Towards Enhanced and Interpretable Clustering/Classification in Integrative Genomics

Yang Young Lu[1],Jinchi Lv[2],Jed A. Fuhrman[3], and Fengzhu Sun[1,4*]

[1]Molecular and Computational Biology Program, Department of Biological Sciences, University of Southern California, CA, USA

[2]Data Sciences and Operations Department, Marshall School of Business, University of Southern California, CA, USA

[3]Department of Biological Sciences and Wrigley Institute for Environmental Studies, University of Southern California, CA, USA

[4]Centre for Computational Systems Biology, School of Mathematical Sciences, Fudan University, Shanghai, China

July 25, 2017

Contents

1 Background **7**
- 1.1 Signed Graph . . . 7
- 1.2 Scaled-LASSO Algorithm . . . 7
- 1.3 Cluster Quality Indices . . . 7
  - 1.3.1 External Indices . . . 7
  - 1.3.2 Internal Indices . . . 8
- 1.4 Feature Selection . . . 8
  - 1.4.1 Correlation-based Feature Selection (CFS) [5] . . . 8
  - 1.4.2 Fast Correlation-Based Filter (FCBF) [16] . . . 9
  - 1.4.3 Sparse Logistic Regression (SLR) [11] . . . 9

2 Details of Hetero-RP **9**

3 Simulation Studies **11**

4 Application to Clustering: Metagenomic Contig Binning **14**
- 4.1 Details of COCACOLA [9] . . . 14
- 4.2 Details of Input Data . . . 15
- 4.3 Details of Features after Hetero-RP . . . 15

5 Application to Classification: RBP Binding Site Prediction **15**
- 5.1 Details of iONMF [12] . . . 15
- 5.2 Details of Input Data . . . 24
- 5.3 Details of Features after Hetero-RP . . . 25
  - 5.3.1 Features of co-binding proteins cDNA counts . . . 26
  - 5.3.2 Features of RNA secondary structures . . . 58
  - 5.3.3 Features of Region types (intron,exon,5'-UTR, 3'-UTR and ORF) . . . 66

List of Figures

S1 The two-dimensional Principal Coordinates Analysis of 20 different synthetic ellipsoidal datasets. Each cluster is displayed with different color as well as different marks. (A)-(J) indicate 10 different repeats of synthetic ellipsoidal datasets with 10 cluster numbers and 100 features. (K)-(T) indicate 10 different repeats of synthetic ellipsoidal datasets with 10 cluster numbers and 1000 features. . . . 12

S2 The comparison between different amount of auxiliary knowledge on 20 synthetic datasets. "P+N" stands for using both "positive-links" and "negative-links" set. "P+N+X" stands for not only using both "positive-links" and "negative-links" set, but also the original data matrix X by pretending the auxiliary knowledge is insufficient. "P" stands for using only "positive-links" set. And "P+X" stands for using "positive-links" set together with the original data matrix X. The evaluation is based on the most commonly used internal cluster index, the silhouette value. 13

S3 The feature scaling of Hetero-RP using co-alignment when sample size is 10, overall 9 replicates, for the simulated "species" dataset. . . . 16

S4 The feature scaling of Hetero-RP using co-alignment when sample size is 20, overall 4 replicates, for the simulated "species" dataset. . . . 17

S5 The feature scaling of Hetero-RP using co-alignment when sample size is 30, overall 3 replicates, for the simulated "species" dataset. . . . 17

S6 The feature scaling of Hetero-RP using co-alignment when sample size is 40, overall 2 replicates, for the simulated "species" dataset. . . . 18

S7 The feature scaling of Hetero-RP using co-alignment when sample size is 50, overall 1 replicate, for the simulated "species" dataset. . . . 18

S8 The feature scaling of Hetero-RP using co-alignment when sample size is 60, overall 1 replicate, for the simulated "species" dataset. . . . 18

S9 The feature scaling of Hetero-RP using co-alignment when sample size is 70, overall 1 replicate, for the simulated "species" dataset. . . . 18

S10 The feature scaling of Hetero-RP using co-alignment when sample size is 80, overall 1 replicate, for the simulated "species" dataset. . . . 19

S11 The feature scaling of Hetero-RP using co-alignment when sample size is 90, overall 1 replicate, for the simulated "species" dataset. . . . 19

S12 The feature scaling of Hetero-RP using co-alignment when sample size is 96, overall 1 replicate, for the simulated "species" dataset. . . . 19

S13 The feature scaling of Hetero-RP using linkage when sample size is 10, overall 9 replicates, for the simulated "species" dataset. . . . 20

S14 The feature scaling of Hetero-RP using linkage when sample size is 20, overall 4 replicates, for the simulated "species" dataset. . . . 21

S15 The feature scaling of Hetero-RP using linkage when sample size is 30, overall 3 replicates, for the simulated "species" dataset. . . . 21

S16 The feature scaling of Hetero-RP using linkage when sample size is 40, overall 2 replicates, for the simulated "species" dataset. . . . 22

S17 The feature scaling of Hetero-RP using linkage when sample size is 50, overall 1 replicate, for the simulated "species" dataset. . . . 22

S18 The feature scaling of Hetero-RP using linkage when sample size is 60, overall 1 replicate, for the simulated "species" dataset. . . . 22

S19 The feature scaling of Hetero-RP using linkage when sample size is 70, overall 1 replicate, for the simulated "species" dataset. . . . 22

S20 The feature scaling of Hetero-RP using linkage when sample size is 80, overall 1 replicate, for the simulated "species" dataset. . . . 23

S21 The feature scaling of Hetero-RP using linkage when sample size is 90, overall 1 replicate, for the simulated "species" dataset. . . . 23

S22 The feature scaling of Hetero-RP using linkage when sample size is 96, overall 1 replicate, for the simulated "species" dataset. . . . 23

S23 The feature scaling of Hetero-RP using co-alignment for the real "MetaHIT" dataset. . . . 23

S24 The feature scaling of Hetero-RP using linkage for the real "MetaHIT" dataset. . . . 23

S25 The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [1]Ago_EIF2C1-4. . . . 27

S26 The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [2]Ago2-MNase. . . . 28

S27 The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [3]Ago2(1). 29

S28 The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [4]Ago2(2). 30

S29 The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [5]Ago2. . . 31
S30 The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [6]eIF4AIII(1). 32
S31 The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [7]eIF4AIII(2). 33
S32 The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [8]ELAVL1. 34
S33 The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [9]ELAVL1-MNase. 35
S34 The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [10]ELAVL1A. 36
S35 The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [11]ELAVL1. 37
S36 The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [12]ESWR1. 38
S37 The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [13]FUS. . . 39
S38 The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [14]Mut_FUS. 40
S39 The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [15]IGF2BP1-3. 41
S40 The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [16]hnRNPC. 42
S41 The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [17]hnRNPC. 43
S42 The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [18]hnRNPL. 44
S43 The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [19]hnRNPL. 45
S44 The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [20]hnRNPL-like. 46
S45 The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [21]MOV10. 47
S46 The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [22]NSUN2. 48
S47 The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [23]PUM2. 49
S48 The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [24]QKI. . . 50
S49 The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [25]SRSF1. 51
S50 The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [26]TAF15. 52
S51 The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [27]TDP-43. 53
S52 The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [28]TIA1. . 54
S53 The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [29]TIAL1. 55
S54 The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [30]U2AF2. 56
S55 The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [31]U2AF2(KD). 57
S56 The features of Hetero-RP for RNA secondary structure in the dataset [1]Ago_EIF2C1-4. 58
S57 The features of Hetero-RP for RNA secondary structure in the dataset [2]Ago2-MNase. 58
S58 The features of Hetero-RP for RNA secondary structure in the dataset [3]Ago2(1). 58
S59 The features of Hetero-RP for RNA secondary structure in the dataset [4]Ago2(2). 59
S60 The features of Hetero-RP for RNA secondary structure in the dataset [5]Ago2. 59
S61 The features of Hetero-RP for RNA secondary structure in the dataset [6]eIF4AIII(1). 59
S62 The features of Hetero-RP for RNA secondary structure in the dataset [7]eIF4AIII(2). 59
S63 The features of Hetero-RP for RNA secondary structure in the dataset [8]ELAVL1. 60
S64 The features of Hetero-RP for RNA secondary structure in the dataset [9]ELAVL1-MNase. . . . 60
S65 The features of Hetero-RP for RNA secondary structure in the dataset [10]ELAVL1A. 60
S66 The features of Hetero-RP for RNA secondary structure in the dataset [11]ELAVL1. 60
S67 The features of Hetero-RP for RNA secondary structure in the dataset [12]ESWR1. 61
S68 The features of Hetero-RP for RNA secondary structure in the dataset [13]FUS. 61
S69 The features of Hetero-RP for RNA secondary structure in the dataset [14]Mut_FUS. 61
S70 The features of Hetero-RP for RNA secondary structure in the dataset [15]IGF2BP1-3. 61
S71 The features of Hetero-RP for RNA secondary structure in the dataset [16]hnRNPC. 62
S72 The features of Hetero-RP for RNA secondary structure in the dataset [17]hnRNPC. 62
S73 The features of Hetero-RP for RNA secondary structure in the dataset [18]hnRNPL. 62
S74 The features of Hetero-RP for RNA secondary structure in the dataset [19]hnRNPL. 62
S75 The features of Hetero-RP for RNA secondary structure in the dataset [20]hnRNPL-like. 63
S76 The features of Hetero-RP for RNA secondary structure in the dataset [21]MOV10. 63
S77 The features of Hetero-RP for RNA secondary structure in the dataset [22]NSUN2. 63
S78 The features of Hetero-RP for RNA secondary structure in the dataset [23]PUM2. 63
S79 The features of Hetero-RP for RNA secondary structure in the dataset [24]QKI. 64
S80 The features of Hetero-RP for RNA secondary structure in the dataset [25]SRSF1. 64
S81 The features of Hetero-RP for RNA secondary structure in the dataset [26]TAF15. 64
S82 The features of Hetero-RP for RNA secondary structure in the dataset [27]TDP-43. 64

S83 The features of Hetero-RP for RNA secondary structure in the dataset [28]TIA1. 65
S84 The features of Hetero-RP for RNA secondary structure in the dataset [29]TIAL1. 65
S85 The features of Hetero-RP for RNA secondary structure in the dataset [30]U2AF2. 65
S86 The features of Hetero-RP for RNA secondary structure in the dataset [31]U2AF2(KD). 65
S87 The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [1]Ago_EIF2C1-4. 66
S88 The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [2]Ago2-MNase. 66
S89 The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [3]Ago2(1). 66
S90 The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [4]Ago2(2). 67
S91 The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [5]Ago2. 67
S92 The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [6]eIF4AIII(1). 67
S93 The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [7]eIF4AIII(2). 67
S94 The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [8]ELAVL1. 68
S95 The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [9]ELAVL1-MNase. 68
S96 The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [10]ELAVL1A. 68
S97 The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [11]ELAVL1. 68
S98 The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [12]ESWR1. 69
S99 The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [13]FUS. 69
S100 The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [14]Mut_FUS. 69
S101 The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [15]IGF2BP1-3. 69
S102 The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [16]hnRNPC. 70
S103 The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [17]hnRNPC. 70
S104 The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [18]hnRNPL. 70
S105 The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [19]hnRNPL. 70
S106 The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [20]hnRNPL-like. 71
S107 The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [21]MOV10. 71
S108 The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [22]NSUN2. 71
S109 The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [23]PUM2. 71
S110 The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [24]QKI. 72
S111 The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [25]SRSF1. 72
S112 The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [26]TAF15. 72
S113 The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [27]TDP-43. 72
S114 The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [28]TIA1. 73

S115 The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [29]TIAL1. 73
S116 The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [30]U2AF2. 73
S117 The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [31]U2AF2(KD). 73

List of Tables

S1 Synthetic ellipsoidal datasets of high dimensionality of features. For each of the two possible combinations of cluster number and feature dimensionality, 10 different repeats were generated, giving 20 datasets in all. 11
S2 The options of sampling probability p in simulated auxiliary knowledge. For each p, 10 different repeats were generated, giving 220 simulated auxiliary knowledge in all. 14

1 Background

1.1 Signed Graph

In graph theory, an undirected signed graph is a graph in which each edge has either a positive or negative sign. Conventionally an undirected signed graph is described as $SG = (V, E^+, E^-)$, where V is the set of nodes, E^+ consists of edges with positive sign and E^- consists of edges with negative sign, respectively.

The signed graph SG can be represented by an adjacency matrix $A \in \mathbb{R}^{|V|\times|V|}$, where $A_{ij} = 1$ if there is a positive edge between node i and node j, $A_{ij} = -1$ if there is a negative edge between node i and node j, and $A_{ij} = 0$ if there is no edge connecting node i and node j. Furthermore, the adjacency matrix A can be decomposed into a positive adjacency matrix $A^+ \in \mathbb{R}^{|V|\times|V|}$ and a negative adjacency matrix $A^- \in \mathbb{R}^{|V|\times|V|}$ that represent the positive part and negative part of A, respectively. Specifically, $A^+_{ij} = A_{ij}$ if $A_{ij} > 0$ and $A^+_{ij} = 0$ otherwise. And $A^-_{ij} = -A_{ij}$ if $A_{ij} < 0$ and $A^-_{ij} = 0$. When signed graph only contains edges with positive sign, then $A = A^+$.

1.2 Scaled-LASSO Algorithm

Suppose we observe a design matrix $X = (x_1, x_2, \cdots, x_p) \in \mathbb{R}^{n\times p}$ and a response vector $y \in \mathbb{R}^n$, the LASSO estimator aims to find the vector of regression coefficients $\beta = (\beta_1, \beta_2, \cdots, \beta_p)^T$ so that the following loss function is minimized:

$$\min_{\beta\in\mathbb{R}^p} \left\{ \frac{1}{2n} \|y - X\beta\|_2^2 + \lambda \|\beta\|_1 \right\} \tag{1}$$

where λ is a tuning parameter indicating the penalty level of regularization term. Scale-invariance considerations and theoretical results suggest using λ proportional to the noise level σ that motivates the scaled-LASSO Algorithm [13]. That is, the estimator $\widehat{\beta}$, the tuning parameter $\widehat{\lambda}$, and the estimation of noise level $\widehat{\sigma}$ can all be obtained through scaled-LASSO. Specifically, the following two steps are iteratively repeated until convergence:

$$\widehat{\beta} \leftarrow \arg\min_{\beta} \left\{ \|y - X\beta\|_2^2 + 2n\lambda_0\widehat{\sigma} \|\beta\|_1 \right\}, \tag{2a}$$

$$\widehat{\sigma}^2 \leftarrow \frac{1}{n} \left\| y - X\widehat{\beta} \right\|_2^2 \tag{2b}$$

where λ_0 is chosen according to the suggestion of [10] and has also been used in [4]. In particular, $\lambda_0 = B/\left(n - 1 + B^2\right)^{1/2}$, where $B = tq(1 - n^{1/2}/(2p\log p), n - 1)$ with $tq(\alpha, d)$ the α-th quantile of a t-distribution with d degrees of freedom.

1.3 Cluster Quality Indices

Due to the variety of application areas, various clustering methods have been developed for the optimization of different criteria, thus clustering quality has been defined ambiguously. For quality measures, the commonly agreed upon cluster quality measures generally could be separated into two different categories: internal and external indices. Specifically, internal cluster indices evaluate the clustering result solely based upon some intrinsic statistical properties of the input data without referring to the ground truth, whereas external cluster indices compare the clustering result to the specified ground truth.

1.3.1 External Indices

To evaluate a clustering result with K obtained clusters against the ground truth with S targeted clusters, a $K \times S$ matrix $A = (a_{ks})$ can be constructed so that a_{ks} indicates the shared number of objects between the k-th obtained cluster and the s-th targeted cluster. Therefore, $a_{k\cdot} = \sum_s a_{ks}$ and $a_{\cdot s} = \sum_k a_{ks}$ stand for the size of the k-th obtained cluster and the s-th targeted cluster, respectively.

The evaluation focuses on if pairs of objects belonging to the same targeted cluster can be clustered together in the obtained cluster. Then the classification of pairs of objects falls into one of the four cases: TP (True Positive) and FP (False Positive) represent the number of pairs of objects that truly belong to the same targeted cluster being clustered into the same obtained cluster and distinct obtained cluster, respectively; FN (False Negative) and TN (True Negative) stand for the number of pairs of objects from different targeted cluster being clustered into the same obtained cluster and distinct obtained cluster, respectively. Let n be the total number of objects. Therefore, the three commonly used external cluster indices: Adjusted Rand Index (ARI), Precision, and Recall can be defined as:

- Adjusted Rand Index (ARI)

$$\begin{aligned}\text{ARI} &= \frac{2\left(TP \times TN - FP \times FN\right)}{FP^2 + FN^2 + 2 \times TP \times TN + (TP + TN) \times (FP + FN)} \\ &= \frac{\sum_{k,s}\binom{a_{ks}}{2} - t_3}{\frac{1}{2}\left(t_1 + t_2\right) - t_3} \\ \textit{where} \quad t_1 &= \sum_k \binom{a_{k\cdot}}{2}, \;\; t_2 = \sum_s \binom{a_{\cdot s}}{2}, \;\; t_3 = \frac{2t_1 t_2}{\binom{n}{2}}\end{aligned}$$

- Precision

$$\text{Precision} = \frac{1}{n}\sum_k \max_s \{a_{ks}\}$$

- Recall

$$\text{Recall} = \frac{1}{n}\sum_s \max_k \{a_{ks}\}$$

1.3.2 Internal Indices

In principle, internal cluster indices measure the compactness and separation of clusters [15]. Specifically, compactness refers to how small the distances among objects within clusters, whereas separation refers to how large the distances among objects across different clusters. Various internal cluster indices usually differ in how they balance these two aspects.

The most commonly used internal cluster index is Silhouette value. The silhouette value has a range of $[-1, 1]$, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters. If most objects have a high value, then the clustering configuration is appropriate. If many points have a low or negative value, then the clustering configuration may have too many or too few clusters. Given a clustering result with K obtained clusters, for each object i, let $a(i)$ be the average distance between object i and other objects within the same cluster. Thus $a(i)$ can be interpreted as how well object i is assigned to its cluster, in the essence of compactness. Let $b(i)$ be the lowest average distance of object i to any other cluster that object i does not belong to. The cluster with lowest average distance is regarded as the "neighbouring cluster" of object i. Then the silhouette value of object i is defined as:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{3}$$

Thus $s(i)$ close to 1 means that the object i is appropriately clustered. If $s(i)$ close to -1 means that the object i would be more appropriate if it was clustered in its "neighbouring cluster". And if $s(i)$ near 0 means that the object i is on the border of two clusters. Finally, the silhouette value over all data takes the average over all $s(i)$.

1.4 Feature Selection

Feature selection, as a data preprocessing strategy, aims to reduce the dimensionality of high-dimensional data and builds simpler and more understandable models [8]. Conventionally, feature Selection techniques select a subset of features that are able to discriminate samples from different classes. With the existence of class labels, the feature relevance is usually assessed via its correlation with class labels. The general procedure to use feature selection in supervised learning is as following. After the data are split into training and testing datasets, classifiers are firstly trained based upon the subset of features selected by feature selection. Note that the feature selection can either be independent of the learning algorithm (i.e., filter methods), or it can depend on the learning algorithm by iteratively assessing the quality of selected features so far (i.e., wrapper methods). Finally, the trained classifier predicts labels of objects in the testing dataset in terms of the selected features.

1.4.1 Correlation-based Feature Selection (CFS) [5]

The idea of CFS is to employ a correlation-based heuristic to evaluate the goodness of feature subset $\mathcal{F}$:

$$\text{CFS}(\mathcal{F}) = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}} \tag{4}$$

where k is the number of features selected in subset $\mathcal{F}$. $\overline{r_{cf}}$ represents the mean feature class correlation and $\overline{r_{ff}}$ represents the average feature-feature intercorrelation, respectively. The intuition of CFS is that a good

feature subset is expected to have strong correlation with class labels and are weakly intercorrelated. For the computational efficiency, CFS adopts a best-search strategy to find a local optimal feature subset. It starts with an empty feature set and then expand the set repeatedly until the stopping criteria is met.

1.4.2 Fast Correlation-Based Filter (FCBF) [16]

FCBF is a filter method that exploits feature-class correlation and feature-feature correlation simultaneously. The algorithm works as following:

(1) it selects a feature subset $\mathcal{F}$ that is highly correlated with the class label with respect to the criterion $SU \geq \delta$, where δ is a predefined threshold and SU is the symmetric uncertainty between a set of features $X_{\mathcal{F}}$ and the class label Y, defined as:

$$SU(X_{\mathcal{F}}, Y) = 2\frac{I(X_{\mathcal{F}}; Y)}{H(X_{\mathcal{F}}) + H(Y)} \tag{5}$$

where $I(X;Y)$ and $H(X)$ are referred as *mutual information* and *entropy*, respectively. A specific feature X_k is called predominant if $SU(X_k, Y) \geq \delta$ and there does not exist a distinct feature $X_j \in \mathcal{F}$ such that $SU(X_j, X_k) \geq SU(X_k, Y)$. Feature X_j is considered to be redundant to feature X_k if $SU(X_j, X_k) \geq SU(X_k, Y)$.

(2) Let $\mathcal{F}_{P_j}$ be the set of redundant features that can further be split into $\mathcal{F}^+_{P_j}$ and $\mathcal{F}^-_{P_j}$ that contain redundant features to feature X_k with $SU(X_j, X_k) > SU(X_k, Y)$ and $SU(X_j, X_k) < SU(X_k, Y)$, respectively.

(3) FCBF uses heuristics on $\mathcal{F}_{P_j}$, $\mathcal{F}^+_{P_j}$, and $\mathcal{F}^-_{P_j}$ to remove redundant features and keep the features that are most relevant features to the class label.

1.4.3 Sparse Logistic Regression (SLR) [11]

SLR considers a logistic regression classifier $\mathbf{w}$ in which the classification of Y can be based on a linear combination of X. By incorporating the regularization, classifier and feature selection are achieved simultaneously by estimating $\mathbf{w}$ with properly tuned penalties. Specifically, the coefficient $\widehat{\mathbf{w}}$ can be estimated as:

$$\widehat{\mathbf{w}} = \arg\min_{\mathbf{w}} c(\mathbf{w}, X) + \alpha\ penalty(\mathbf{w}) \tag{6}$$

where $c(\cdot)$ is the classification objective function, $penalty(\mathbf{w})$ is a regularization term, and α s the regularization parameter balancing the classification objective function and the regularization penalty. SLR uses the logistic loss as the classification objective function and the LASSO regularization as penalty, that is:

$$c(\mathbf{w}, X) = \sum_{i=1}^{n} \log\left(1 + \exp(-y_i(\mathbf{w}^T x_i + b))\right) \tag{7a}$$

$$penalty(\mathbf{w}) = \sum_{i=1}^{p} |w_i| \tag{7b}$$

An important property of LASSO regularization is that it can get an estimation of $\mathbf{w}$ with exact zero coefficient. Since each coefficient of $\mathbf{w}$ corresponds to one feature, features with coefficients that are close to 0 are then eliminated. And feature selection is achieved and only features with nonzero coefficient in $\mathbf{w}$ will be used in the classifier.

2 Details of Hetero-RP

Let $\mathcal{O} = o_1, o_2, \cdots, o_n$ be the set of objects that possibly indicate metagenomic contigs for binning, RBP interaction sites for prediction, etc. Each object is represented by a feature vector, for features from a single data source or concatenation of multiple data sources. Mathematically, each data source i with feature dimensionality p_i on the same set of n objects is represented by a data matrix $X_i \in \mathbb{R}^{p_i \times n}$. When the number of data sources $m > 1$, we stack them together by $X = \begin{pmatrix} X_1 \\ \cdots \\ X_m \end{pmatrix}$. $X \in \mathbb{R}^{p \times n}$ is the stacked data matrix illustrated in Figure **??**(B) and $p \triangleq p_1 + p_2 + \cdots + p_m$, where $p_1, p_2, \cdots, p_m$ are the feature dimensionality of each data source, respectively. Then the feature vector of o_i is denoted as $X_{\cdot i}$, for $i = 1, 2, \cdots, n$.

We encode "positive-links" and "negative-links" by an undirected signed graph $\mathcal{G} = (\mathcal{O}, \mathcal{P}, \mathcal{N})$, where $\mathcal{O}$ is the set of objects, $\mathcal{P}$ and $\mathcal{N}$ consists of "positive-links" and "negative-links", respectively. $\mathcal{G}$ can be represented by adjacency matrix $A \in \mathbb{R}^{n \times n}$, where $A_{ij} = 1$ if there is a "positive-link" between o_i and o_j, $A_{ij} = -1$ if there is a "negative-link" between o_i and o_j, and $A_{ij} = 0$ otherwise. With these notations, Hetero-RP aims to find a p-dimensional vector $W = [w_1, w_2, \cdots, w_p]$ for overall p features, so as to minimize the *inconsistency* between

the signed graph $\mathcal{G}$ and the feature-wise rescaled data matrix $\text{diag}(W)X$, where $\text{diag}(\cdot)$ diagonalizes the vector into a diagonal matrix.

$$\begin{aligned}\min_W L(W) &= \sum_{i,j} A_{ij} \left\|\text{diag}(W)X_{\cdot i} - \text{diag}(W)X_{\cdot j}\right\|^2 \\ &= \text{tr}(\text{diag}(W)XLX^T\text{diag}(W)), \\ \text{s.t.} \quad & W \geq 0, \text{ and } \sum_i W_i = p\end{aligned} \tag{8}$$

where $L = D - A$ denotes the *Laplacian matrix* [2] of adjacency matrix A and D indicates the diagonal matrix whose d_{ii} entry equals the sum of the i-row (or column due to symmetry) of A. In the above formulation, inconsistency decreases when object pairs joined by *positive-links* are pulled closer after data matrix is rescaled. To avoid trivial solution, we we enforce W to be nonnegative and conserved in sum, i.e. $\sum_i W_i = p$.

Unlike conventional feature selection that assumes most features are irrelevant, Hetero-RP assumes the majority of features are useful. Among those useful features, only a subset of them are more or less informative (scaling $\neq 1$) whereas the rest are neutral (scaling $= 1$). In comparison, conventional feature selection treats features either relevant (scaling $= 1$) or irrelevant (scaling $= 0$). To examine whether the assumption of Hetero-RP holds, for the clustering scenario, the dip test [7] can be used to check if each feature is multi-modal. If not, that feature is regarded uninformative and thus excluded. For the classification task, univariate metrics such as t-test can also be applied to score each feature and the resulting p-values are obtained. Features whose p-value exceeds a certain threshold are not considered as well. The assumption of Hetero-RP naturally leads to the regularization of $\Delta W = W - 1$, the deviation from unit scaling. Thus, Equation (8) can be represented in terms of ΔW:

$$\begin{aligned}\min_W L(\Delta W) &= \text{tr}(\text{diag}(1+\Delta W)XLX^T\text{diag}(1+\Delta W)) + \lambda \left\|\Delta W\right\|^2 \\ \text{s.t.} \quad & \Delta W_i \geq -1, \text{ and } \sum_i \Delta W_i = 0\end{aligned} \tag{9}$$

where the parameter $\lambda > 0$ shrinks scaling towards unit and towards each other. It is easy to verify that the constraints on W and ΔW are essentially the same. Specifically, $\sum_i W_i = p$ and $W_i \geq 0$, naturally become the constraints on ΔW, $\sum_i \Delta W_i = 0$ and $\Delta W_i \geq -1$.

Equation (9) can be rewritten into the following form:

$$\begin{aligned}\min_W L(\Delta W) &= \left\|Z + Zdiag(\Delta W)\right\|_F^2 + \lambda \left\|\Delta W\right\|^2 \\ \text{s.t.} \quad & \Delta W \geq -1, \text{ and } \sum_i \Delta W_i = 0\end{aligned} \tag{10}$$

where $XLX^T = ZZ^T$ holds when only "positive-links" are available. In that case, $L = UU^T$ is decomposable by eigen-decomposition due to positive semi-definiteness of L. U is a $n \times r$ matrix, where r represents the rank of L, and $Z = U^TX^T$ is an $r \times p$ matrix. Note that when "negative-links" are available, the positive semi-definiteness of L may not necessarily holds. In that case, $\left[XLX^T\right]_+ = ZZ^T$ still holds, where $[x]_+ = \max(0, x)$ is the entry-wise hinge loss, keeping the positive semi-definiteness of $\left[XLX^T\right]_+$.

According to the scaled-LASSO algorithm, the tuning parameter λ can be selected automatically [13] by handling the following two steps iteratively until convergence:

$$\Delta\widehat{W} \leftarrow \arg \min_{\substack{\Delta W \geq -1 \\ \sum_i \Delta W_i = 0}} \left\|Z + Zdiag(\Delta W)\right\|_F^2 + 2p\lambda_0\widehat{\sigma} \left\|\Delta W\right\|^2 \tag{11a}$$

$$\widehat{\sigma} \leftarrow \frac{1}{\sqrt{p}} \left\|Z + Zdiag(\Delta\widehat{W})\right\|_F \tag{11b}$$

λ_0 is chosen according to the suggestion of [10]. In particular, $\lambda_0 = B/\left(p - 1 + B^2\right)^{1/2}$, where $B = tq(1 - p^{1/2}/(2r\log r), p-1)$ with $tq(\alpha, d)$ the α-th quantile of a t-distribution with d degrees of freedom.

However, one major drawback of Equation 10 involves eigen-decomposition, which is computationally unaffordable when large n. In the implementation details, we do not explicitly decompose L, but solve an equivalent quadratic programming (QP) problem to speed up the process. In particular, by letting Y be the diagonal vector of XLX, i.e., $Y_i = (XLX)_{ii}$, $i = 1, 2, \cdots, n$. Also, Y_i can be analytically expressed by $Y_i = (X_{i\cdot}L)X_{i\cdot}^T$ where $X_{i\cdot}$ is the i-th row of X. The analytic expression of Y helps to avoid the colossal computation of XLX when p is large. Therefore, we reformulate Equation 10 into the equivalent quadratic programming form:

$$\begin{aligned}\arg\min L(\Delta W) &= \sum_i Y_i(\Delta W_i + 1)^2 + \lambda\Delta W_i^2 \\ &= \Delta W^T diag(Y + \lambda)\Delta W + 2Y^T\Delta W + const \\ \text{s.t.} \quad & \Delta W \geq -1, \text{ and } \sum_i \Delta W_i = 0\end{aligned} \tag{12}$$

where $diag(Y+\lambda)$ is a diagonalize vector with i-th entry equals to $Y_i+\lambda$. Therefore, the iterative algorithm described by Equation 11 can be changed to the following form:

$$\Delta\widehat{W} \leftarrow \arg \min_{\substack{\Delta W \geq -1 \\ \sum_i \Delta W_i = 0}} \sum_i Y_i(\Delta W_i + 1)^2 + 2p\lambda_0\widehat{\sigma} \left\|\Delta W\right\|^2, \tag{13a}$$

$$\widehat{\sigma} \leftarrow \sqrt{\frac{1}{p}\sum_i Y_i(\Delta W_i + 1)^2}, \tag{13b}$$

3 Simulation Studies

We first evaluated Hetero-RP based upon the synthetic datasets from simulation. The advantage of using synthetic datasets is because the correct cluster assignments are known a priori, which can be helpful to evaluate the performance of Hetero-RP with respect to the amount of auxiliary knowledge available.

We used the ellipsoidal cluster generators designed for large high-dimensional datasets and large numbers of clusters [6]. Specifically, the generator creates ellipsoidal clusters with the major axis at an arbitrary orientation. For each cluster, data points are generated at a Gaussian-distributed distance from a uniformly random point on the major axis, in a uniformly random direction. Using this method, 20 different datasets with varying combination of cluster number and feature dimensionality were generated, as described in Table S1. Also, the two-dimensional Principal Coordinates Analysis (PCoA) plot of each dataset is illustrated in Figure S1.

Parameter	range
Number of Clusters	10
Dimensionality of Features	100, 1000
Size of each cluster	uniformly in $[50, 500]$

Table S1: Synthetic ellipsoidal datasets of high dimensionality of features. For each of the two possible combinations of cluster number and feature dimensionality, 10 different repeats were generated, giving 20 datasets in all.

We next simulated the auxiliary knowledge for each synthetic dataset. Recall that the auxiliary knowledge can be decomposed into the set of "positive-links" $\mathcal{P}$ and the set of "negative-links" $\mathcal{N}$. We aim to construct the balanced "positive-links" and "negative-links" set, in other words, $\|\mathcal{P}\| = \|\mathcal{N}\|$. Given the knowledge of cluster assignments a priori, for each possible pair of objects (o_i, o_j) belonging to the same cluster, we uniformly chose object o_k so that (o_i, o_k) belongs to different cluster. Therefore, the obtained triple of objects (o_i, o_j, o_k) would be picked at random with probability $0 < p < 1$. Once picked, we added the undirect link (o_i, o_j) and (o_i, o_k) to the "positive-links" and the "negative-links" set, respectively. The options of sampling probability p contain $0.1\%, 1\%, 5\%, 10\%, 15\%, 20\%, 25\%, 30\%, 35\%, 40\%, 45\%, 50\%, 55\%, 60\%, 65\%, 70\%, 75\%, 80\%, 85\%, 90\%, 95\%, 99\%$. For each p, 10 different repeats were generated, giving 220 simulated auxiliary knowledge in all, as described in Table S2.

We considered four different scenario to utilize the auxiliary knowledge. (1) both "positive-links" and "negative-links" are available; (2) though both "positive-links" and "negative-links" are available, the amount is insufficient, so the original data matrix X is also incorporated; (3) only "positive-links" are available; (4) only insufficient "positive-links" are available, so the original data matrix X is also incorporated. Since we aimed to evaluate the effect of the auxiliary knowledge independently from the clustering methods involved later on, we therefore used the internal cluster indices to measure the compactness and separation of clusters, with respect to the correct cluster assignments known a priori. As shown in Figure S2, we have the following conclusions:

1. When there is sufficient amount of auxiliary knowledge available, incorporating the original data matrix X undermines the performance. However, when the auxiliary knowledge is insufficient ($\leq 1\%$), incorporating the original data matrix X doesn't hurt.

2. In most cases, utilizing both "positive-links" and "negative-links" are preferable than using "positive-links" alone.

3. With the increase amount of auxiliary knowledge, the increasing trend of the performance using "positive-links" only starts to slow down. In that case, progressively collecting the auxiliary knowledge may not worth the effort.

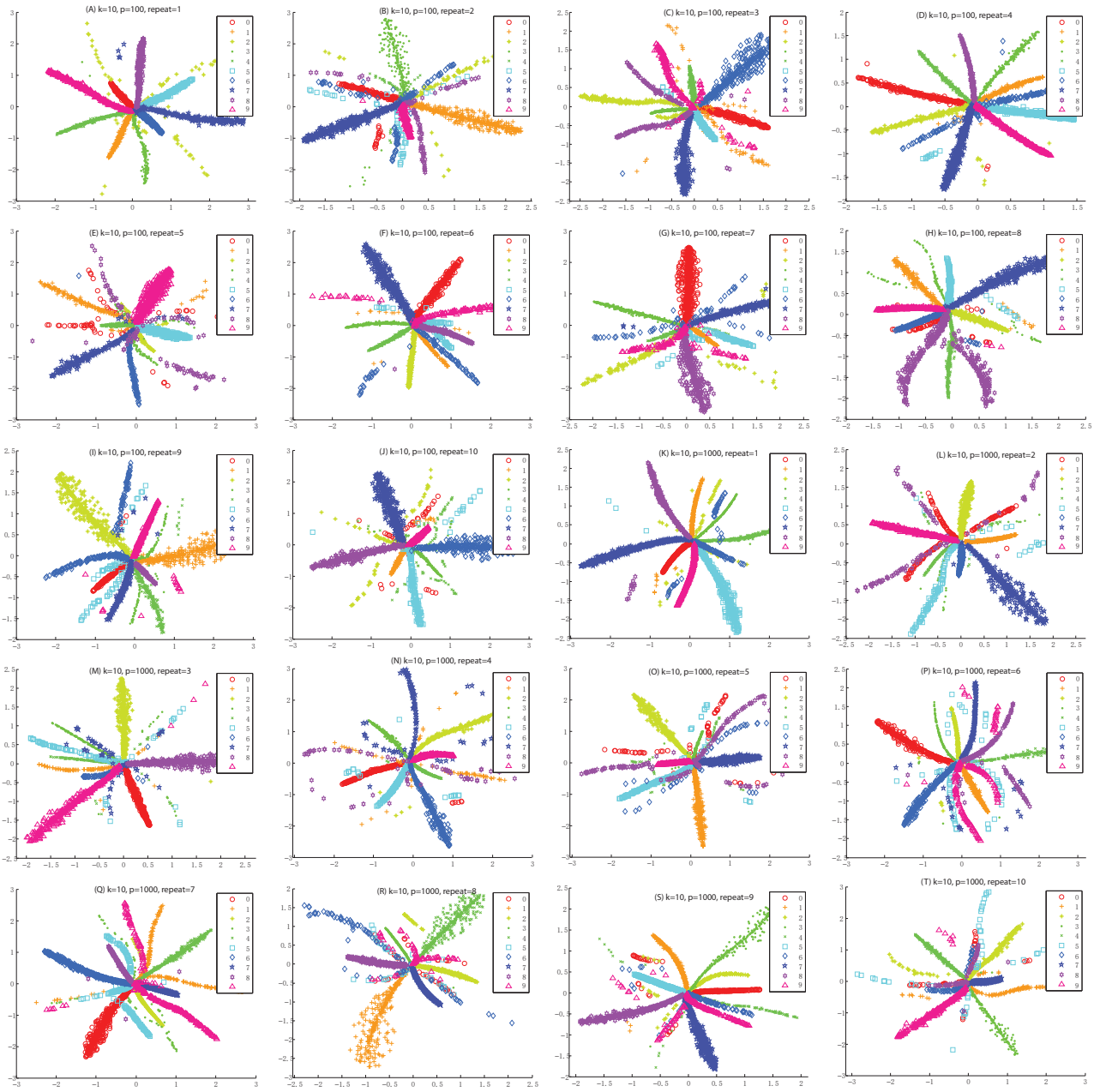

Figure S1: The two-dimensional Principal Coordinates Analysis of 20 different synthetic ellipsoidal datasets. Each cluster is displayed with different color as well as different marks. (A)-(J) indicate 10 different repeats of synthetic ellipsoidal datasets with 10 cluster numbers and 100 features. (K)-(T) indicate 10 different repeats of synthetic ellipsoidal datasets with 10 cluster numbers and 1000 features.

Figure S2: The comparison between different amount of auxiliary knowledge on 20 synthetic datasets. "P+N" stands for using both "positive-links" and "negative-links" set. "P+N+X" stands for not only using both "positive-links" and "negative-links" set, but also the original data matrix X by pretending the auxiliary knowledge is insufficient. "P" stands for using only "positive-links" set. And "P+X" stands for using "positive-links" set together with the original data matrix X. The evaluation is based on the most commonly used internal cluster index, the silhouette value.

Parameter	range
p	0.1%,1%,5%,10%,15%,20%,25%,30%,35%,40%,45%,50%,55%,60%,65%,70%,75%,80%,85%,90%,95%,99%

Table S2: The options of sampling probability p in simulated auxiliary knowledge. For each p, 10 different repeats were generated, giving 220 simulated auxiliary knowledge in all.

4 Application to Clustering: Metagenomic Contig Binning

4.1 Details of COCACOLA [9]

A microbial community is comprised of a set of operational taxonomic units (OTUs) at different abundance levels, and our objective is to group assembled sequence fragments, also known as contigs, into the genomic OTU bins from which they were originally derived. OTUs are expected to be disentangled based on contigs comprising either the discriminative abundance or dissimilarity among sequences in terms of tetra-mer composition. The rationale of binning contigs into OTUs relies on the underlying assumption that contigs originating from the same OTU share similar relative abundance as well as sequence composition.

Formally, COCACOLA encodes the abundance and composition of the n-th contig by a $P \triangleq p_1 + p_2$ dimensional feature vector, $X_{\cdot n}$, $n = 1, 2, \cdots, N$, where p_1 is the number of samples, p_2 is the number of distinct tetra-mers, and N is the total contigs number. Specifically, X_{mn} represents the abundance of the n-th contig in the m-th sample, $m = 1, 2, \cdots, p_1$, respectively. And $X_{p_1+v,n}$ stands for the tetra-mer relative frequency composition of the n-th contig, $v = 1, 2, \cdots, p_2$. Similarly, the feature vector of the k-th OTU is denoted as $W_{\cdot k}$, $k = 1, 2, \cdots, K$. Let $\mathbb{H}_{kn}$ be the indicator function describing whether the n-th contig belongs to the k-th OTU, i.e., $\mathbb{H}_{kn} = 1$ means the n-th contig from the k-th OTU and $\mathbb{H}_{kn} = 0$ otherwise. Therefore, the OTU-contig membership can be written as:

$$X \approx W\mathbb{H} \quad s.t.\ W \geq 0,\ \ \mathbb{H} \in \{0,1\}^{K\times N}, \|\mathbb{H}_{\cdot n}\|_0 = 1 \tag{14}$$

where $W = (W_{\cdot 1}, W_{\cdot 2}, \cdots, W_{\cdot K})$ is a $P \times K$ nonnegative matrix with each column encoding the feature vector of the corresponding OTU. And $\mathbb{H} = (\mathbb{H}_{\cdot 1}, \mathbb{H}_{\cdot 2}, \cdots, \mathbb{H}_{\cdot N})$ is a $K \times N$ binary matrix with each column encoding the indicator function of the corresponding contig. $\|\mathbb{H}_{\cdot n}\|_0 = \sum_{k=1}^{K} \mathbb{H}_{kn} = 1$ ensures the n-th contig belongs exclusively to only one particular OTU.

The matrices W and $\mathbb{H}$ can be achieved by minimizing the Frobenius norm:

$$\arg\min_{W,\mathbb{H}} \|X - W\mathbb{H}\|_F^2 \quad s.t.\ \ \mathbb{H} \in \{0,1\}^{K\times N}, \|\mathbb{H}_{\cdot n}\|_0 = 1 \tag{15}$$

Note that the binary constraint makes Equation (15) NP-hard to optimize. Thus, a relaxation of $\mathbb{H}$ is taken by:

$$\arg\min_{W,H} \|X - WH\|_F^2 \quad s.t.\ W, H \geq 0 \tag{16}$$

where H becomes a coefficient matrix instead of an indicator matrix. It has been observed that by imposing sparsity on each column of H, the hard clustering behavior can be facilitated [**?**]. Therefore, Equation (16) is further formulated into the Sparse Nonnegative Matrix Factorization (SNMF) form [**?**]:

$$\arg\min_{W,H\geq 0} \|X - WH\|_F^2 + \alpha \sum_{n=1}^{N} \|H_{\cdot n}\|_1^2 \tag{17}$$

where $\|\cdot\|_1$ indicates L_1-norm. Due to non-negativity constraints of H, $\|H_{\cdot n}\|_1$ stands for the column sum of the n-th column vector of H. The parameter $\alpha > 0$ balances the trade-off between approximation accuracy and the sparseness of H.

Equation (23) is solved by alternating nonnegative least squares (ANLS) [**?**], which iteratively handles two nonnegative least square (NNLS) subproblems in Equation (22) until convergence:

$$H \leftarrow \arg\min_{H\geq 0} \|X - WH\|_F^2 + \alpha \sum_{n=1}^{N} \|H_{\cdot n}\|_1^2 \tag{18a}$$

$$W \leftarrow \arg\min_{W\geq 0} \left\|X^T - H^T W^T\right\|_F^2 \tag{18b}$$

Given the aforementioned $P \times N$ nonnegative matrix X as input, the upgraded COCACOLA constructs an $N \times N$ similarity matrix $\mathcal{A}$ from k-nearest neighbor network, as the local embedding of X. Here the goal is to connect vertex $X_{\cdot i}$, $i = 1, 2, \cdots, n$, with its k nearest neighbors. Concretely, if vertex i and vertex j are connected, the edge weight is quantified by Gaussian similarity function $\exp\left\{-\frac{\|X_{\cdot i} - X_{\cdot j}\|^2}{2\sigma^2}\right\}$, where the

parameter σ controls the width of the neighborhoods. Inspired by similar ideas in reproducing kernel Hilbert spaces (RKHS) methods [**?**], one option is to take the 5% quantile of the values $\|X_{\cdot i} - X_{\cdot j}\|^2$. However, this definition leads to a directed graph, as the neighborhood relationship is not symmetric. Therefore, one way is simply ignore the directions of the edges.

Similar to spectral clustering [14], eigenvectors corresponding to the K largest eigenvalues of the similarity matrix $\mathcal{A}$ is computed, denoted as $u_1, u_2, \cdots, u_K$. Let U be the $K \times N$ matrix containing the eigenvectors $u_1, u_2, \cdots, u_K$ as columns. Furthermore, a step called spectral rotation [**?**] is applied to transform U into a $K \times N$ matrix $\tilde{U}$. Since $\tilde{U}$ may contains negative values, a constant is added to each entry of $\tilde{U}$, and resulting the new nonnegative matrix $\hat{U}$ as the input.

4.2 Details of Input Data

Each contig is represented by a $P \triangleq p_1 + p_2$ dimensional column feature vector including p_1 dimensional coverage and p_2 dimensional tetra-mer composition. The coverage denotes the average number of mapped reads per base pair from each of p_1 different samples. While the tetra-mer composition denotes the tetra-mer frequency for the contig itself plus its reverse complement. Due to palindromic tetra-mers, $p_2 = 136$.

The coverage of all the N contigs is represented by an $N \times p_1$ matrix Y, where N is the number of contigs of interest and Y_{nm} indicates the coverage of the n-th contig from the m-th sample. Whereas the tetra-mer composition of the N contigs are represented by an $N \times p_2$ matrix Z where Z_{nv} indicates the count of v-th tetra-mer found in the n-th contig. Before normalization, a pseudo-count is added to each entry of the coverage matrix Y and composition matrix Z, respectively. As for the coverage, a small value is added, i.e., $Y'_{nm} = Y_{nm} + 100/L_n$, analogous to a single read aligned to each contig as prior, where L_n is the length of the n-th contig. As for the composition, a single count is simply added, i.e., $Z'_{nv} = Z_{nv} + 1$.

The coverage matrix Y is firstly column-wise normalized (i.e., normalization within each individual sample), followed by row-wise normalization (i.e., normalization across p_1 samples) to obtain coverage profile M. The row-wise normalization aims to mitigate sequencing efficiency heterogeneity among contigs.

$$Y''_{nm} = \frac{Y'_{nm}}{\sum_{n=1}^{N} Y'_{nm}} \qquad M_{nm} = \frac{Y''_{nm}}{\sum_{m=1}^{p_1} Y''_{nm}} \tag{19}$$

The composition matrix Z is row-wise normalized for each contig (i.e., normalization across p_2 tetra-mer count) to obtain composition profile V:

$$V_{nv} = \frac{Z'_{nv}}{\sum_{v=1}^{p_2} Z'_{nv}} \tag{20}$$

The feature matrix of contigs is denoted as $X = [M\ V]^T$, as the combination of coverage profile M and composition profile V. To be specific, X is a $P \times N$ nonnegative matrix of which each column represents the feature vector of a particular contig. And COCACOLA takes X as the input data.

4.3 Details of Features after Hetero-RP

For the simulated "species" dataset, the scaling brought from Hetero-RP using co-alignment is shown from Figure S3 to Figure S12, respectively. The blue shadow on the left side of the dash line indicates the scaling of abundance profile, whereas the red shadow on the right side of the dash line indicates the scaling of composition profile. And the green horizontal line indicates the scaling of 1, in other word, the scaling above or below the green line means that the correspond features are more or less informative. As observed by [9, 1], when sample size is small, the binning result is unstable. When sample size increase, the binning result performs better. Consistent with the observation, the scaling of abundance is of the smallest magnitude when sample size is 10, which increase with respect to the increase of sample size. When sample size is large enough (> 60), the scaling of abundance becomes comparable to the scaling of composition. the scaling brought from Hetero-RP using linkage is shown from Figure S13 to Figure S22, respectively. Compared to co-alignment, Hetero-RP using linkage cannot achieve as significant changes. This is within the expectation because the *attraction-link* set of co-alignment is $\sim$ 1800x larger than the set of linkage.

For the real "MetaHIT" dataset, the scaling brought from Hetero-RP using co-alignment and linkage is shown in Figure S23 to Figure S24, respectively.

5 Application to Classification: RBP Binding Site Prediction

5.1 Details of iONMF [12]

Let matrices $X_i \in \mathbb{R}^{m \times n_i}$ represent the data from each source, which contains m rows representing samples and n_i columns representing features. The classic NMF [**?**] seeks a factorization of X_i by the product of a

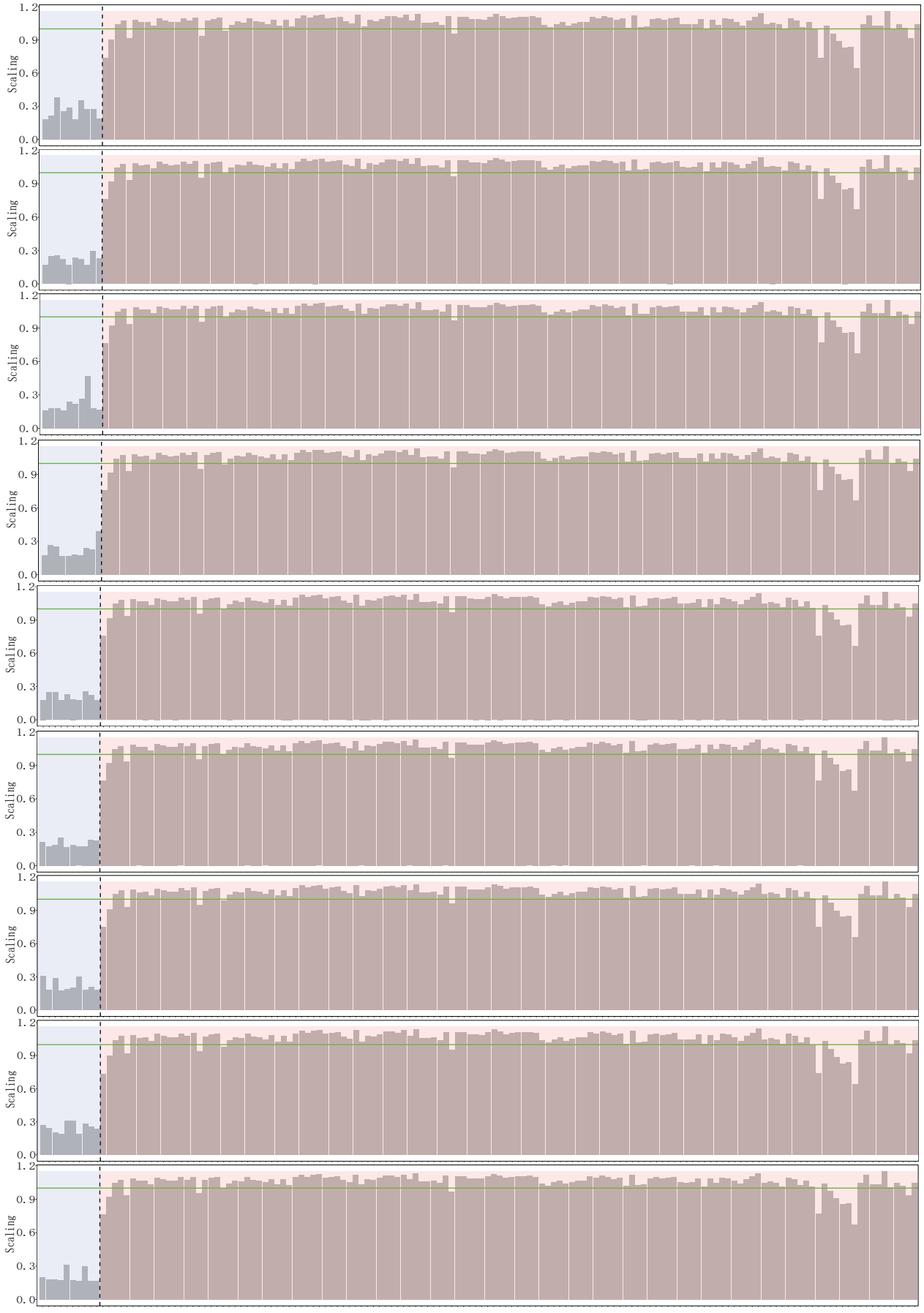

Figure S3: The feature scaling of Hetero-RP using co-alignment when sample size is 10, overall 9 replicates, for the simulated "species" dataset.

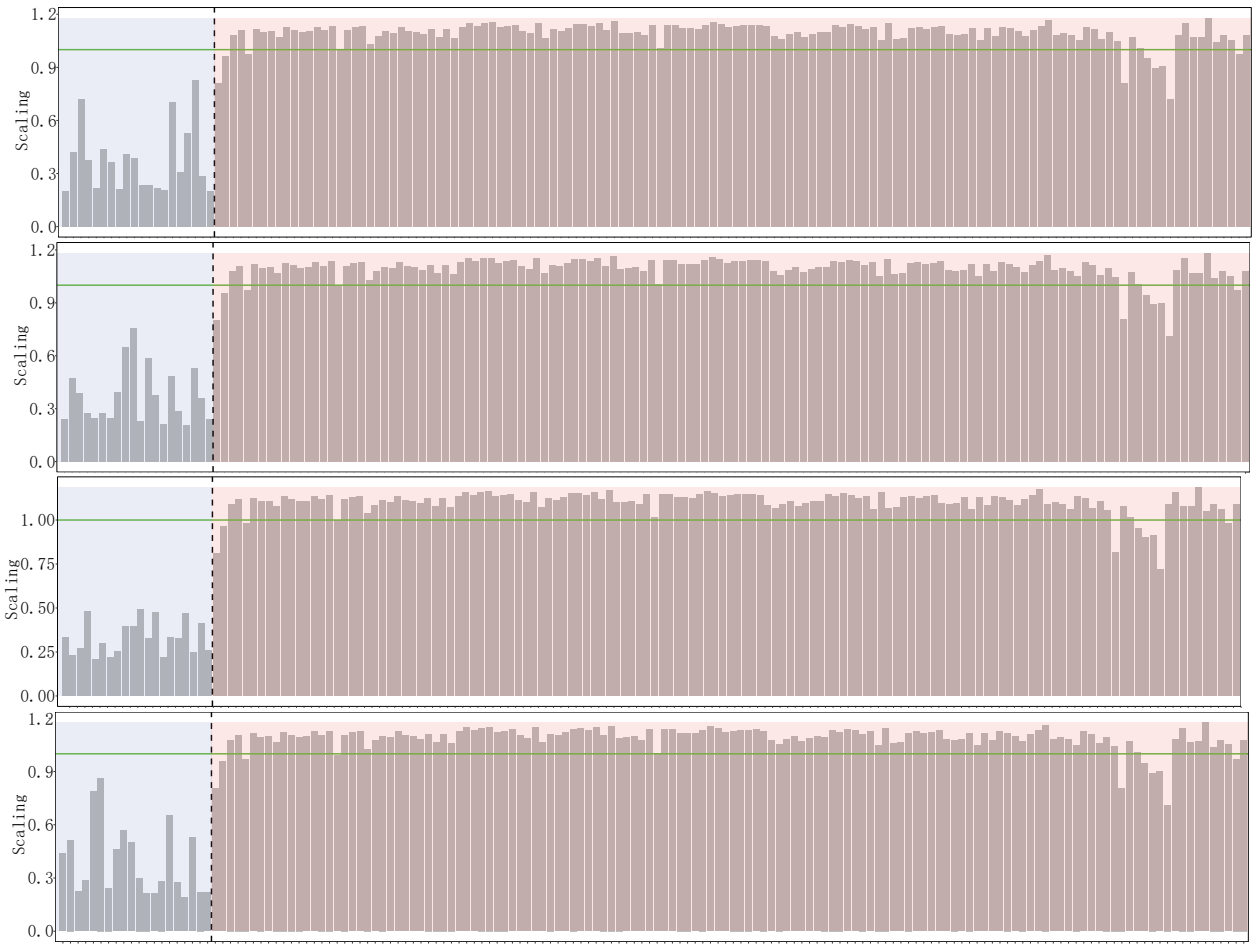

Figure S4: The feature scaling of Hetero-RP using co-alignment when sample size is 20, overall 4 replicates, for the simulated "species" dataset.

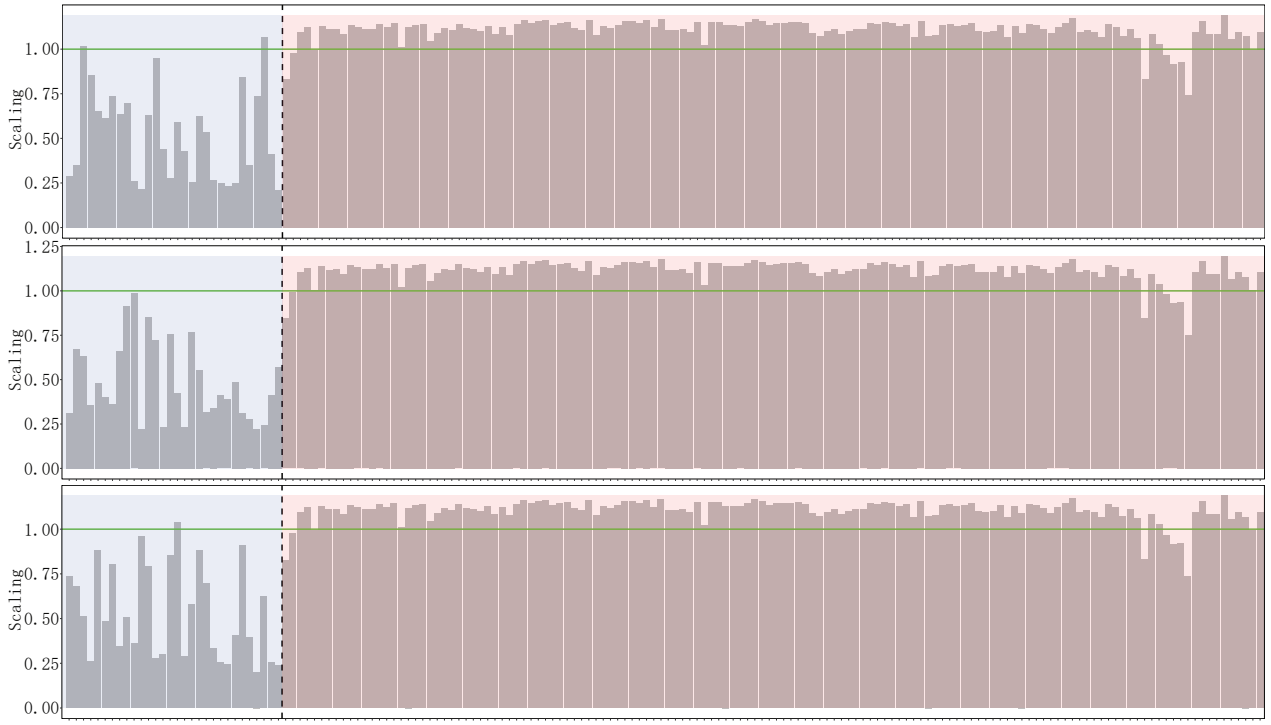

Figure S5: The feature scaling of Hetero-RP using co-alignment when sample size is 30, overall 3 replicates, for the simulated "species" dataset.

Figure S6: The feature scaling of Hetero-RP using co-alignment when sample size is 40, overall 2 replicates, for the simulated "species" dataset.

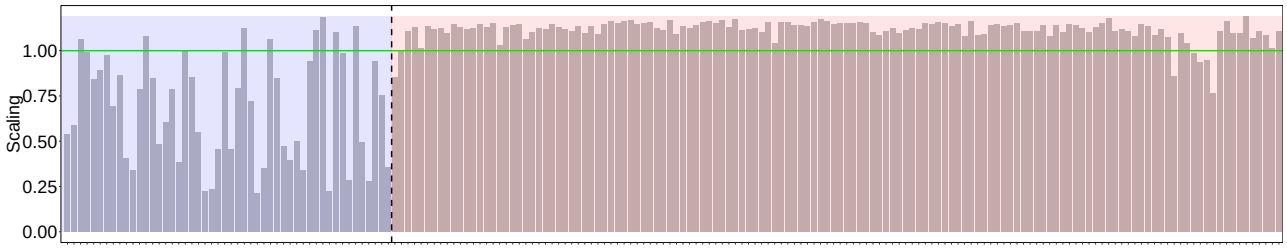

Figure S7: The feature scaling of Hetero-RP using co-alignment when sample size is 50, overall 1 replicate, for the simulated "species" dataset.

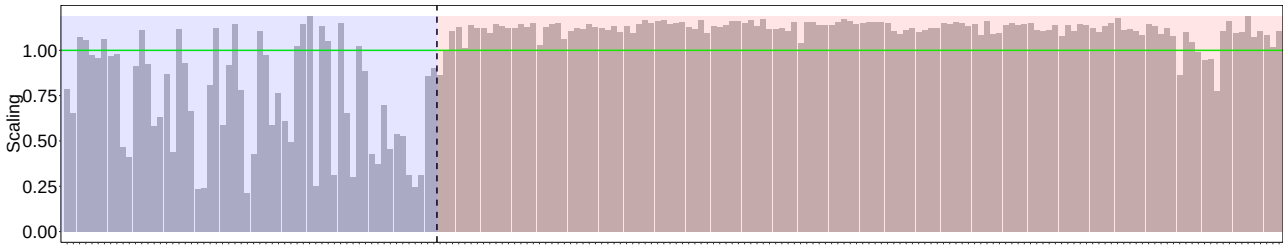

Figure S8: The feature scaling of Hetero-RP using co-alignment when sample size is 60, overall 1 replicate, for the simulated "species" dataset.

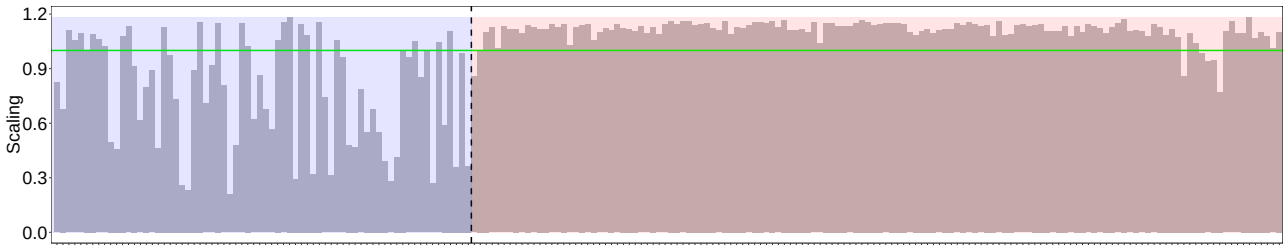

Figure S9: The feature scaling of Hetero-RP using co-alignment when sample size is 70, overall 1 replicate, for the simulated "species" dataset.

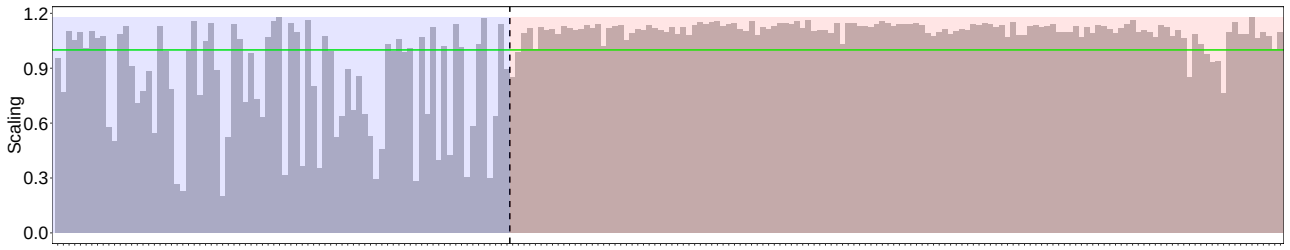

Figure S10: The feature scaling of Hetero-RP using co-alignment when sample size is 80, overall 1 replicate, for the simulated "species" dataset.

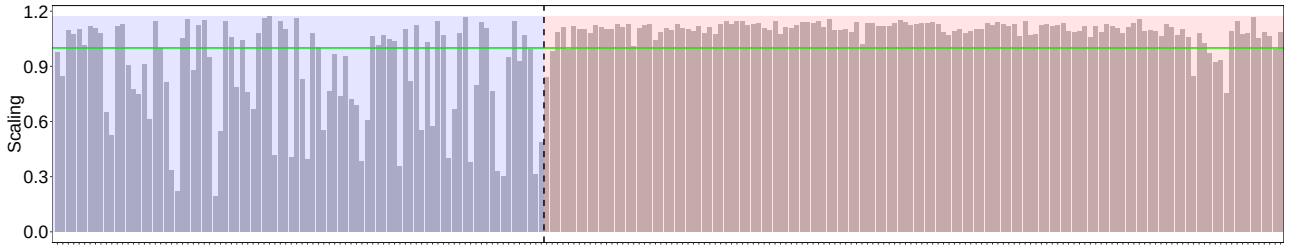

Figure S11: The feature scaling of Hetero-RP using co-alignment when sample size is 90, overall 1 replicate, for the simulated "species" dataset.

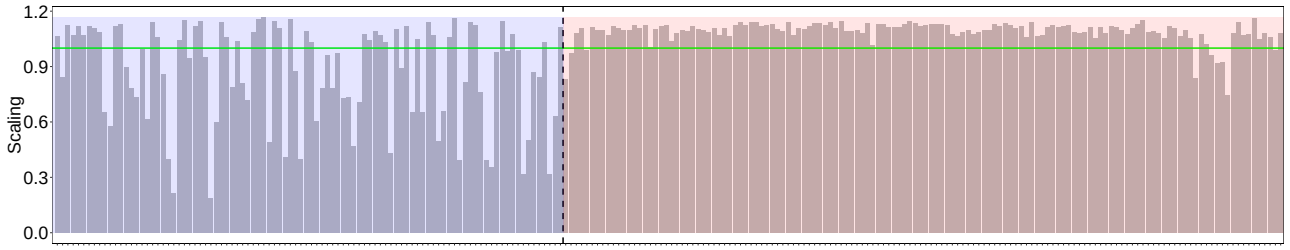

Figure S12: The feature scaling of Hetero-RP using co-alignment when sample size is 96, overall 1 replicate, for the simulated "species" dataset.

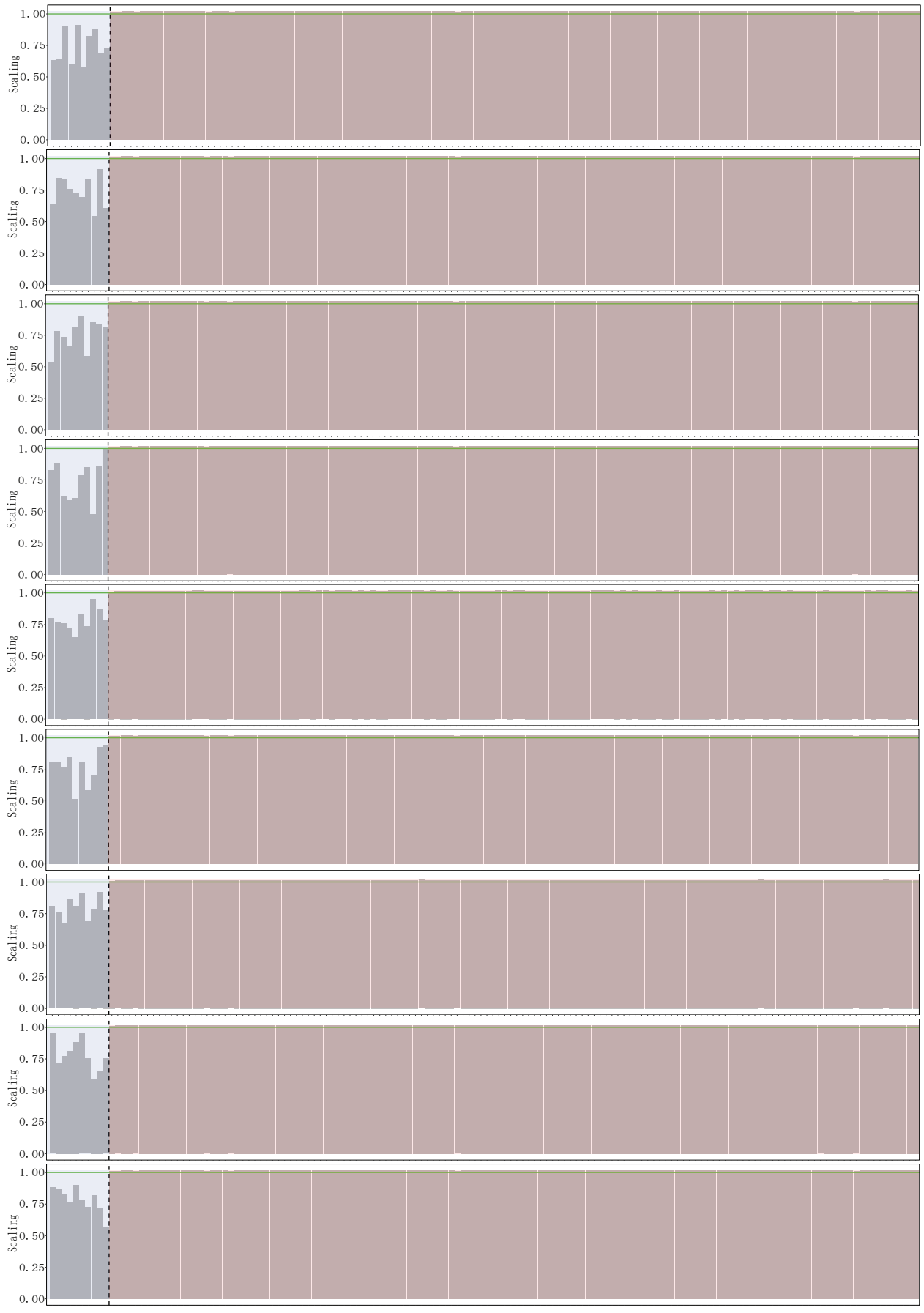

Figure S13: The feature scaling of Hetero-RP using linkage when sample size is 10, overall 9 replicates, for the simulated "species" dataset.

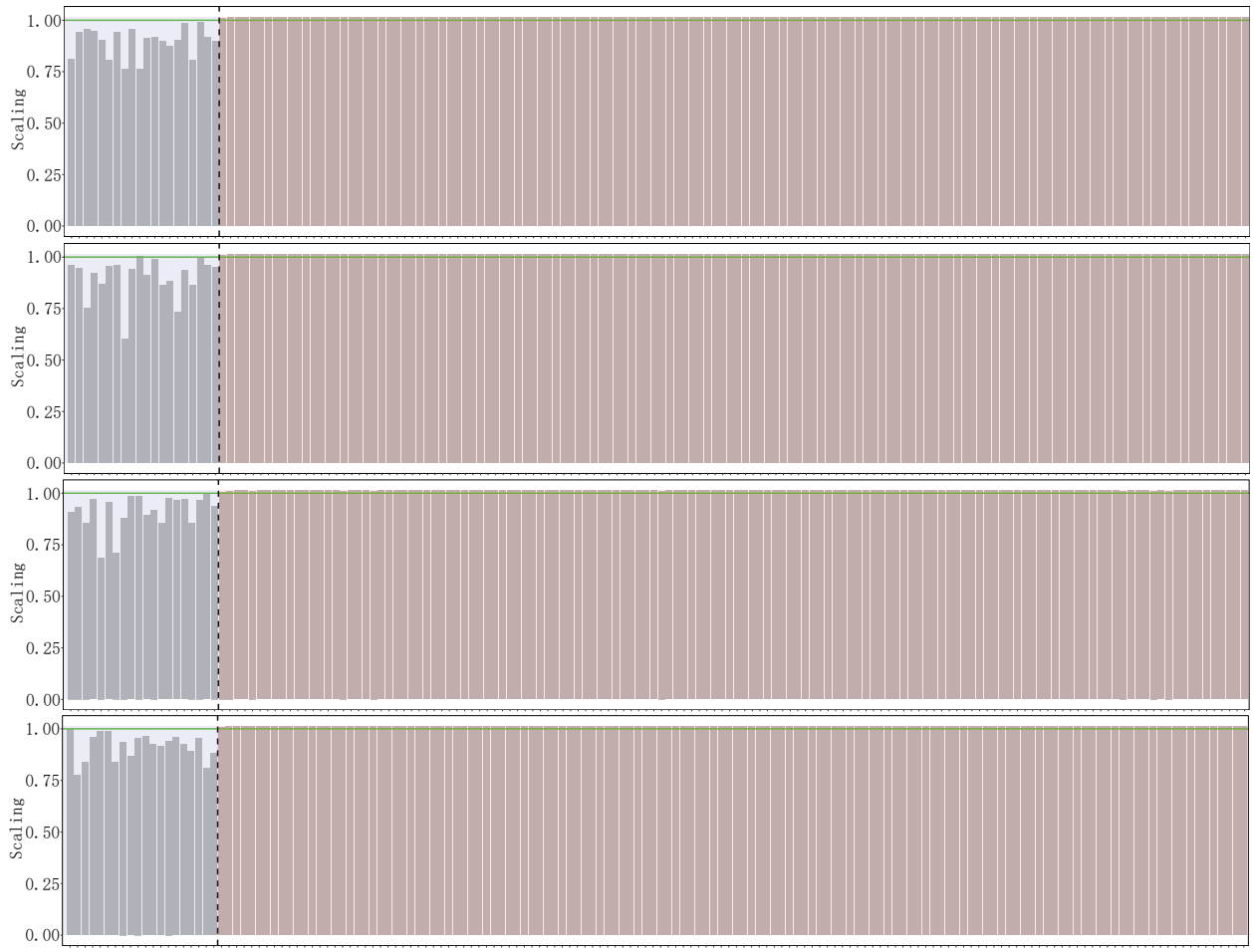

Figure S14: The feature scaling of Hetero-RP using linkage when sample size is 20, overall 4 replicates, for the simulated "species" dataset.

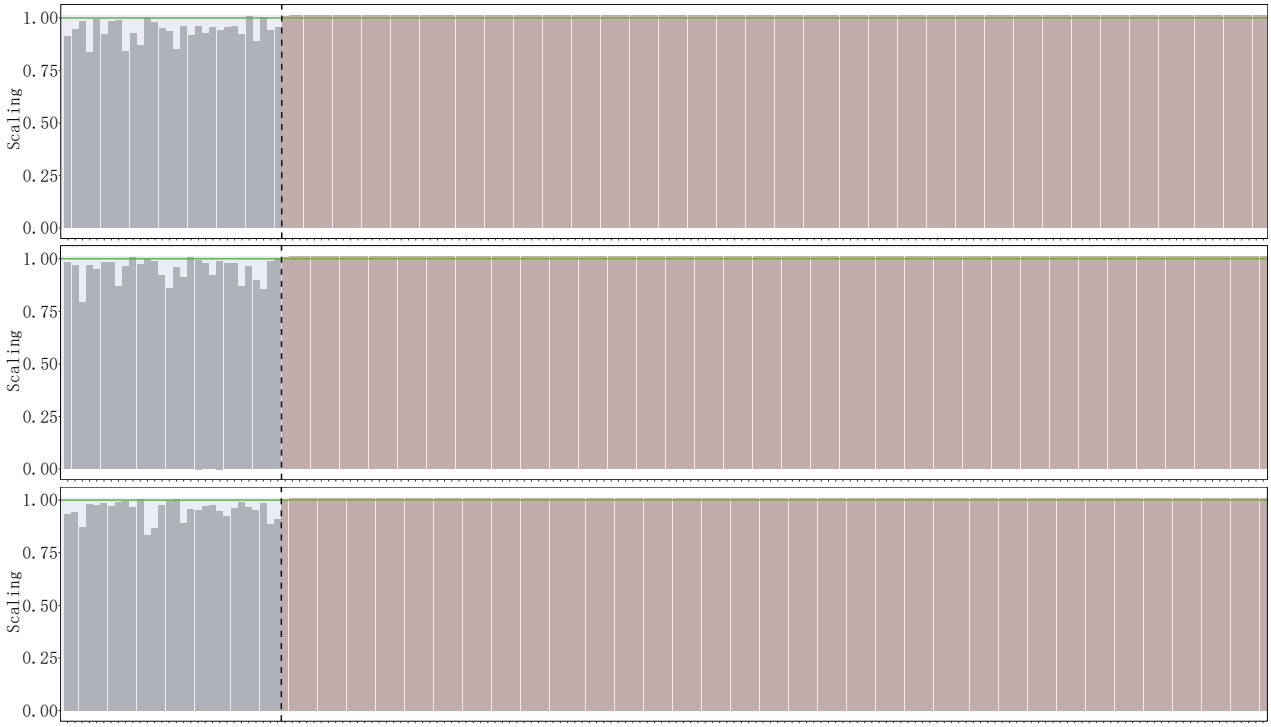

Figure S15: The feature scaling of Hetero-RP using linkage when sample size is 30, overall 3 replicates, for the simulated "species" dataset.

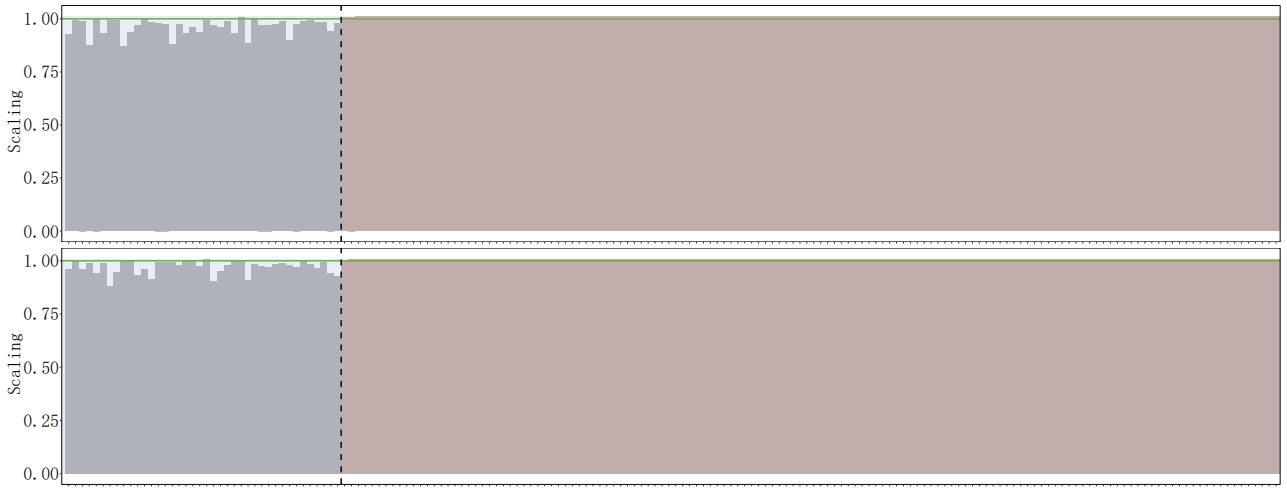

Figure S16: The feature scaling of Hetero-RP using linkage when sample size is 40, overall 2 replicates, for the simulated "species" dataset.

Figure S17: The feature scaling of Hetero-RP using linkage when sample size is 50, overall 1 replicate, for the simulated "species" dataset.

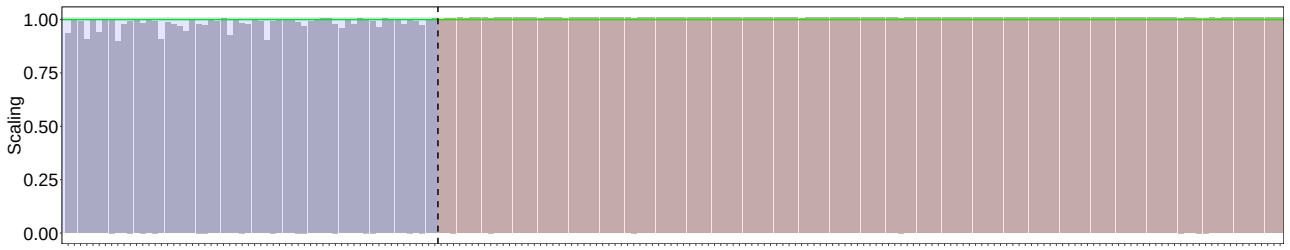

Figure S18: The feature scaling of Hetero-RP using linkage when sample size is 60, overall 1 replicate, for the simulated "species" dataset.

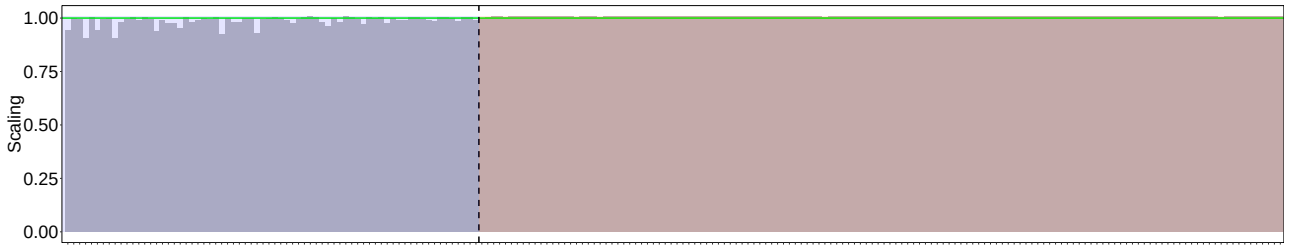

Figure S19: The feature scaling of Hetero-RP using linkage when sample size is 70, overall 1 replicate, for the simulated "species" dataset.

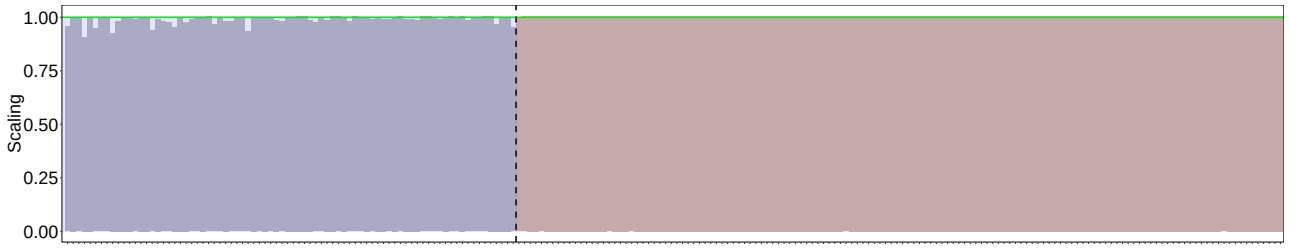

Figure S20: The feature scaling of Hetero-RP using linkage when sample size is 80, overall 1 replicate, for the simulated "species" dataset.

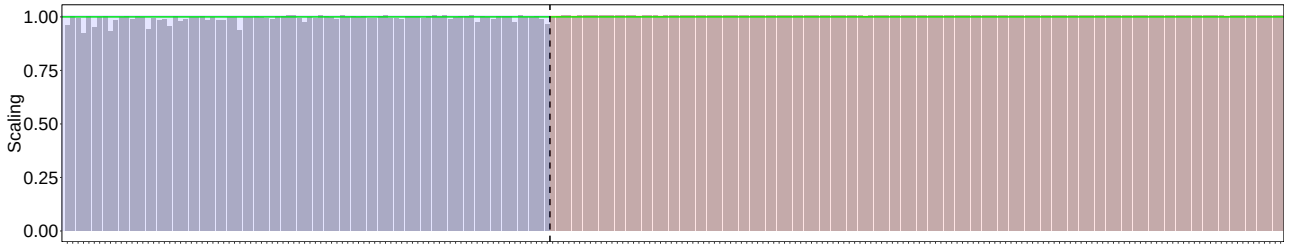

Figure S21: The feature scaling of Hetero-RP using linkage when sample size is 90, overall 1 replicate, for the simulated "species" dataset.

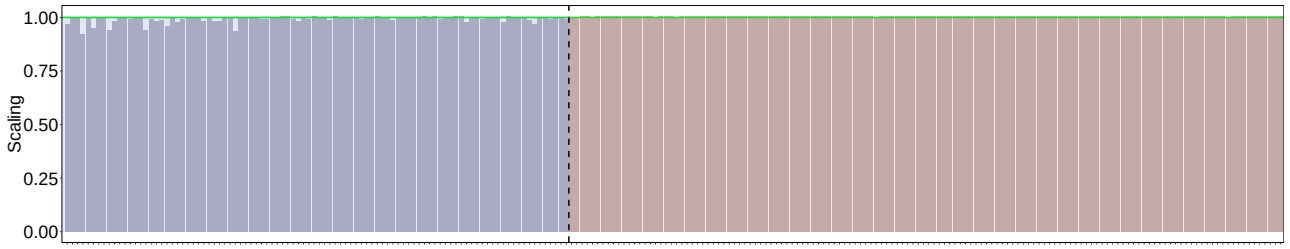

Figure S22: The feature scaling of Hetero-RP using linkage when sample size is 96, overall 1 replicate, for the simulated "species" dataset.

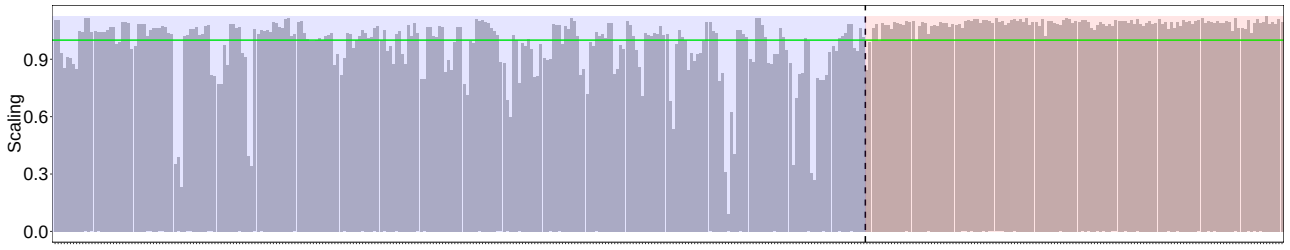

Figure S23: The feature scaling of Hetero-RP using co-alignment for the real "MetaHIT" dataset.

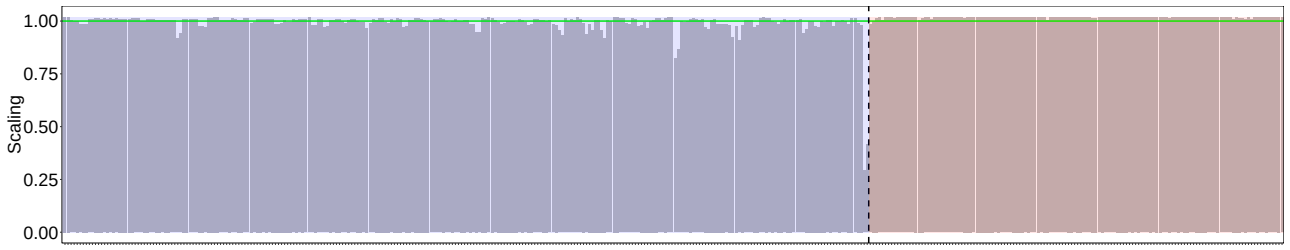

Figure S24: The feature scaling of Hetero-RP using linkage for the real "MetaHIT" dataset.

coefficient matrix $W \in \mathbb{R}^{m\times r}$ and basis matrices $H_i \in \mathbb{R}^{r\times n_i}$. Thus, each sample is assigned to one or more of the r modules, revealed by the coefficient matrix W. Features relevant to each module are encoded in corresponding basis vectors in H_i. iONMF (Orthogonality-regularized NMF) aims to discover non-overlapping features relevant to each module by imposing orthogonality on the basis vectors. Given multiple data matrices X_i, iONMF solves the following optimization problem:

$$J(W, H_i) = \sum_{i=1}^{N} \left\| X_i - W H_i^T \right\|_F^2 + \alpha \left\| H_i^T H_i - I \right\|_F^2 \tag{21}$$

where $W, H_i \geq 0$ and I is the identity matrix. The target column vector Y is treated just as regular matrices, thus having its corresponding basis H_Y. Note that H_Y does not have orthogonality constraints because H_Y only contains one single column. In the training process, iONMF iteratively handles the following updating rule until convergence:

$$W \leftarrow W \circ \sqrt{\frac{\sum_i X_i H_i + Y H_Y}{\sum_i W H_i^T H_i + W H_Y^T H_Y}} \tag{22a}$$

$$H_i \leftarrow H_i \circ \sqrt{\frac{X_i W + \alpha H_i}{H_i W^T W + \alpha H_i H_i^T H_i}} \tag{22b}$$

$$H_Y \leftarrow H_Y \circ \sqrt{\frac{Y^T W}{H_Y W^T W}} \tag{22c}$$

where $\circ$ is the element-wise product. Once the basis matrices H_Y and H_Y are learned through the training process, it is kept fixed in the test process. Specifically, given the data matrices $\hat{X}_i$ in the test set, iONMF solves the following optimization problem:

$$J(\hat{W}) = \sum_{i=1}^{N} \left\| \hat{X}_i - \hat{W} H_i^T \right\|_F^2 \tag{23}$$

Concretely, iONMF iteratively perform the following updating rule until convergence:

$$\hat{W} \leftarrow \hat{W} \circ \sqrt{\frac{\sum_i \hat{X}_i H_i}{\sum_i \hat{W} H_i^T H_i}} \tag{24}$$

Once $\hat{W}$ has been obtained, the predicted cDNA counts for all other genomic positions can be achieved by $\hat{Y} = \hat{W} H_Y^T$. Then, the predicted cDNA counts were in turn used to classify positions as cross- linked or not crosslinked.

5.2 Details of Input Data

For each dataset, the training data matrix contains up to 50000 rows, where each row represents contains features from various data sources for a particular nucleotide position. The data sources include:

Y**: Protein-RNA cDNA count**, 50000×1. Protein-RNA cDNA counts are reported on the current nucleotide position, resulting in 1 column. This column will be used for model fitting and to evaluate the predictive performance.

X_{KMER}**: RNA k-mers**, 50000×25856. Positions $[-50, 50]$ relative to the current nucleotide position are checked for the presence of RNA tetra-mers, resulting in $256 \times 101 = 25856$ columns. The value is binary representing the presence of particular RNA tetra-mer.

X_{RNA}**: RNA secondary structure**, 50000×101. Positions $[-50, 50]$ relative to the current nucleotide position are processed with RNAfold software [3], resulting in 101 columns. The value is numerical representing the probabilities of double-stranded RNA secondary structure at the particular position.

X_{RG}**: Region type**, 50000×505. Position $[-50, 50]$ relative to the current nucleotide position are assigned into one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF, resulting in $5 \times 101 = 505$ columns. The value is binary representing the presence of one particular gene region type at the particular position.

X_{CLIP}**: co-binding proteins cDNA counts**, 50000×3030. For each of the remaining (up to 30) RBP experiments not in the same replicate group, the protein-RNA cDNA counts at positions $[-50, 50]$ relative to the current nucleotide position are reported as 1 for nonzero cDNA counts or 0 otherwise, resulting in up to $30 \times 101 = 3030$ columns. The value is binary.

X_{GO}**: Gene annotation**, 50000×39560. Genomic positions within known genes are annotated with Gene Ontology terms, overall 39560 terms.

For each dataset, the test data matrices $\left(\hat{Y}, \hat{X}_{KMER}, \hat{X}_{RNA}, \hat{X}_{RG}, \hat{X}_{CLIP}, \hat{X}_{GO}\right)$ share the same structure, including 50000 rows as well. It is guaranteed that the nucleotide positions used in the training set are not included in the test set.

5.3 Details of Features after Hetero-RP

5.3.1 Features of co-binding proteins cDNA counts

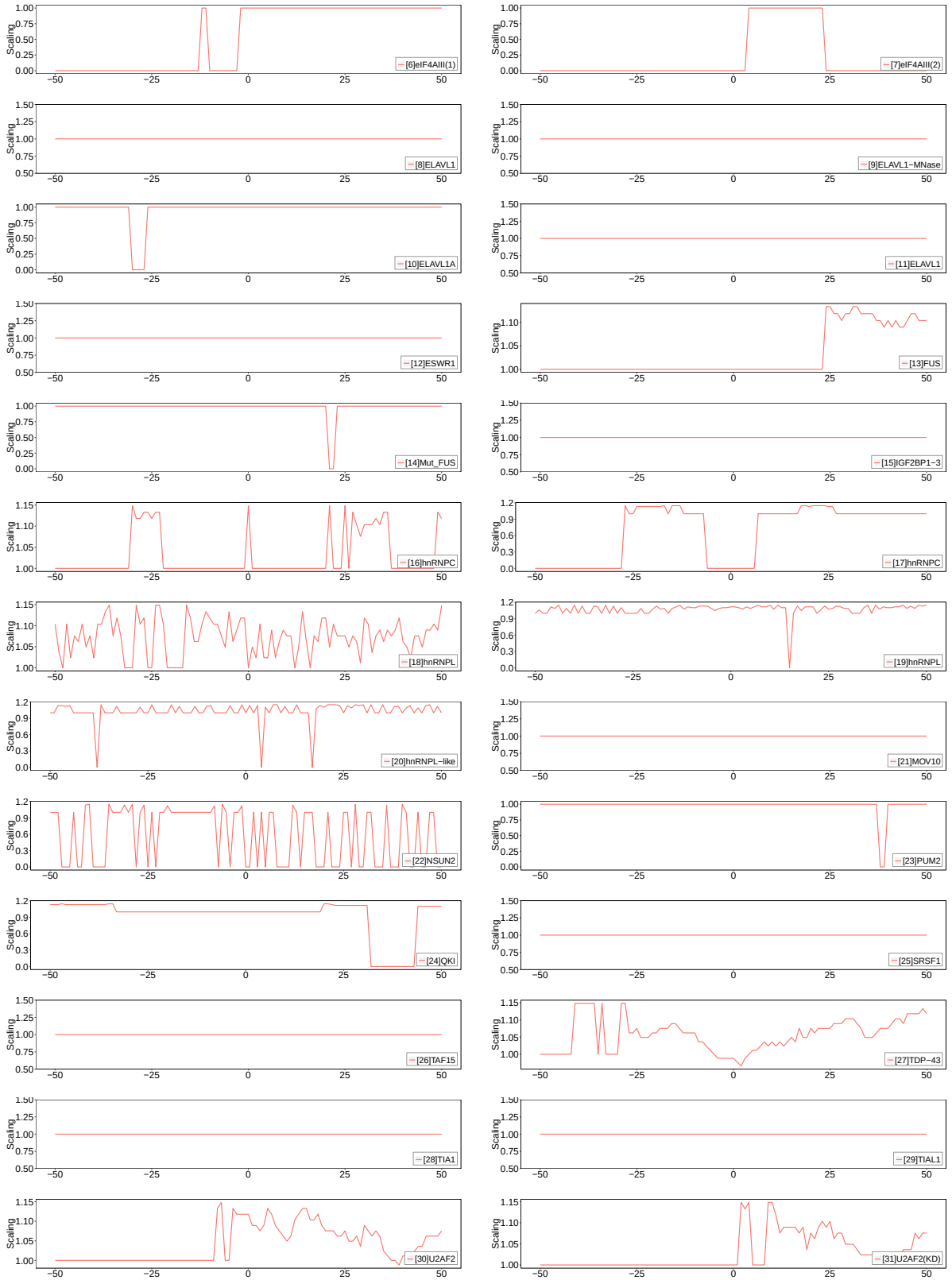

Figure S25: The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [1]Ago_EIF2C1-4.

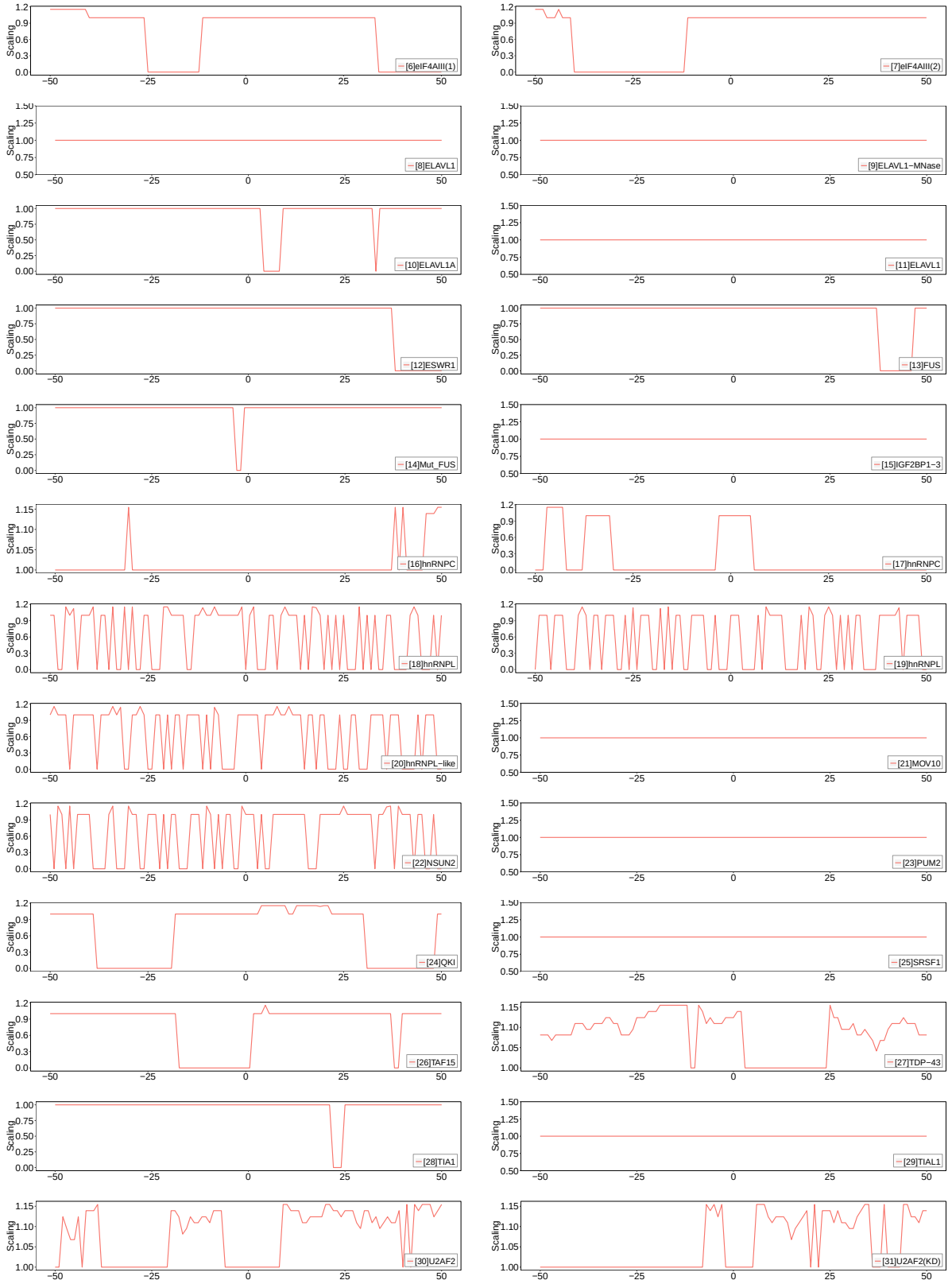

Figure S26: The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [2]Ago2-MNase.

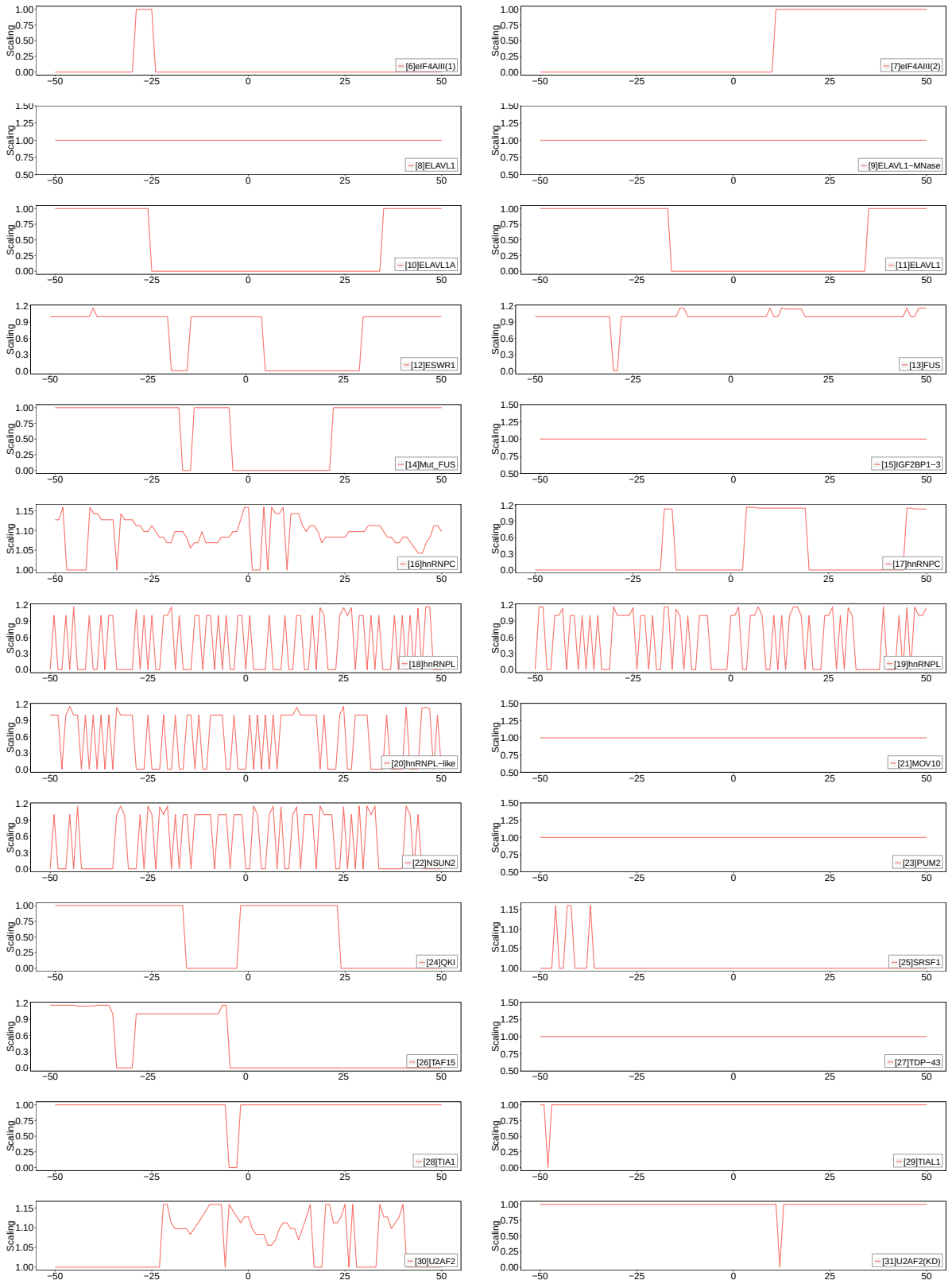

Figure S27: The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [3]Ago2(1).

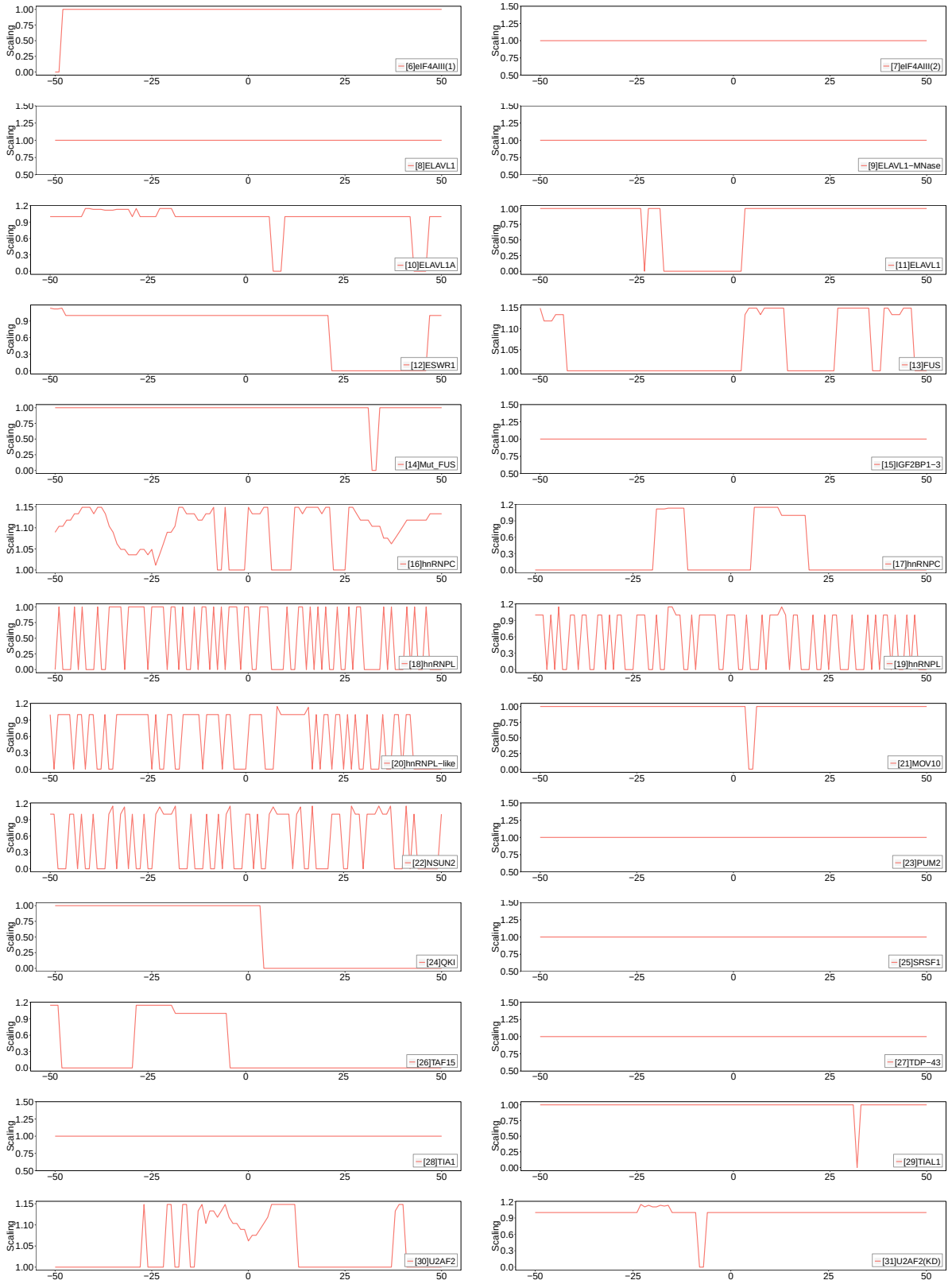

Figure S28: The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [4]Ago2(2).

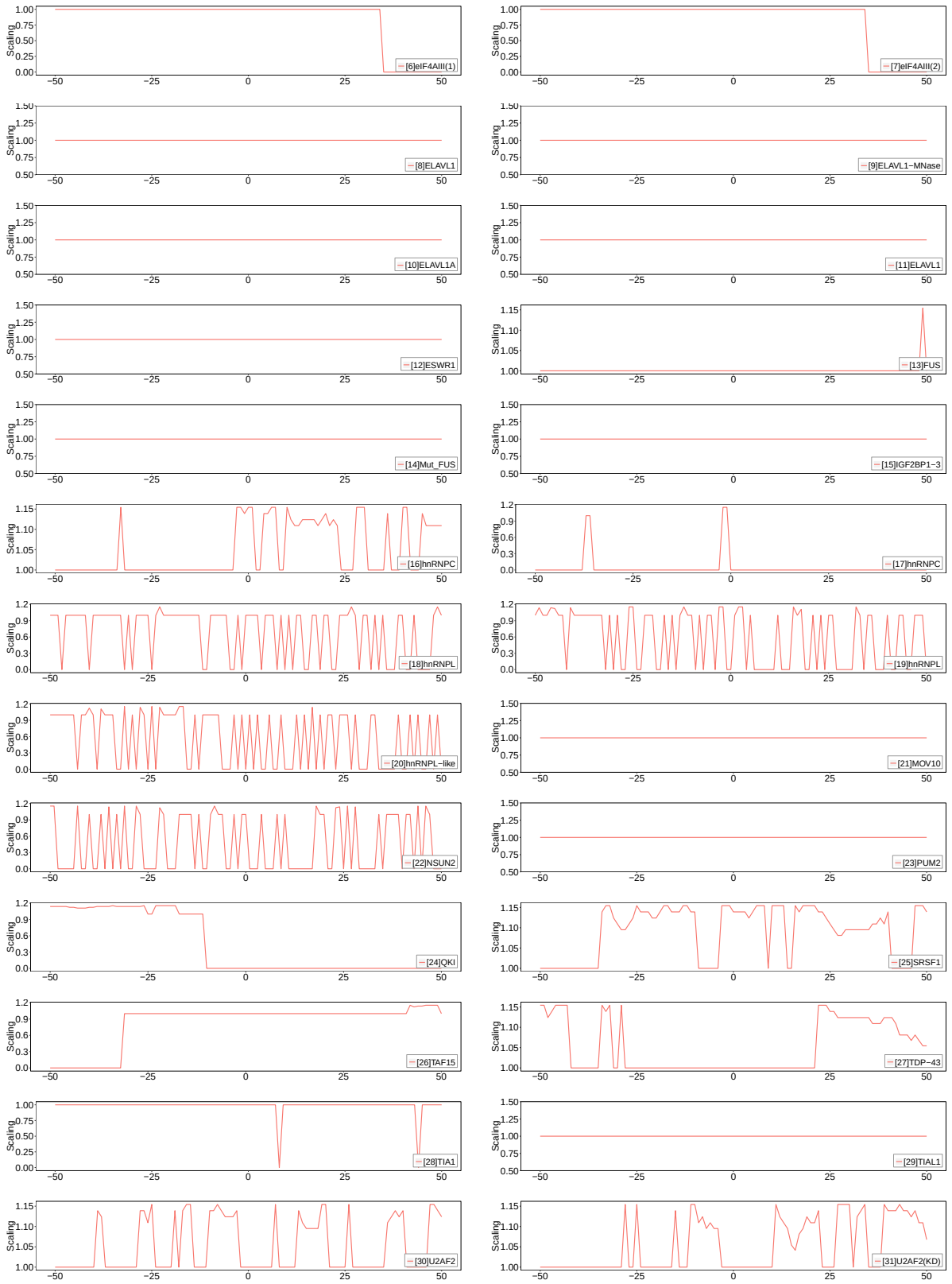

Figure S29: The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [5]Ago2.

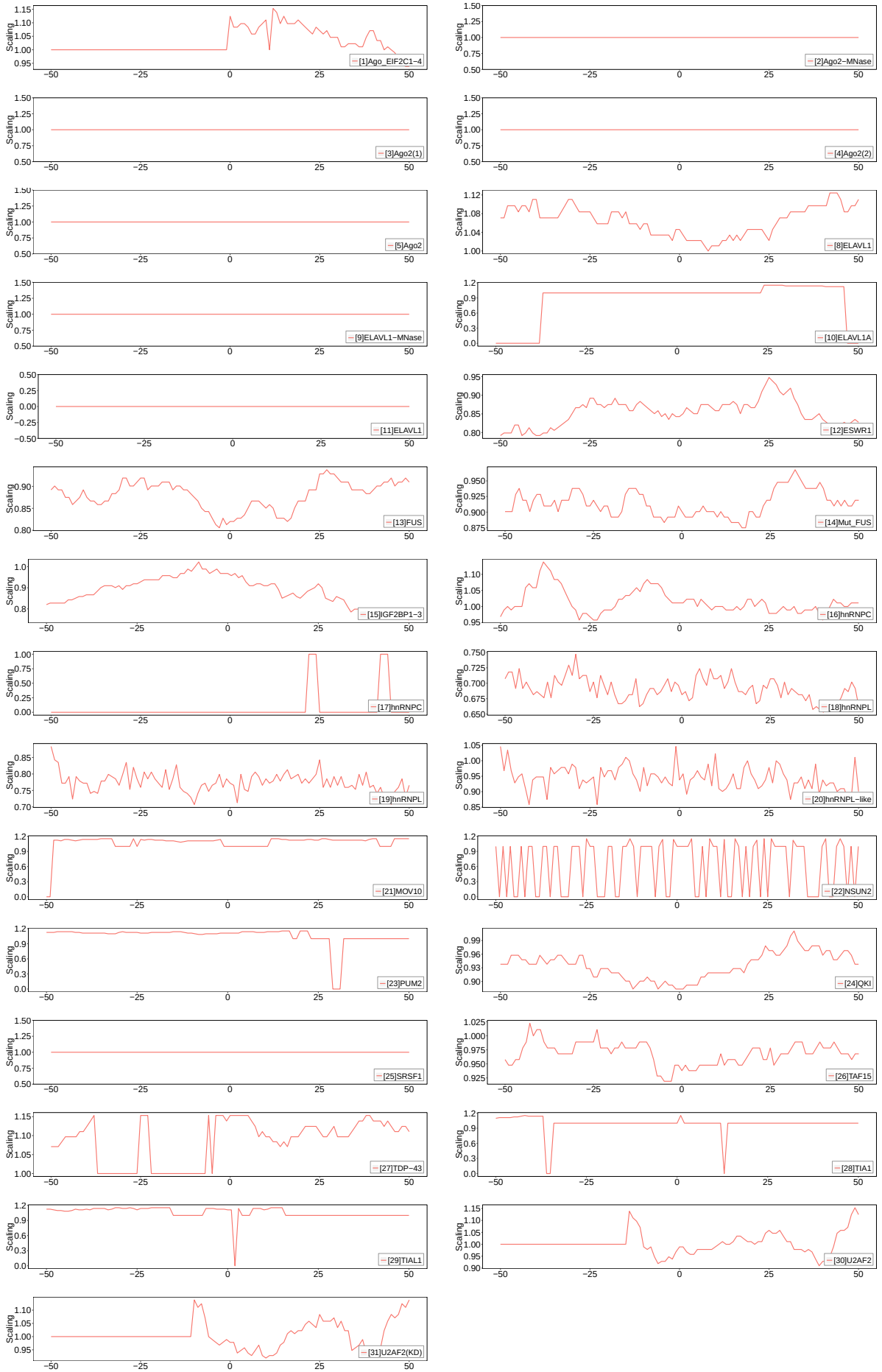

Figure S30: The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [6]eIF4AIII(1).

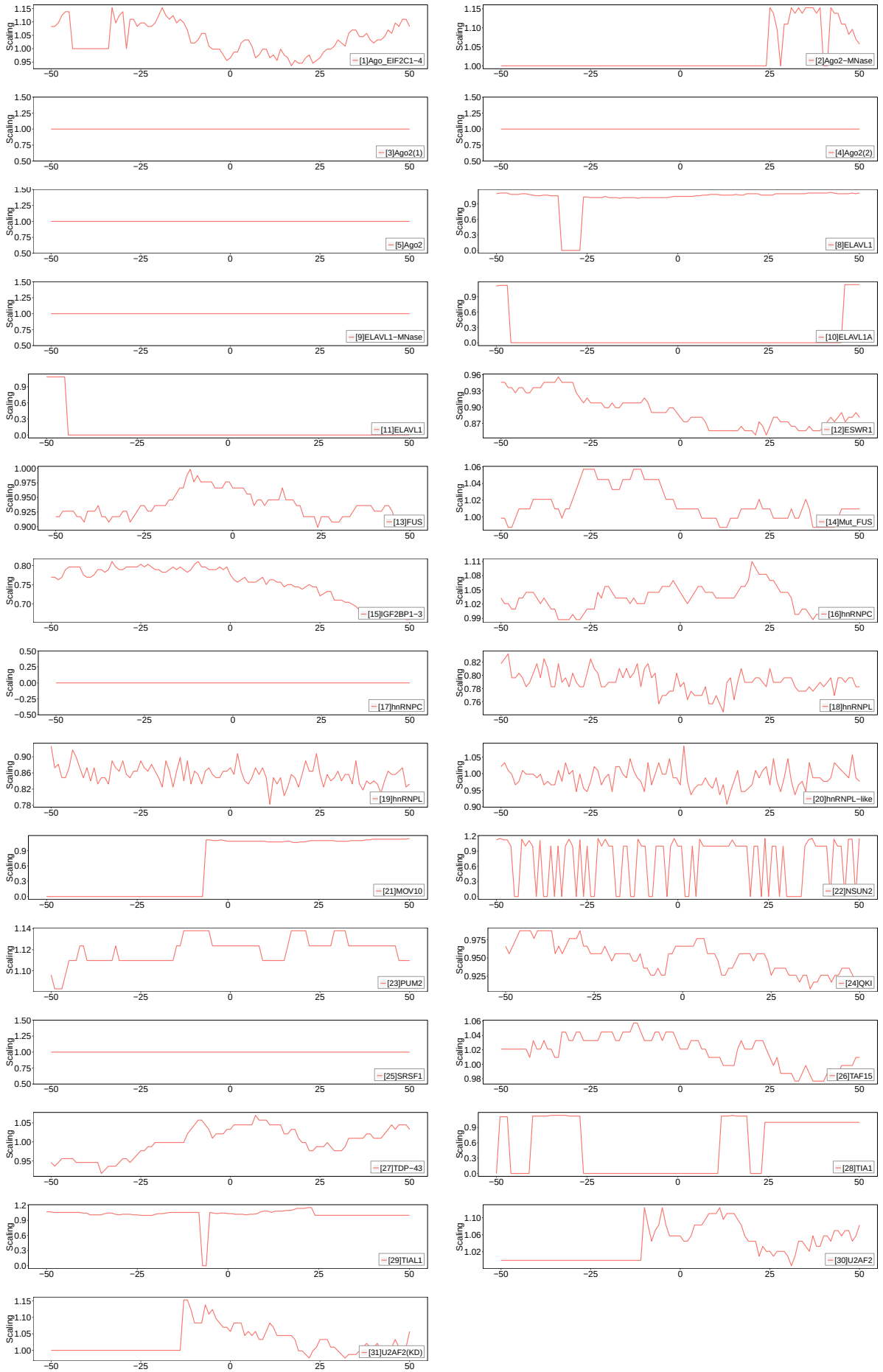

Figure S31: The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [7]eIF4AIII(2).

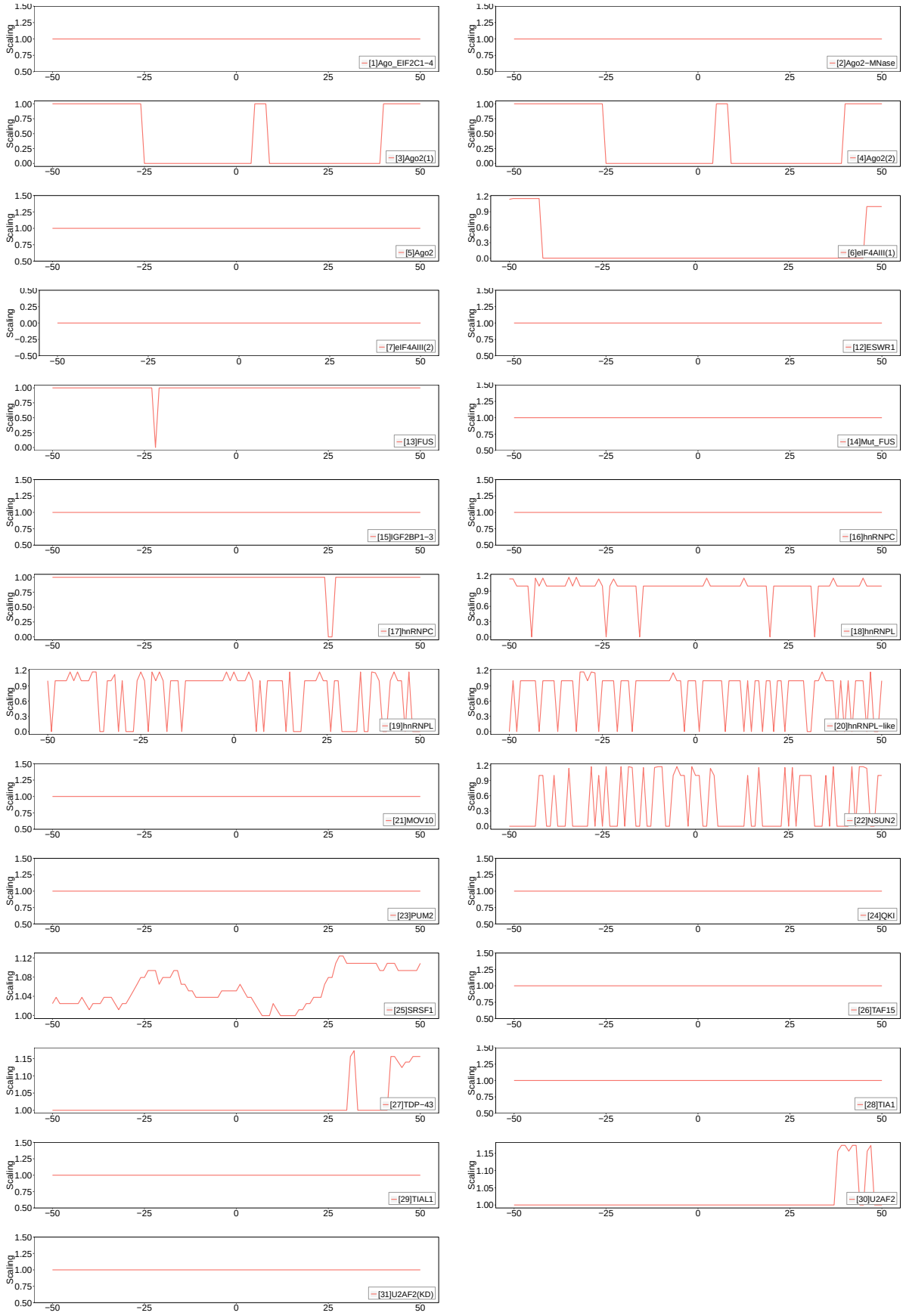

Figure S32: The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [8]ELAVL1.

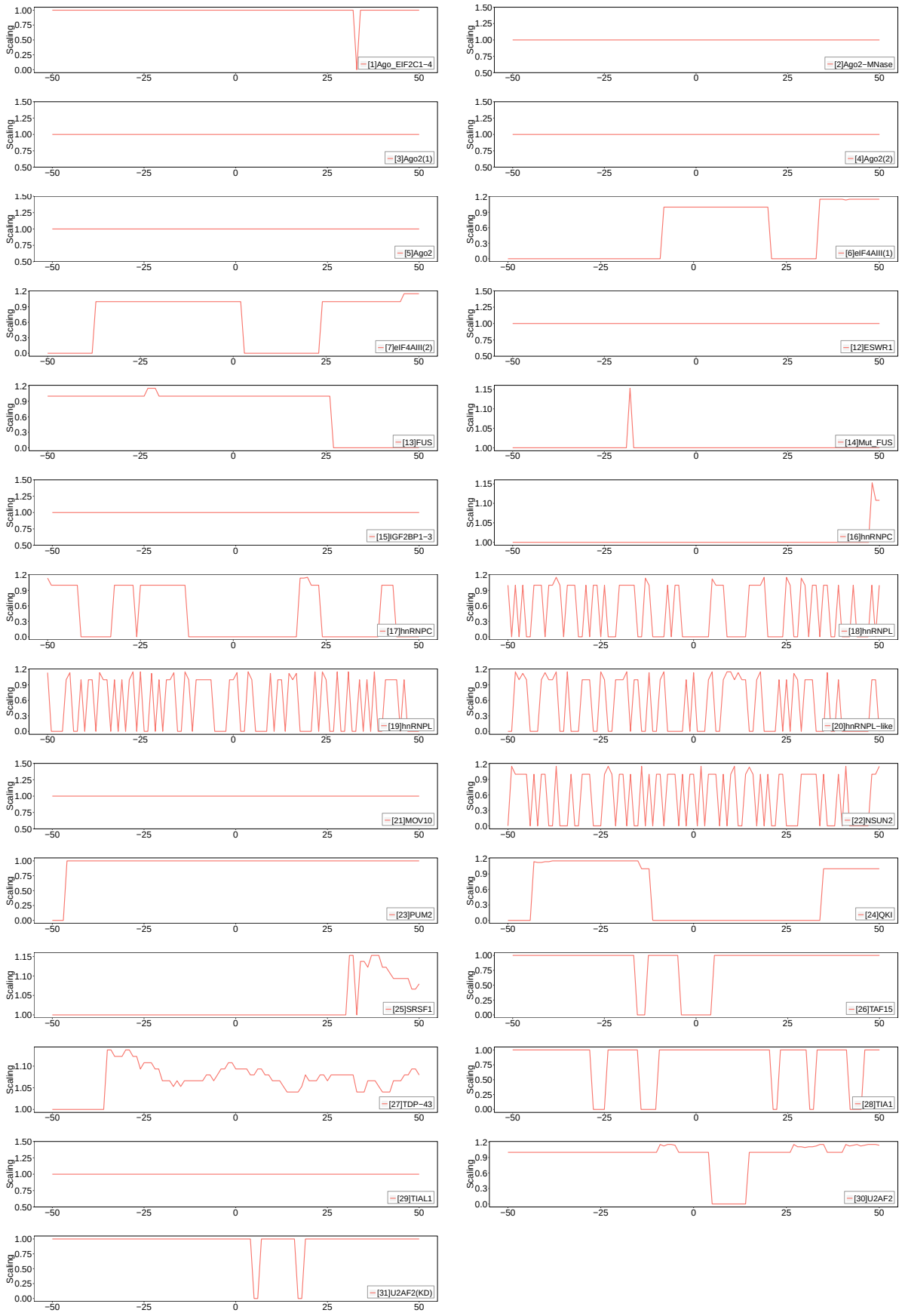

Figure S33: The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [9]ELAVL1-MNase.

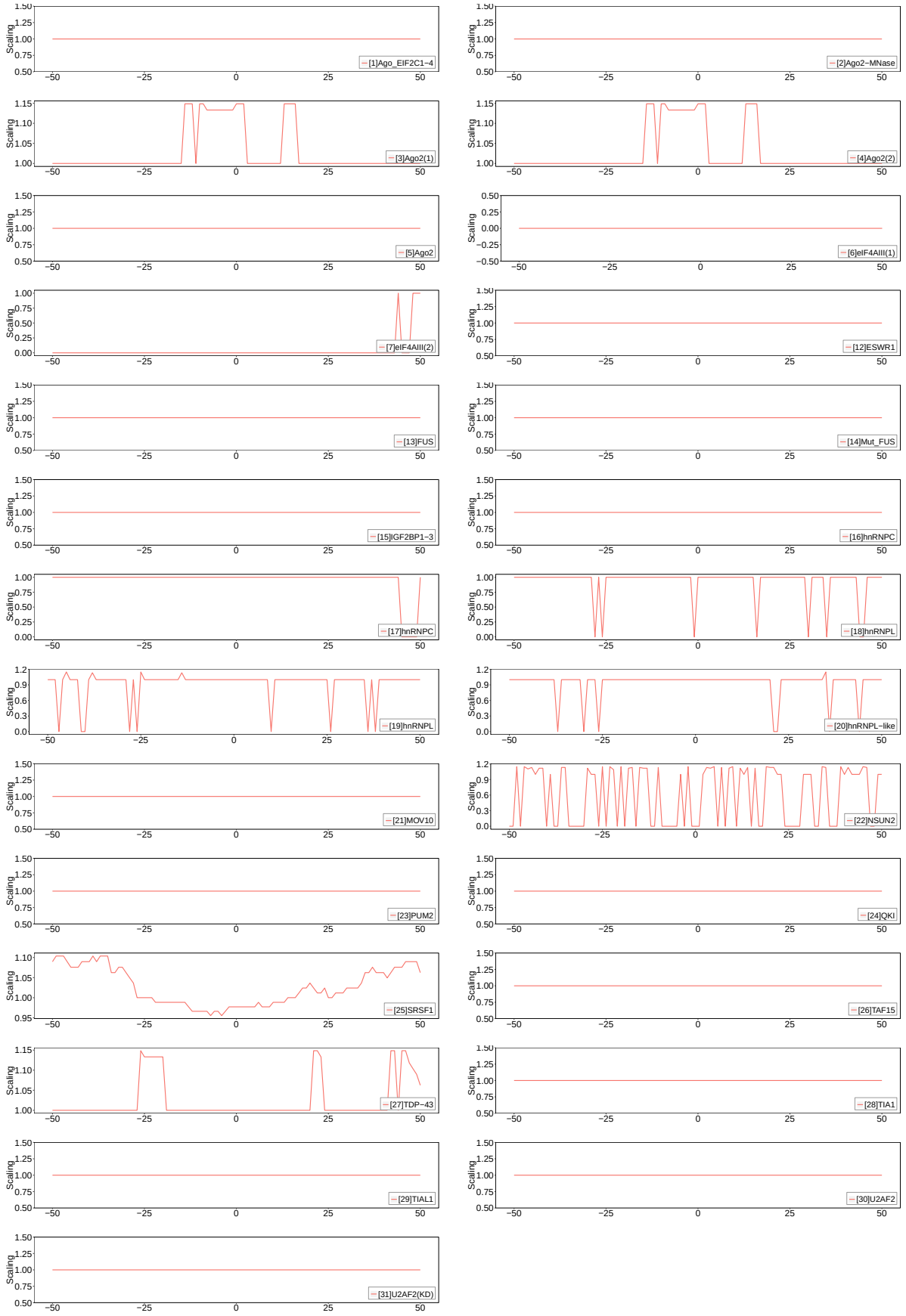

Figure S34: The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [10]ELAVL1A.

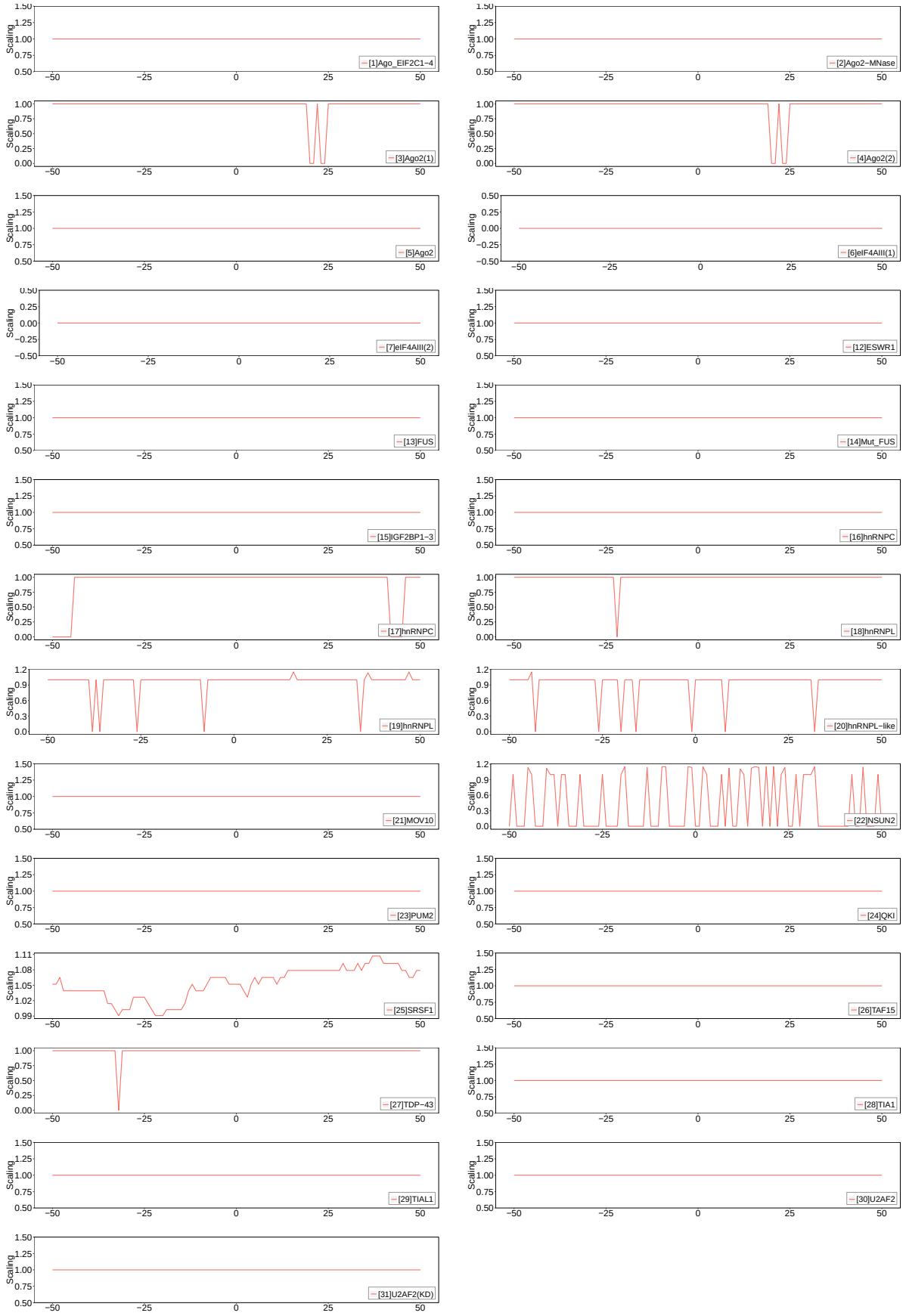

Figure S35: The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [11]ELAVL1.

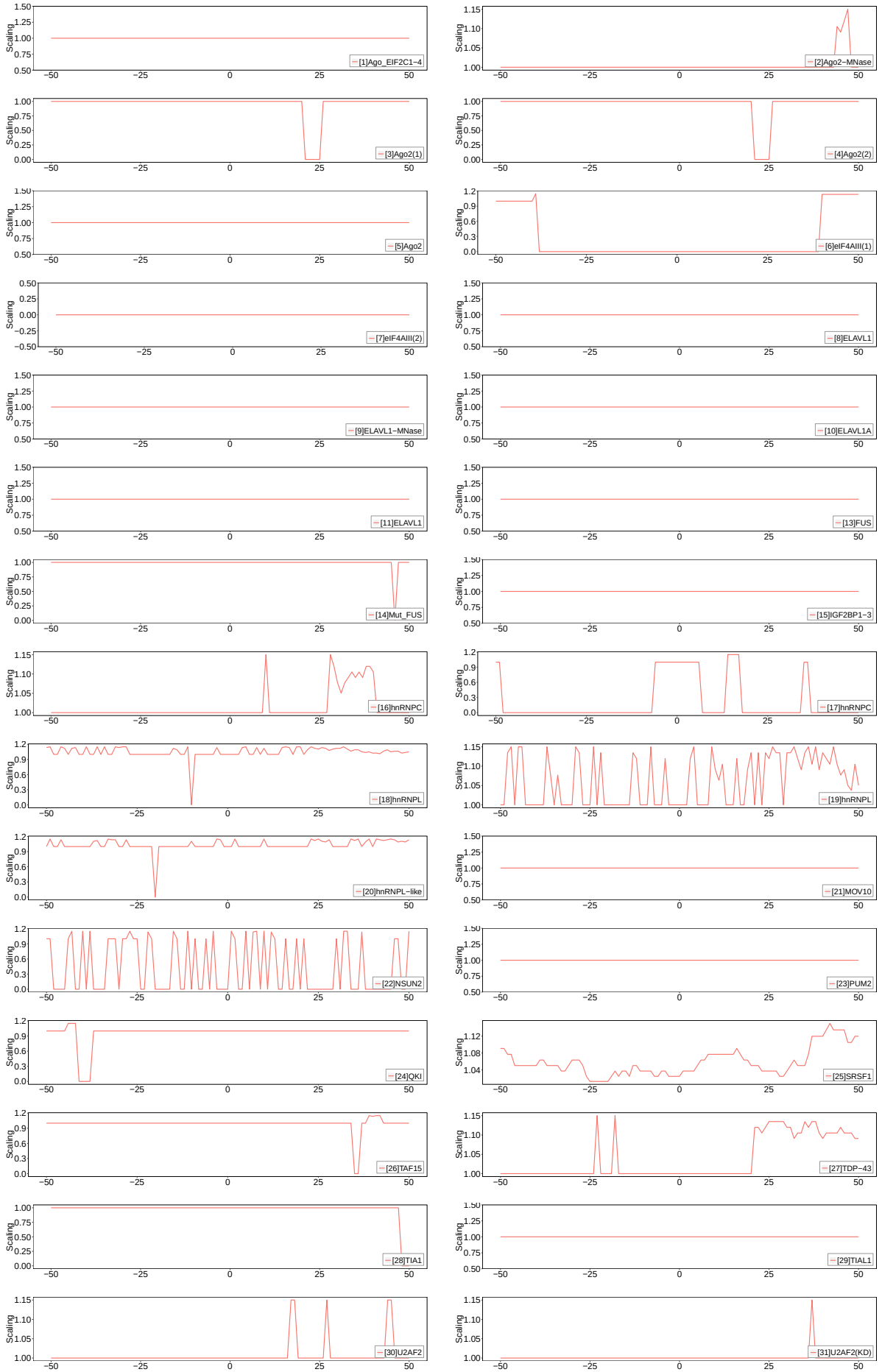

Figure S36: The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [12]ESWR1.

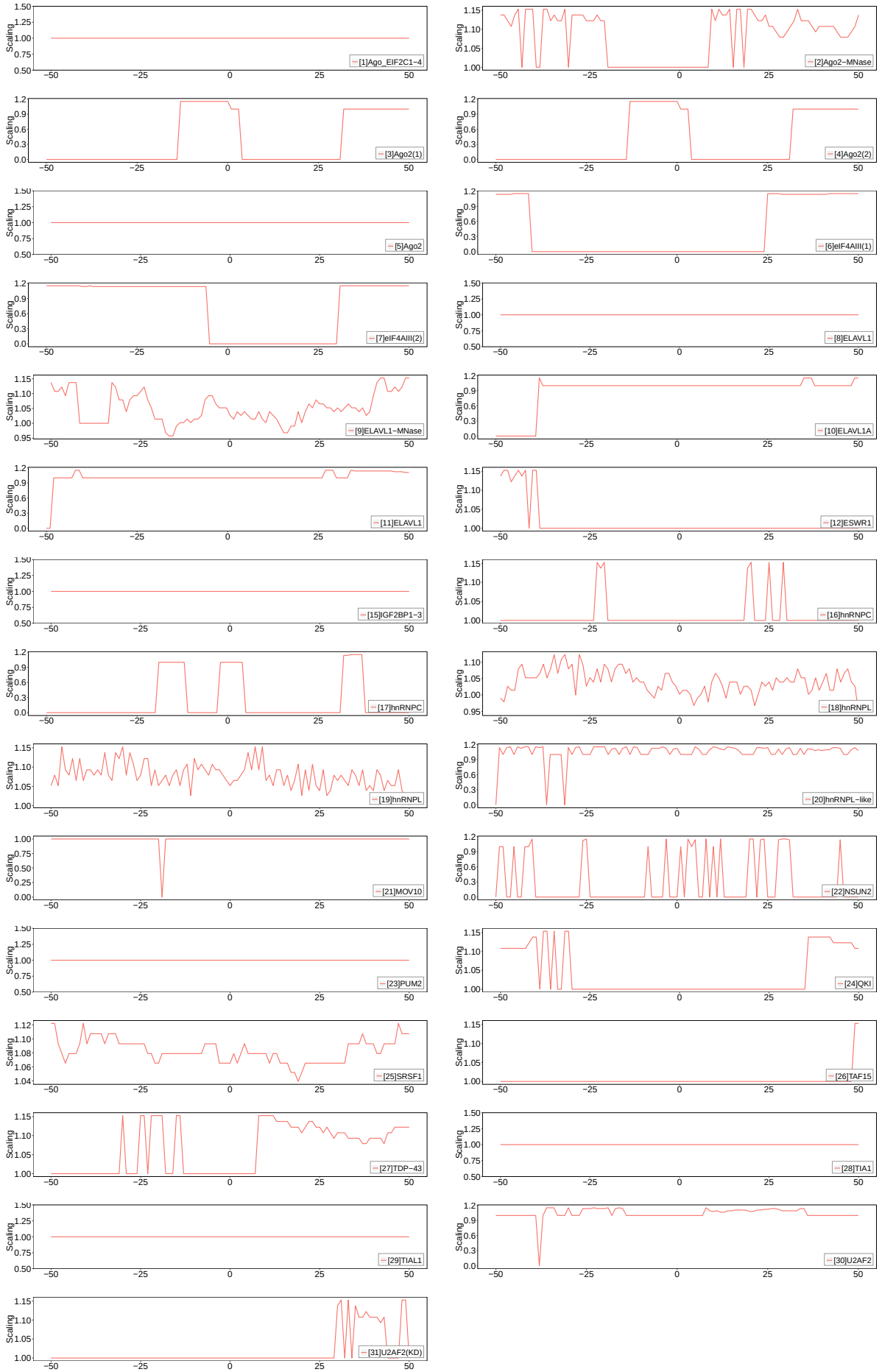

Figure S37: The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [13]FUS.

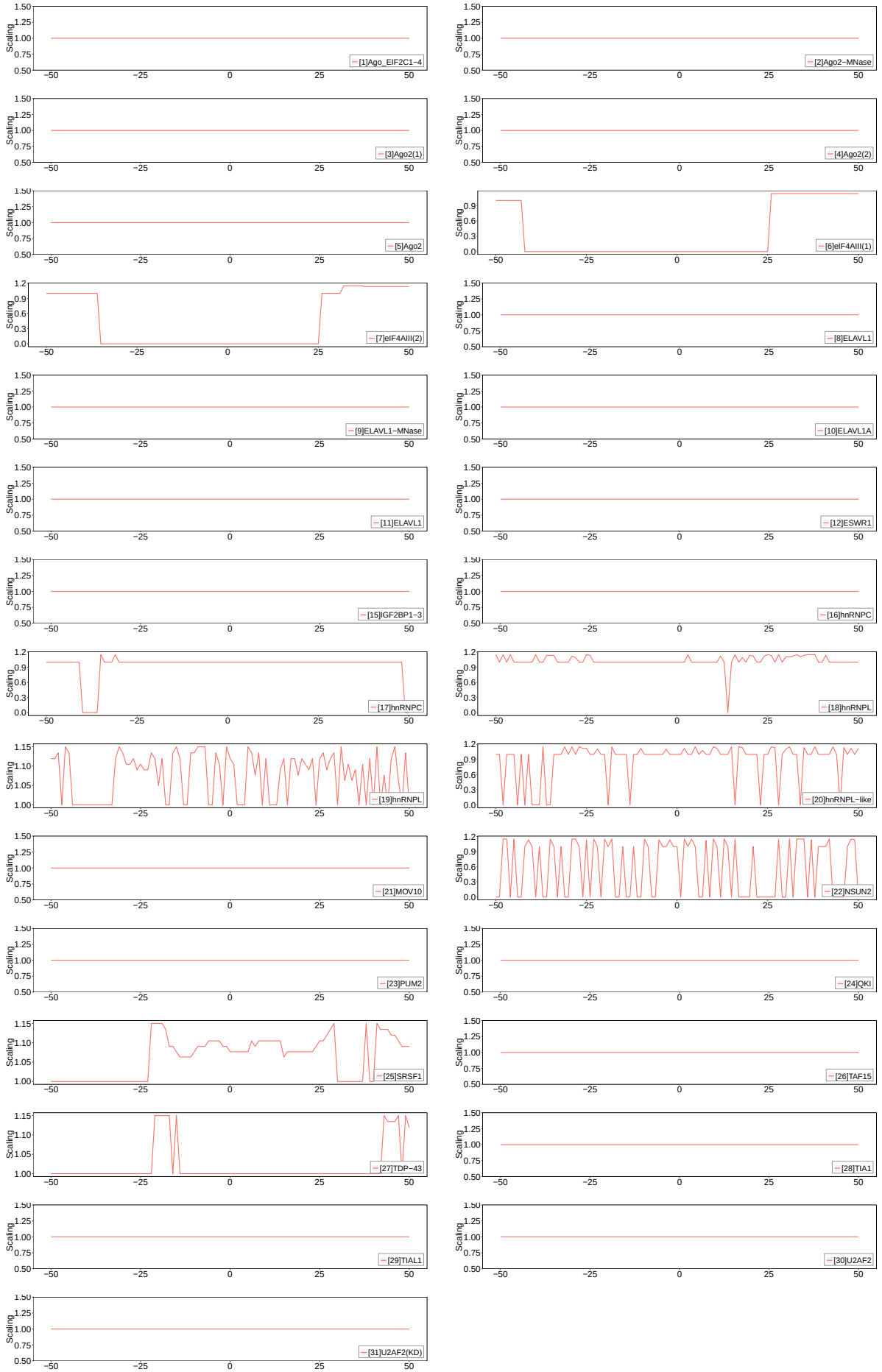

Figure S38: The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [14]Mut_FUS.

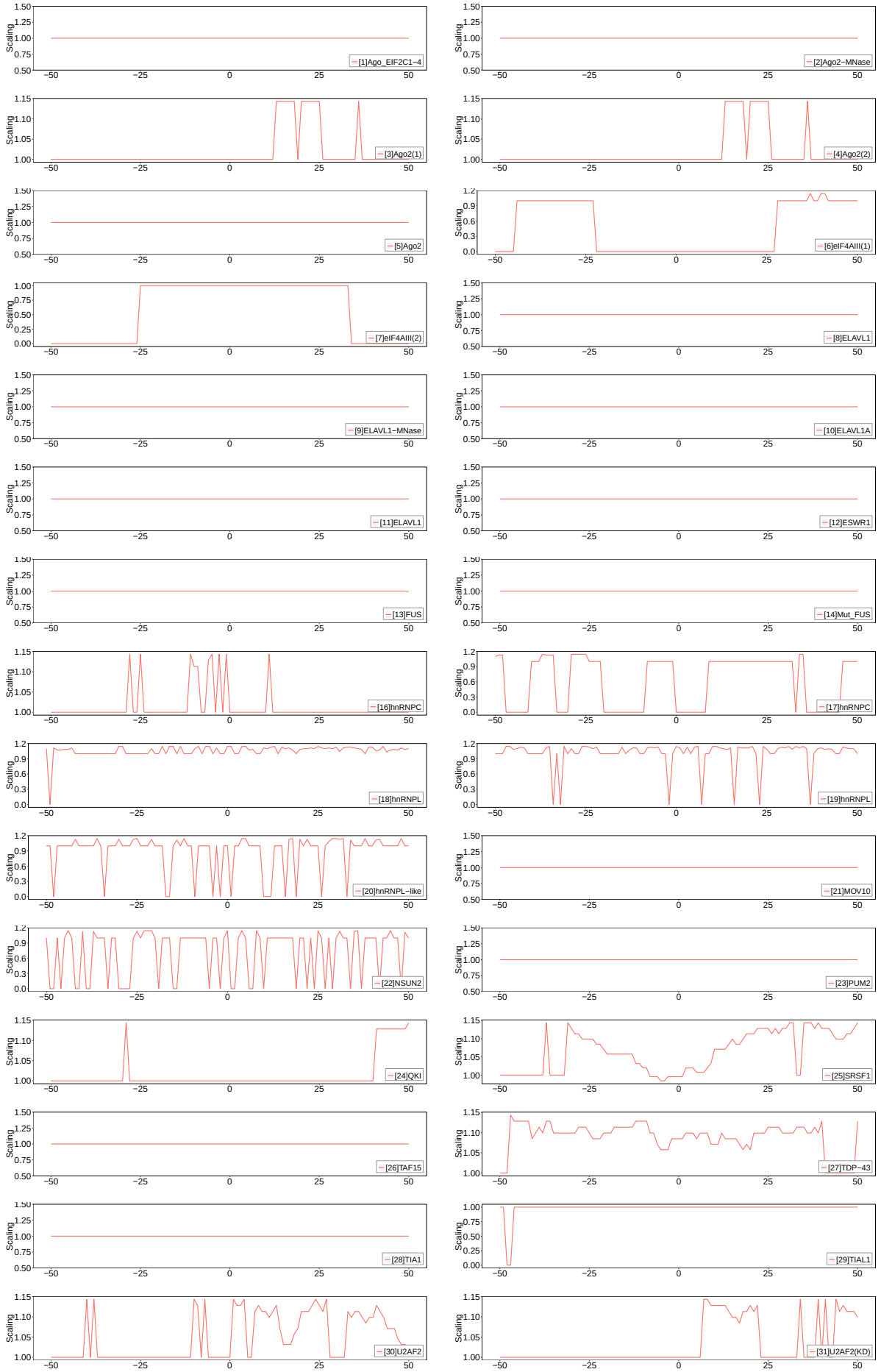

Figure S39: The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [15]IGF2BP1-3.

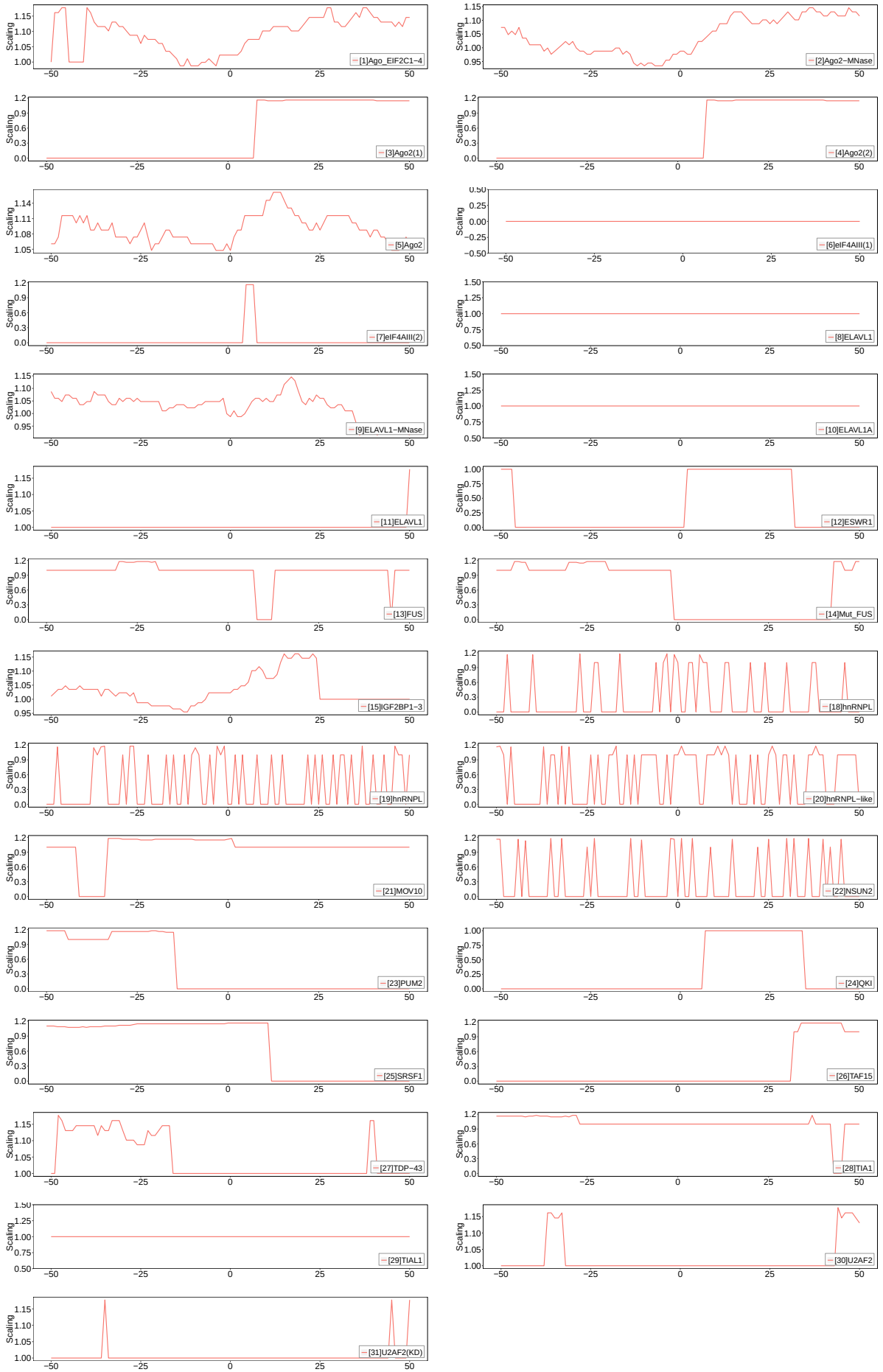

Figure S40: The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [16]hnRNPC.

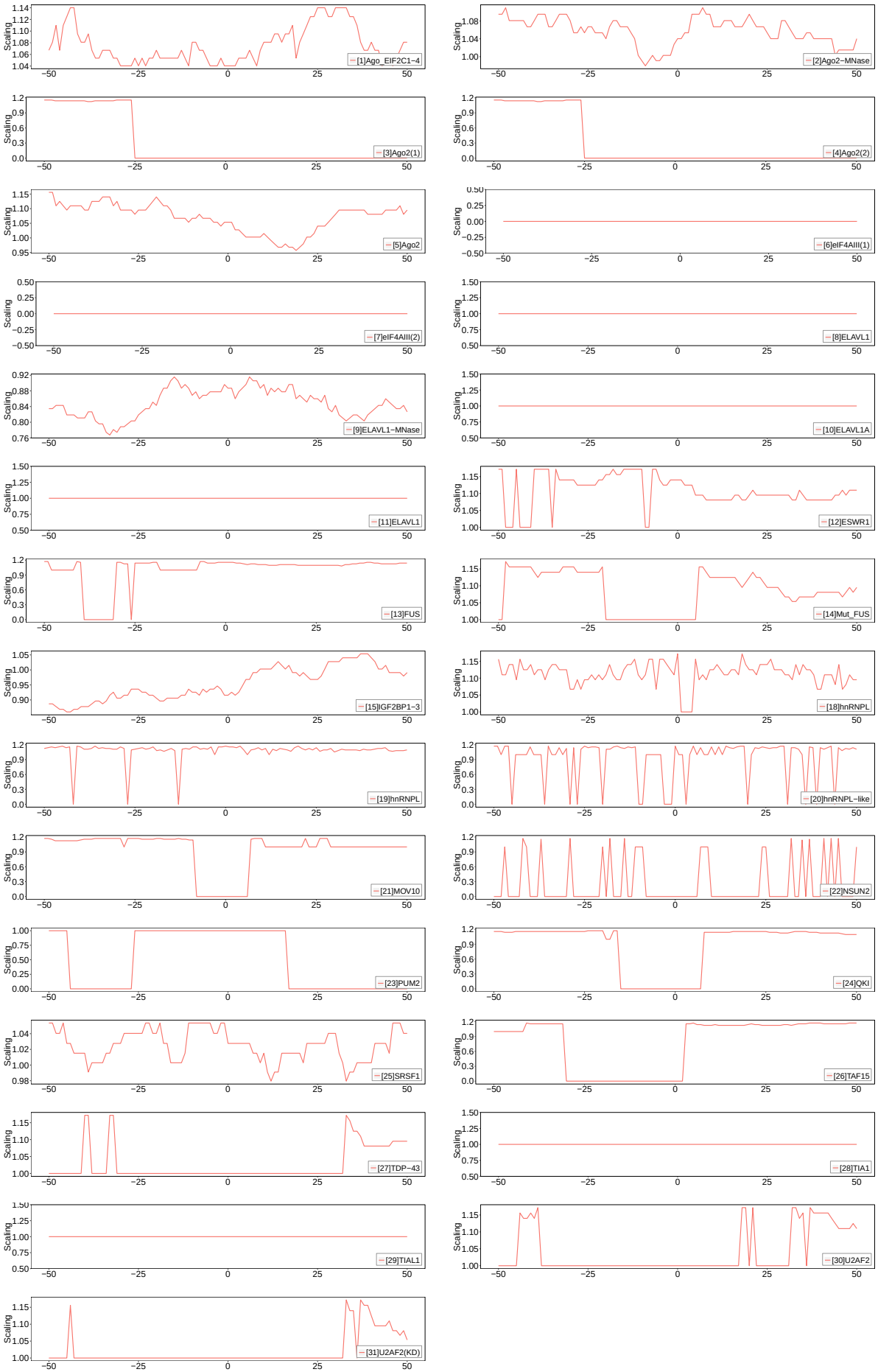

Figure S41: The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [17]hnRNPC.

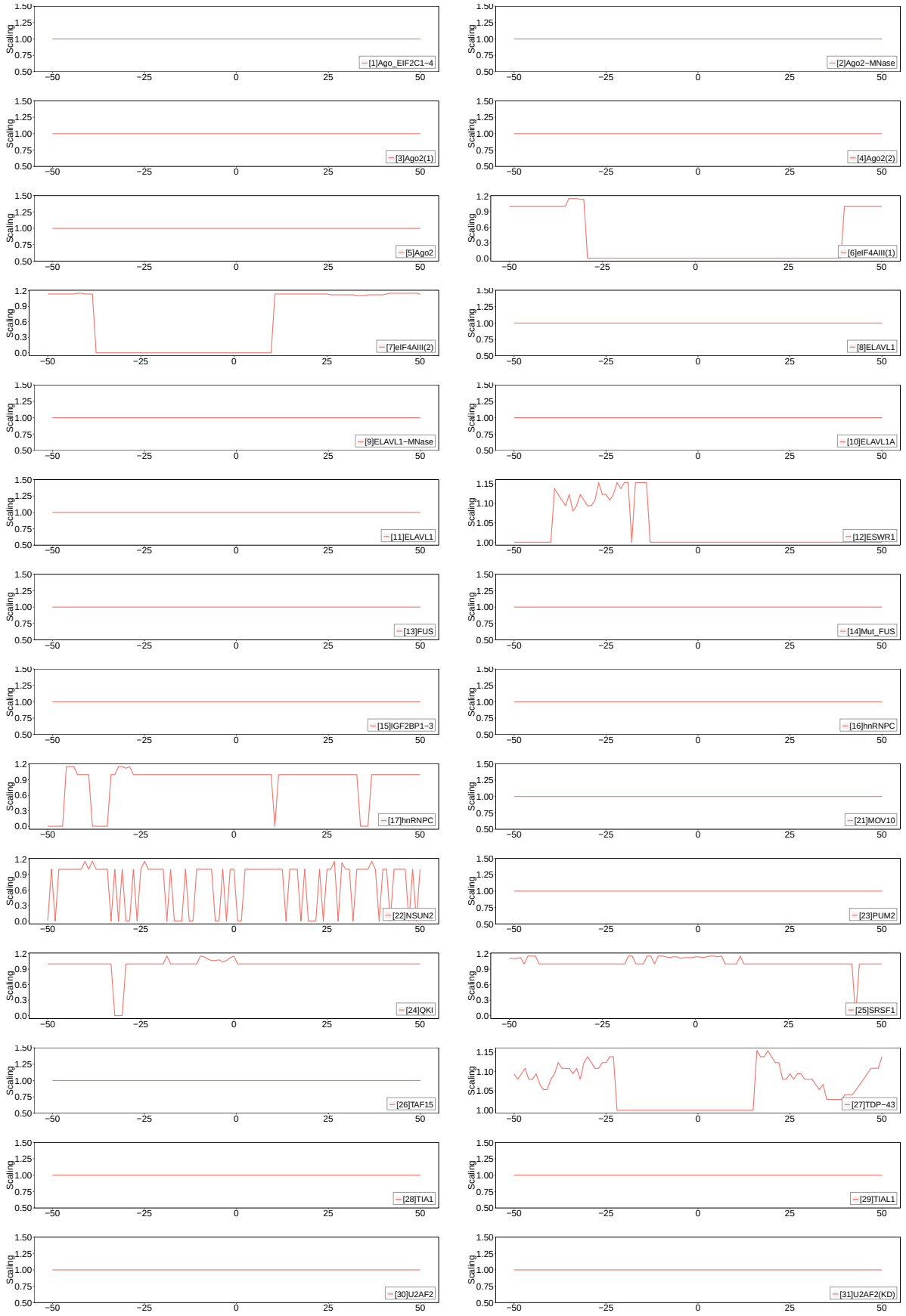

Figure S42: The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [18]hnRNPL.

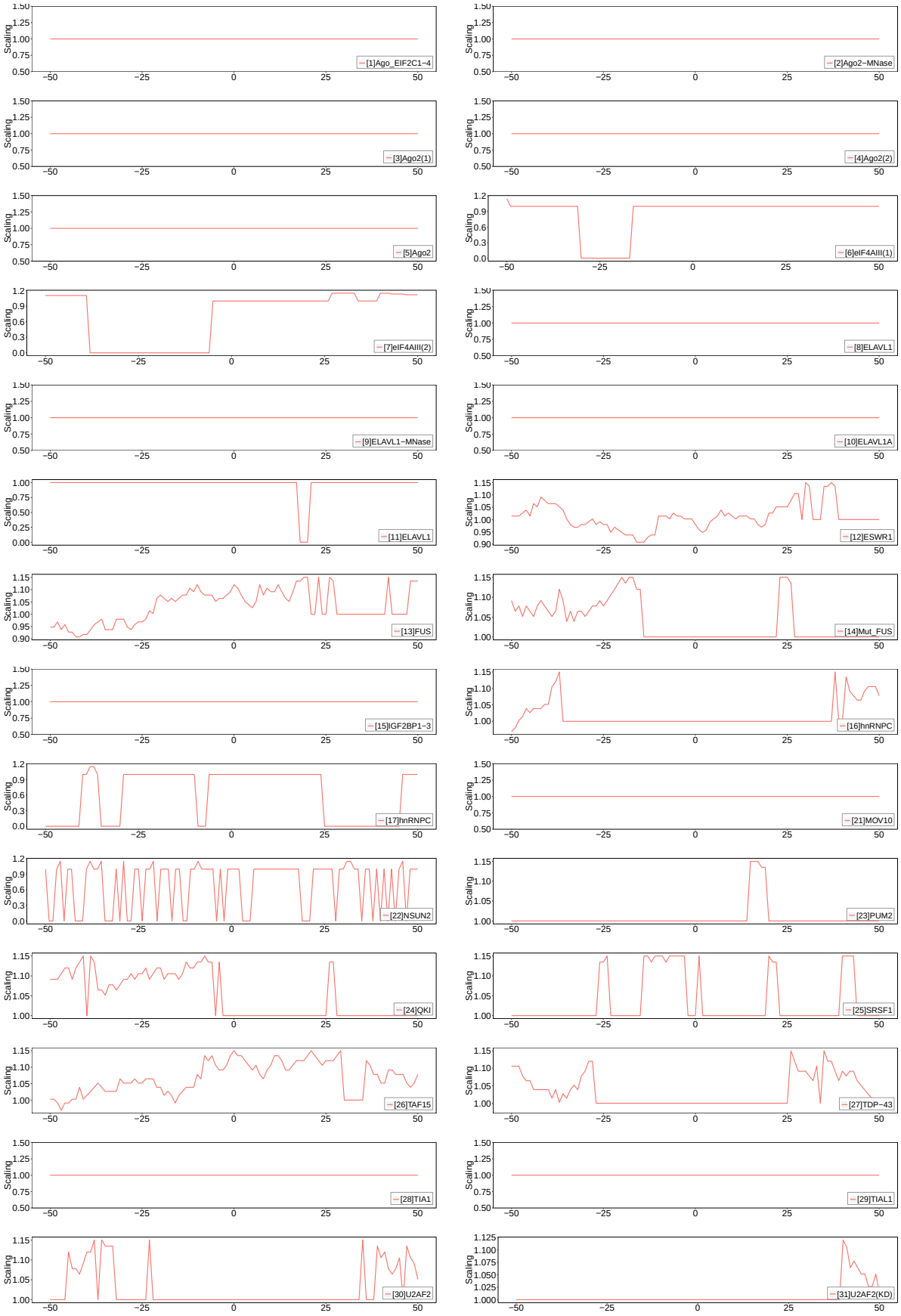

Figure S43: The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [19]hnRNPL.

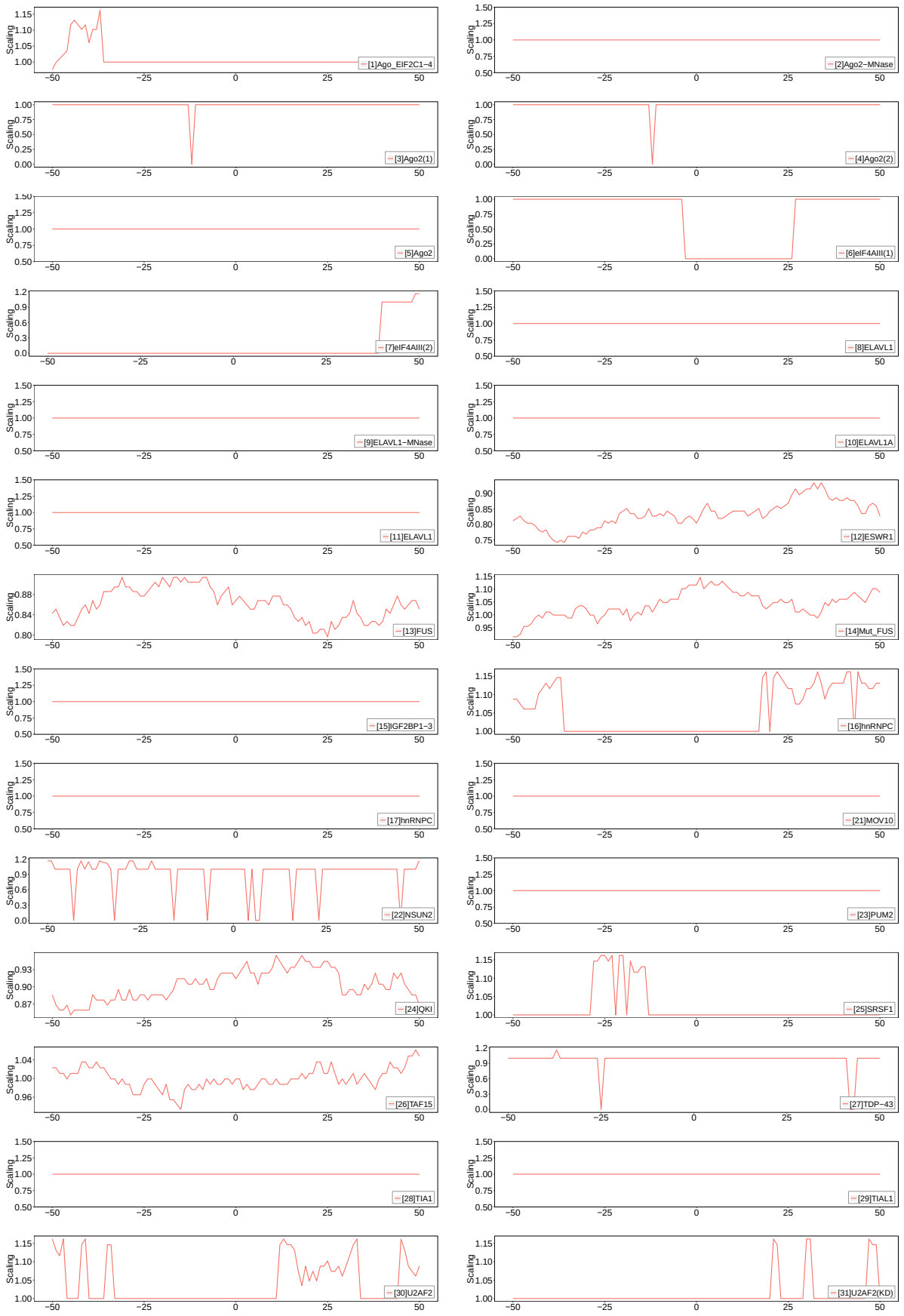

Figure S44: The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [20]hnRNPL-like.

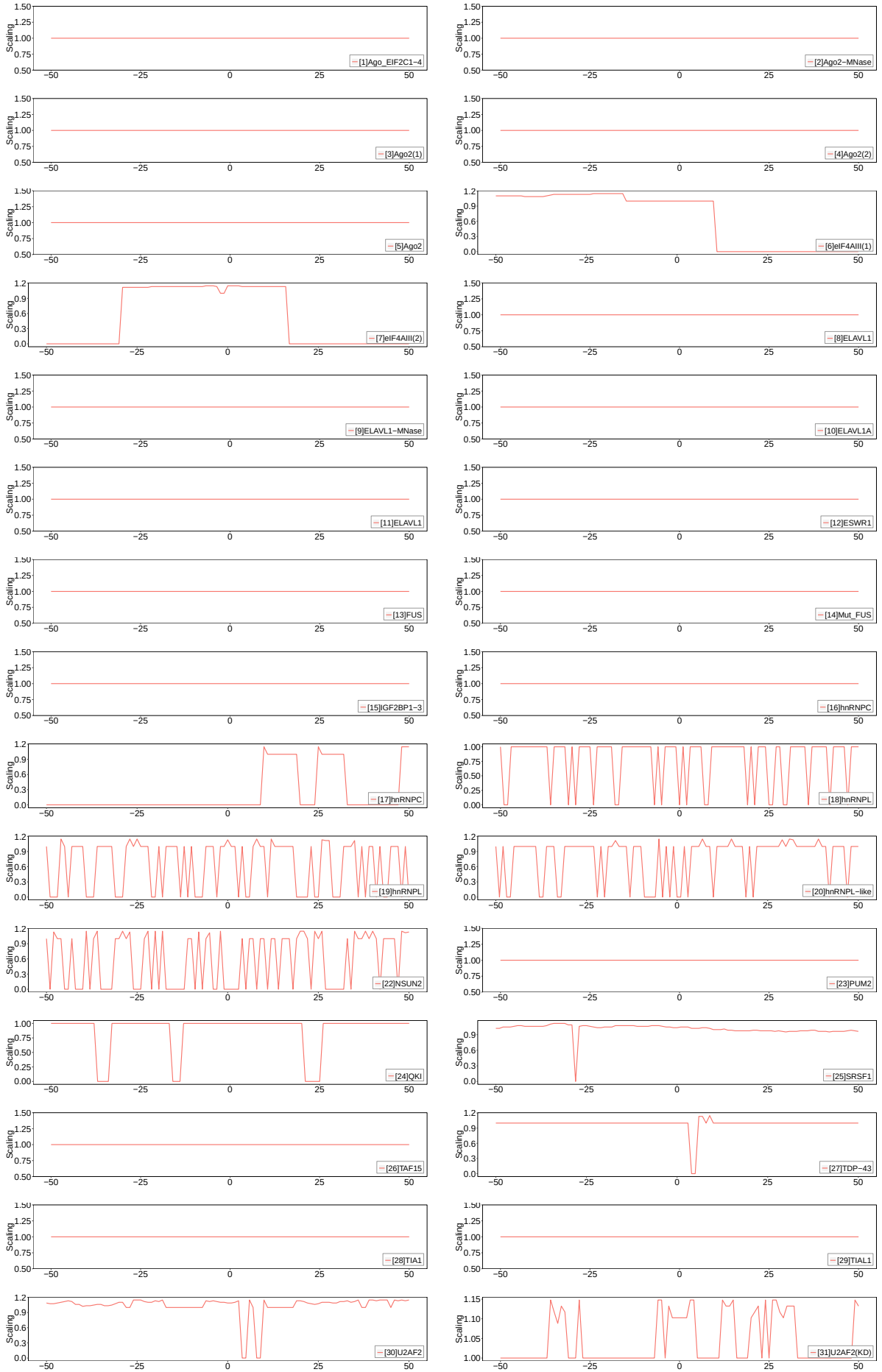

Figure S45: The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [21]MOV10.

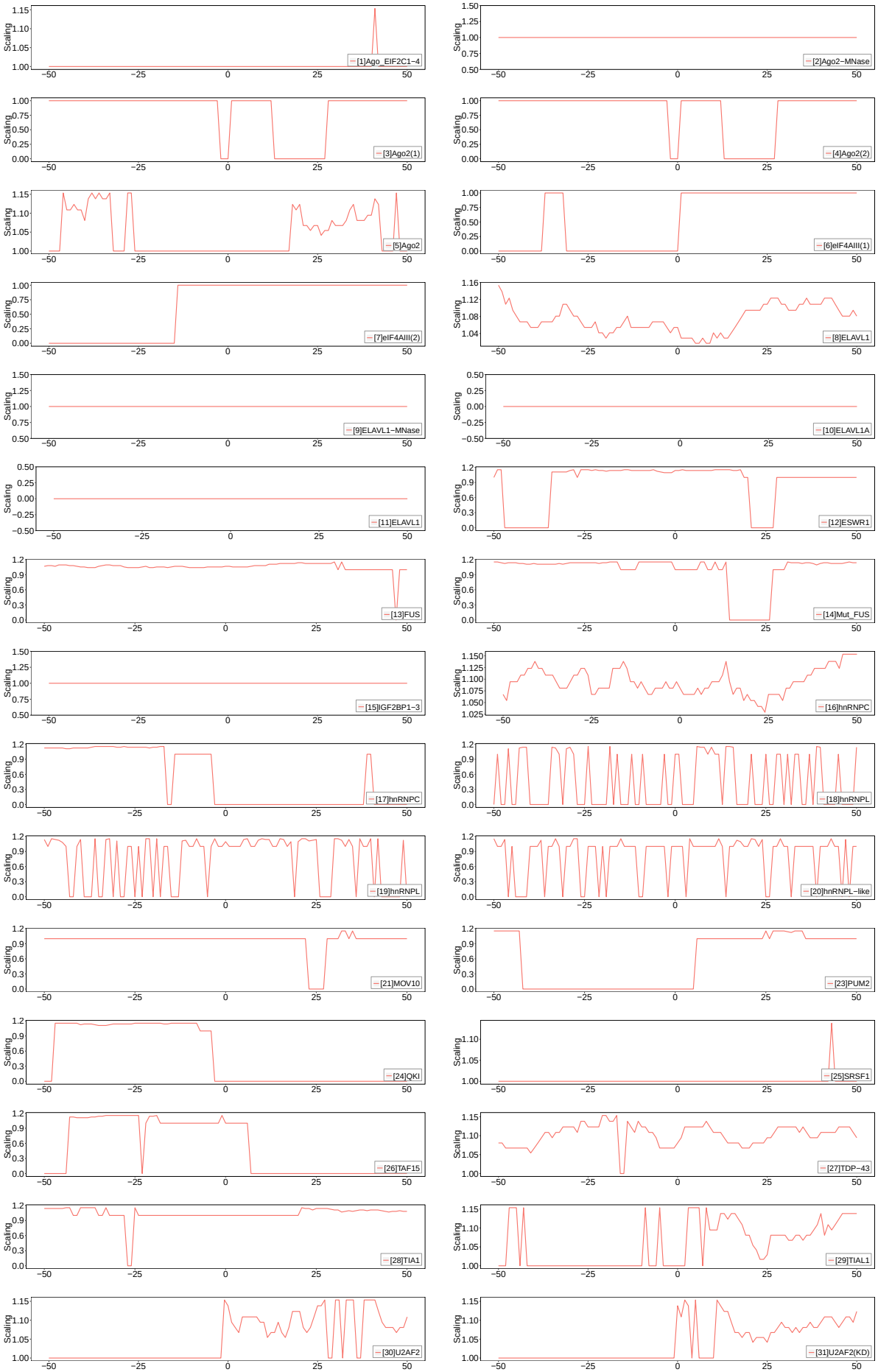

Figure S46: The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [22]NSUN2.

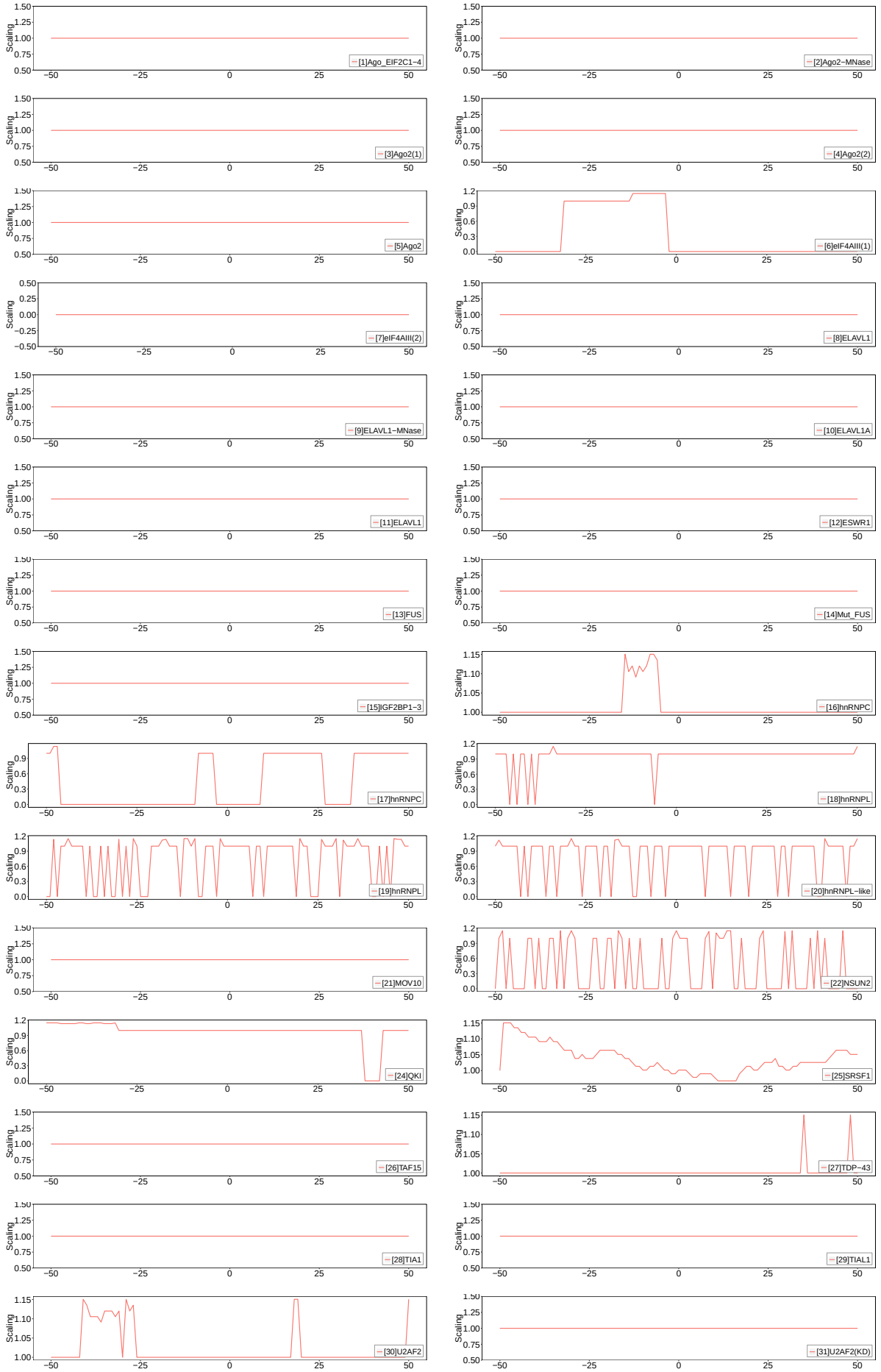

Figure S47: The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [23]PUM2.

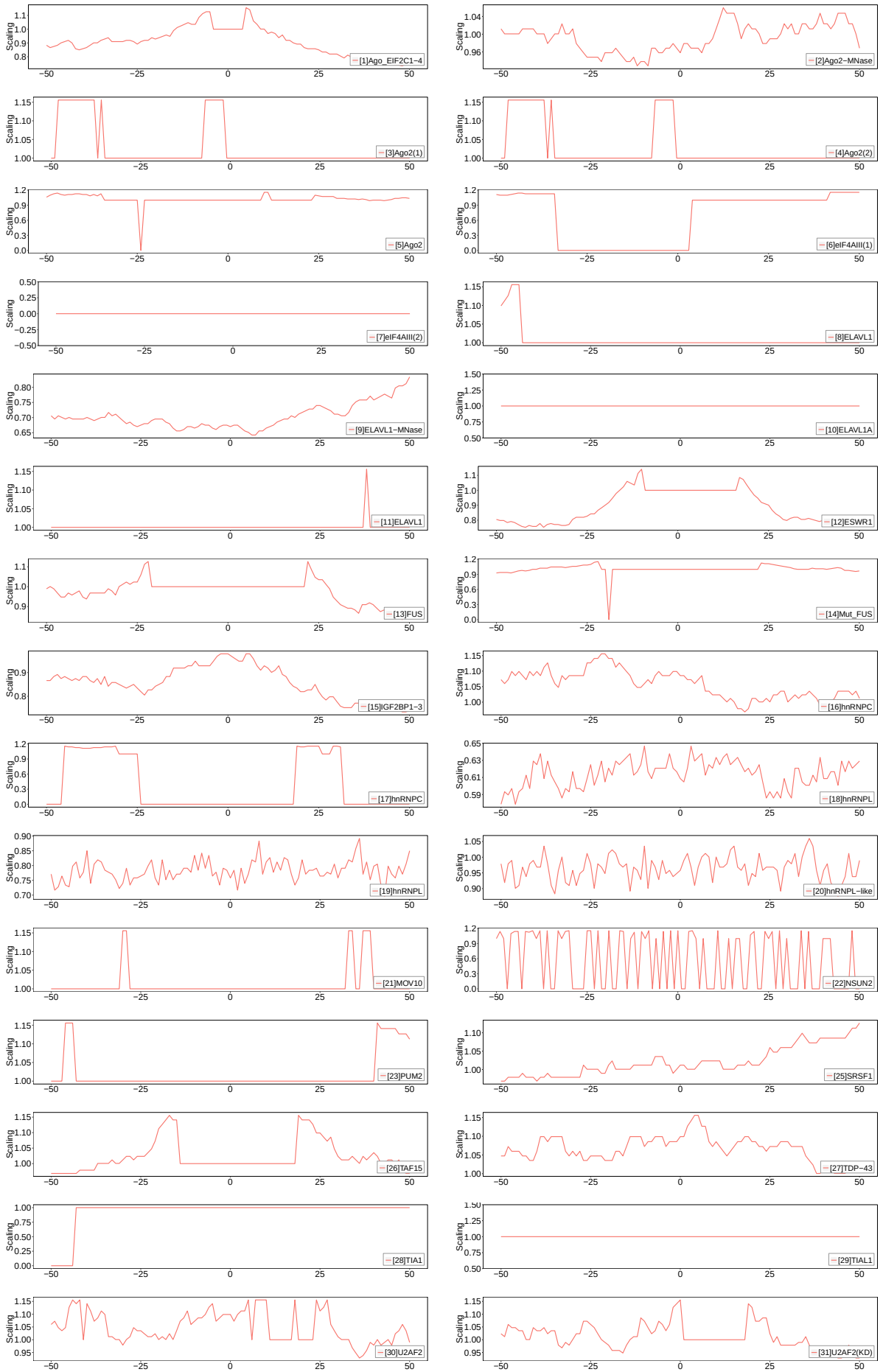

Figure S48: The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [24]QKI.

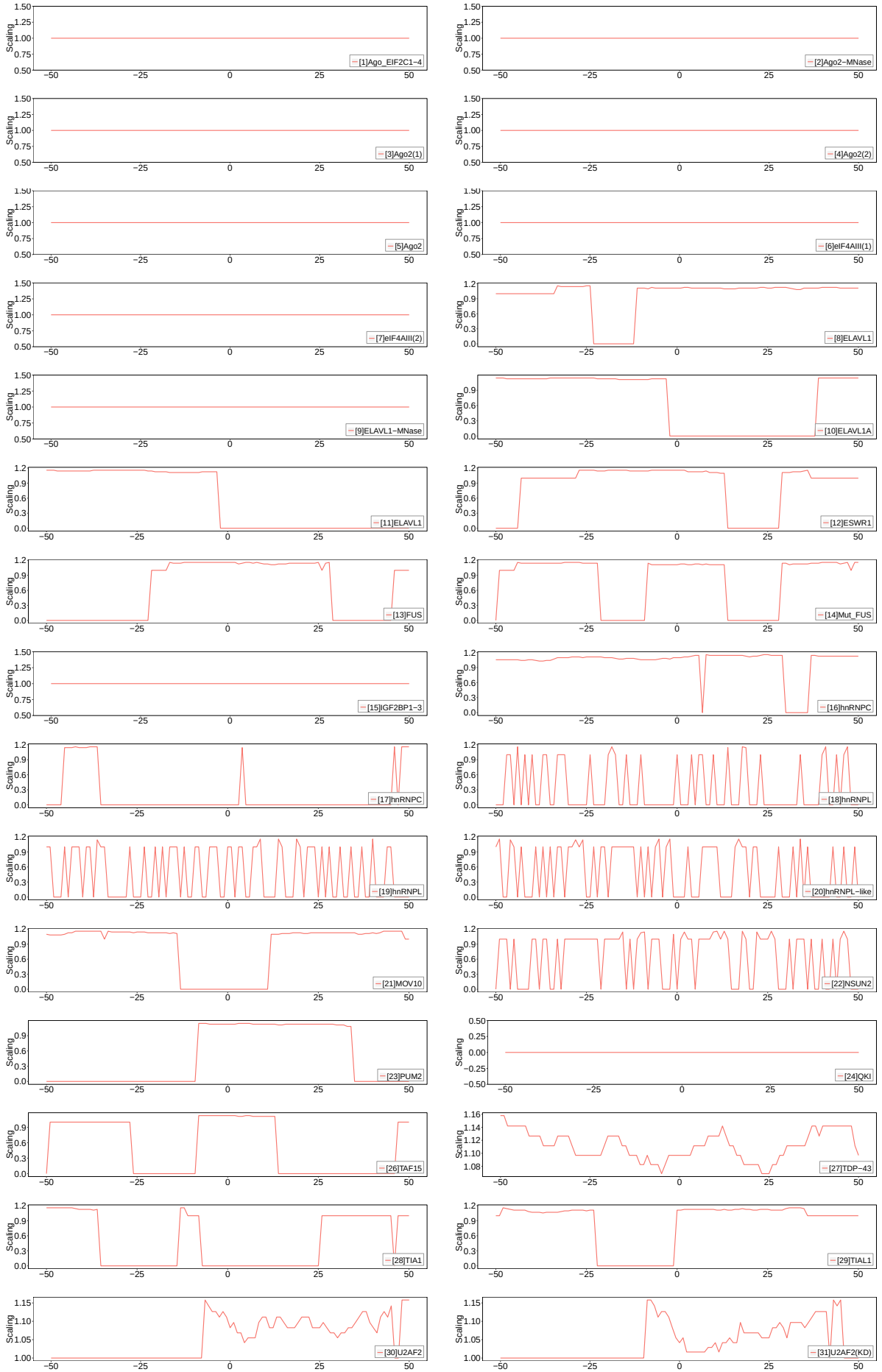

Figure S49: The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [25]SRSF1.

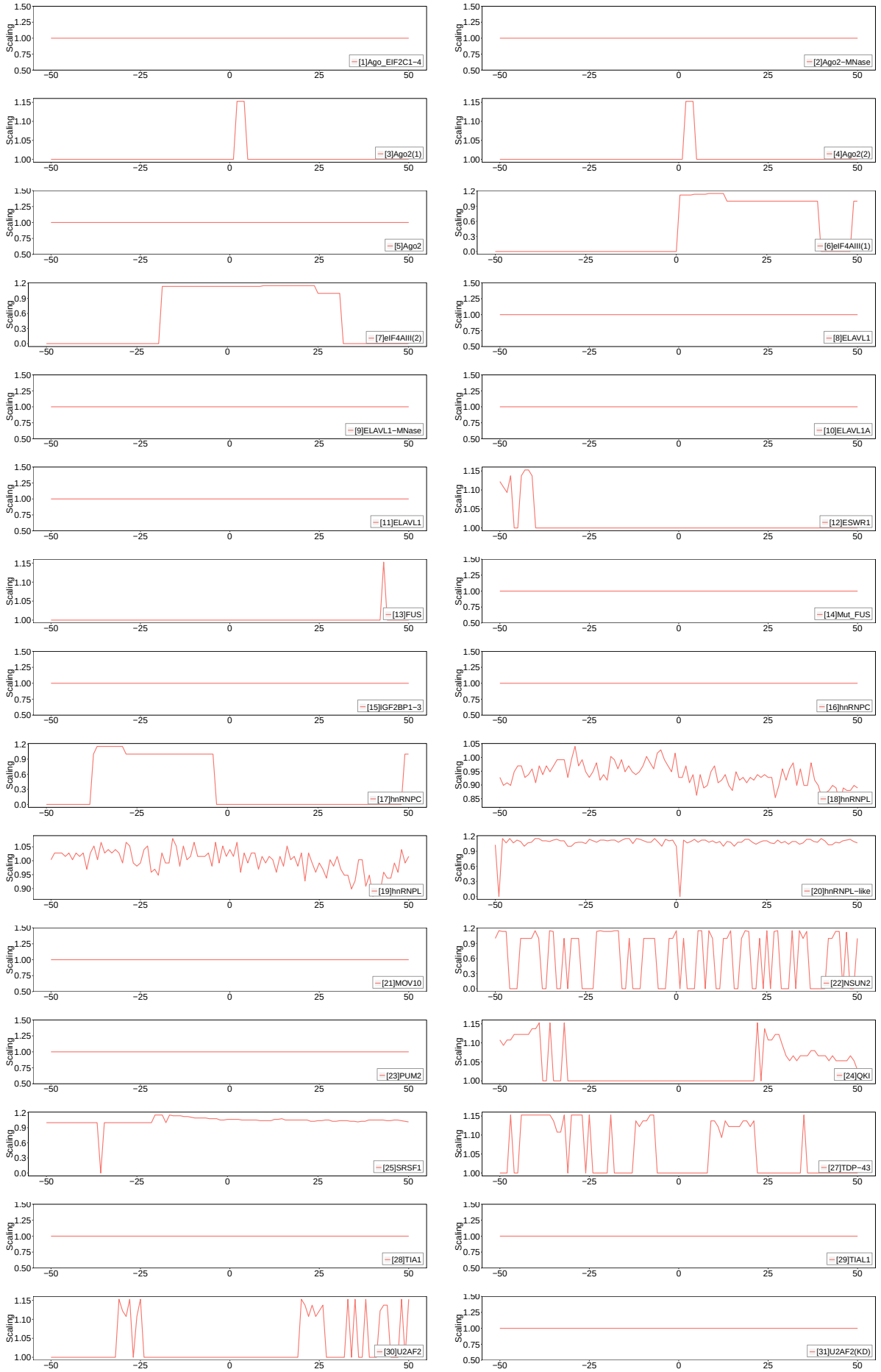

Figure S50: The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [26]TAF15.

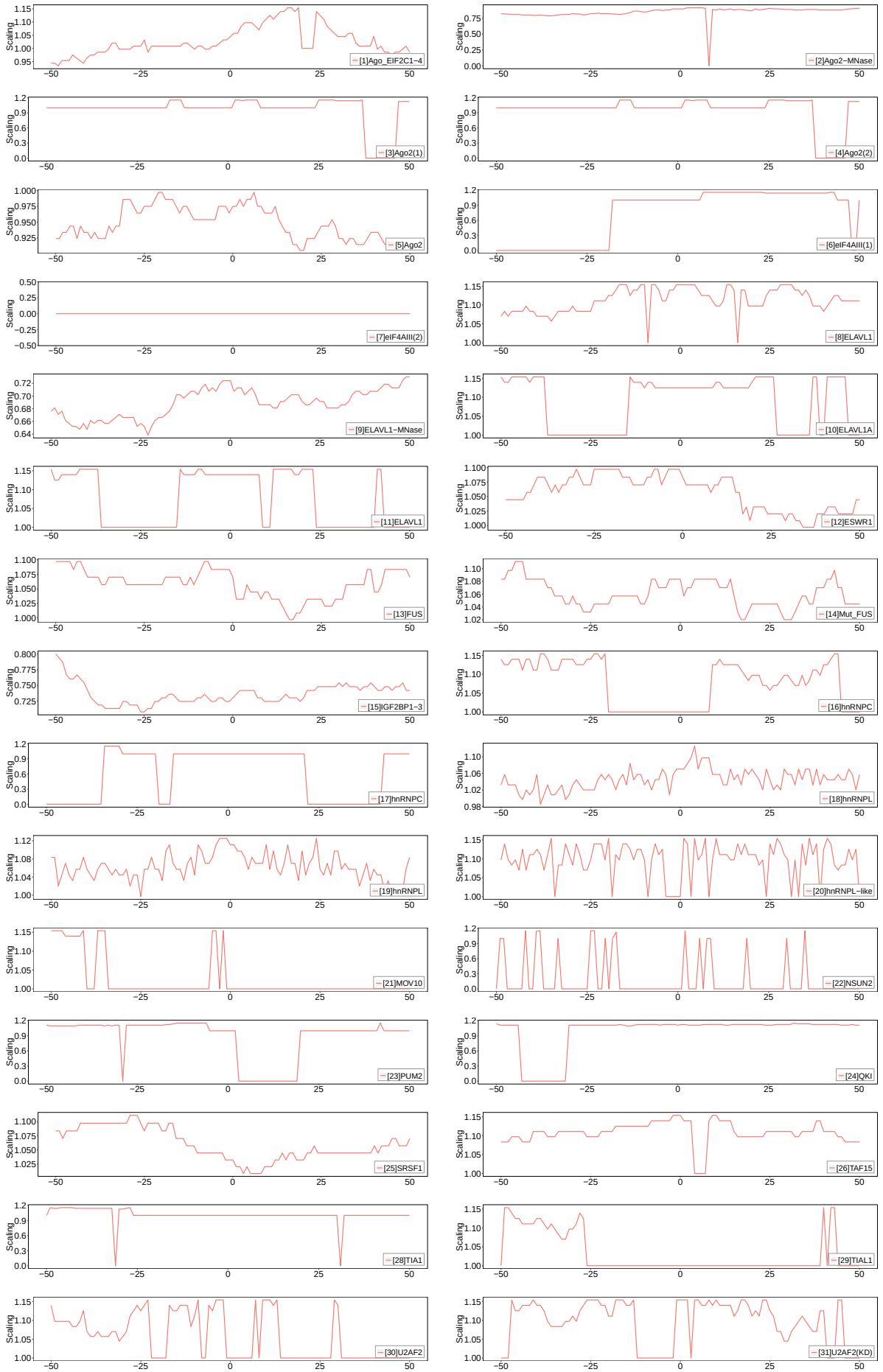

Figure S51: The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [27]TDP-43.

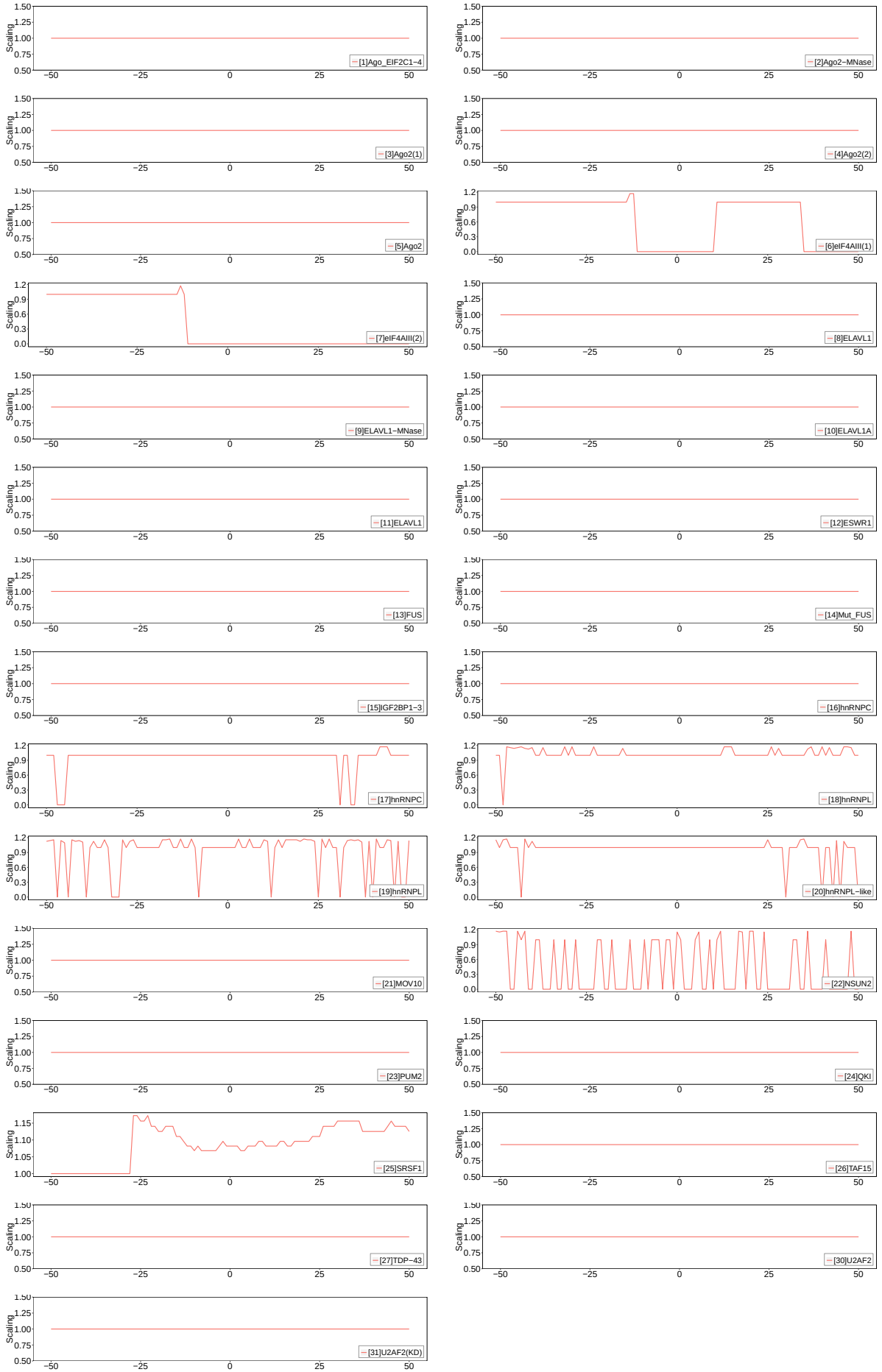

Figure S52: The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [28]TIA1.

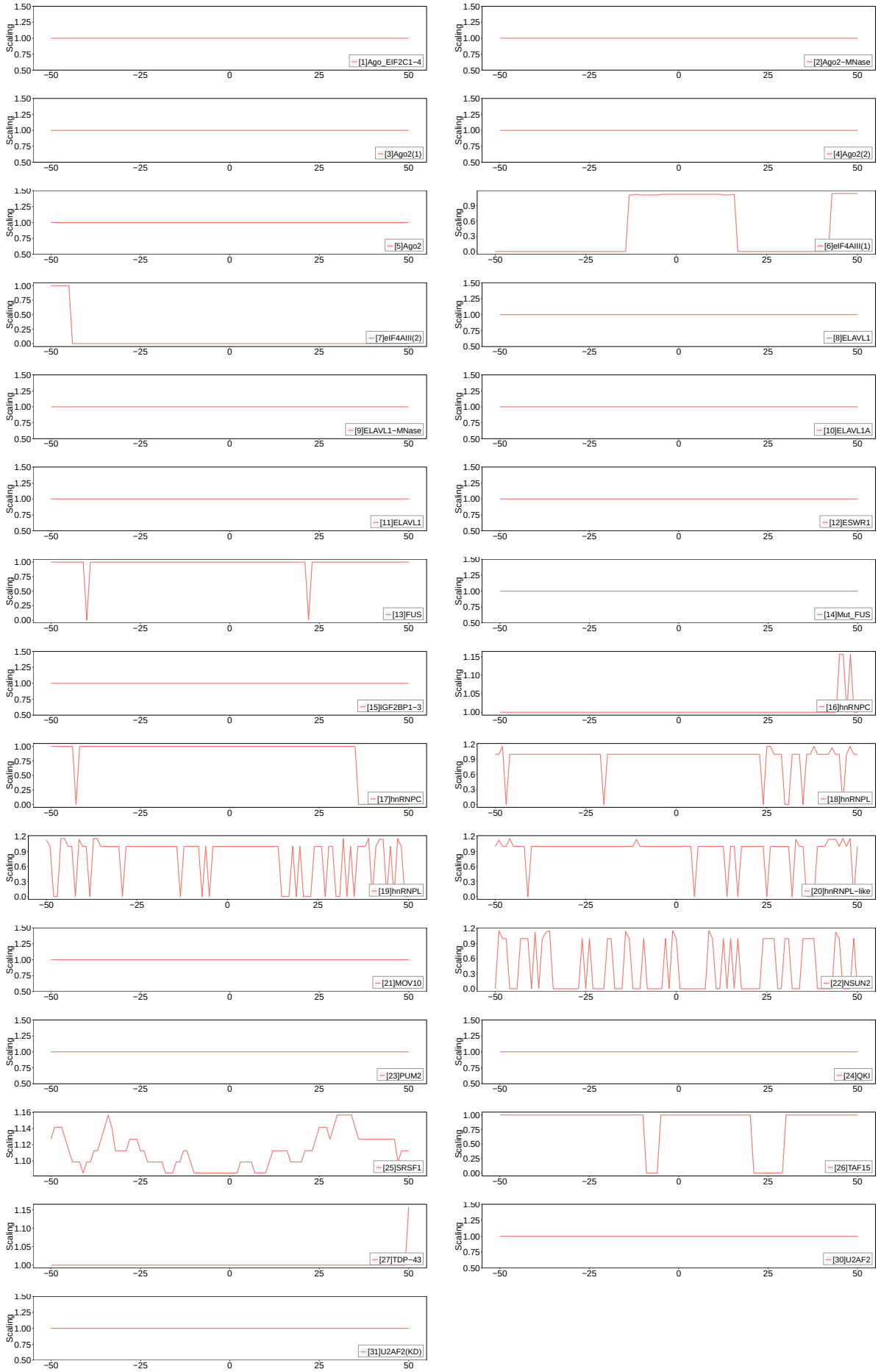

Figure S53: The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [29]TIAL1.

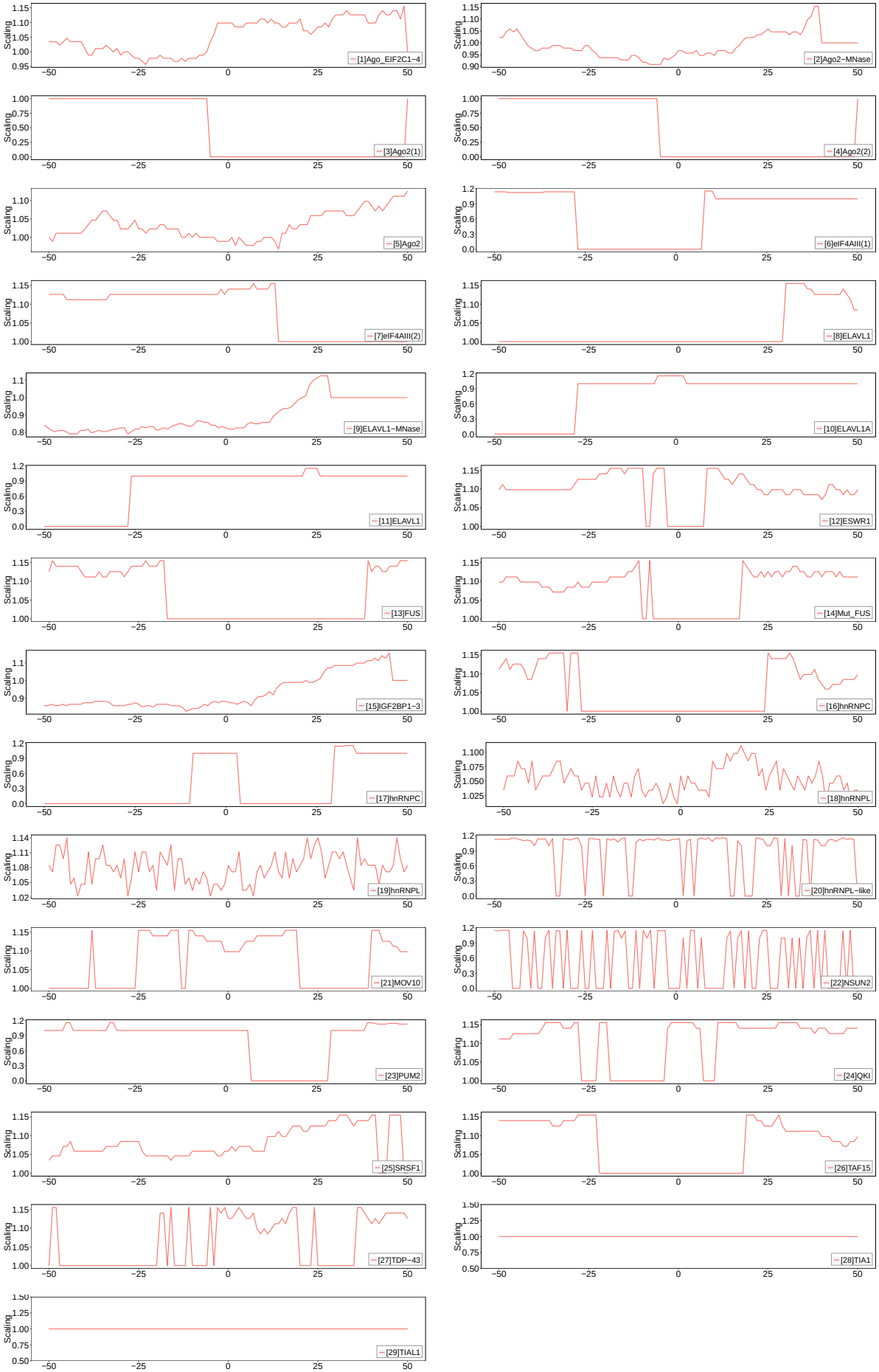

Figure S54: The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [30]U2AF2.

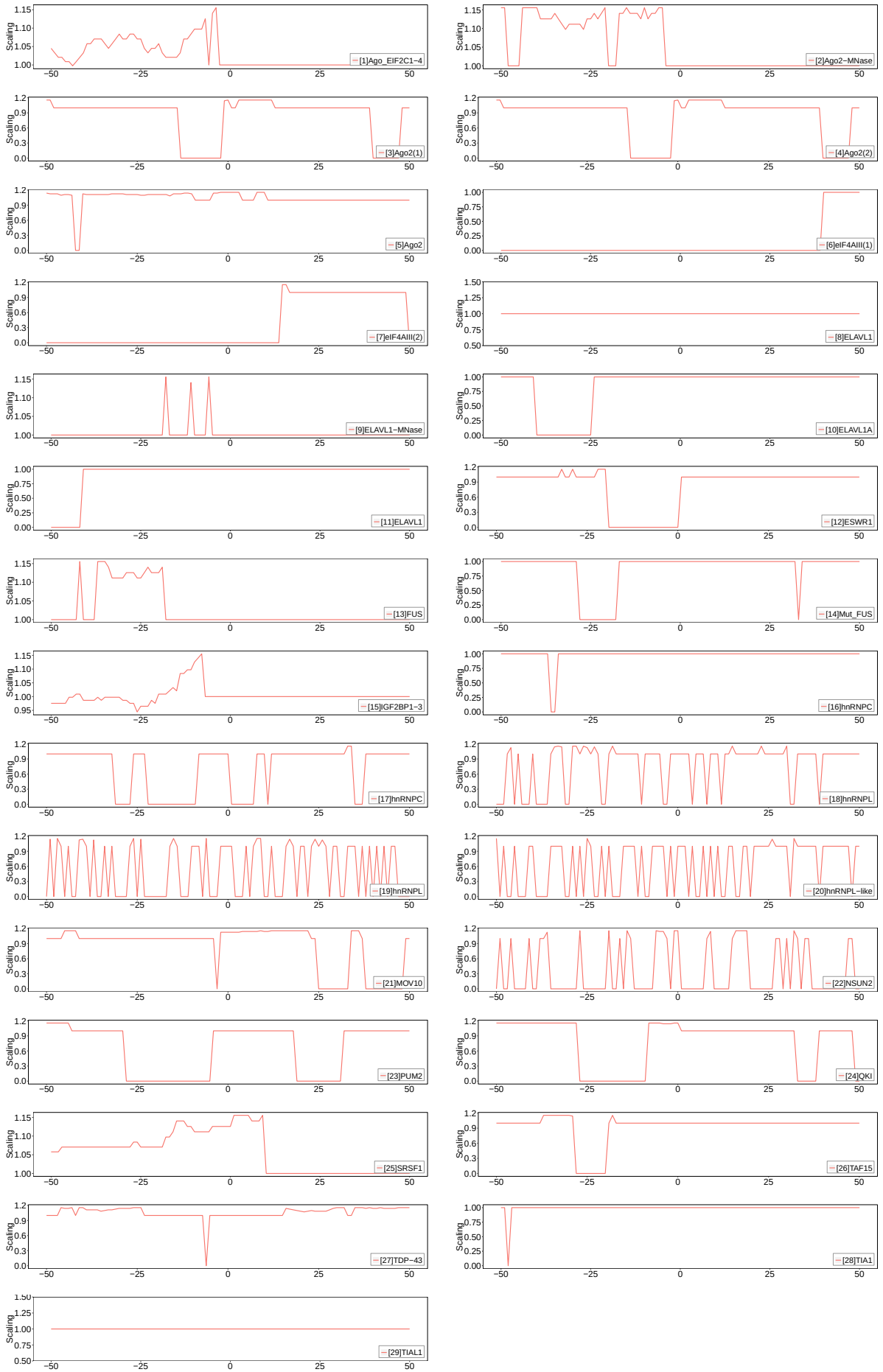

Figure S55: The features of Hetero-RP for co-binding protein-RNA cDNA counts in the dataset [31]U2AF2(KD).

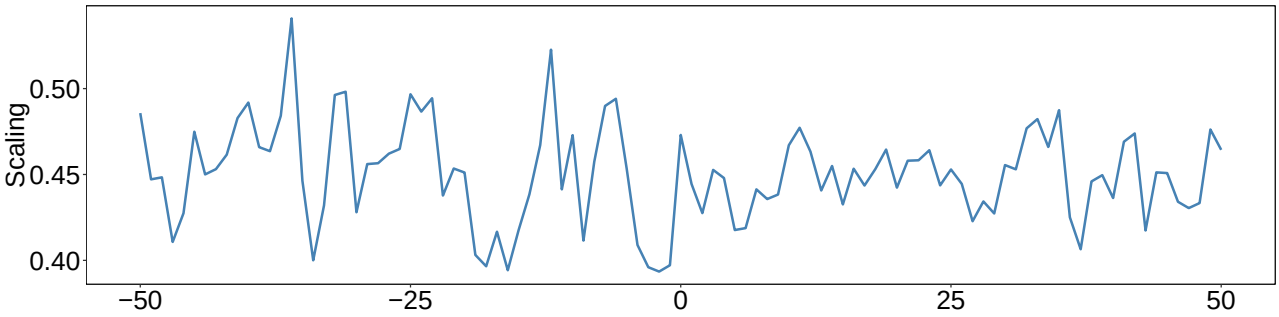

Figure S56: The features of Hetero-RP for RNA secondary structure in the dataset [1]Ago_EIF2C1-4.

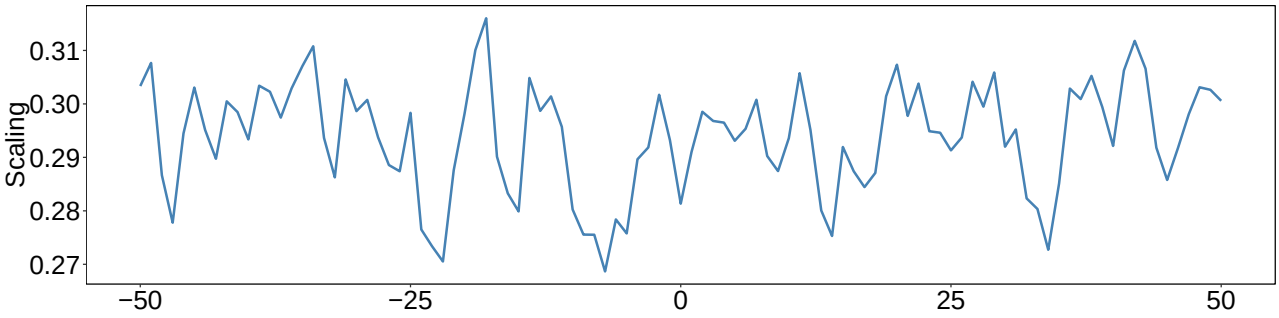

Figure S57: The features of Hetero-RP for RNA secondary structure in the dataset [2]Ago2-MNase.

5.3.2 Features of RNA secondary structures

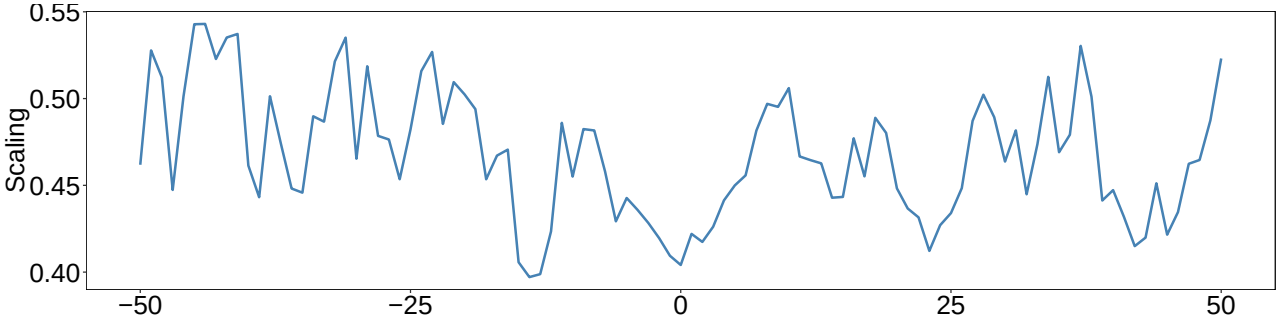

Figure S58: The features of Hetero-RP for RNA secondary structure in the dataset [3]Ago2(1).

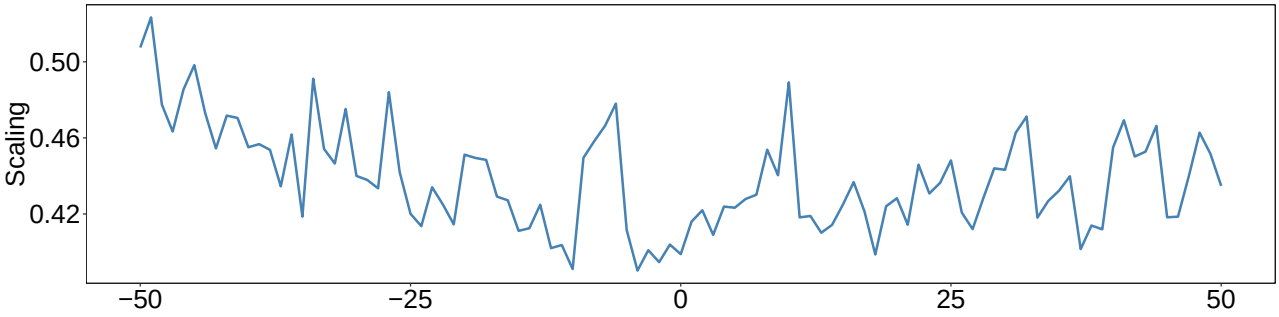

Figure S59: The features of Hetero-RP for RNA secondary structure in the dataset [4]Ago2(2).

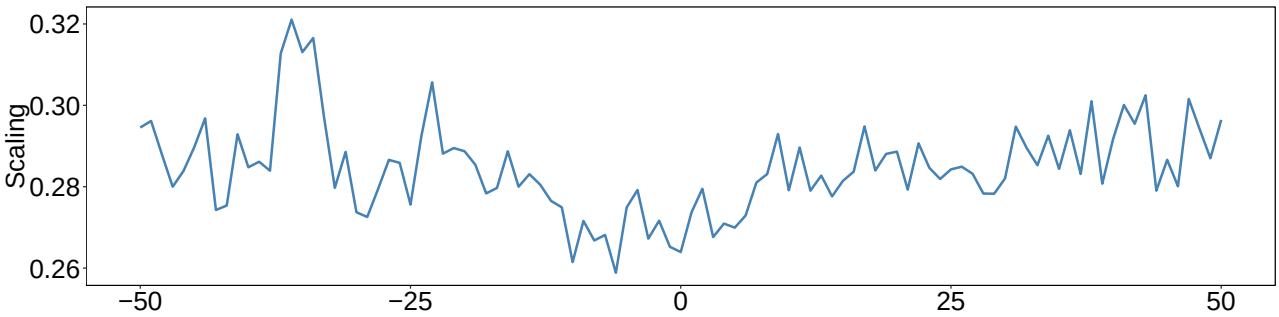

Figure S60: The features of Hetero-RP for RNA secondary structure in the dataset [5]Ago2.

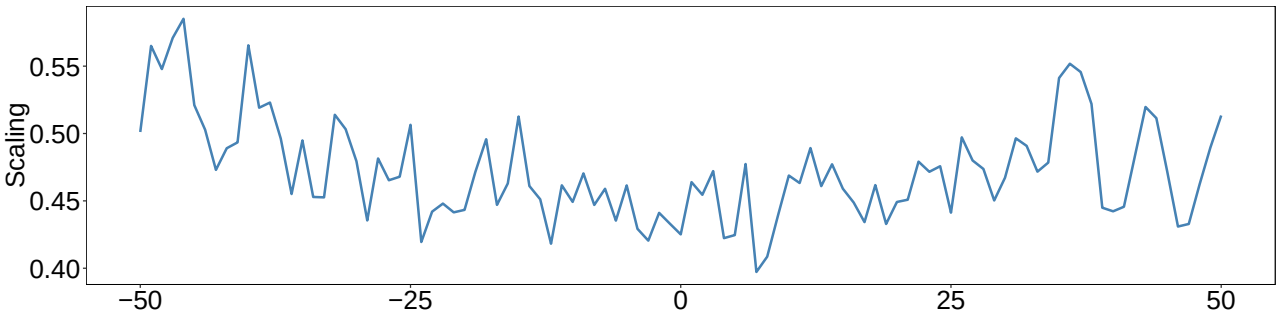

Figure S61: The features of Hetero-RP for RNA secondary structure in the dataset [6]eIF4AIII(1).

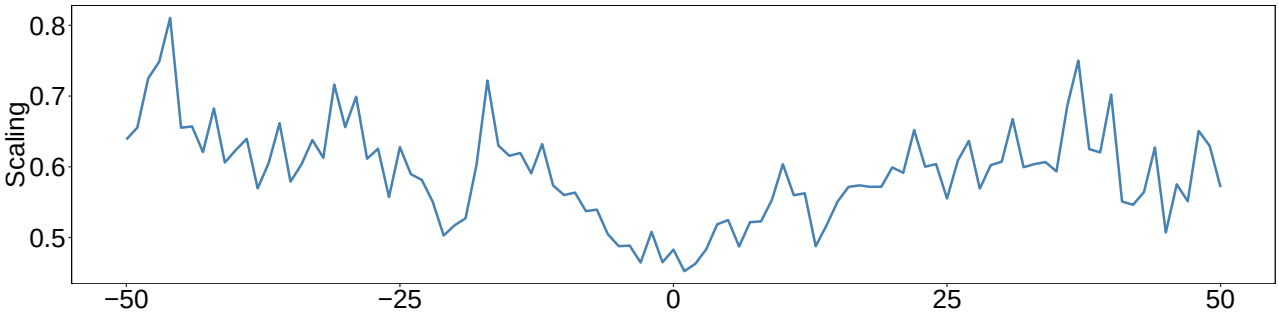

Figure S62: The features of Hetero-RP for RNA secondary structure in the dataset [7]eIF4AIII(2).

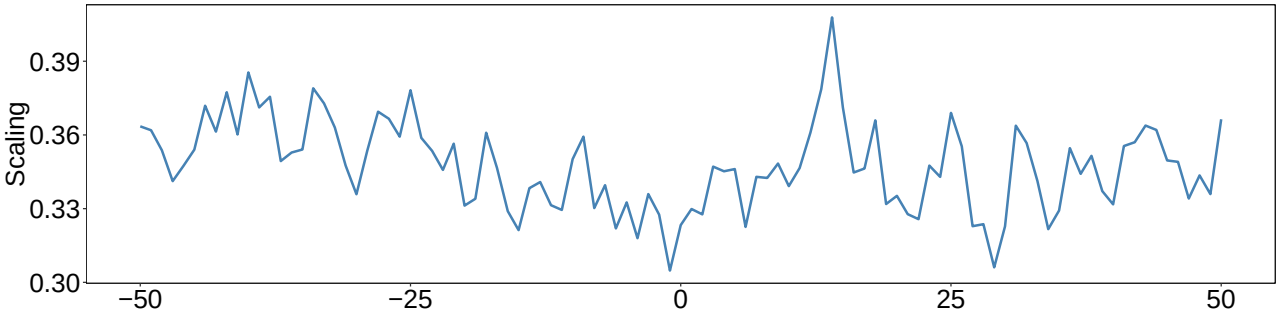

Figure S63: The features of Hetero-RP for RNA secondary structure in the dataset [8]ELAVL1.

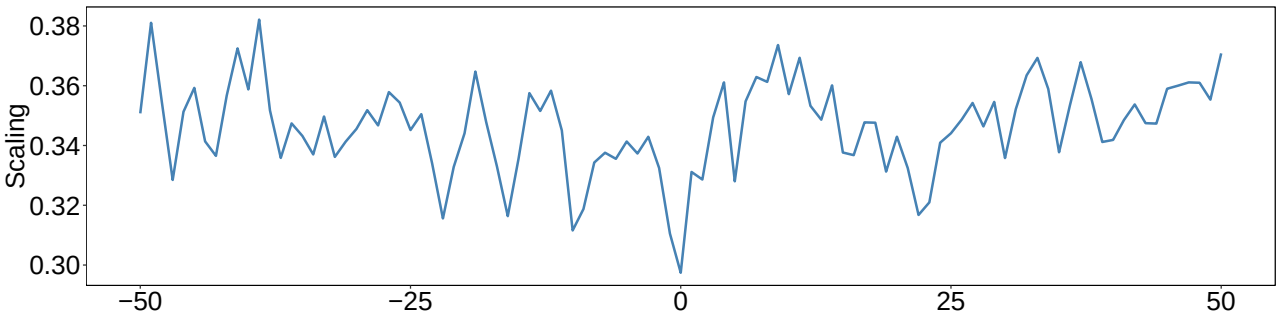

Figure S64: The features of Hetero-RP for RNA secondary structure in the dataset [9]ELAVL1-MNase.

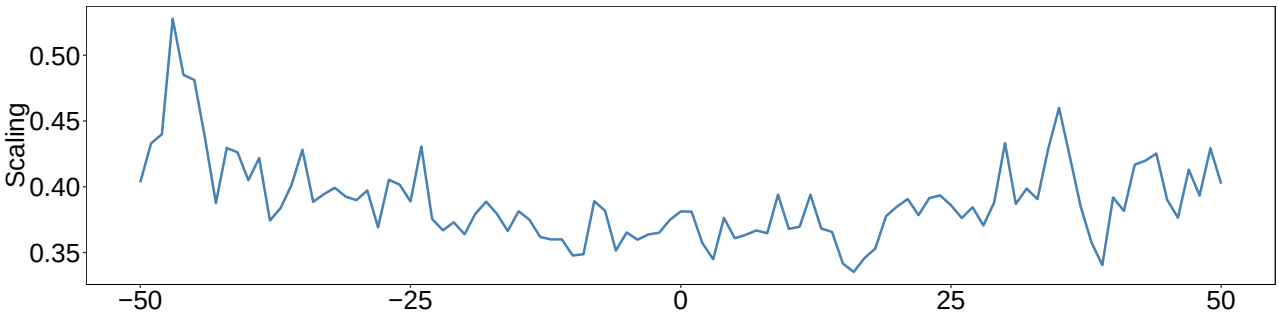

Figure S65: The features of Hetero-RP for RNA secondary structure in the dataset [10]ELAVL1A.

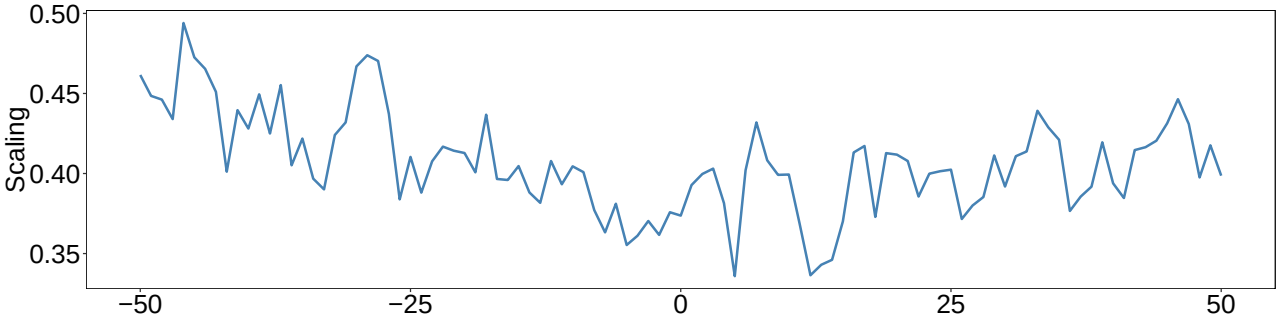

Figure S66: The features of Hetero-RP for RNA secondary structure in the dataset [11]ELAVL1.

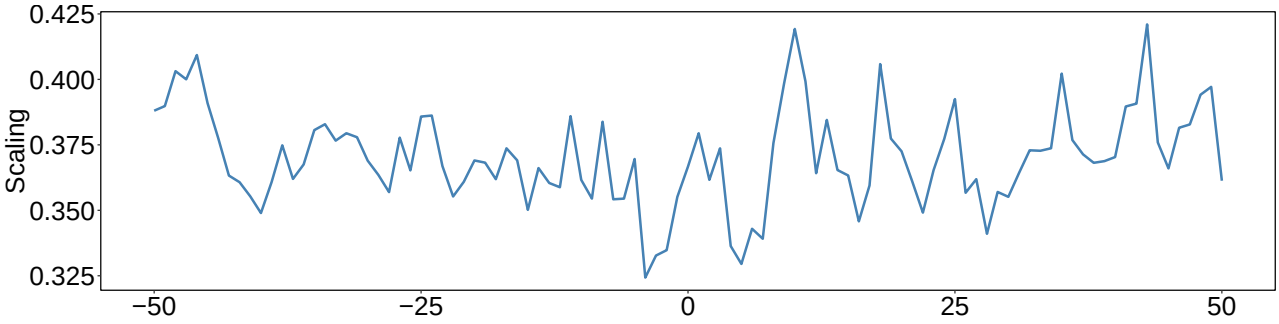

Figure S67: The features of Hetero-RP for RNA secondary structure in the dataset [12]ESWR1.

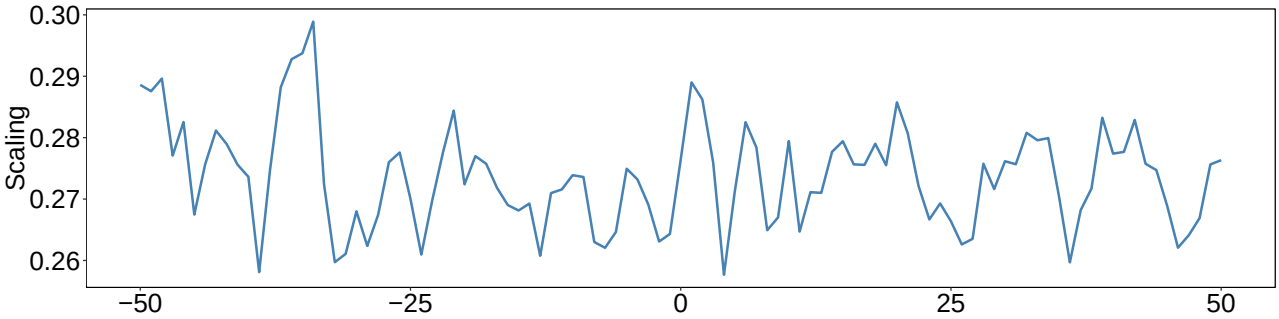

Figure S68: The features of Hetero-RP for RNA secondary structure in the dataset [13]FUS.

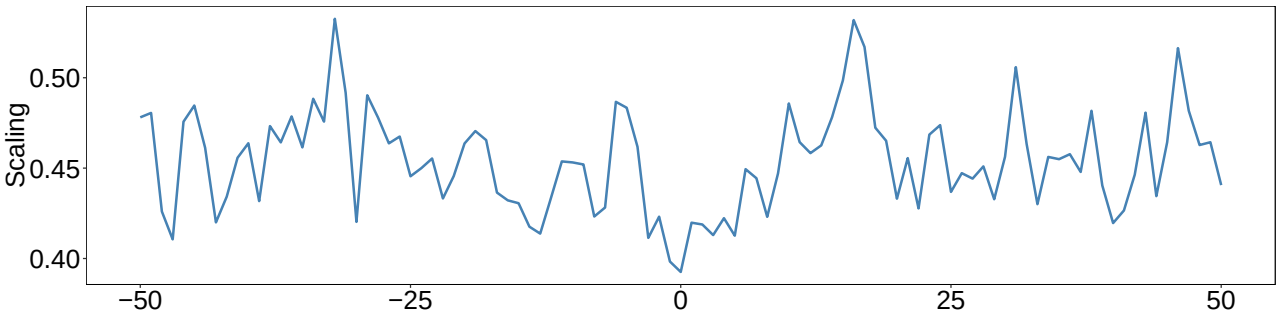

Figure S69: The features of Hetero-RP for RNA secondary structure in the dataset [14]Mut_FUS.

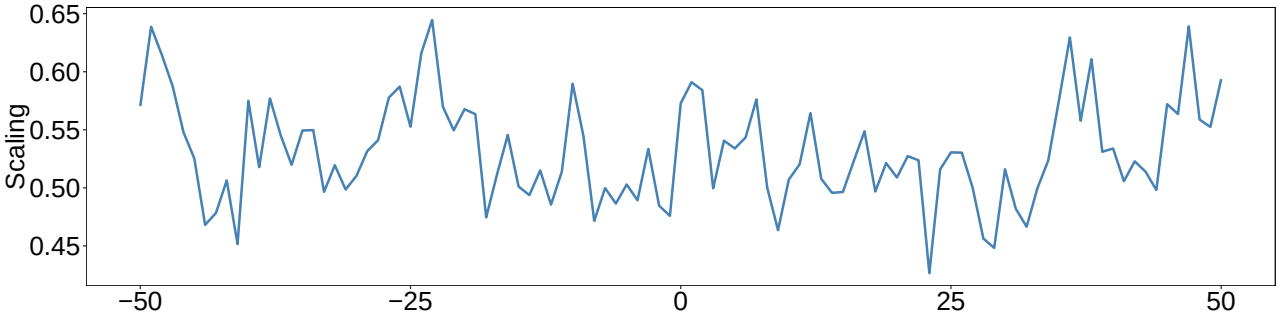

Figure S70: The features of Hetero-RP for RNA secondary structure in the dataset [15]IGF2BP1-3.

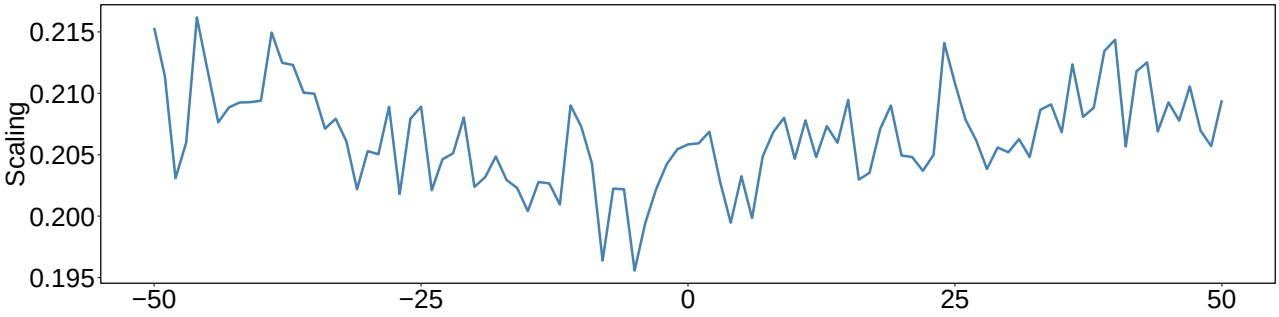

Figure S71: The features of Hetero-RP for RNA secondary structure in the dataset [16]hnRNPC.

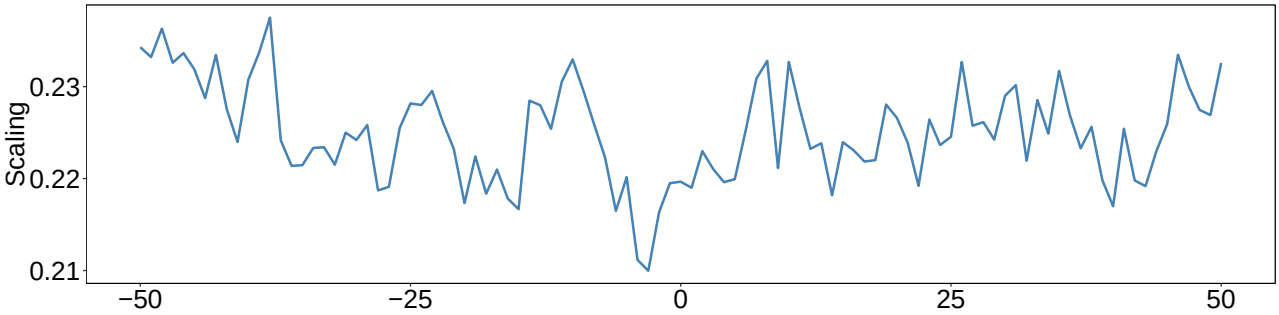

Figure S72: The features of Hetero-RP for RNA secondary structure in the dataset [17]hnRNPC.

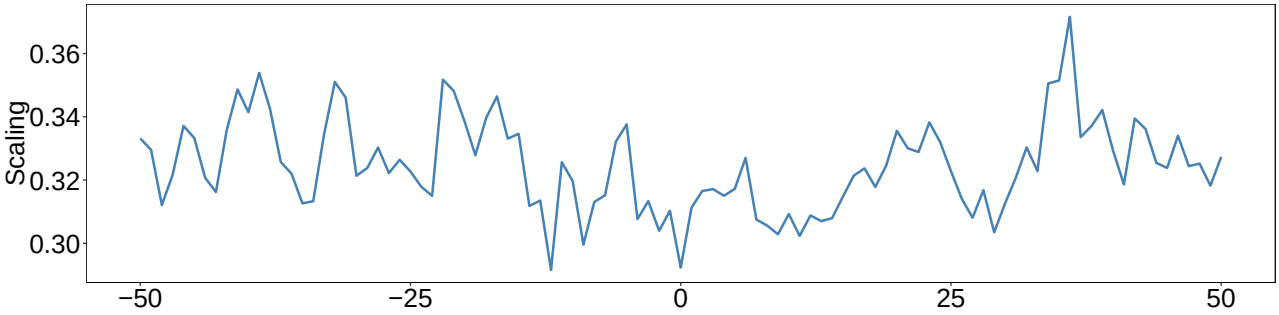

Figure S73: The features of Hetero-RP for RNA secondary structure in the dataset [18]hnRNPL.

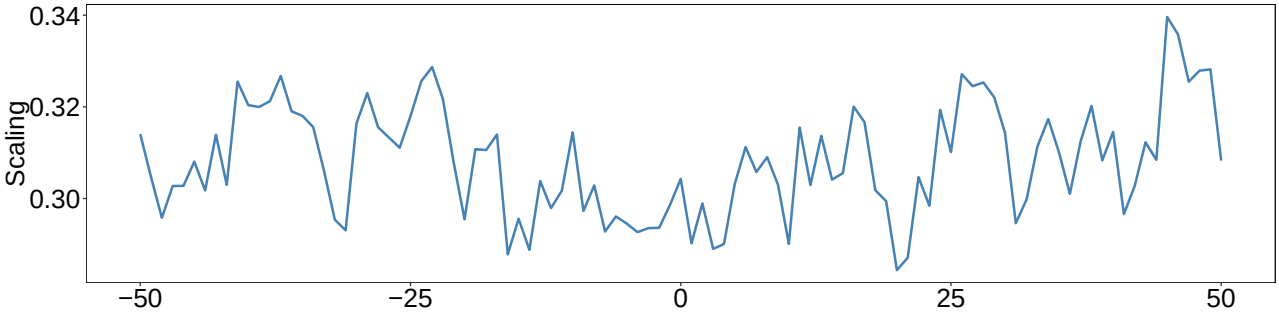

Figure S74: The features of Hetero-RP for RNA secondary structure in the dataset [19]hnRNPL.

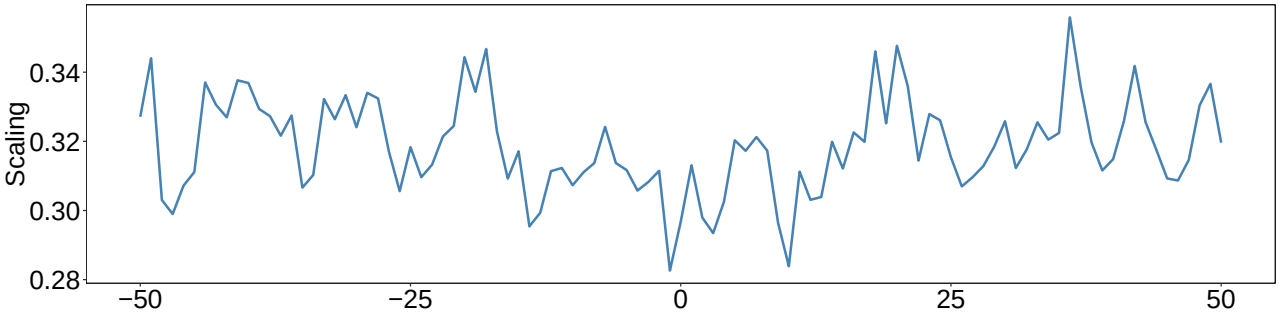

Figure S75: The features of Hetero-RP for RNA secondary structure in the dataset [20]hnRNPL-like.

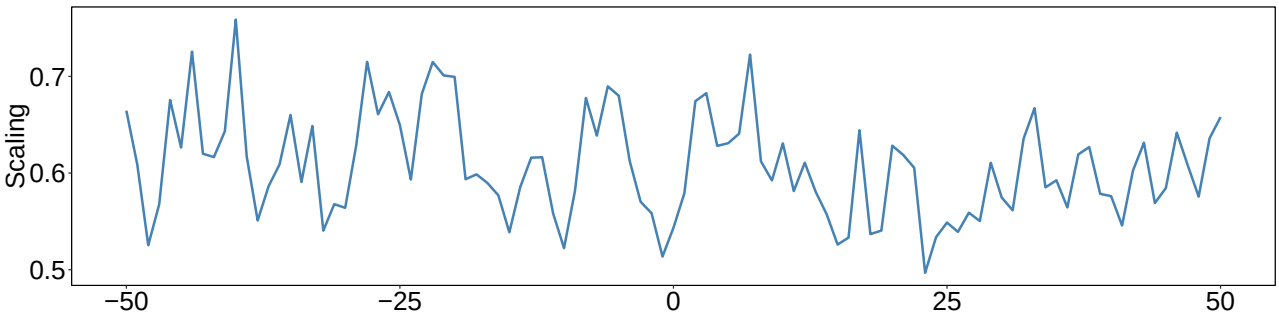

Figure S76: The features of Hetero-RP for RNA secondary structure in the dataset [21]MOV10.

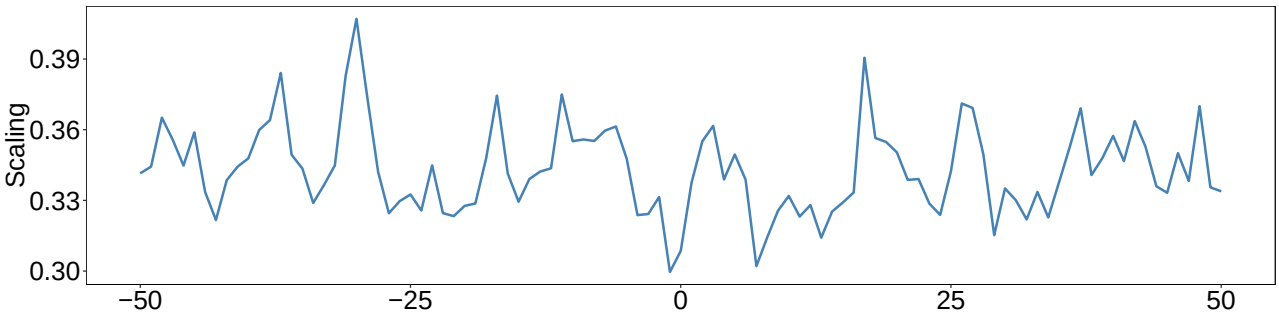

Figure S77: The features of Hetero-RP for RNA secondary structure in the dataset [22]NSUN2.

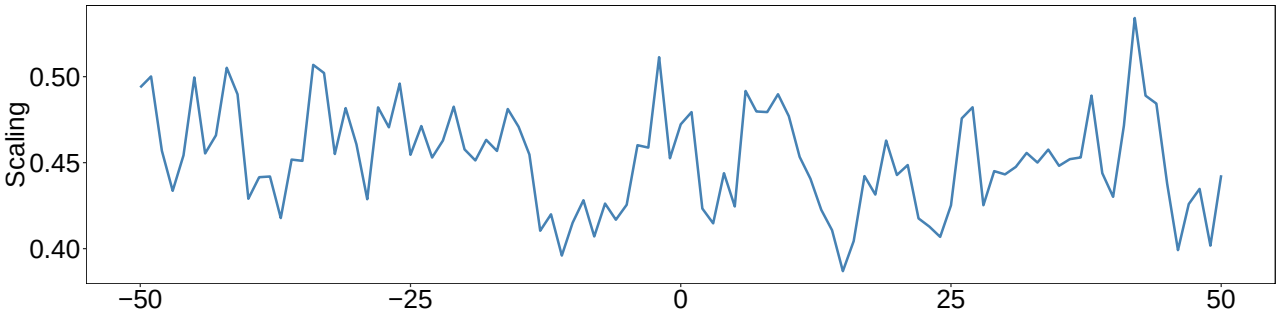

Figure S78: The features of Hetero-RP for RNA secondary structure in the dataset [23]PUM2.

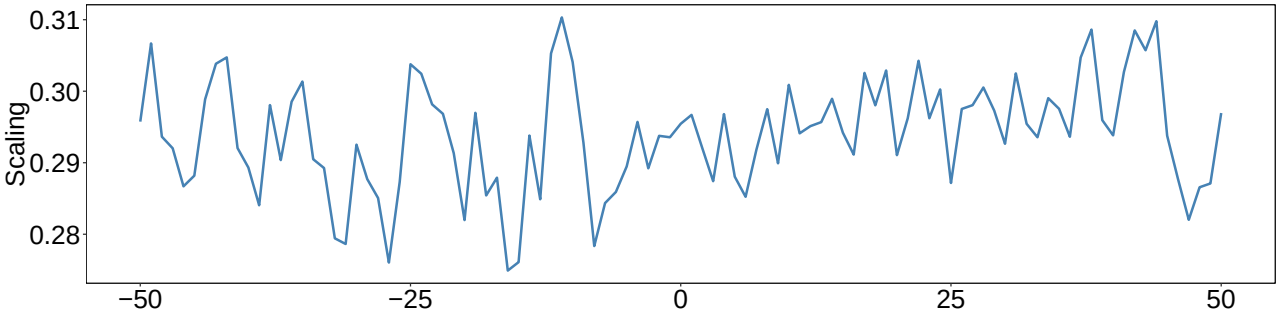

Figure S79: The features of Hetero-RP for RNA secondary structure in the dataset [24]QKI.

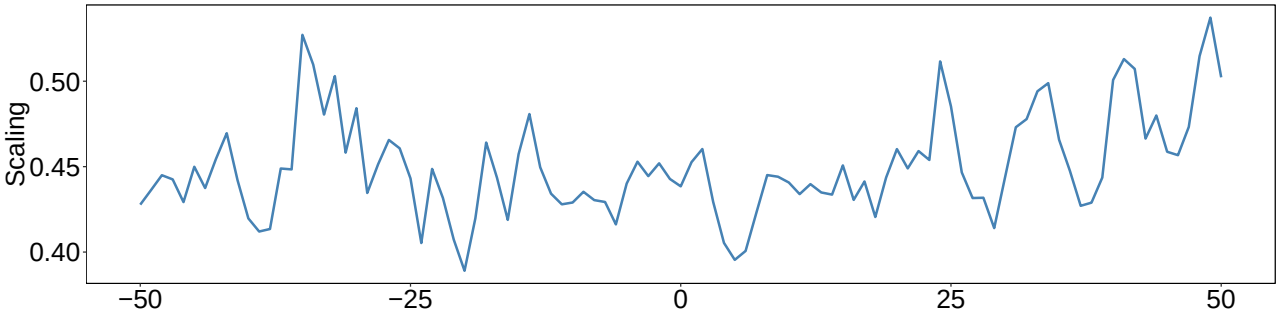

Figure S80: The features of Hetero-RP for RNA secondary structure in the dataset [25]SRSF1.

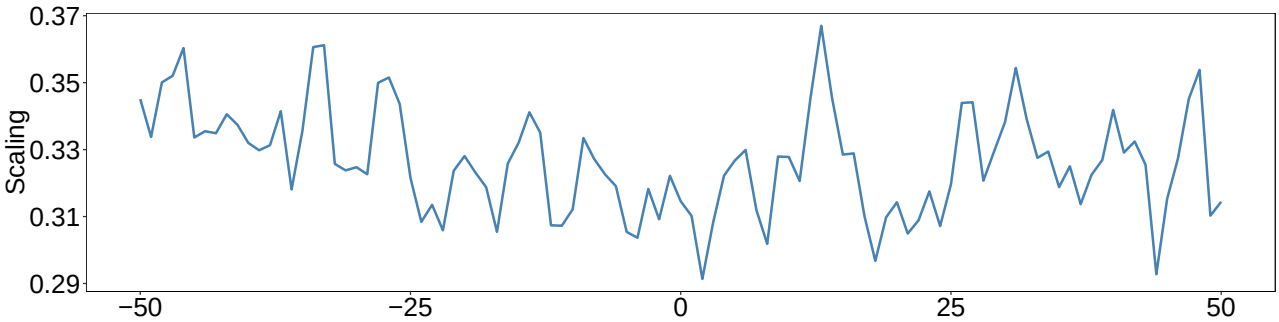

Figure S81: The features of Hetero-RP for RNA secondary structure in the dataset [26]TAF15.

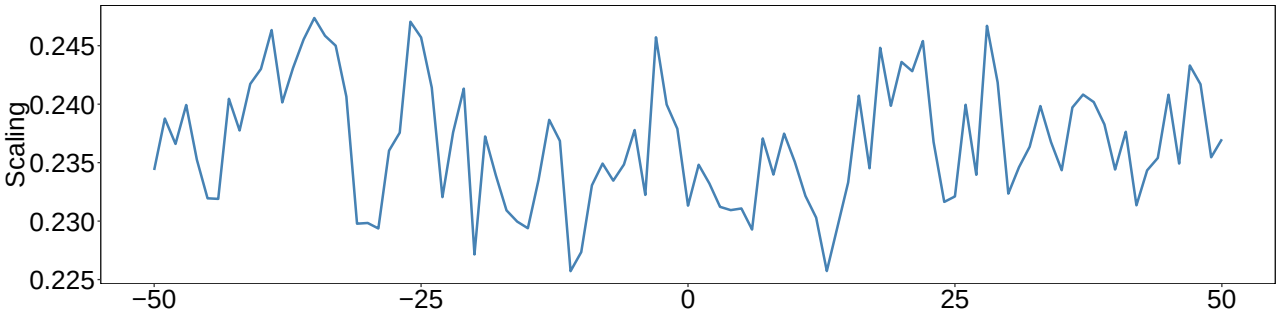

Figure S82: The features of Hetero-RP for RNA secondary structure in the dataset [27]TDP-43.

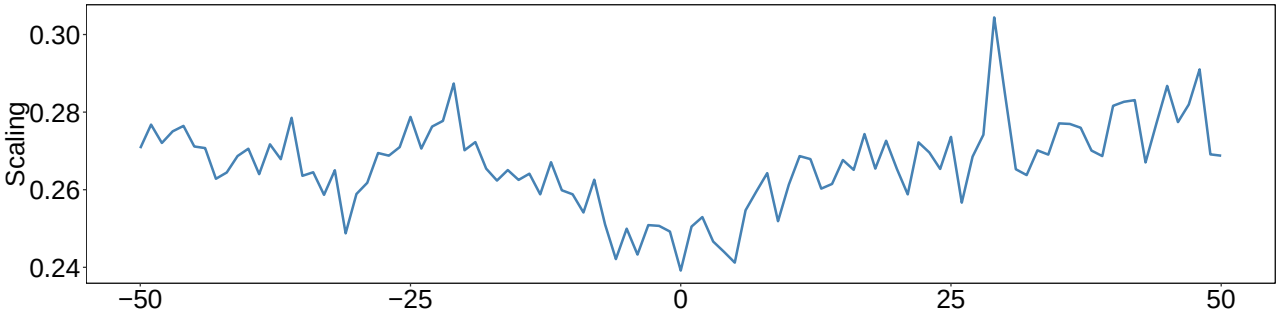

Figure S83: The features of Hetero-RP for RNA secondary structure in the dataset [28]TIA1.

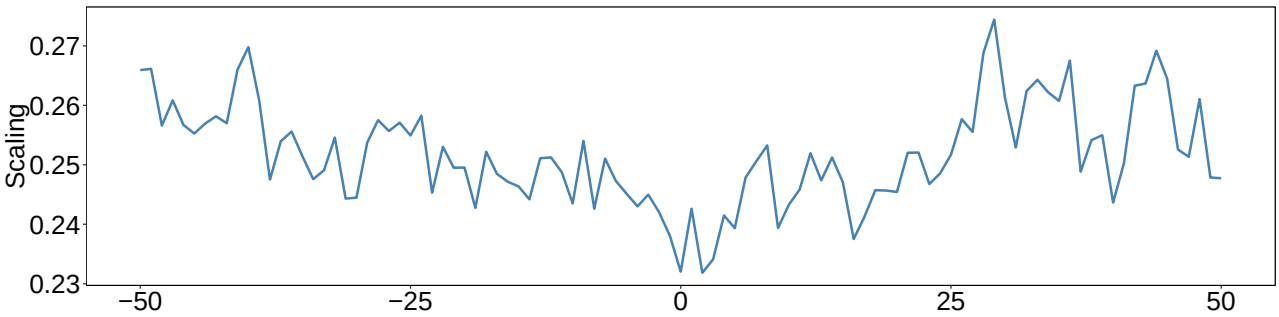

Figure S84: The features of Hetero-RP for RNA secondary structure in the dataset [29]TIAL1.

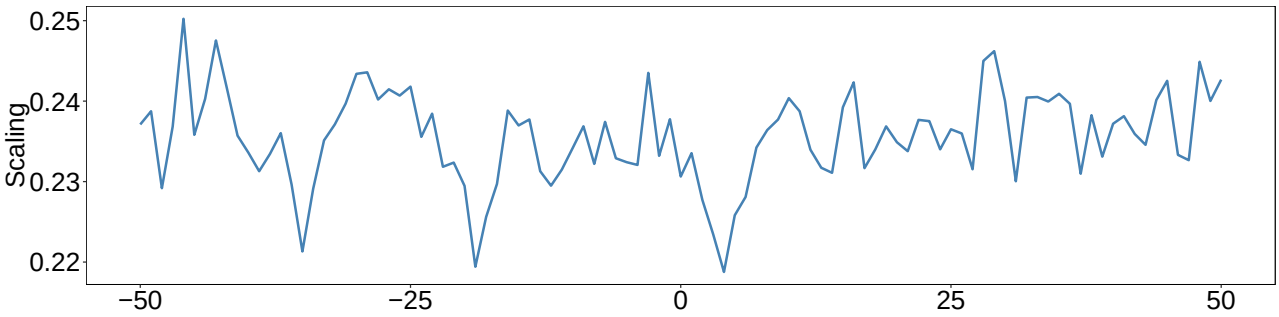

Figure S85: The features of Hetero-RP for RNA secondary structure in the dataset [30]U2AF2.

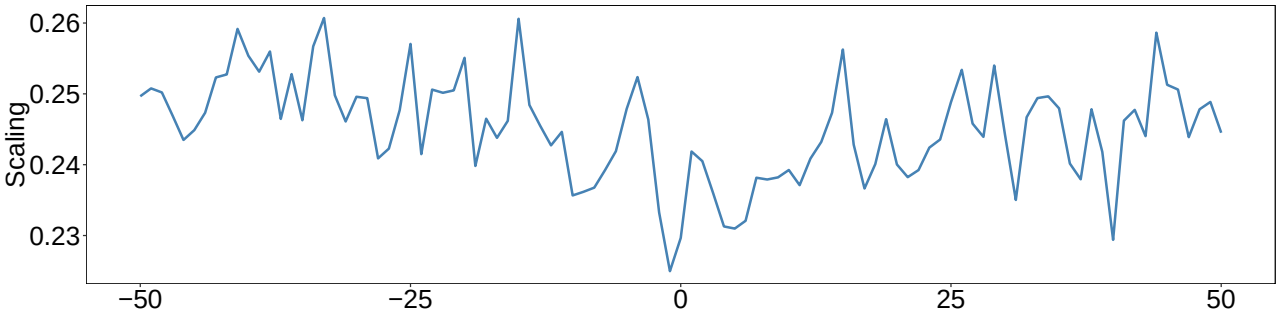

Figure S86: The features of Hetero-RP for RNA secondary structure in the dataset [31]U2AF2(KD).

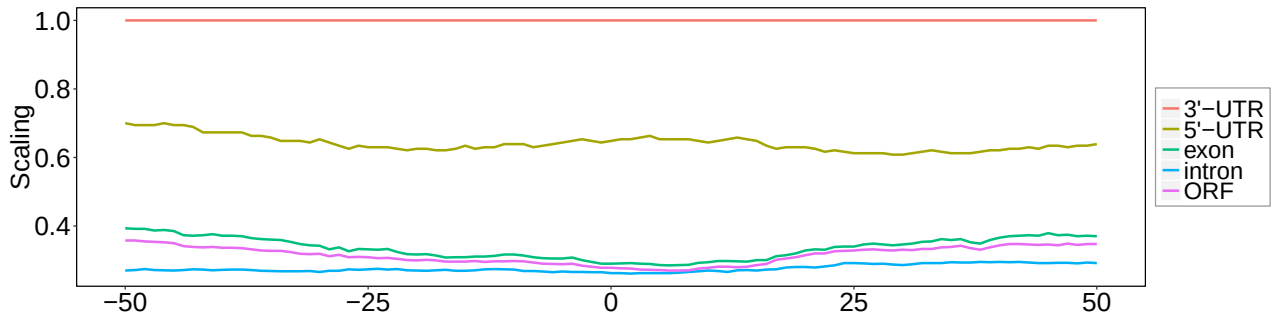

Figure S87: The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [1]Ago_EIF2C1-4.

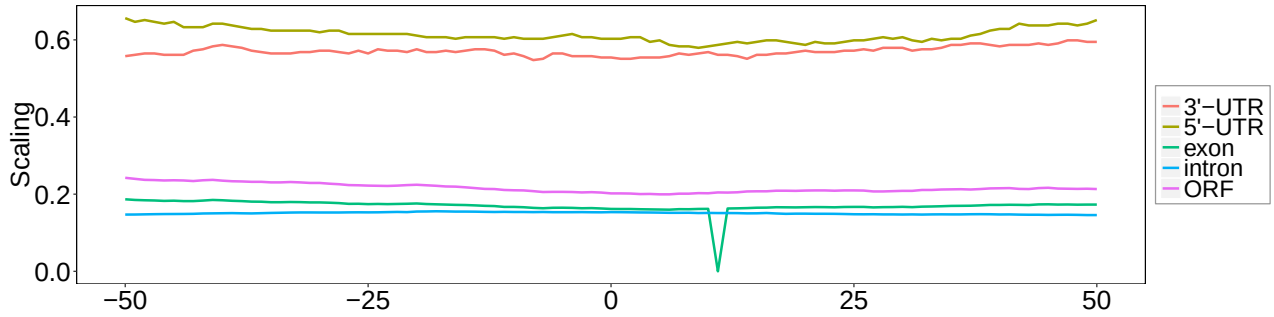

Figure S88: The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [2]Ago2-MNase.

5.3.3 Features of Region types (intron,exon,5'-UTR, 3'-UTR and ORF)

References

[1] Johannes Alneberg, Brynjar Smári Bjarnason, Ino de Bruijn, Melanie Schirmer, Joshua Quick, Umer Z Ijaz, Leo Lahti, Nicholas J Loman, Anders F Andersson, and Christopher Quince. Binning metagenomic contigs by coverage and composition. *Nature Methods*, 11(11):1144–1146, 2014. [PubMed:25218180] [doi:10.1038/nmeth.3103].

[2] Fan RK Chung. *Spectral graph theory*, volume 92. American Mathematical Society., 1997.

[3] Robert B Denman. Using rnafold to predict the activity of small catalytic rnas. *Biotechniques*, 15(6):1090–1095, 1993. [PubMed:8292343].

[4] Yingying Fan and Jinchi Lv. Innovated scalable efficient estimation in ultra-large Gaussian graphical models. *The Annals of Statistics*, 44:2098–2126, 2016. [doi:10.1214/15-AOS1416].

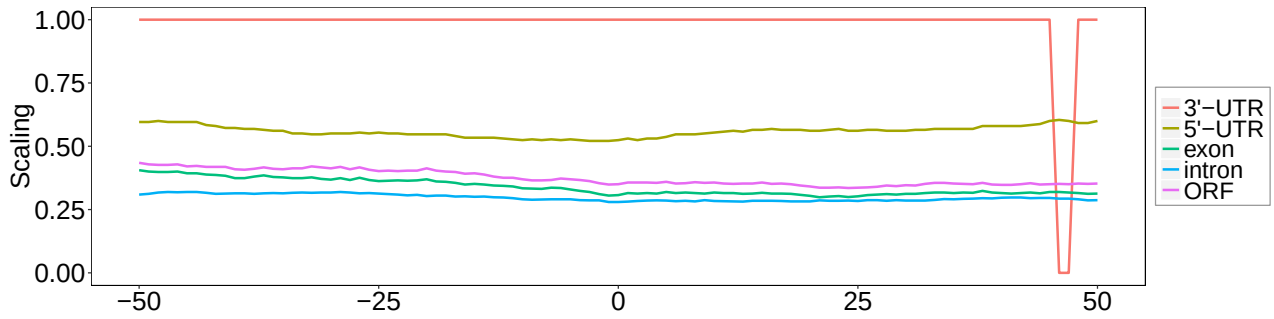

Figure S89: The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [3]Ago2(1).

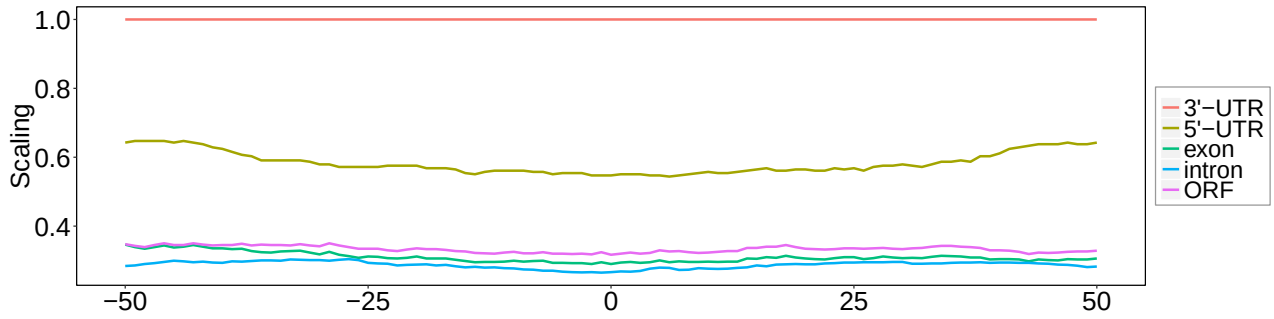

Figure S90: The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [4]Ago2(2).

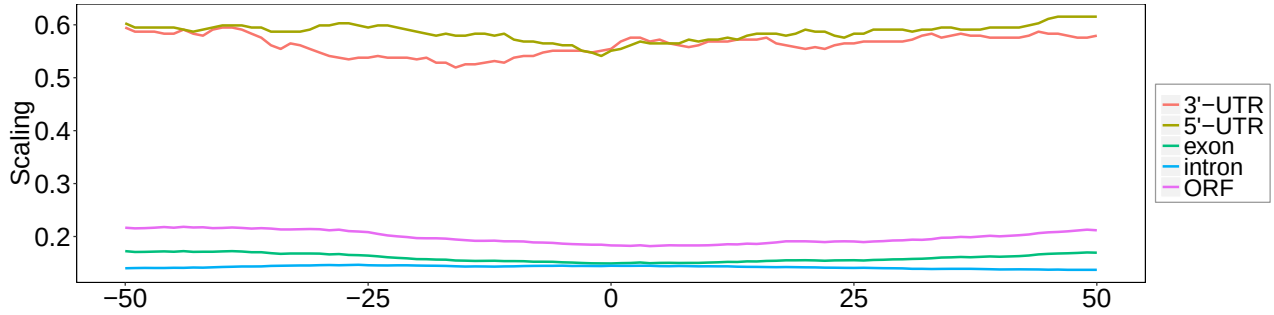

Figure S91: The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [5]Ago2.

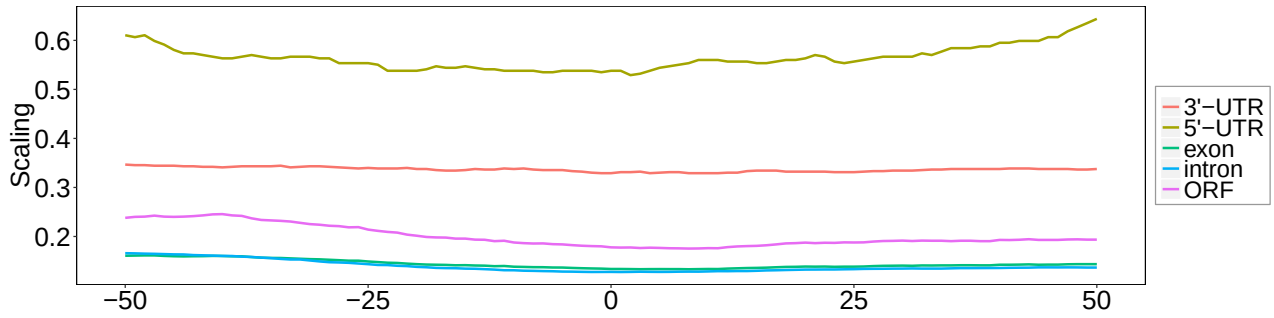

Figure S92: The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [6]eIF4AIII(1).

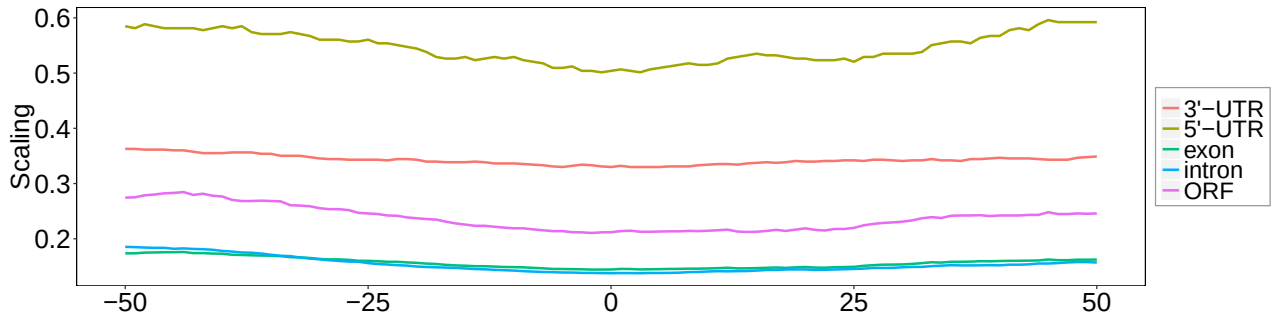

Figure S93: The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [7]eIF4AIII(2).

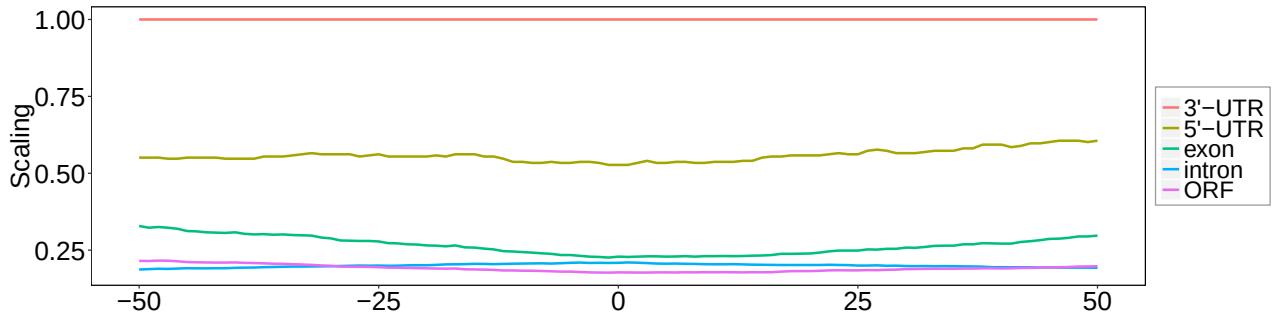

Figure S94: The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [8]ELAVL1.

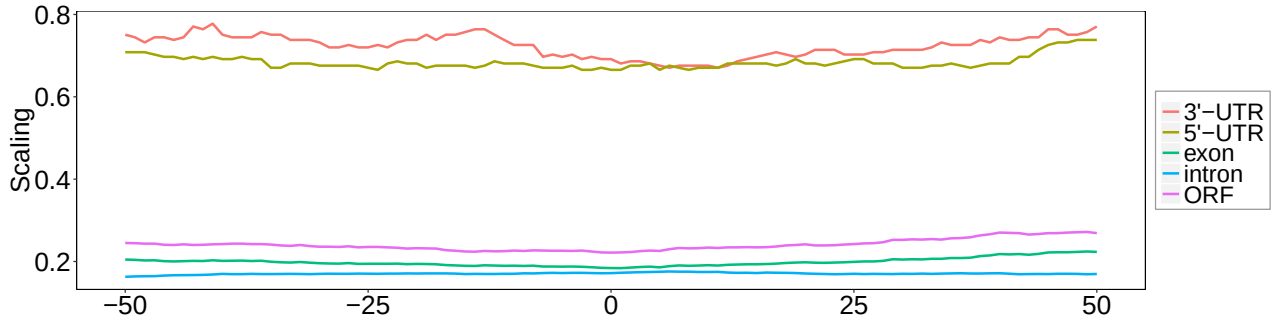

Figure S95: The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [9]ELAVL1-MNase.

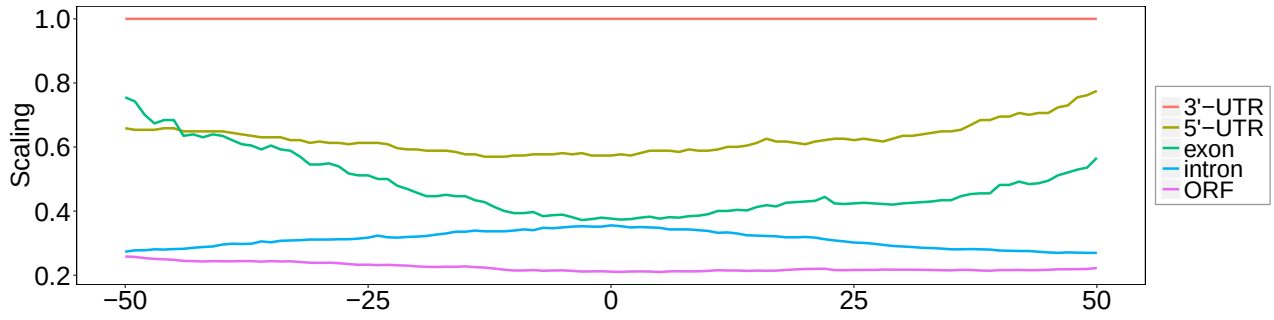

Figure S96: The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [10]ELAVL1A.

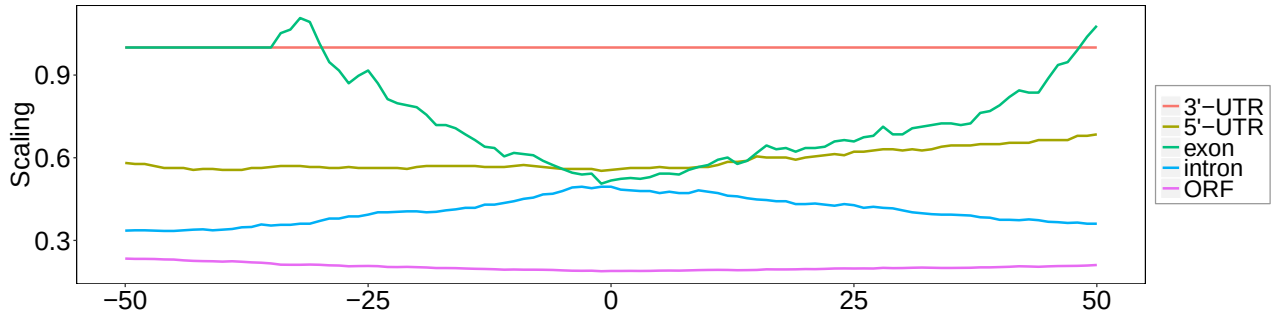

Figure S97: The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [11]ELAVL1.

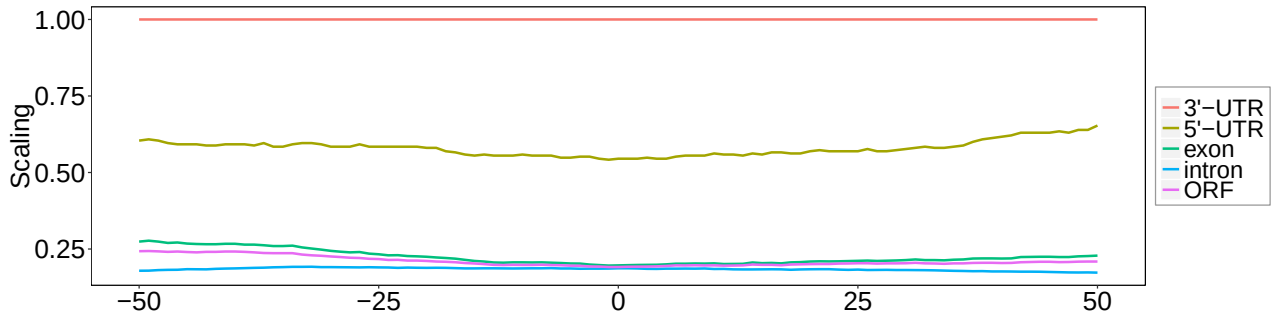

Figure S98: The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [12]ESWR1.

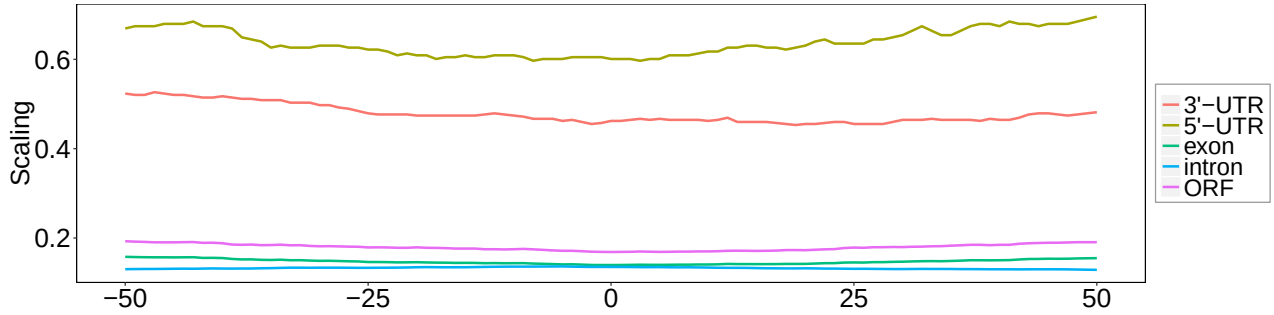

Figure S99: The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [13]FUS.

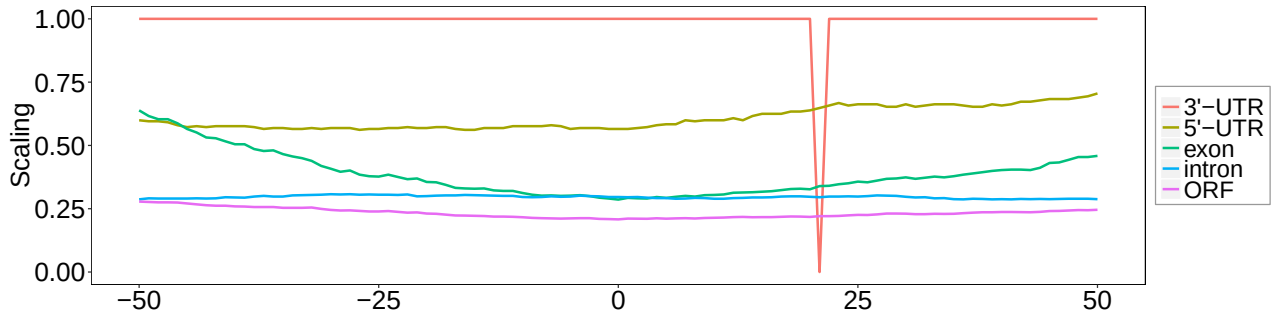

Figure S100: The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [14]Mut_FUS.

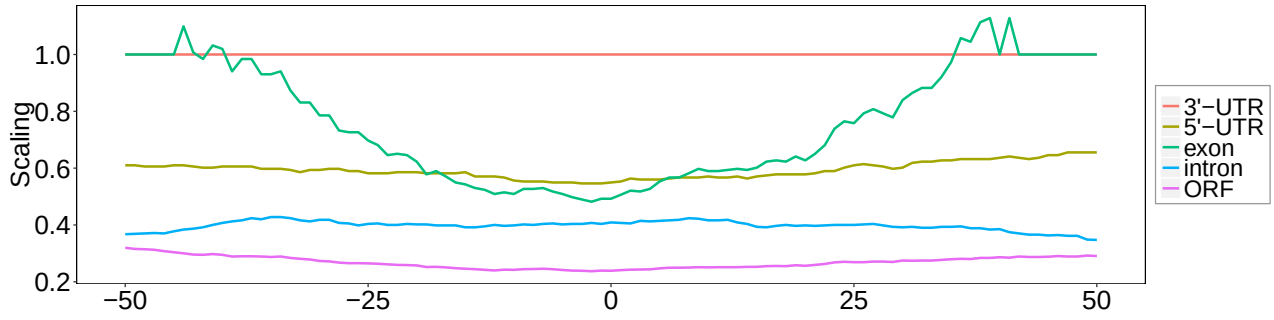

Figure S101: The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [15]IGF2BP1-3.

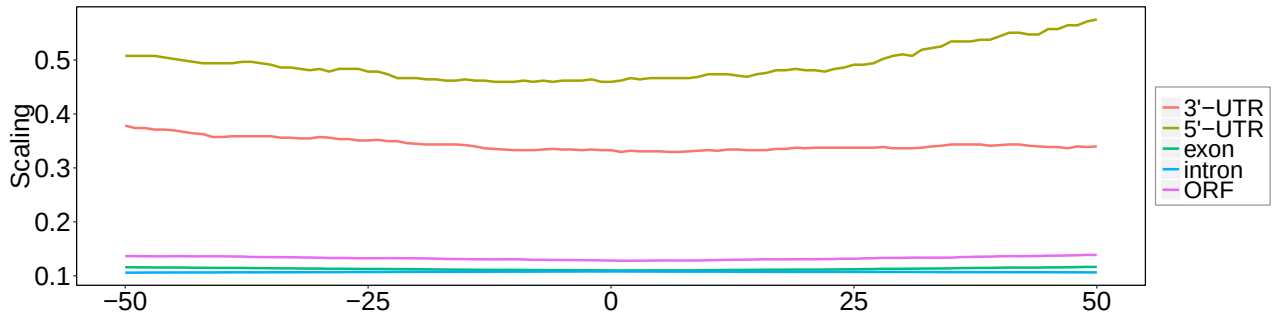

Figure S102: The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [16]hnRNPC.

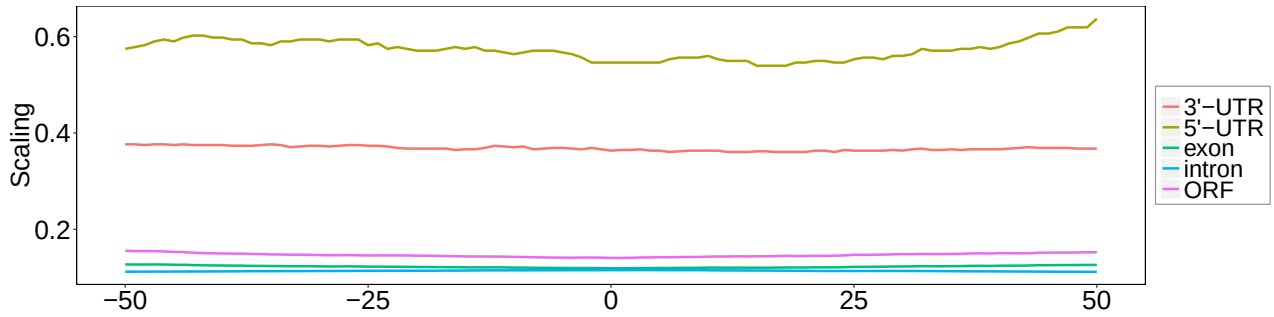

Figure S103: The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [17]hnRNPC.

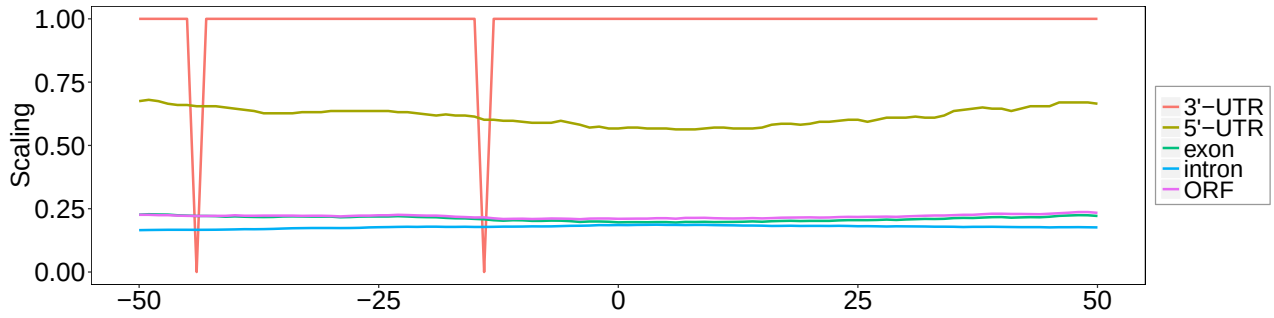

Figure S104: The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [18]hnRNPL.

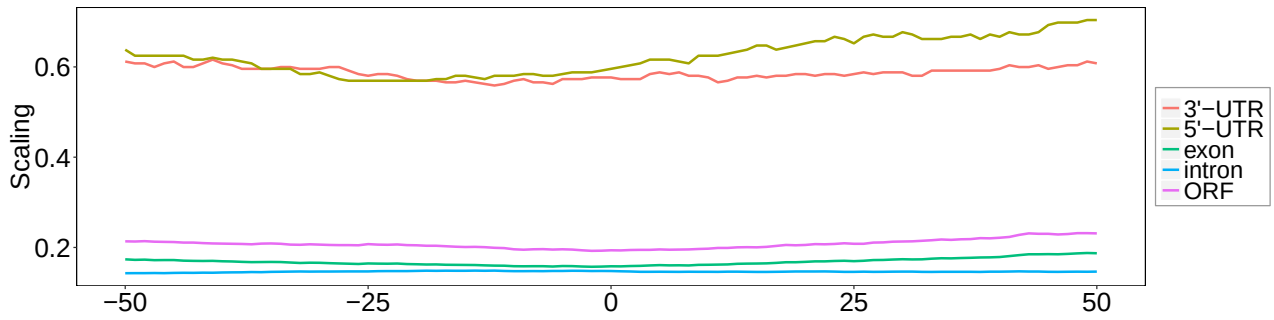

Figure S105: The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [19]hnRNPL.

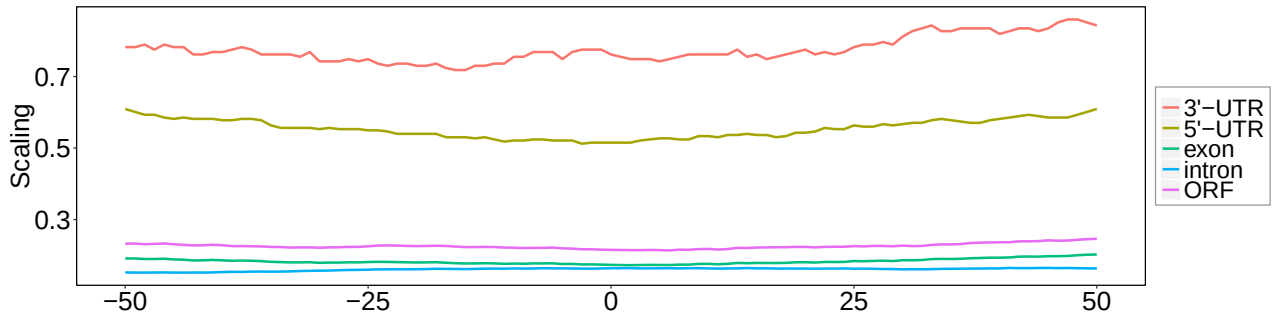

Figure S106: The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [20]hnRNPL-like.

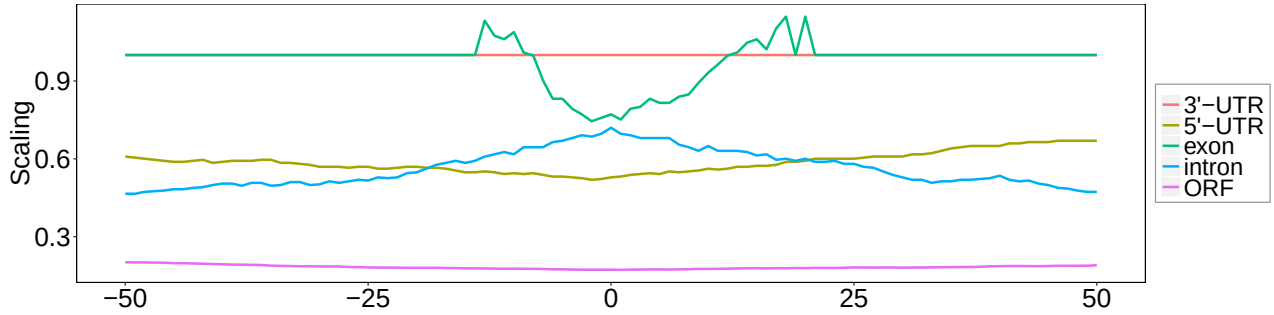

Figure S107: The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [21]MOV10.

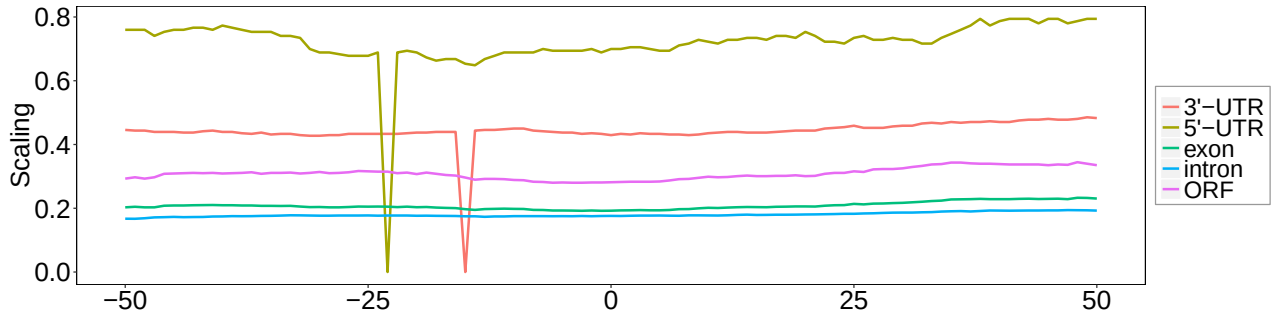

Figure S108: The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [22]NSUN2.

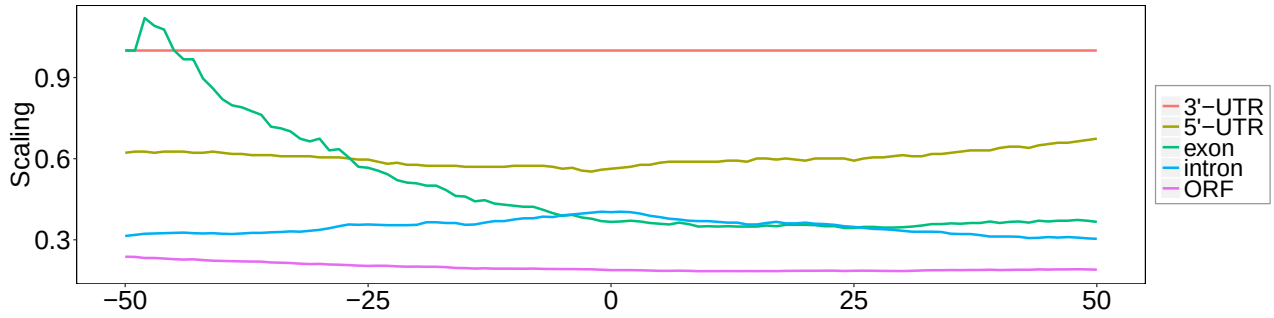

Figure S109: The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [23]PUM2.

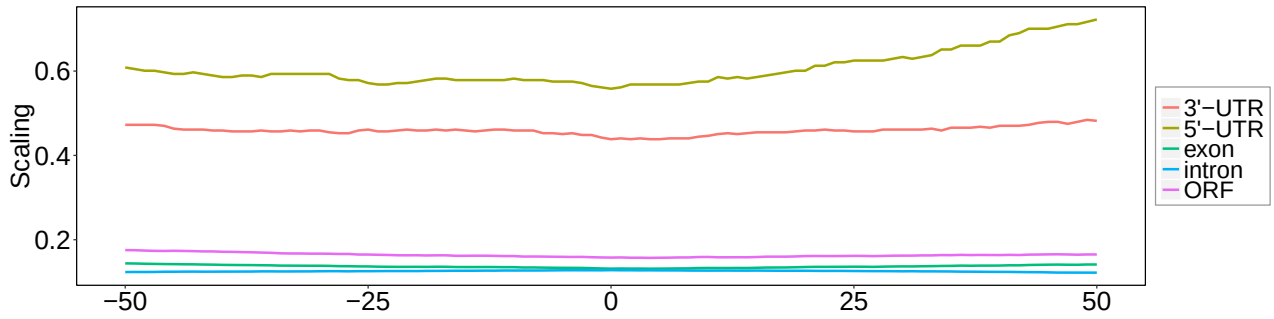

Figure S110: The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [24]QKI.

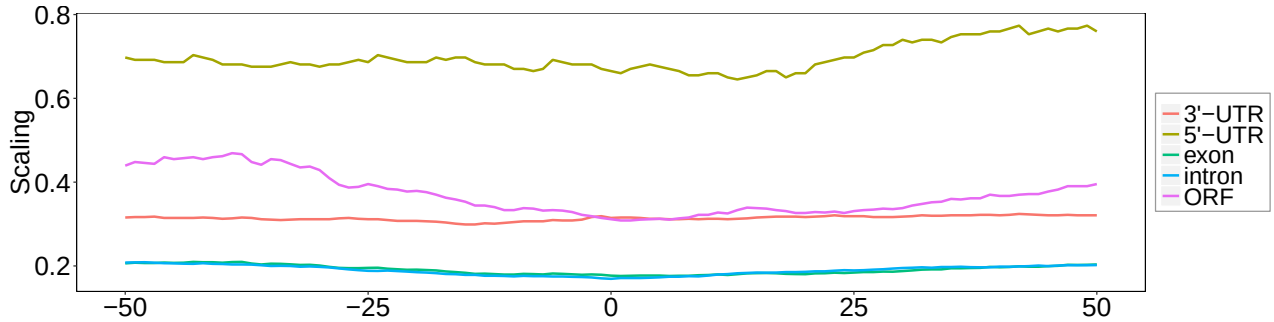

Figure S111: The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [25]SRSF1.

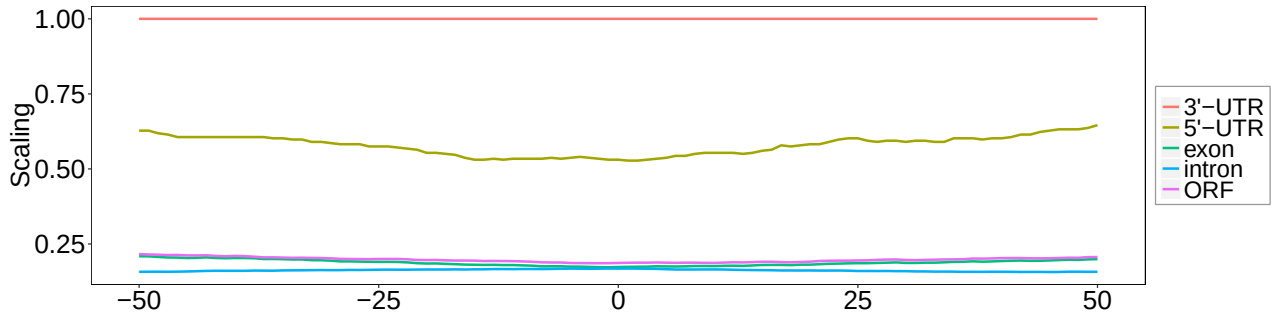

Figure S112: The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [26]TAF15.

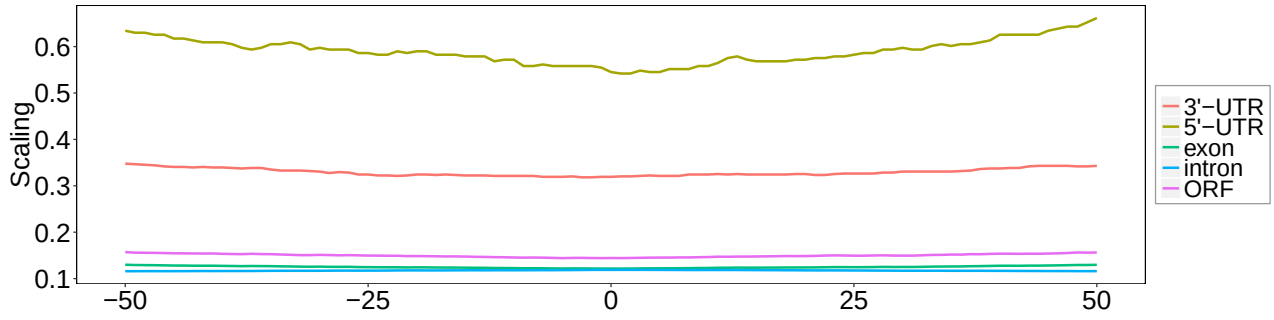

Figure S113: The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [27]TDP-43.

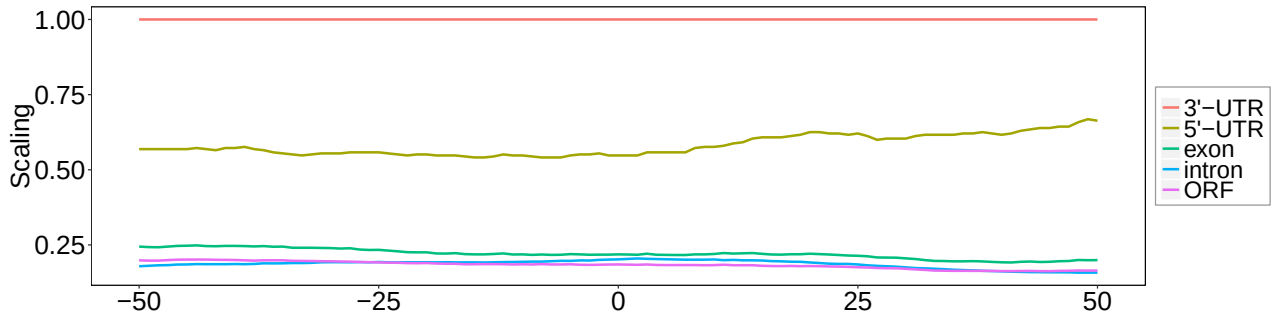

Figure S114: The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [28]TIA1.

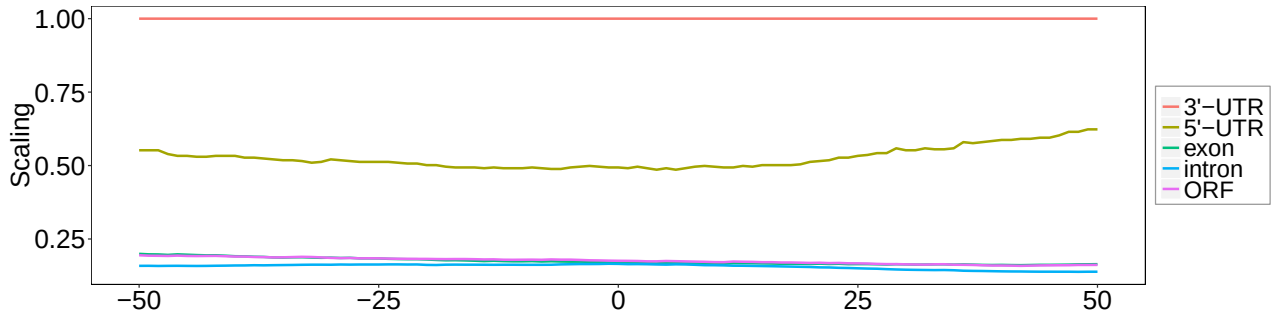

Figure S115: The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [29]TIAL1.

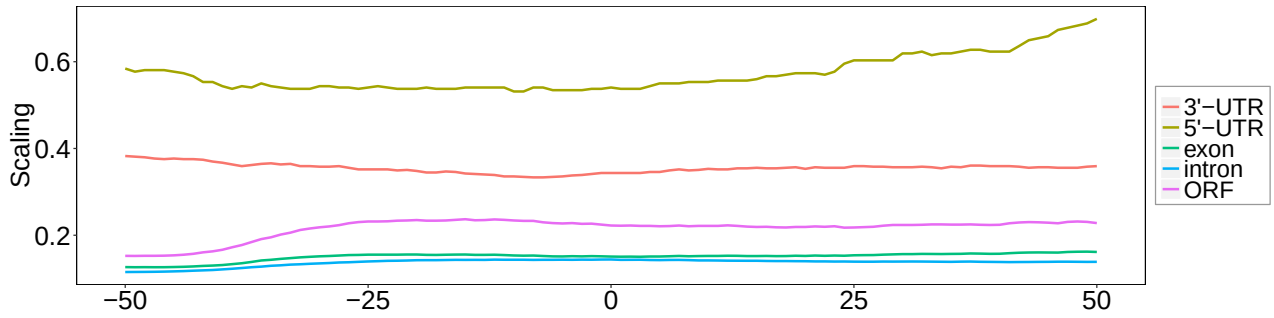

Figure S116: The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [30]U2AF2.

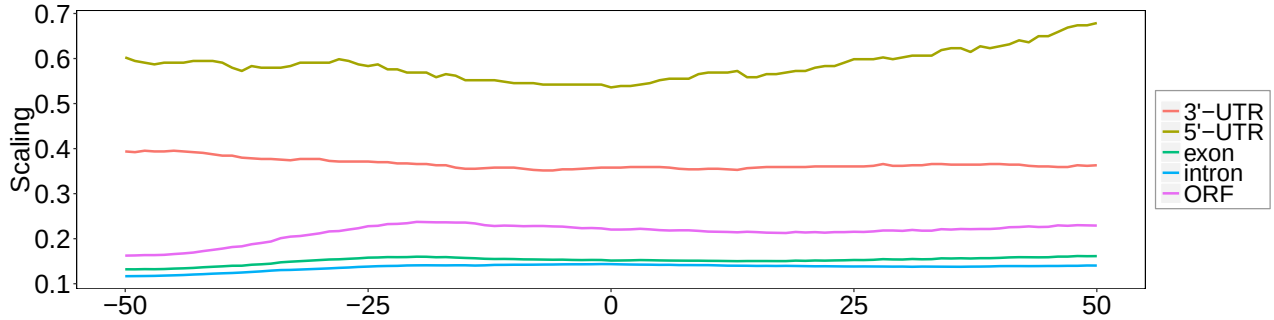

Figure S117: The features of Hetero-RP for Region type (one of the five gene regions: intron,exon,5'-UTR, 3'-UTR and ORF) in the dataset [31]U2AF2(KD).

[5] Mark A Hall and Lloyd A Smith. Feature Selection for Machine Learning: Comparing a Correlation-Based Filter Approach to the Wrapper. In *FLAIRS Conference*, volume 1999, pages 235–239, 1999.

[6] Julia Handl and Joshua Knowles. Cluster generators for large high-dimensional data sets with large numbers of clusters. *Dimension*, 2:20, 2005.

[7] John A Hartigan and PM Hartigan. The Dip Test of Unimodality. *The Annals of Statistics*, pages 70–84, 1985. [doi:10.1214/aos/1176346577].

[8] Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P Trevino, Jiliang Tang, and Huan Liu. Feature selection: A data perspective. *arXiv preprint arXiv:1601.07996*, 2016.

[9] Yang Young Lu, Ting Chen, Jed A Fuhrman, and Fengzhu Sun. COCACOLA: binning metagenomic contigs using sequence COmposition, read CoverAge, CO-alignment, and paired-end read LinkAge. *Bioinformatics*, 33(6):791–798, 2017. [PubMed:27256312] [doi:10.1093/bioinformatics/btw290].

[10] Zhao Ren, Tingni Sun, Cun-Hui Zhang, Harrison H Zhou, et al. Asymptotic normality and optimalities in estimation of large gaussian graphical models. *The Annals of Statistics*, 43(3):991–1026, 2015. [doi:10.1214/14-AOS1286].

[11] Shirish Krishnaj Shevade and S Sathiya Keerthi. A simple and efficient algorithm for gene selection using sparse logistic regression. *Bioinformatics*, 19(17):2246–2253, 2003. [PubMed:14630653] [doi:10.1093/bioinformatics/btg308].

[12] Martin Stražar, Marinka Žitnik, Blaž Zupan, Jernej Ule, and Tomaž Curk. Orthogonal matrix factorization enables integrative analysis of multiple RNA binding proteins. *Bioinformatics*, 32(10):1527–1535, 2016. [PubMed:26787667] [PubMed Central:PMC4894278] [doi:10.1093/bioinformatics/btw003].

[13] Tingni Sun and Cun-Hui Zhang. Scaled sparse linear regression. *Biometrika*, 99:879–898, 2012. [doi:10.1093/biomet/ass043].

[14] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007. [doi:10.1007/s11222-007-9033-z].

[15] Christian Wiwie, Jan Baumbach, and Richard Röttger. Comparing the performance of biomedical clustering methods. *Nature Methods*, 12(11):1033–1038, 2015. [PubMed:26389570] [doi:10.1038/nmeth.3583].

[16] Lei Yu and Huan Liu. Feature selection for high-dimensional data: A fast correlation-based filter solution. In *ICML*, volume 3, pages 856–863, 2003.