AGEAS: Automated Machine Learning based Genetic Featrue Extraction System

Jack Yu[1,2,*1,+], **Masayoshi Nakamoto**[2,+], **and Jiawang Tao**[1,*2,+]

[1]Center for Health Research, Guangzhou Institutes of Biomedicine and Health, Chinese Academy of Sciences, Guangzhou 510530, China
[2]Shenzhen Mozhou Technology Co., Ltd, Shenzhen, China
[*1]Correspondence: gyu17@alumni.jh.edu
[*2]Correspondence: tao_jiawang@gibh.ac.cn
[+]these authors contributed equally to this work

ABSTRACT

TBD

Introduction

TBD

Method

The basic principle of *AGEAS* is to find key genetic features regulating phenotype of interest through analyzing how well-performing classification models distinguish gene regulatory networks (GRNs) of sample with the phenotype from GRNs of samples without. To reconstruct sufficient GRNs for each class, RNA-Seq based expression data is segmented into subsets while each one is analogized as a pseudo-sample having discrete expression data. The pseudo-sample GRNs (psGRNs) are reconstructed accordingly; thus, classification models can be trained, evaluated, and interpreted having gene regulatory pathways (GRPs) as input factors. With heavily weighted GRPs and associated genes repeatedly obtained from interpretations of success sample class predictions, GRNs could be formed and inferred as genetic feature complexes playing important role in differentiating the studying sample classes. The overall workflow can be summarized as Figure 1. By default setting, four seperate extractor units in workflow run parallelly after data preprocessing part, and all extracted genetic features are used to form GRNs later combined into one atlas. Following sections describe each step of *AGEAS* in more depth.

Step 1:Data preprocessing

The main purpose of this step is to build pseudo-samples and reconstruct corresponding pseudo-sample GRNs (psGRNs). For each sample class, gene expression matrices (GEMs) labeled with same class label are concatenated as one comprehensive expression matrix. With comprehensive GEMs, a meta-level GRN (meta-GRN) is reconstructed in advance of psGRNs to provide generic guidance on reconstruction. In general, the workflow of this step can be represented as Figure 2.

Reconstruct meta-GRN

Firstly, genes included in the comprehensive GEMs are assessed and determined whether having potential to form informative GRPs with other genes. Commonly, differentially expressed genes (DEGs) would be considered as important factors of studying feature. Here we apply the Mann-Whitney U rank test (MWW) implemented by *SciPy*[1] to exclude genes having indistinguishable expression level distribution across GEMs of different classes. The p-value for rejecting null hypothesis, that expression distribution underlying class 1 samples is the same as the expression distribution underlying class 2 samples, is set to 0.05 by default. Furthermore, a $\log_2$ fold change (log2FC) filter is also implemented in *AGEAS*. However, enabling the log2FC filter is not encouraged considering upstream transcription factors (TFs) indirectly regulating key genes associated with feature of interest may not always have significant expression level difference between sample classes. The lof2FC filter shall mostly be used in order to decrease meta-GRN's GRP total amount in a compromising position caused by limited computational resources. After differential expression based filters, a standard deviation (σ) filter is applied to exclude genes either having low expression value or merely affected by dynamic expression status of other genes. By default, the σ threshold is set to 1.0, the lowest expression level in raw gene count matrix gained from RNA-Seq data. The threshold value should be adjusted corresponding to prior knowledge of input GEMs, for example of what normalization method was applied to the GEMs.

With candidate genes passed filters above, some gene pairs are formed and assessed potential of representing GRPs. To reduce overall computational complexity, a gene pair shall be formed with at least one TF which could be the regulatory source of GRP. If not further specified, TF list will be retrieved from integrated *TRANSFAC*[2] datasets according to the provided species information. Utilizing genetic interaction database like *GTRD*[3] and *BioGRID*[4], *AGEAS* checks whether the binding ability of gene pair is confirmed or not. By default, if TF being recorded to have binding site within target gene's promoter range (-1000 to +100) by Chromatin Immunoprecipitation Sequencing (ChIP-Seq) dataset retrieved from *GTRD*[3], the potential GRP gene pair will be passed to expression correlation assessment. For TFs not covered by interaction database, *GRNBoost2*[5]-like algorithm is initiated to predict potential regulatory target genes. The prediction importance threshold can either be set manually or automatically based on recorded interactions as:

$$Threshold = min(GA(M_1,t) \uplus GA(M_2,t), \frac{1}{g})$$

Here $GA()$ denotes *GRNBoost2*[5]-like algorithm; M_1 is concatenated class 1 samples GEM with genes passed filters; M_2 is concatenated class 2 samples GEM with genes passed filters; t is the TF having largest amount of recorded interaction in dataset; g is total amount of unique genes in all samples passed filters.

After potential GRP gene pairs obtained, a expression correlation filter is used to exclude gene pairs having low covariance which implies weak relationship. To assess expression correlation, *AGEAS* applies Pearson's Correlation coefficient[6] (PCC) whihc is one of the widely adopted methods[7]. With default setting, gene pairs can reach absolute correlation coefficient of 0.2 while keeping correspond p-value lower than 0.05 are included in meta-GRN as validated GRP.

Reconstruct psGRNs

The comprehensive GEMs is divided into few sample subsets with sliding window algorithm (SWA) to build pseudo-samples. With SWA, we can gain GEM of *i-th* pseudo-sample through:

$$SWA(i) = \left\{ x_j{}_{j=i*p}^{j+k} \right\}, j+k < l$$

Here l is number of samples in a comprehensive GEM which can be expressed as $\left\{ x_j{}_{j=0}^{l} \right\}$; k denotes window size; p denotes padding stride.

If sample amount is considerably low or imbalanced, customized window size and padding stride can be applied to generate sufficient amount of pseudo-samples for later classifier training and assessment processes. Utilizing meta-GRN, psGRNs are reconstructed with GEMs of pseudo-sample. Each GRP gene pair in meta-GRN is formed with expression data in pseudo-sample and filtered by same PCC filter in meta-GRN reconstruction process.

After all pseudo-samples have been used to reconstruct psGRN, every psGRN can be represented as matrix comprising GRPs' PCC values and corresponding genes' expression values. The order of GRPs in each psGRN is also unified through adding GRPs included in other psGRNs with 0.0 PCC value.

Step 2: Classification model selection

Since *AGEAS* is not aiming to develop optimized model architectures for psGRN classification but gain insights from multiple models as divergent as possible, the main goal of this step is to select configurations of models capable to make correct psGRN classifications from provided configuration set. Regarding the fact that computational power is limited resource, portion of less efficient model configurations shall be pruned although interpreting success predictions of more classification models would lead to more comprehensive insight into sample class differences. Therefore, we apply a simple Hyperband[8]-based algorithm 1 which performs grid search for well-performing classification models with varying model training resource and pruning aggressiveness.

The model selection algorithm requires five inputs (1)R, the maximum amount of training resource, equivalent to all avaliable psGRNs (2)C, the total set of provided classification model configurations (3)I, the number of iterations for model pruning (default set as 3) (4)α_{max}, the maximum portion of R can be fed to single model (default set as 0.95) (5)k_{min}, the minimum portion of remaining model configurations will be kept by single pruning iteration (default set as 0.5). Furthermore, two functions are also required while need to be defined based on input model configurations:

- $run\_then\_evaluate(c,r,R)$: trains classification model initialized using configuration c for the allocated resource r, then returns prediction accuracy (ACC), the area under a receiver operating characteristic curve (AUROC)[9] score, and total cross-entropy loss (L_{CE}) calculated through predicting sample class for all psGRNs R.
- $top\_configs(C,P,k)$: takes a set of model configurations C with associated evaluation results P and returns configurations having ACC, AUROC score, or L_{CE} reaching top k portion.

By default, *AGEAS* initializes with 128 model configurations utilizing 9 integrated classification algorithms listed in Table 1.

The general architecture designs of 1D-CNN and Hybrid-CNN are implemented referring to 1D-CNN and 2D-Hybrid-CNN applied in recent cancer type prediction study[13]. However, taking one convolution layer and adjacent max-pooling layer as a

Algorithm 1: Model selection algorithm

Input: R, C, I (default $I = 3$), α_{max}(default $\alpha_{max} = 0.9$), k_{min}(default $k_{min} = 0.5$)
$\alpha_{low} = \frac{1}{(2^I - 1)}$;
for $i \in \{0, 1, ..., I-1\}$ **do**
  if $i == I - 1$ **then**
    $\alpha = \alpha_{max}$;
  else
    $alpha = 2^i \alpha_{low}$;
  end
  $r = \alpha R$;
  $P = \emptyset$;
  for $c \in C$ **do**
    $p = run\_then\_evaluate(c, r, R)$;
    Append p to P;
  end
  $k = max(1 - \alpha, k_{min})$;
  $C = top\_configs(C, P, k)$;
end
Return: C

layer set, we implemented both CNN models with flexibilities on number of layer set, which is fixed as 1 in original research. An example of 1D-CNN with 2 convolution layer sets is illustrated in Figure 3.

For transformer models, considering psGRNs are already be represented by numerical data matrix while GRPs should barely have positional relationships in the matrix, the embedding layer and positional encoding layer designed for input data tokenization in standard architecture[14] are replaced with a single linear layer in *AGEAS*.

Step 3: Feature selection

With selected well-performing models, *AGEAS* already can start repetition of model training and interpretation in Step 4 to extract key GRPs. However, few uninformative GRPs in training psGRNs merely relied by any model to make classification could be pruned in advance for saving computational power. Furthermore, to prevent classification models focusing on a small group of GRPs regardless of training psGRNs, GRPs draw excessive attention shall also be seperated from psGRNs to improve extraction comprehensiveness. Thus, in this step, *AGEAS* iteratively train classificaion models with dynamic α_{max} portion of psGRNs and obtain feature importance scores as described in subsection below to find GRPs either scored extremely high or considerably low.

More specifically, at each iteration, GRPs having z-scores ranked as bottom b portion (default set as 0.1) are discarded. Also, an i-th ranked GRP will be seperated from psGRNs and passed to Step 5 directly if having z-score fulfilling the condition:

$$Z^i_{score} \geq max(Z^{thread}_{score}, \frac{Z^{i-1}_{score}}{3}, 3 \cdot IQR)$$

The Z^{thread}_{score} is input z-score threshold (default set as 3.0), and IQR stands for interquartile range calculated by the beginning of each iteration. With this criterion, *AGEAS* ensures only GRPs draw significantly more attention shall be selected by each iteration despite the data distribution of varying z-score scalled importance values.

By default, *AGEAS* iterates this feature selection step for 3 times.

Feature importance estimation

AGEAS applies concept of The Shapley value[15] for estimating importance of each input feature, equivalent to GRP of input psGRN, in any kind of classification model while making predictions. Specific Shapley value calculation or approximating methods are implemented with *SHAP*[16] and applied to different algorithms as shown in Table 2. Regarding the standard differences between feature importance estimating methods, we utilize *softmax* function to normalize feature importances and define the normalized importance calculation function as:

$$T(X) = softmax(\{F(x)\}, x \in X)$$

Here X is total set of all features, and $F(x)$ is the importance estimating method of individual classification model being interpreted. If the feature importances can be approached with internalized method $f(x)$, $F(x)$ is set as:

$$F(x) = \frac{f(x)}{\sum_{x' \in X} f(x')}$$

Algorithm	#λ	Categorical	Continuous	# Configs
Implemented with Pytorch[10]				
Transformer	**14**	4	10	32
1D Convolutional Neural Network (1D-CNN)	**10**	2	8	32
Hybrid Convolutional Neural Network (Hybrid-CNN)	**10**	2	8	32
Gated Recurrent Unit (GRU)	**10**	4	6	4
Long Short-Term Memory (LSTM)	**11**	4	7	4
Standard Recurrent Neural Network (RNN)	**11**	5	6	4
Implemented with XGBoost[11]				
Gradient Boosted Decision Trees (GBDT)	**18**	5(1)	13(4)	16
Implemented with scikit-learn[12]				
Random Forests (RF)	**14**	5(1)	9(1)	2
Support Vector Machine (SVM)	**7**	4	3(3)	2

Table 1. Classification model algorithms integrated in *AGEAS* with correspond numbers of hyperparameter and preset model configurations. Categorical hyperparameters and continuous numerical hyperparameters are clarifed beside total number of hyperparameters (#λ). Conditional hyperparameters which are required for selected other hyperparameters are shown in brackets if there is any. # Configs indicates total amount of configurations *AGEAS* applies by default.

Otherwise, $F(x)$ is defined utilizing correctly classified input samples S, equivalent to psGRNs, with Shapley values ϕ of feature x when predicting sample s as class c_1 or class c_2:

$$F(x) = \sum_{s \in S} \frac{\left|\phi^x_{c_1,s}\right| + \left|\phi^x_{c_2,s}\right|}{2}$$

After all selected classification models M have been interpreted, we can intergrate the feature importances matrices weighted with corresponding models' L_{CE} to one matrix A as:

$$A = \left\{\sum_{m \in M}(1 - L^m_{CE})T_m(x)\right\}, x \in X$$

Then, generalized importance values are obtained through z-score calculation and later sorted by descending order:

$$Z_{score} = \left\{\frac{a - \bar{A}}{\sigma_A}\right\}, a \in A$$

Algorithm	SHAP[16] Method
Transformer	Gradient Explainer
1D-CNN	Deep Explainer
Hybrid-CNN	Deep Explainer
GRU	Gradient Explainer
LSTM	Gradient Explainer
RNN	Gradient Explainer
GBDT*	Tree Explainer
RF	Tree Explainer
SVM*	Linear Explainer / Kernel Explainer

Table 2. Classifier algorithms with applicable Shapley value approximating methods. Algorithms marked with * have internalized feature importance estimating methods which will be applied with higher priority than Shapley value based methods. GBDT implemented with *XGBoost*[11] can have feature importance approximated with average weight gain at each split involving the feature. Linear SVM implemented with *scikit-learn*[12] can have the importance estimated with feature coefficient or using Linear Explainer. However, for SVM with kernel function, the feature coefficient estimation would be inappropriate, and feature importances should be approximated with Kernel Explainer.

Step 4: Top GRP extraction

To extract GRPs can effectively define sample class differences, *AGEAS* iteratively initializes classification models with configurations gained from Step 2, trains them with randomly selected α_{max} portion of psGRNs scaled after Step 3, and interpret every models' correct predictions as mentioned in subsection above. At each iteration, *AGEAS* receives a z-score scaled feature

importance matrix A and add up each GRP's score accordingly from matrix A' kept from previous iteration if there is one. Then, top a (default set as 100) ranked GRPs having z-score greater than 0.0 extracted from A are compared with GRPs extracted from A' by same setting. If less than d (default set as 0.05) portion of GRPs are distinct for GRP sets stratified from A and A', *AGEAS* will consider the GRP extration result from this iteration being consistent with previous one. Extraction iteration will terminate if either encountering n (default set as 3) continuous consistent result or running out of preset iteration number (default set as 10). All feature importance scores in matrix A from last iteration are divided by total extraction iteration number processed, and top a ranked GRPs are considered as key GRPs for sample class differentiation and passed to next step.

Step 5: Key network reconstruction

Analoging every GRP previously extracted or seperated with high z-score in Step 3 as a directional edge connecting two gene vertices, equivalent to a TF and a gene, in graph theory, *AGEAS* attempts to reconstruct a network graph representing regulatory differences between query sample classes. Since there is no guarantee on all GRP edges can be connected, some regulatory relationships between gene vertices could be missed. Hence, *AGEAS* utilizes meta-GRN gained from Step 1 to find GRPs which can further elucidate the regulatory relationships and adds the GRPs back to the network graph.

For a limited iteration time (default set as 1), *AGEAS* exhaustively search meta-GRN for TFs which can directly regulate any genes or TFs already covered in the network graph and add the returned TFs as new vertices. Next, any GRP in meta-GRN capable to connect two distinct vertices will be added if it is not covered yet. The network graph after expansion above represents the key genetic regulatory differences *AGEAS* extracted from input sample classes.

Results

To assess *AGEAS*'s predictive power, we first applied *AGEAS* to study somatic reprogramming to induced pluripotent stem cells (iPSC) achieved in mice by the year 2006[17], milestone discovery of cell plasticity[18]. With σ thread set to 2.0 for constraining total GRP amount in psGRNs, single-cell RNA sequencing (scRNA-Seq) based gene expression matrices (GEMs) of embryonic stem cell (ESC) and mouse embryonic fibroblast (MEF) were used as input. The extraction result is summarized as TF regulons, collections of a TF and corresponding direct regulatory targets, in ... Since Pou5f1 (Oct4), Nanog, and Sox2 not only are forming sizeable regulons with extracted GRPs but also have most significant expression changes between MEF and ESC, we can infer them as important factors closely related with cell type differences. Multiple previous studies have confirmed that all of Oct4, Nanog, and Sox2 are playing important role in maintaing pluripotency[19] and inducing cell conversion[17,20–23].

To further address *AGEAS*'s applicability and limitation,

radial glial coculture to dopaminergic neurons...

Postnatal p7-p28 cardiomyocyte mature

HSC activation in CCl4 injury model

To demonstrate empirically the predictive capabilities...

Discussion

References

1. Virtanen, P. *et al.* SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nat. Methods* **17**, 261–272, DOI: 10.1038/s41592-019-0686-2 (2020).
2. Matys, V. Transfac(r) and its module transcompel(r): Transcriptional gene regulation in eukaryotes. *Nucleic Acids Res.* **34**, DOI: 10.1093/nar/gkj143 (2006).
3. Kolmykov, S. *et al.* GTRD: an integrated view of transcription regulation. *Nucleic Acids Res.* **49**, D104–D111, DOI: 10.1093/nar/gkaa1057 (2020). https://academic.oup.com/nar/article-pdf/49/D1/D104/35364856/gkaa1057.pdf.
4. Stark, C. Biogrid: A general repository for interaction datasets. *Nucleic Acids Res.* **34**, DOI: 10.1093/nar/gkj109 (2006).
5. Moerman, T. *et al.* Grnboost2 and arboreto: Efficient and scalable inference of gene regulatory networks. *Bioinforma. (Oxford, England)* **35**, DOI: 10.1093/bioinformatics/bty916 (2018).
6. Song, L., Langfelder, P. & Horvath, S. Comparison of co-expression measures: Mutual information, correlation, and model based indices. *BMC Bioinforma.* **13**, DOI: 10.1186/1471-2105-13-328 (2012).
7. Liu, L.-y. D., Hsiao, Y.-C., Chen, H.-C., Yang, Y.-W. & Chang, M.-C. Construction of gene causal regulatory networks using microarray data with the coefficient of intrinsic dependence. *Bot. Stud.* **60**, DOI: 10.1186/s40529-019-0268-8 (2019).
8. Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A. & Talwalkar, A. Hyperband: A novel bandit-based approach to hyperparameter optimization (2018).

9. Hanley, J. A. & McNeil, B. J. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology* **143**, 29–36, DOI: 10.1148/radiology.143.1.7063747 (1982).

10. Paszke, A. *et al.* Pytorch: An imperative style, high-performance deep learning library. In Wallach, H. *et al.* (eds.) *Advances in Neural Information Processing Systems 32*, 8024–8035 (Curran Associates, Inc., 2019).

11. Chen, T. & Guestrin, C. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 785–794 (2016).

12. Pedregosa, F. *et al.* Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011).

13. Mostavi, M., Chiu, Y.-C., Huang, Y. & Chen, Y. Convolutional neural network models for cancer type prediction based on gene expression - bmc medical genomics (2020).

14. Vaswani, A. *et al.* Attention is all you need (2017).

15. Roth, A. E. & Shapley, L. S. The shapley value : essays in honor of lloyd s. shapley. *Economica* **101**, 123 (1991).

16. Lundberg, S. M. & Lee, S.-I. A unified approach to interpreting model predictions. In *Proceedings of the 31st international conference on neural information processing systems*, 4768–4777 (2017).

17. Takahashi, K. & Yamanaka, S. Induction of pluripotent stem cells from mouse embryonic and adult fibroblast cultures by defined factors. *Cell* **126**, 663–676, DOI: 10.1016/j.cell.2006.07.024 (2006).

18. Wang, H., Yang, Y., Liu, J. & Qian, L. Direct cell reprogramming: Approaches, mechanisms and progress. *Nat. Rev. Mol. Cell Biol.* **22**, 410–424, DOI: 10.1038/s41580-021-00335-z (2021).

19. Niwa, H. How is pluripotency determined and maintained? *Development* **134**, 635–646, DOI: 10.1242/dev.02787 (2007).

20. Wang, B. *et al.* Induction of pluripotent stem cells from mouse embryonic fibroblasts by jdp2-jhdm1b-mkk6-glis1-nanog-essrb-sall4. *Cell Reports* **27**, DOI: 10.1016/j.celrep.2019.05.068 (2019).

21. Shi, Y. *et al.* Induction of pluripotent stem cells from mouse embryonic fibroblasts by oct4 and klf4 with small-molecule compounds. *Cell Stem Cell* **3**, 568–574, DOI: 10.1016/j.stem.2008.10.004 (2008).

22. Yu, J. *et al.* Induced pluripotent stem cell lines derived from human somatic cells. *Science* **318**, 1917–1920, DOI: 10.1126/science.1151526 (2007).

23. Wang, Y. *et al.* Reprogramming of mouse and human somatic cells by high-performance engineered factors. *EMBO reports* **12**, 373–378, DOI: 10.1038/embor.2011.11 (2011).

Author contributions statement

J.Y.: Methodology, Software, Writing- Original draft preparation, Project administration **M.N.**: Methodology, Software **J.T.**: Writing - Review & Editing, Project administration

Additional information

All scRNA-Seq datasets are retrieved from Gene Expression Omnibus(GEO) as described in the Table below.

Sample Class	Accession number
GSE103221	
MEF	GSM3629847
ESC	GSM3629848
GSE137720	
a6w hsc	GSM4085625
a6w pf	GSM4085627
healthy hsc	GSM4085623
healthy pf	GSM4085626
GSE185275	
neural co-culture	GSM5609927
purified neurons	GSM5609930
GSE156482	
p4 cm	GSM4732219
p7 cm	GSM4732221
p28 cm	GSM4732225
p28 ncm	GSM4732226

Table 3. GEO data source table.

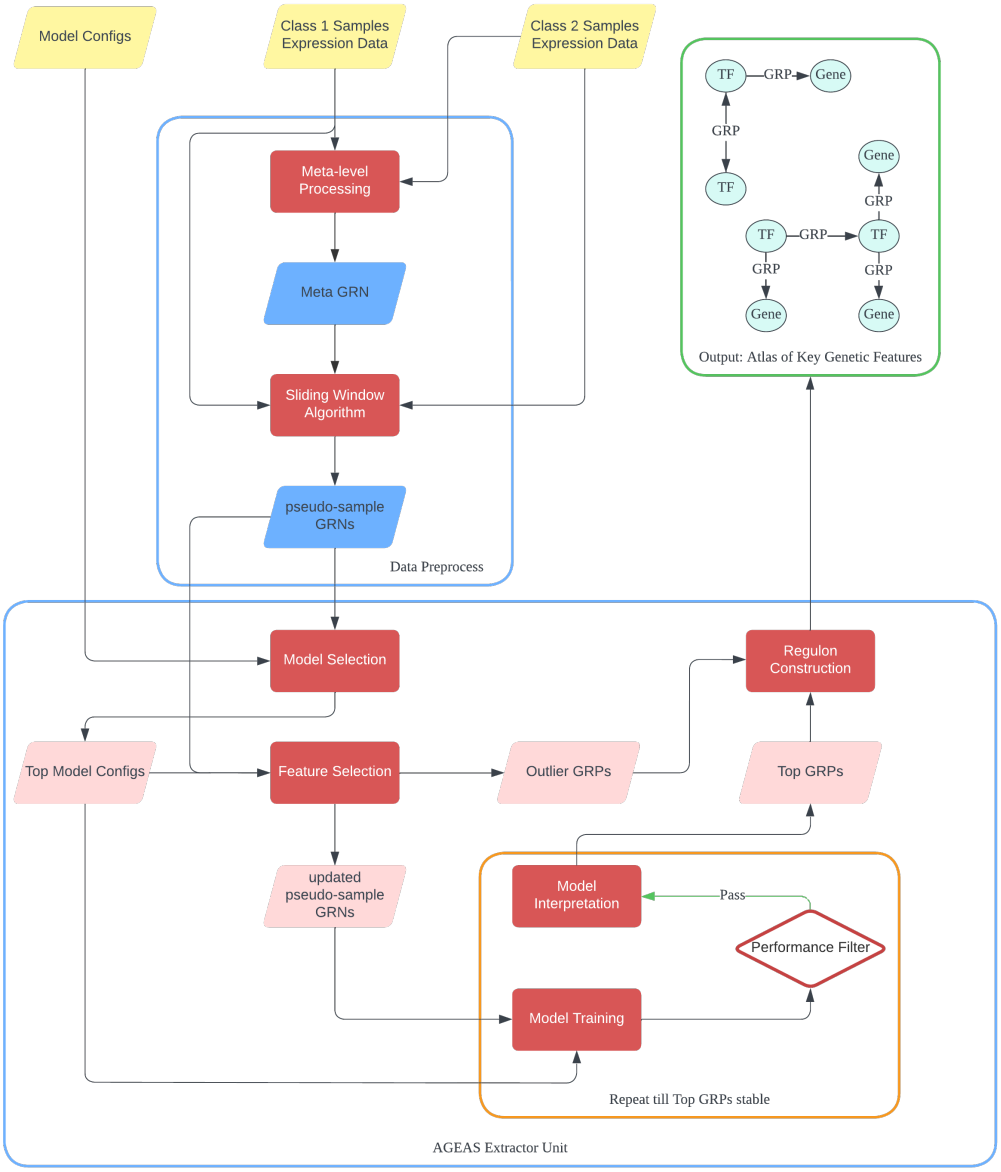

Figure 1. The overall workflow of *AGEAS*: ***(1)*** Reconstruct meta-level GRN (meta-GRN) with expression data of all samples. ***(2)*** Build pseudo-samples with sliding window algorithm and reconstruct GRNs with GRPs identified in meta-GRN accordingly. ***(3)*** Select best performing classifiers in predicting class labels of pseudo-sample GRNs (psGRNs). ***(4)*** Interpret how top models make classifications and gradually exclude GRPs with low weights or outlier-level high weights. ***(5)*** Repeatedly train classifiers with differnt set of psGRNs as training data to extract GRPs frequently ranked as top important features for decision. ***(6)*** Reconstruct GRNs with extracted GRPs and GRPs excluded as significant outliers.

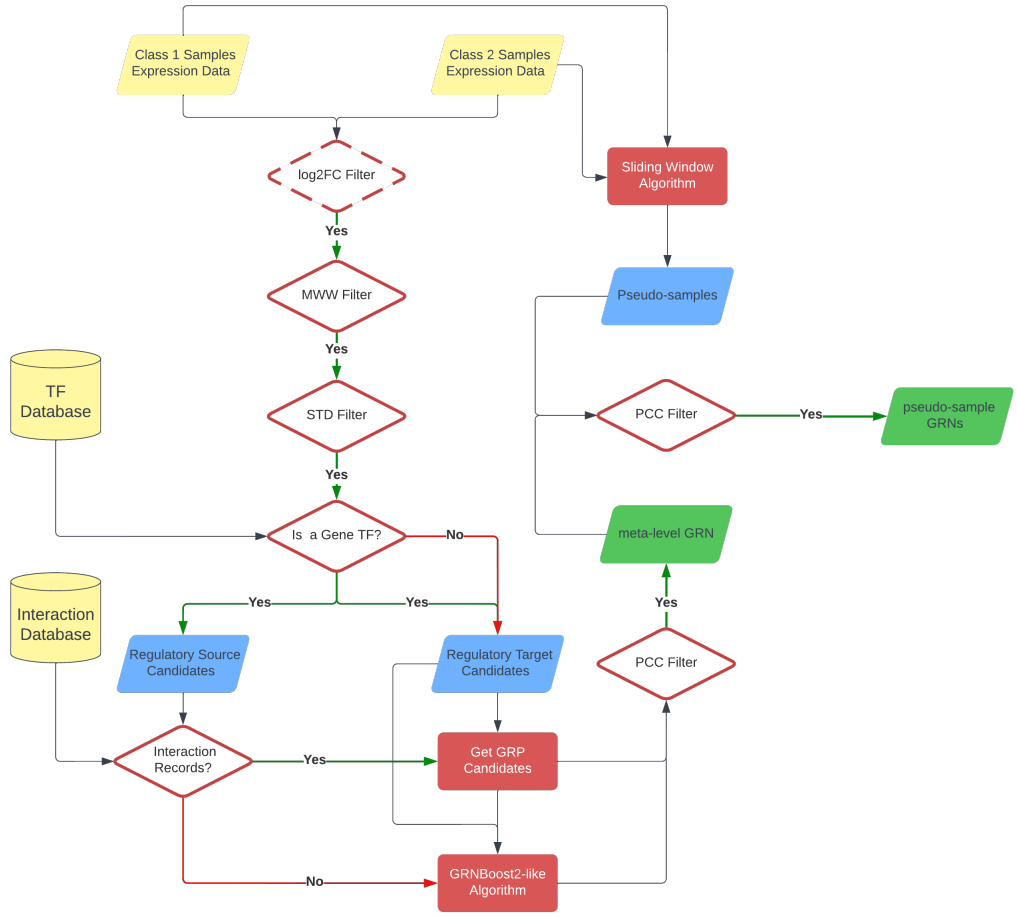

Figure 2. Workflow to reconstruct meta-GRN and psGRNs. ***(1)*** Filter genes from GEMs with log2FC filter(optional), MWW filter, and σ filter. ***(2)*** Find candidate GRP gene pairs from either interaction database or predictions made by *GRNBoost2*[5]-like algorithm. ***(3)*** Filter candidate GRP gene pairs with PCC filter and reconstruct meta-GRN with validated GRPs. ***(4)*** Generate pseudo-samples with SWA. ***(5)*** Utilize GRPs in meta-GRN as generic guidance to form candidate GRPs for pseudo-samples. ***(6)*** Filter every candidate GRP for all pseudo-samples with PCC filter and reconstruct psGRNs with validated GRPs.

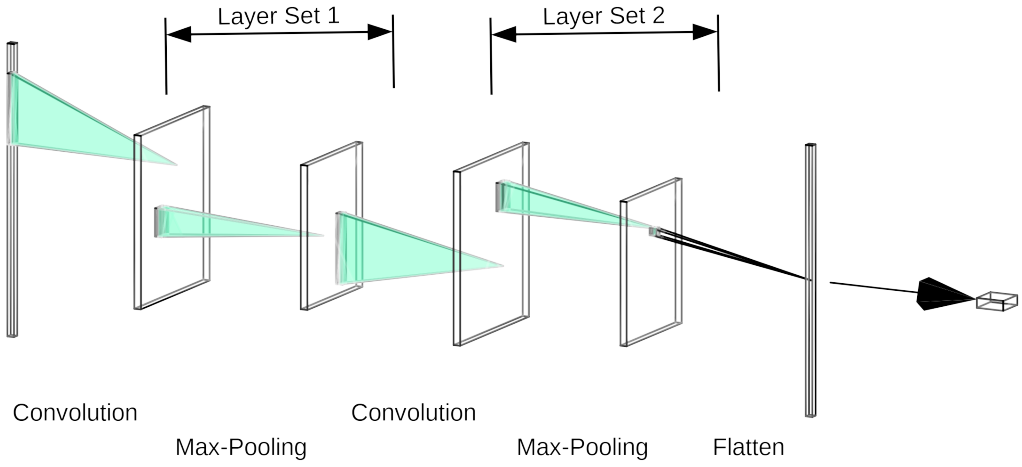

Figure 3. 1D-CNN with 2 convolution layer set.