Supplementary Notes for "Learning sequence patterns of AGO-sRNA affinity from high-throughput sequencing libraries to improve *in silico* functional small RNA detection and classification in plants"

Lionel Morgado*, Ritsert C. Jansen and Frank Johannes

S1 Supplementary Notes

S1.1 Support Vector Machines

The Support Vector Machine (SVM) is a popular machine learning algorithm with strong bonds to statistical learning theory. During the training phase, the SVM searches for the decision hyperplane that maximizes the margin for the training set $(\vec{x}_1, y_1)$ ... $(\vec{x}_i, y_i)$ composed of n training instances $\vec{x}_i$ with labels $y_i \in \{-1, 1\}$ in 2-class problems. Maximum margin classifiers are frequently the ones that guarantee the highest generalization capacity, which explains the higher performance of models obtained via SVM compared with other machine learning algorithms (1).

To attain a classifier of the form $\vec{w} \cdot \vec{x} + b = 0$, the learning process demands to minimize the norm of the decision hyperplane normal vector $\|\vec{w}\|$ subject to the function $y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1$, for $i = 1, \ldots, n$. A classification is then given by the signal of the output from the classifier, this in the binary case.

To accommodate points that lay in the wrong side of the decision frontier, a soft-margin classifier can be trained instead of a hard-margin version, considering a hinge loss function of the type

$$\zeta_i = max\left(0, 1 - y_i(\vec{w} \cdot \vec{x}_i + b)\right)$$

, and minimizing

$$\frac{1}{n}\sum_{i=1}^{n} \zeta_i + \lambda\|\vec{w}\|^2$$

, subject to $y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1 - \zeta_i$ and $\zeta_i \geq 0$, for all i. Variable λ defines the tradeoff between having a larger margin and the number of training points lying in the wrong side of the margin. For $\lambda = 0$, the soft margin formulation becomes the hard-margin version. Most frequently, the trade-off is tuned not through λ but a directly related cost parameter $C = \frac{1}{\lambda}$. The above equation states the optimization problem in a form known as the primal. Classically, it can be solved using quadratic programming, but there are other more sophisticated approaches such as the sub-gradient descent or coordinate descent methods that can reduce the training time several orders of magnitude when learning from very large datasets.

Alternatively, the optimization problem can be stated in the dual form

$$maximize\ f(\alpha_1 \ldots \alpha_n) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{i=1}^{n} y_i \alpha_i (x_i \cdot x_j) y_j \alpha_j$$

, subject to $\sum_{i=1}^{n} \alpha_i y_i = 0$ and $0 \leq \alpha_i \leq \frac{1}{2n\lambda}$, for all i; and can be solved through the Lagrangian method. In this case the decision hyperplane is described in terms of the training instances $\vec{x_i}$ that lie closer to it and for which the Lagrangian multiplier $\alpha_i \neq 0$, being $\vec{w} = \sum_{i=1}^{n} \alpha_i y_i \vec{x_i}$. These instances placed in the margin are called support vectors (SV).

When more complex relationships between the classes are encoded in the data, it is beneficial to use a search space with higher dimensionality where the discrimination is more feasible. Data can be projected into such space using non-linear functions classically employed in machine learning like polynomials, sigmoids or radial basis functions (RBF) (2, 3). RBF are a special and common choice, since other kernel functions like the linear and the sigmoid can be designed as special cases of RBF (2, 3).

S1.2 SVM-RFE

Recursive Feature Elimination (RFE) is a backward feature elimination method that can be defined in three simple steps:

1. Training of a classifier (optimization of weights w_i according to an objective function J)
2. Determine the ranking criteria for all features (the cost $DJ(i)$ or ${w_i}^2$)
3. Remove the feature(s) with the smallest ranking criterion

The whole process must be repeated until a stopping criterion is met, such as achieving a minimum number of features desired, an empty feature set or other.

In the case of SVM-RFE, the principles of RFE are combined with SVM model optimization in a wrapper approach.

S1.3 SVM learning in very large datasets

SVM learning is frequently mentioned as one of the most accurate machine learning methods, thanks to the maximum margin principle that it employs. However, searching for the maximum margin hyperplane in a large dataset comes with a computational burden from loading large matrices for which it is necessary to solve extensive optimization problems. This makes traditional SVM formulations with a training complexity dependent on the number n of training samples in between $O(n^2)$ and $O(n^3)$ (4), of difficult use when exploring large sets. The machine learning community has been struggling to overcome such adversity, mostly by optimizing the mathematical background of SVM learning or by presenting “out of the box” solutions based on the theory behind SVM learning and experimental observations.

Empirical studies showed that linear SVM learning can be in some cases sufficient and allows to obtain classifiers with accuracy values comparable with the ones achieved using more complex kernel transformations. There are solvers specialized in linear SVM learning, such as LibLinear (5) with a training time complexity of $O(n)$, for which the training time grows linearly with the number of training instances and that therefore has a drastically reduced learning time when comparing to other SVM formulations. Additionally, using such approach does not demand optimizing any other

hyperparameters beyond the cost, further decreasing the training rounds necessary to tune the algorithm.

When non-linear SVM learning is necessary, the computational burden can be attenuated by using special learning schemes, such as the cascade configuration (6), which take advantage of theoretical properties of the SVM algorithm. This architecture breaks the global optimization process into smaller subproblems that are solved separately by multiple SVM with subsets of data. The partial solutions composed by the support vectors can then be joined and reprocessed in a cascade of SVM that end in a single model guaranteed to converge to the global optimum after multiple passes through the cascade. Empirical tests showed that a single pass over the cascade can already produce decision rules with good generalization. In addition, this design can be parallelized further dropping the training time.

SVM formulations for multiclass problems are available. However, since the complexity of the problem increases as the number of classes are included in the same optimization problem, multiclass SVM classification uses typically binary models. Two flavors can be found: one-vs-one or one-vs-all (7). In the first case, binary classifiers are trained for each of the pairwise combinations among the K classes, forming a total of $\frac{K(K-1)}{2}$ classifiers (1). One-vs-all schemes use less classifiers since each of the classes is trained to separate class i from all remaining groups, so a total of K. A final prediction can be obtained by combining the inference of each binary classifier through mechanisms as simple as an unweighted voting system, where a final decision is made in favor of the group that gathers the highest number of positive predictions from all the individual rules. Multiclass predictions through binary schemes is proven to achieve accuracy levels comparable to that of straight multiclass models (1), plus it brings extra advantages associated with the parallelization of the training process and its ease of implementation.

S1.4 Cascade SVM

The cascade SVM (6) is a learning architecture in which training is performed sequentially in subsets of data. Each partial SVM filters uninformative data by finding sub-solutions that through the cascade are sequentially joined in a global solution. Former SVM sub-solutions which are given by the respective SV are fed as input to a later SVM that further simplifies the result in case of data redundancy. This hierarchy ends-up in a single SVM that joins the SV from previously optimized problems in a final model.

This divide-and-conquer strategy which explores principles of SVM learning in the dual form, allows reducing the number of training points used to find the global solution by dealing with smaller and therefore computationally more manageable subsets of data. The computational problems faced by the SVM algorithm when dealing with large datasets are circumvented since the training complexity is dependent on the number n of training samples and estimated between $O(n^2)$ and $O(n^3)$, with obvious gains for smaller datasets.

S1.5 Motif analysis

All sRNA sets used to develop the classifiers here presented were subject to a motif search to clarify the eventual presence of conserved patterns in the sRNA sequences. This task used the MEME-ChIP platform (8) and was directed towards the list of unique sRNA in the “AGO”, “noAGO” and AGO-IP sets. In the beginning, a *de novo* motif discovery search was executed for long ungapped motifs using MEME (6 to 27 nt) and short ungapped motifs with DREME (3 to 8 nt). The motifs found to be significant (e-value≤0.05) were then analysed for enrichment in terms of their relative position inside the sequence with CentriMo (e-value≤0.05).

Next, all motifs found were aligned to two libraries of known motifs from *Arabidopsis thaliana*, using Tomtom (e-value≤1). The first library contained a set of 113 motifs detected by protein-binding microarrays (PBM, 9) and the second one is composed by 872 motifs captured by DNA affinity purification sequencing (DAP-seq, 10).

Finally, the motifs from each set were compared pairwisely to identify library specific motifs and shared ones.

The k-mers selected by SVM-RFE were matched with perfect similarity (no mismatches, no gaps) to subsequences retrieved with FIMO (p-value<0.0001) using the motifs previously detected against each sRNA set from which they derived. For a matter of simplicity and to avoid ubiquitous matches, only 5 nt k-mers were used and matched against the sRNA subsequences. To get a more reliable set of matches, we compared the probability to get a certain 5-mer by chance against the probability of finding the segment in the motif where that 5-mer matches. The probability of getting a given k-mer by chance was determined using the frequency f_i for each of the nucleotides $i = \{A, C, G, T\}$ in the *A. thaliana* genome, and then dividing it by the total number of nucleotides in the genome. These values were then combined to get a probability for each k-mer through the product of the probabilities for each of the letters in the specific 5-mer. The motif-associated probability was determined in similar way but using the position-specific probability matrix from the mapping motif. K-mers with a motif probability lower than random were left behind.

Supplementary References

1. Hsu,C.-W. and Lin,C.-J. (2002) A comparison of methods for multi-class support vector machines. IEEE Trans. Neural Networks, 13, 415-425.

2. Keerthi,S.S. and Lin,C.-J. (2003) Asymptotic behaviors of support vector machines with Gaussian kernel. Neural Computation, 15, 1667-1689.

3. Lin,K.-M. and Lin,C.-J. (2003) A study on reduced support vector machines. IEEE Trans. Neural Networks, 14, 1449-1559.

4. Bottou,L. and Lin,C.-J. (2007) Support Vector Machine solvers. in large scale Kernel Machines. Weston editors, MIT Press, Cambridge, MA, pp. 1– 28.

5. Fan,R.-E., Chang,K.-W., Hsieh,C.-J., Wang,X.-R. and Lin,C.-J. (2008) LIBLINEAR: A library for large linear classification. J. Machine Learn. Res., 9, 1871-1874

6. Graf,H., Cosatto,E., Bottou,.L., Dourdanovic,I. and Vapnik,V. (2004) Parallel support vector machines: the cascade SVM. In NIPS: 521-528.

7. Knerr,S., Personnaz,L. and Dreyfus,G. (1990) Single-layer learning revisited: a stepwise procedure for building and training neural network. Neurocomputing: Algorithms, Architectures and Applications, NATO ASI, Berlin: Springer-Verlag.

8. Machanick,P. and Bailey,T.L. (2011) MEME-ChIP: motif analysis of large DNA datasets. Bioinformatics, 2712, 1696-1697.

9. Franco-Zorrilla,J.M., López-Vidriero,I., Carrasco,J.L., Godoy,M., Vera,P. and Solano,R. (2014) DNA-binding specificities of plant transcription factors and their potential to define target genes. Proc. Natl. Acad. Sci. U.S.A., 111, 2367-2372. doi: 10.1073/pnas.1316278111.

10. O'Malley,R.C., Huang,S.S., Song,L., Lewsey,M.G., Bartlett,A., Nery,J.R., Galli,M., Gallavotti,A., Ecker,J.R. (2106) Cistrome and epicistrome features shape the regulatory DNA landscape. Cell, 165, 1280-1292.