Supplementary Material

# Details of XGBoost Algorithm:

The details of XGBoost algorithm is introduced as follows ([Chen T and Guestrin C, 2016](#r11xgboost)):

Integrate the tree model with addition method, assuming a total of K trees, and use *F* to represent the basic tree model, then:

 (1)

The objective function is:

 (2)

where $l$ is the loss function, which represents the error between the predictive value and the true value;$Ω$ is the function used for regularization to prevent overfitting:

 (3)

where$ T $represents the number of leaves per tree, and *w* represents the weight of the leaves of each tree.

$\hat{y}\_{i}^{(t)}$ represents the predictive value after the $t\_{th} $iteration, then:

 (4)

Therefore, the objective function can be expressed as:

 (5)

The second-order Taylor expansion of the objective function is:

 (6)

where:

 (7)

 (8)

Ignore constant term, (6) can be written as:

 (9)

Define $I\_{j}=\left\{i\left|q(x\_{i})=j\right.\right\}$ as the $j\_{th} $leaf node and (9) can be written as:

 (10)

When the derivative of the objective function is equal to 0 , the optimal weight is:

 (11)

When $w\_{j}=w\_{j}^{\*}$, the objective function is:

 (12)

XGBoost uses a greedy algorithm instead of an enumeration method to improve computational efficiency. $I\_{L} $represents the set of all left nodes after each split, and $I\_{R}$ is the set of all right nodes after each split, then after each split, the information gain of the objective function is:

 (13)

**Data**

The data used in our paper was kindly provided by the author of paper “Gene expression inference with deep learning”, and we don't have the ownership of the data. The readers who want to access the data can contact the author of this article.

**References:**

Chen, T., and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785-794). ACM.[doi:10.1145/2939672.2939785](https://doi.org/10.1145/2939672.2939785)

Chen, Y., Li, Y., Narayan, R., Subramanian, A., and Xie, X. (2016). Gene expression inference with deep learning. *Bioinformatics*, *32*(12), 1832-1839. [doi: 10.1093/bioinformatics/btw074](https://doi.org/10.1093/bioinformatics/btw074)