-based methods (and logistic regression methods alike) could provide good classification performances in these cases if applied in at least 2-dimensional spaces to account for correlations. Moreover, the “linearity” of these methods (that generate very simple class separation surfaces, i.e., linear or quadratic) guarantees that the construction of a multidimensional signature based on 2-dimensional initial signatures (our feature pairs) is feasible.

A possible way to overcome these issues was introduced by Geman et al.\textsuperscript{10} via the Top Scoring Pair (TSP) classifier and its further refinements\textsuperscript{11,12} and extensions\textsuperscript{13}. The TSP algorithm is based on a bottom-up combinatorial approach that exploits the discriminant power of all feature pairs tailored for gene expression classification problems: TSP algorithm identifies pairs of features whose relative expressions/values are upturned between two classes, i.e., it tries to find couples of genes whose relative rankings are inverted in most samples of the two classes. The simplicity of the method guarantees an easy biological interpretation of the results, but it does not provide any criteria to combine several gene-pairs into a higher-dimensional signature.
DNetPRO - Discriminant Analysis with Network PROcessing - generates multivariate signatures starting from all the feature pairs, tested with Discriminant Analysis (ref. Fig. 2 and Supplementary Material for detailed method description), extending the backbone idea behind the TSP algorithm: a couple of omics features (e.g. gene, miRNA protein expression levels, etc.) constitutes two nodes of a network, and a link between them is created if their classification performance exceeds a selected threshold (see Methods Section). Given this set of links, every connected subnetwork generated results in a possible signature from the given feature set. Extensive exploration of all possible feature combinations (all K-tuples over N possible features) is known to be an NP-hard problem\cite{14}; the DNetPRO method is an attempt to overcome single feature selection without the computational burden of the full combinatorial exploration, with a computing time for feature space exploration proportional to the square of the number of features (ranging from $10^3$ to $10^5$ in a typical high-throughput omics study). Moreover, the geometrical simplicity of the resulting linear class-separation surfaces allows an easier interpretation of the results compared to very powerful methods like nonlinear-kernel SVM or Neural Networks that suffer from hard-to-explain decision boundaries. DNetPRO method belongs to the category of network-based algorithms, a class of methods recently applied for dimensionality reduction, visualization and clustering tasks.\cite{15-17}.

![Figure 1](image.png)

**Figure 1.** (a) An example in which single-feature classification performance fails in predicting higher-dimension classification performance. Both features (gene expression 1 and gene expression 2) badly classify in 1D but have a very good performance in 2D. Moreover, the classification can be easily interpreted in terms of relative higher/lower expression of both probes. (b) Activity of a biological feature (e.g. a gene) as a function of its expression level: top - monotonically increasing, often also dichotomized to an on/off state; bottom - “windowed” behavior, in which there are two or more activity states that do not depend monotonically on expression. X axis: expression level, Y axis: biological state (arbitrary scales).

**Results**

We tested the proposed DNetPRO algorithm on synthetic data and on real omics datasets of different types (mRNA, miRNA, and RPPA) in comparison with current state-of-art classifiers applied on public cancer gene expression datasets (Synapse datasets).\cite{18} To compare our results on the Synapse datasets, we used the AUC (Area Under the Curve) score, provided in the paper as the result of their analyses. All datasets were analyzed considering the pipeline proposed in Fig. 2.

**Synthetic data**

We compare the accuracy performance of a classical incremental feature selection algorithm, i.e., single feature selection based on ANOVA, with the proposed DNetPRO algorithm, on a synthetic toy model dataset (see Methods Section for its description). For both the features selection methods we used a diag-quadratic Discriminant Analysis for the evaluation of classification performances. The simulations were performed according to the procedure A showed in Fig. 2.

For the same number of features (Fig. 3a-c) one can see that the two methods perform quite similarly, but the DNetPRO obtains better performances as the number of samples increases.

Despite the simplicity of our toy model, the DNetPRO highlights its efficiency in terms of performances against the single-feature method. A slightly different behavior is shown by varying the number of features and keeping fixed the number
Figure 2. Scheme of the DNetPRO algorithm. On the “Training set”, all possible pairs of features are used for Discriminant Analysis, generating the fully connected network with links weighted by pair classification performances. By thresholding the weighted links, one or more signatures are identified as connected components, and then their performance is evaluated on the “Whole Test set” (procedure A). A unique best signature can be identified on a “Validation set” and tested in a “Scoring set”, obtained by further splitting the “Whole Test set” (procedure B).

Figure 3. Synthetic dataset simulation. Comparison of accuracy performances obtained by the DNetPRO algorithm and the K-best algorithm. (a) Performances obtained in function of the number of samples, keeping fixed the number of features (4000 features). (b) Performances obtained in function of the number of features, keeping fixed the number of samples (500 samples). (c-d) Differences between the performances obtained by the DNetPRO and the K-best algorithms on the same simulations.
of samples (Fig. 3b). In this case DNetPRO always outperforms the \( K \)-best algorithm in terms of median accuracy (black line in the plot, ref. Fig. 3d). With a small number of features (left part of the plot) the \( K \)-best algorithm performances are more stable, and we can notice how the medians of the distribution related to the DNetPRO results are still higher compared to the \( K \)-best ones. As the number of features increases, the efficiency of the DNetPRO algorithm also increases, until it exceeds the \( K \)-best algorithm (and its distribution is narrowed). We reached this situation quite rapidly in our simulations, since we constrained our toy model with a forced unbalance between the number of samples and features, i.e., the so-called ill-posed problems. Moreover, DnetPRO allows to identify features with good performance in couples, but with low ranking singularly, highlighting their synergistic behavior, also useful for a biological interpretation of the resulting signature.

## Synapse dataset

TCGA (The Cancer Genome Atlas) core sets of data used are available at the Synapse homepage (accession number syn300013, doi:10.7303/syn300013) created by Yuan et al.\(^\text{18}\), and are composed of four cancer datasets: kidney renal clear cell carcinoma (KIRC), glioblastoma multiforme (GBM), ovarian serous cystadenocarcinoma (OV) and lung squamous cell carcinoma (LUSC) (see Material and Methods and Supplementary Materials for further details).

For each cancer type we analyzed the mRNA, miRNA, and RPPA datasets, performing the classification of dichotomized outcomes\(^\text{18}\) via the same train/test subdivisions provided by Yuan et al. We report the results in terms of maximum AUC (ref. Fig. 4[a, c, e]) and AUC distributions (ref. Fig. 4[b, d, f]). We show the results obtained with DNetPRO according to the procedure \( A \) (corresponding to the validation approach used by Yuan et al.) and procedure \( B \) (full double cross-validation procedure to avoid data contamination) in comparison with a series of more complex methods based on single-feature evaluations as proposed by Yuan et al.

![Figure 4](image)

**Figure 4.** Results obtained by DNetPRO on the mRNA, miRNA, and RPPA samples related to the four cancer types in the Synapse dataset. [a, c, e] Comparison of the DNetPRO results with the methods used in the work of Yuan et al., in terms of the maximum AUC value obtained on a 10-fold cross-validation procedure (bold red: top-performing method). [b, d, f] Distributions of the AUC values related to each analyzed dataset. Green boxplots: results using procedure \( A \); yellow boxplots: results using procedure \( B \). The results obtained by the two procedures are not directly comparable due to the different data subdivisions and therefore the two distributions are plotted individually.

The results on the mRNA datasets using procedure \( A \) are comparable (LUSC) or better (KIRC, GBM) than the results reported in\(^\text{18}\), except for the OV dataset. This ranking is maintained even with the more conservative procedure \( B \), involving a further cross-validation step. The size of the extracted signatures is approximately constant between cross-validations, and typically smaller than 500 genes in each pipeline execution.

Performances decrease with the introduction of the second cross validation step (procedure \( B \), as expected, but they remain...
The validation procedure used in the reference paper by Yuan et al. corresponds to our approach without the second validation step (procedure A).

The results obtained on the miRNA datasets are comparable to the reference ones, while for the RPPA datasets only the LUSC shows AUC values comparable with the others. Moving from procedure A to procedure B, i.e., adding a second cross-validation step, RPPA performances drastically decrease for the KIRC and OV, while they remain stable for the LUSC dataset. The same behavior is shown in the miRNA datasets, in which however both performances are still comparable or better (KIRC, GBM, LUSC) than the reference ones.

From these analyses it can be seen that, given a similar (or better) classification performance, DNetPRO allows a simple biological interpretation of the identified signature, in terms of up/down regulation of its elements. The use of a network-based approach for the merging of top scoring pairs provides an automated (and natural) solution for the gene-pairs combination, extending the classical TSP algorithm.

**Signature overlap**

In our analysis on the Synapse dataset, we used a complex pipeline of cross-validations (ref Fig. 2) to obtain a sufficient sample size: we replicated our simulation for 100 repetitions, extracting a set of 1000 totally independent signatures. As a case study, we focused on the KIRC mRNA dataset, in which the extracted signatures ranged from 4 to 654 genes (and an average of \( \mu = 382 \) genes). For each gene we counted its occurrences among the 1000 signatures. The same analysis was performed considering the signatures generated using the \( K \)-best score features (based on ANOVA) and a random feature extraction (as a null model).

Both DNetPRO and \( K \)-best identified a core set of genes common to all the signatures, significantly differing from the null model.

We observed a set of 74 genes in 90% of DNetPRO signature (20 of which are common to all signatures). This list of 74 genes was mapped into the TISIDB\(^1\) on-line database, confirming the relationship between most of them and KIRC tumor (see Supplementary Material for further details).

**Discussion**

In this work we proposed a network-based feature extraction method tailored to omics data, the DNetPRO algorithm, that combines top ranking pairs of omics entities (e.g. gene transcripts or proteins) into a multidimensional classification signature, allowing a simple biological interpretation of the results in terms of up/down regulation. We tested our method on synthetic data, showing how its efficiency increases on ill-posed problems (similar to those encountered in omics analysis) in comparison with classical incremental feature selection methods. Moreover, the proposed DNetPRO method was also tested on benchmark real datasets, with results in general better or comparable with state-of-art nonlinear methods. The method is easily scalable on parallel architectures, allowing fast processing of high-dimensional data (in the order of \( 10^4 \) elements in 1 minute on server grade machines, see Supplementary Material) and a version of the algorithm is publicly available on Github\(^2\).
Methods

**DNetPRO algorithm**

The pseudo-code of the proposed DNetPRO algorithm could be sketched as:

**Data:** Data matrix \((N, S)\)

**Result:** List of putative signatures

Divide the data into training and test by a Hold-Out method;

\[
\text{for} \quad \text{couple} \leftarrow (\text{feature}_1, \text{feature}_2) \in \text{Couples} \quad \text{do}
\]

Leave-One-Out cross validation;

Score estimation using a Classifier;

end

Sorting of the couples in ascending order according to their score;

Threshold over the couples score \((K\)-best couples); 

\[
\text{for} \quad \text{component} \in \text{connected\_components} \quad \text{do}
\]

\[
\text{if} \quad \text{reduction} \quad \text{then}
\]

Iteratively pendant node removal;

end

\[
\text{else}
\]

continue;

end

Signature evaluation using a second Classifier;

end

**Algorithm 1:** DNetPRO algorithm for Feature Selection.

Given a dataset, consisting of \(S\) samples (e.g., cells, patients) with \(N\) observations each (our features, e.g., omics measurements such as gene, protein or metabolite expression) the signature identification procedure is summarized with the following pipeline:

- Separation of available data into a training and a test set (typically 66/33, or 80/20).
- Estimation of the classification performance according to the desired metric on the training set of all \(S(S - 1)/2\) feature pairs through a computationally fast and reproducible cross-validation procedure (leave-one-out cross validation was chosen). The results are mapped into a completely connected symmetric weighted network, with nodes corresponding to features and link weights corresponding to performance of the node couples.
- Selection of top-performing pairs through a hard-thresholding procedure, that removes links (and nodes) from the initial completely connected network: every connected component obtained is considered as a putative classification signature. The threshold value can be tuned according to a desired minimum-performance value or considering a minimum number of nodes/features in the signature. The threshold value can be determined also via cross validation of the entire signature extraction procedure.
- (Optional) In the hypothesis that node degree is associated to the global feature performance in combination with the other features, to reduce the size of an identified signature, the pendant nodes of the signature network, i.e., nodes with degree equal to one, can be removed. This procedure can be applied once, or recursively until the core network, i.e., a network with all nodes with at least two links, is reached. We have tested the efficacy of this empirical approach in some real cases\(^{21,22}\), obtaining a smaller-dimensional signature with comparable performance, even if there is not a solid theoretical basis supporting this procedure.
- All signatures are applied onto the test set to estimate their performance, producing more than one final signature. This corresponds to procedure \(A\) in Fig. 2.
- To identify a unique best performing signature, a further cross validation step can be applied, with a further dataset splitting into training (to identify the multiple signatures), test (to identify the best signature) and validation set (to evaluate the best signature performance). This corresponds to procedure \(B\) in Fig. 2.

To test the performance of all feature pairs, we used a diag-quadratic Discriminant Analysis, a robust classifier that allows fast computation. We remark that the signatures have a purely statistical relevance, being generated with a purpose of maximal classification performance, but previous applications of a simplified version of the DNetPRO algorithm\(^{1,23–25}\) allowed to gain knowledge on the biological mechanisms associated to the studied phenomena.
A variant of DNetPRO method has been also applied for dimensional reduction of network structures, where sub-modules of the network were identified by studying the correlation between links\textsuperscript{26,27}. Further information about the implementation of the algorithm are available in the Supplementary Material. DNetPRO code is publicly available on Github\textsuperscript{20} as C++ library and Python module.

**Synthetic data**

The most common feature selection algorithms usually treat features as individual and independent entities. Starting from the features ranked according to their scores, a signature is obtained selecting the top ones according to an incremental addition of features until a desired output performance is reached. These kinds of methods are called $K$-best algorithms and they select the features without any information on their mutual interaction or correlation. The proposed DNetPRO algorithm tries instead to extract the more statistically significant features considering the interaction between them, i.e., their combination in pairs for a 2-dimensional discriminant supervised classification.

To compare the two methods on a synthetic dataset, we used a toy model generator provided by the scikit-learn package, generating normally-distributed clusters of points and introducing interdependence between the features. The model generator creates clusters of points normally-distributed about vertices of a pre-determined number of informative dimensional hypercube and assigns an equal number of clusters to each class (2 in our case). The model generator allows to set the number of sample classes, distinguishing between informative features, i.e., features which easily separate the class populations, and non-informative features, i.e., features which represent noise in our problem. The number of informative features should be realistically small compared to the noise, so in our simulations we chose to introduce a maximum of 1\% informative features in each simulation.

We randomly generated data from Gaussian uncorrelated distributions with an increasing number of samples and features, i.e., dimensions. We want to remark that this configuration of data would tend to prefer the single feature methods like the $K$-best one: the inclusion of noise sources and a high number of dimensions would stress the $K$-best efficiency, representing a good benchmark for the DNetPRO application. In each simulation we split the number of samples in training and test sets (Hold-Out method, with 66\% of data as training and 33\% as test) and we applied the DNetPRO algorithm. From each simulation we tested the extracted signatures on the test set, keeping the best performing one. On the same data-subdivision we applied the $K$-best algorithm, filtering the same number of features of the DNetPRO best signature, i.e., $K$ equal to the number of nodes in the DNetPRO best signature. In this way, we can compare the performances obtained on the test set by the two methods, using the same number of features/dimensions for both the algorithms. We would highlight that, in general, there is no stop criteria for the $K$-best algorithm, so the number of features selected could be smaller or greater than the number of DNetPRO signature nodes. However, we can reasonably assume that, according to the $K$-best interpretation, the selected features should be the best performing ones, and the addition of more features should introduce only an increasing amount of noise. We used as threshold criteria for the DNetPRO algorithm a maximum number of features: keeping the top scoring pairs, we progressively added groups of features until a maximum number of 100 pairs was reached (ref. Supplementary Materials for a description of ranked distributions). We intentionally did not tune the threshold parameter of the DNetPRO algorithm to keep the comparison of the two methods unbiased.

**Synapse dataset**

We processed each cancer dataset by adding a zero-mean Gaussian random noise ($\sigma = 10^{-4}$) to remove possible zero values, which could produce numerical errors in the evaluation of the distance between gene/protein expression values. Then, we randomly split each dataset in training and test sets with a stratified (i.e., balanced for class sample ratio) 10-fold procedure: with this stratification each training set is representative of the whole dataset. The choice of a 10-fold splitting is aimed to reproduce the analysis pipeline presented by Yuan et al.\textsuperscript{18} with an analogous cross-validation procedure. Since we do not have exact details of their data splitting, the cross-validation was repeated 100 times, for a total of 1000 training procedures for each tumor (OV, LUSC, KIRC, GB) and data type (mRNA, miRNA, RPPA). Each training procedure led to the extraction of multiple signatures.

We chose threshold values to obtain a resulting number of features in the signatures in the order of $10^2 - 10^3$. According to each dataset, an appropriated threshold was estimated to achieve this requirement. If more than one connected component existed, each one was considered as a different signature.

The final multidimensional signatures were tested by a nonlinear Discriminant Analysis with a diag-quadratic distance, to avoid possible problems deriving from covariance matrix non-invertibility (as for the Mahalanobis distance in which there is a higher number of coefficients to be estimated from the data).

**Cross validation procedures A and B**

To allow a fair comparison with the results presented by Yuan et al.\textsuperscript{18} we performed an identical cross-validation procedure, referred to as PROCEDURE A. This procedure however does not allow an unbiased comparison of the signature performance,
lacking a second cross-validation step. To obtain such an unbiased performance estimation we also proceeded to evaluate the results using a double cross-validation approach, referred to as Procedure B.

In the single cross-validation pipeline (procedure A, ref. Fig. 2) the best signature was extracted as the one reaching the highest accuracy score during the training step. This best signature was then tested over the available test set. The introduction of the second cross-validation step (procedure B in Fig. 2) led to choose the best signature as the one with the best performances over a subset of the whole test set (renamed test set), evaluating the final performance on the remaining validation set.

Signature overlap

The DNetPRO algorithm can provide more than one signature as outcome, given by the various connected components found in the feature network, and a unique top-performing signature can be obtained by a further cross-validation step (procedure A and B in Fig. 2, respectively).

In our applications, we divided the datasets into a training-test subdivision and the signatures were extracted along a 10-fold cross-validation over the training sets. This kind of setup could, in the worst case, extract up to 10 totally different signatures (one for each split).

Starting from this large number of signatures, we evaluated the robustness of the DNetPRO algorithm in the feature identification, studying the overlap between them. From a statistical point-of-view, it is quite unlikely that the same set of features would be included into all the extracted signatures, especially on this application in which features represent gene expressions. On the other hand, the overlap of these signatures could highlight a statistical significance of some features, and thus genes related to the associated tumor.

For each fold we evaluated the average dimension of the signatures, and we computed the distribution of these dimensions for each cancer type (see Supplementary Material for further details).

References


**Acknowledgements**
The authors acknowledge IMI-2 HARMONY n. 116026 EU Project and IMforFUTURE Horizon 2020 (EU) Project.

**Author contributions statement**
N.C.: Formal analysis; N.C., G.L., and E.G.: Methodology and Software; E.G., G.C., and D.R.: Supervision and Conceptualization; G.C. and D.R.: Project administration; All the authors reviewed the manuscript.

**Competing interests**
The authors declare no competing interests.

**Availability of Data and Materials**
The synthetic datasets used for the analyses are available from the corresponding author on reasonable request. The Synapse dataset used for the analyses is available at the Synapse homepage (accession number syn300013, doi:10.7303/syn300013). The code for the reproducibility of the results is available on Github20.

**Additional information**
*Correspondence* and requests for materials and methods should be addressed to E.G. at enrico.giampieri@unibo.it.
Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- supplementaryScientificReports.pdf