CoVIRA: Consensus by Voting with Iterative Re-weighting based on Agreement

Frederico Schmitt Kremer, André Alex Grassmann & Luciano da Silva Pinto

Laboratório de Bioinformática e Proteômica, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, Capão do Leão, Rio Grande do Sul, Brazil

contact: fred.s.kremer@gmail.com

About

CoVIRA (Consensus by Voting with Iterative Re-weighting based on Agreement) is an algorithm to generate a consensus prediction based on the results from independent binary predictors by using an unsupervisioned and iterative weighted-voting system. Different from supervisioned ensemble learning methods, such as AdaBoost, CoVIRA doesn't require a dataset for training, as it is based on the measure of the "agreement" between the predictor. As we have no prior knownledge of accuracy of each predictor, we assume that the most accurate will be confirmed more times than those with smaller accuracy, requiring that the degree of accuracy in the set of predictors must be uniform. The algorithm was empirically designed on top of four logical postulates:

- Each predictor may have it's own accuracy. Therefore, an naive voting may be more likely to generate dubios results if compared to an weighted voting.
- The more predictors are used, more likely it is to achieve a reliable prediction by voting.
- The weight a predictor receives in the voting must be higher if its results where confirmed by other programs, and lower if it tends to deviate from the majority of the of the programs.
- The accuracy of a predictor is proportional to the weight it receives.

Table 1. Example of a dataset with results from three different predictions for the same protein.

Protein ID	Predictor 1	Predictor 2	Predictor 3
LIC10010	1	0	0
LIC10011	0	1	0
LIC10024	1	1	0
LIC10125	1	0	0
LIC10307	1	0	0
LIC10371	1	0	0
LIC10468	1	0	0
LIC10647	1	0	1

Note: The dataset file must not contain the header row. See the file in the `TEST` directory for more examples.

In this example it is possible to see that the results from the Predictor 1 were confirmed by the Predictors 2 and 3 only for a few proteins. Therefore, it is expected that its weight in the final voting might be relatively smaller than the weights of the other programs. When running CoVIRA whith this dataset set, the weight calculated for each predictor is:

- Predictor 1: 0.1875
- Predictor 2: 0.375
- Predictor 3: 0.4375

Based on the weights calculated for each predictor it is possible to perform an weighted voting and select those results that are more likely to be negatives or positives (eg:threshold = 0.5).

Protein ID	Predictor 1	Predictor 2	Predictor 3	CoVIRA score
LIC10010	1	0	0	0.1875
LIC10011	0	1	0	0.375
LIC10024	**1**	**1**	**0**	**0.5625**
LIC10125	1	0	0	0.1875
LIC10307	1	0	0	0.1875
LIC10371	1	0	0	0.1875
LIC10468	1	0	0	0.1875
LIC10647	**1**	**0**	**1**	**0.625**

It is also possible to use the weights calculated by CoVIRA to idenetify those predictors that are more likely to generate erroneous or dubious results (eg: Predictor 1) or those that are more likely to be confirmed by other programs (eg: Predictor 3), which can also be seen as a kind of "centroid" predictor in the ensemble of programs.

Using

The input must be a tab-delimited text file where the first field is an unique identifier of the entry (eg: protein ID), and the others are predictions generated by independent programs. The `test` directory contains some examples of inputs.

Two scripts are included as part of the CoVIRA package: `covira.py` (calculates the weight of each predictor) and `covira_score.py` (calculates the score for each entry in the original dataset based on the scores calculated by `covira.py`).

covira.py: results to STDOUT

```
$ python covira.py -i dataset.txt
```

covira.py: results to file

```
$ python covira.py -i dataset.txt -o weights.txt
```

covira_score.py: results to STDOUT

```
$ python covira.py -i dataset.txt -w weights.txt
```

covira_score.py: results to file

```
$ python covira.py -i dataset.txt -w weights.txt -o scores.txt
```

covira_score.py: results to file ranked by score

```
$ python covira.py -i dataset.txt -w weights.txt -o scores.txt -r
```