Supporting Information of
GAFSE: Towards Explicit Understanding the Modelling of Ligand Bioactivities Through Deep Graph Learning

Yueming Yin[1,2], Haifeng Hu [*1], Jitao Yang[1], Chun Ye[1], Wilson Wen Bin Goh [4,5], Adams Wai-Kin Kong[2], and Jiansheng Wu [†6,7]

[1]School of Telecommunications and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China.
[2]School of Computer Science and Engineering, Nanyang Technological University 639798, Singapore.
[4]Lee Kong Chian School of Medicine, Nanyang Technological University 637551, Singapore.
[5]Center for Biomedical Informatics, 636921, Singapore.
[6]School of Geographic and Biologic Information, Nanjing University of Posts and Telecommunications, Nanjing 210023, China.
[7]Smart Health Big Data Analysis and Location Services Engineering Research Center of Jiangsu Province, Nanjing University of Posts and Telecommunications, Nanjing 210023, China.

October 6, 2023

[*]Corresponding author: huhf@njupt.edu.cn
[†]Corresponding author: jansen@njupt.edu.cn

Table S1: Notations

Functions	
CE	Cross-entropy loss
dropout	Randomly drop part of neural network nodes in the training stage
D	Discrepancy function
elu	Exponential linear unit
E	Graph encoder
G	Graph decoder
GRU	Gated recurrent unit
L	Error function between predictions and assays
leaky_relu	Leaky rectified linear unit
$\mathcal{L}_{AFSE}$	Adversarial feature subspace enhancement loss function
$\mathcal{L}_{Bio.}$	Biological property loss function
$\mathcal{L}_{Recon.}$	Molecular reconstruction loss function
$\mathcal{L}_{Val.}$	Chemical validity loss function
N	Multi-layer perceptron
ϕ_a, ϕ_b	Mapping function of initial atomic/bond feature
relu	Rectified linear unit
softmax	Softmax activate function
Valid	Validity function
WCE	Weighted cross-entropy loss
Indices	
i^*	Index of the key atom
k_i	Index of the original symbol of atom $\mathbf{a}_i$
k^*	Index of the replaced chemical element
L	Number of atomic message passing steps ($l \in \{1, \cdots, L\}$)
N_a	Number of atoms in a molecule ($i \in \{1, \cdots, N_a\}$)
N_s	Number of element types ($k \in \{1, \cdots, N_s\}$)
T	Number of molecular massage passing steps ($t \in \{1, \cdots, T\}$)
Matrices	
$\mathbf{W}$	Weight matrices of independent one-layer neural networks, their column and row dimensions depend on their respective input and output vectors
Operators	
$[\cdot]$	Matrix indexing
$[\cdot, \cdot]$	Row concatenation
$\|\| \cdot \|\|$	l_2-norm
∇	Gradient operator
Parameters	

λ_1	Balance coefficient between the biological property loss and the AFSE loss
λ_2	Balance coefficient between the representation learning and the molecule optimization

Sets

$\mathcal{B}_i$, $\hat{\mathcal{B}}_i$, $\widetilde{\mathcal{B}}_i$	Set of $\mathbf{b}_{i,j}$, $\hat{\mathbf{b}}_{i,j}$ and $\widetilde{\mathbf{b}}_{i,j}$ for all $j \in \{1, \cdots, N_{N(i)}\}$, respectively
$\mathcal{H}$	Set of $\mathbf{h}_i$ for all $i \in \{1, \cdots, N_a\}$
$\mathcal{K}_i$	Set of original and confusing element on the i-th atom
$\mathcal{K}_i^*$	Set of candidate element on the i-th atom
$N(i)$	set of adjacent atom of the i-th atom ($j \in N(i)$)

Scalars

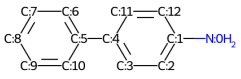
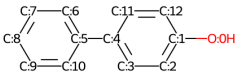
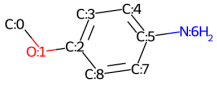
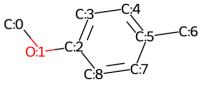
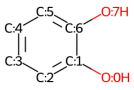
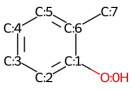
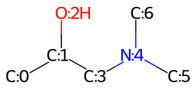
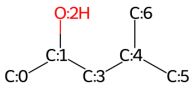
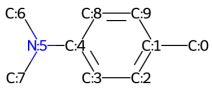
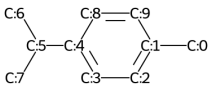
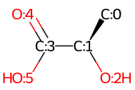
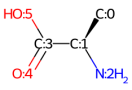
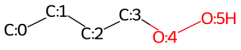
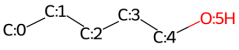
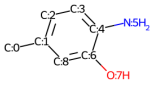
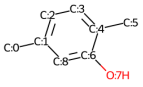
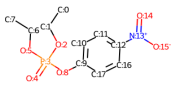
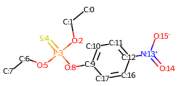
$\mathbf{a}_{i,k}$, $\hat{\mathbf{a}}_{i,k}$, $\widetilde{\mathbf{a}}_{i,k}$	True/reconstructed/optimized probability of the i-th atom belonging to the k-th element
η	Learning rate of N
γ_i	Projection of atomic embedding on molecular embedding
$P_{\mathbf{f}}(s\|a)$	The posterior probability that the atom a is predicted to be the element s According to the embedding $\mathbf{f}$, equal to $\hat{\mathbf{a}}_{i=a,k=s}$
$P_{\mathbf{f}+\mathbf{d}}(s\|a)$	The posterior probability that the atom a is predicted to be the element s according to the embedding $\mathbf{f}+\mathbf{d}$, equal to $\widetilde{\mathbf{a}}_{i=a,k=s}$
ε	Small positive rational number
$w_{N(i)}$	Attention weight between the i-th atom and its adjacent atoms
y	The experimentally determined molecular activity or property values through chemical wet experiments
$\hat{y}$	Predicted molecular bioactivities or properties

Column Vectors

$\mathbf{a}_i$, $\hat{\mathbf{a}}_i$, $\widetilde{\mathbf{a}}_i$	True/reconstructed/optimized chemical feature vector of the i-th atom
$\mathbf{b}_{i,j}$, $\hat{\mathbf{b}}_{i,j}$, $\widetilde{\mathbf{b}}_{i,j}$	True/reconstructed/optimized chemical feature vector of the bond between the i-th and the j-th atom
$\mathbf{c}$	Bias vector of independent one-layer neural networks
$\mathbf{C}_i^l$	Context feature of the i-th atom
$\mathbf{d}$	Adversarial perturbation generated by AFSE algorithm
$\mathbf{f}$	Molecular embedding
$\mathbf{f}_t$	Molecular embedding at the t-th training step
$\mathbf{g}_i$	Atomic embedding of the i-th atom for molecular reconstruction and optimization
$\mathbf{g}_i^t$, $\mathbf{g}_i^l$	$\mathbf{g}_i$ at the t-th molecular or the l-th atomic message passing step
$\mathbf{g}_{N(i)}$	Atomic embedding of the neighbor atom of the i-th atom for molecular reconstruction and optimization
$\mathbf{g}_{N(i)}^t$, $\mathbf{g}_{N(i)}^l$	$\mathbf{g}_{N(i)}$ at the t-th molecular or the l-th atomic message passing step
$\mathbf{h}_i$	Atomic embedding of the i-th atom for molecular activity and property prediction
$\mathbf{h}_i^t$, $\mathbf{h}_i^l$	$\mathbf{h}_i$ at the t-th molecular or the l-th atomic message passing step

$\mathbf{h}_{N(i)}$	Atomic embedding of the neighbor atom of the i-th atom for molecular activity and property prediction
$\mathbf{h}^t_{N(i)}$, $\mathbf{h}^l_{N(i)}$	$\mathbf{h}_{N(i)}$ at the t-th molecular or the l-th atomic message passing step
$\mathbf{r}$	Random vector
$\mathbf{r}_i$	The relationship embedding between the i-th atom and the whole molecule

Table S2: Generation of matched non-toxic molecules from toxic molecules by GAFSE.

Targets	Anchors	Anchor Properties	Optimized Molecules	Optimized Properties
NR-AhR Toxicity		Toxicity: High QED: 0.6155 SA: 1.2681 logP: 3.8570		Atom#0: N→O Toxicity: Non-Toxic QED: 0.6969 (+) SA: 1.2254 logP: 3.0592 (+)
		Toxicity: High QED: 0.5630 SA: 1.4461 logP: 2.4220		Atom#6: N→C Toxicity: Non-Toxic QED: 0.5532 SA: 1.0100 (+) logP: 2.0036
		Toxicity: High QED: 0.3762 SA: 1.8671 logP: 1.4854		Atom#7: O→C Toxicity: Non-Toxic QED: 0.5359 (++) SA: 1.4050 (+) logP: 1.7006
NR-ER Toxicity		Toxicity: High QED: 0.5285 SA: 2.7983 logP: -0.0712		Atom#4: N→C Toxicity: Non-Toxic QED: 0.5586 (+) SA: 2.6752 logP: 1.4133 (++)
		Toxicity: High QED: 0.5694 SA: 1.3958 logP: 2.0610		Atom#5: N→C Toxicity: Non-Toxic QED: 0.5533 SA: 1.2512 (+) logP: 3.1184 (−)
		Toxicity: High QED: 0.4539 SA: 2.3257 logP: -0.5482		Atom#2: O→N Toxicity: Non-Toxic QED: 0.4514 SA: 2.3196 logP: -0.5818
AMES Toxicity		Toxicity: High QED: 0.3211 SA: 2.3758 logP: 1.2761		Atom#4: N→C Toxicity: Non-Toxic QED: 0.5133 (++) SA: 1.5665 (+) logP: 1.1689
		Toxicity: High QED: 0.4030 SA: 1.8514 logP: 1.2828		Atom#5: N→C Toxicity: Non-Toxic QED: 0.5577 (++) SA: 1.5860 (+) logP: 2.0090
		Toxicity: High QED: 0.4312 SA: 2.3371 logP: 3.1547		Atom#4: O→S Toxicity: Non-Toxic QED: 0.4360 SA: 2.5031 logP: 3.2711

Algorithm S1 Generate molecular embeddings

Input: molecule $\mathbf{m}$, the step size L of the message passing of the graph neural network, the number of steps T for aggregating molecular features.

Output: Molecular embeddings $\mathbf{f}$. Initialization: $l \leftarrow 0$,$t \leftarrow 0$

1: Extract chemical features $\mathbf{a}_i$ and $\mathbf{b}_{i,j}$ for each atom and bond from $\mathbf{m}$, where i and j are atomic numbers

2: Obtain the initially hidden feature of each atom $\mathbf{a}_i$:

$$\mathbf{h}_i^0 = \mathbf{W}_0 \cdot \mathbf{a}_i + \mathbf{c}_0$$

3: Get the chemical features of the adjacent atoms $\mathbf{a}_{N(i)}$ and bonds $\mathbf{b}_{i,N(i)}$:

$$\mathbf{h}_{N(i)}^0 = \text{leaky\_relu}(\mathbf{W}_1 \cdot [\mathbf{a}_{N(i)}, \mathbf{b}_{i,N(i)}] + \mathbf{c}_1)$$

4: **while** $l < L$ **do**

5: Get the attention weight between each atom and its adjacent atoms:

$$w_{N(i)} = \text{softmax}(\text{leaky\_relu}(\mathbf{W}_2 \cdot \text{dropout}([\mathbf{h}_i^l, \mathbf{h}_{N(i)}^l]) + \mathbf{c}_2))$$

6: Get the context feature of each atom:

$$\mathbf{C}_i^l = \text{elu}(\sum_{N(i)} w_{N(i)} \cdot \mathbf{W}_3(\text{dropout}(\mathbf{h}_{N(i)}^l)) + \mathbf{c}_3)$$

7: Readout the next hidden feature of each atom:

$$\mathbf{h}_i^{l+1} = \text{relu}(\text{GRU}(\mathbf{C}_i^l, \mathbf{h}_i^l))$$

8: $l \leftarrow l + 1$

9: Update the adjacent features for each atom:

$$\mathbf{h}_{N(i)}^l = \text{leaky\_relu}(\mathbf{W}_4 \cdot \text{dropout}([\mathbf{h}_i^{l-1}, \mathbf{h}_{N(i)}^{l-1}]) + \mathbf{c}_4)$$

10: **end while**

11: Aggregate the hidden features of all atoms to get the molecular feature:

$$\mathbf{h}^L = \sum_i \mathbf{h}_i^L$$

12: Let molecules be the supernodes: $\mathbf{h}^0 \leftarrow \mathbf{h}^L$

13: Let all atoms be the adjacent nodes: $\mathbf{h}_N^0 \leftarrow [\mathbf{h}_1^L, \mathbf{h}_2^L, \cdots, \mathbf{h}_{N_a}^L]$

14: **while** $t \leq T$ **do**

15: Obtain molecular hidden features $\mathbf{h}^t$ by performing steps 5-9 with $\mathbf{W}_5$, $\mathbf{W}_6$, $\mathbf{W}_7$ and $\mathbf{b}_5$, $\mathbf{b}_6$, $\mathbf{b}_7$

16: $t \leftarrow t + 1$

17: **end while**

18: Obtain the molecular embedding $\mathbf{f} = \mathbf{h}^T$

Algorithm S2 Generate molecular graphs by AGRNs

Input: Attentive FP $\mathbf{f}$, the change of feature $\mathbf{d}$ ($\mathbf{d} = 0$ for reconstruction, and $\mathbf{d} \neq 0$ for generation), the step size L of the message passing of the graph neural network, the number of steps T for aggregating molecular features, and the hidden features of each atom $\mathbf{h}_i^L$.

Output: Chemical features $\mathbf{a}_i$ and $\mathbf{b}_{i,j}$ for each atom and bond in the molecule $\mathbf{m}$, where i and j are atoms 's number.

Initialization: $t \leftarrow T, l \leftarrow L$

Calculate the component of the molecular feature on each atom:

$$\gamma_i = \frac{\exp(\langle \mathbf{f} + \mathbf{d}, \mathbf{h}_i^L \rangle)}{\sum_j \exp(\langle \mathbf{f} + \mathbf{d}, \mathbf{h}_j^L \rangle)}$$

Obtain the hidden features of each atom used to reconstruct the molecule:

$$\mathbf{g}_i^T = \gamma_i(\mathbf{f} + \mathbf{d}) + \mathbf{h}_i^L$$

while $t \geq 1$ **do**

  Get the relationship information $\mathbf{r}_i^t$ between the molecule and its atoms:

$$\mathbf{r}_i^t = \text{elu}(\mathbf{W}_1 \cdot (\text{dropout}([\mathbf{f} + \mathbf{d}, \mathbf{g}_i^t])) + \mathbf{c}_1)$$

  Deduce the hidden feature on each atom: $\mathbf{g}_i^{t-1} = \text{relu}(\text{GRU}(\mathbf{r}_i^t, \mathbf{g}_i^t))$

  $t \leftarrow t - 1$

end while

Assigning atom features: $\mathbf{g}^L \leftarrow \mathbf{g}_i^0$

Assigning adjacent features: $\mathbf{g}_{N(i)}^L \leftarrow \mathbf{g}_{N(i)}^0$

while $l \geq 1$ **do**

  Get the attention weight between each atom and its adjacent atoms

$$w_{N(i)} = \text{softmax}(\text{leaky\_relu}(\mathbf{W}_2 \cdot \text{dropout}([\mathbf{g}_i^l, \mathbf{g}_{N(i)}^l]) + \mathbf{c}_2))$$

  Get the context feature of each atom:

$$\mathbf{C}_i^l = \text{elu}(\sum_{N(i)} w_{N(i)} \cdot \mathbf{W}_3(\text{dropout}(\mathbf{g}_{N(i)}^l)) + \mathbf{c}_3)$$

  Deduce the hidden feature on each atom: $\mathbf{g}_i^{l-1} = \text{relu}(\text{GRU}(\mathbf{C}_i^l, \mathbf{g}_i^l))$

  $l \leftarrow l - 1$

  Update the adjacent features for each atom:

$$\mathbf{g}_{N(i)}^l = \text{leaky\_relu}(\mathbf{W}_4 \cdot \text{dropout}([\mathbf{g}_i^{l+1}, \mathbf{g}_{N(i)}^{l+1}]) + \mathbf{c}_4)$$

end while

Deduce the chemical feature of each atom: $\widetilde{\mathbf{a}}_i = \phi_a(\mathbf{W}_4 \cdot \mathbf{g}_i^0 + \mathbf{c}_4)$

Deduce the chemical features of each bond:

$$\widetilde{\mathbf{b}}_{i,j} = \phi_b(\text{leaky\_relu}(\mathbf{W}_5 \cdot \text{dropout}([\mathbf{g}_i^0, \mathbf{g}_j^0]) + \mathbf{c}_5))$$

Algorithm S3 Generate MMP-Cliffs by AGRNs and GAFSE

Input: Attentive FP $\mathbf{f}$ and the hidden feature $\mathbf{h}_i^r$ of each atom $\mathbf{a}_i$ on step L.

Output: Generated chemical features of atoms and bonds $\widetilde{\mathbf{a}}_i$ and $\widetilde{\mathbf{b}}_{i,j}$, where i and j are adjacent atoms's number.

1: Generate random unit vector $\mathbf{r}_0 \in \mathbb{R}^{d_f}$ using i.i.d. Gaussian distributions, where d_f is the dimension of $\mathbf{f}$.

2: Calculate $\mathbf{d}$ via taking the gradient of D with respect to $\mathbf{r}$ on $\mathbf{f}$ at $\varepsilon\mathbf{r}_0$:

$$\mathbf{g} \leftarrow \nabla_{\mathbf{r}} D(\mathbf{W}[\mathbf{f},\mathbf{f}], \mathbf{W}[\mathbf{f}, \mathbf{f} \oplus \mathbf{r}])|_{\mathbf{r}=\varepsilon\mathbf{r}_0}$$

$$\mathbf{d} \leftarrow \eta \frac{\mathbf{g}}{\|\mathbf{g}\|_2}$$

$$D(\mathbf{W}[\mathbf{f},\mathbf{f}], \mathbf{W}[\mathbf{f}, \mathbf{f} \oplus \mathbf{r}]) \triangleq (\sigma(\frac{\mathbf{W}[\mathbf{f}, \mathbf{f}+\mathbf{r}]}{\mathbf{W}[\mathbf{f},\mathbf{f}]}) - \gamma)^2$$

$$+(\sigma(\frac{\mathbf{W}[\mathbf{f}, \mathbf{f}-\mathbf{r}]}{\mathbf{W}[\mathbf{f},\mathbf{f}]}) - \gamma)^2$$

$$\sigma(x) \triangleq \mathrm{Sigmoid}(x) = 1/(1+e^{-x})$$

$$\gamma = \mathrm{Sigmoid}(1)$$

3: Execute Algorithm S2 to generate the atom and bond features of the molecule near the activity cliff:

$$[\widetilde{\mathbf{a}}_i, \widetilde{\mathbf{b}}_{i,j}] = \mathrm{AGRNs}(\mathbf{f}+\mathbf{d})$$

Text S1: Evaluation metrics

1) Enhancement Factor (EF) is a widely used metric in virtual screening of drug hits and is of great concern to pharmacists [1]. To address the situation where the test set includes a large number of inactive compounds, the EF metric was extended in [2] as follows:

$$\begin{aligned} \mathrm{EF}_\gamma &= \frac{NTB_\gamma}{\min\{NTB_{\text{total}} \times \gamma, NTB_{\text{total}}\}}, \\ \gamma &= \frac{N_{\text{total}}}{NTB_{\text{total}}} \times \alpha,\ 0 \le \alpha \le 1, \end{aligned} \tag{1}$$

where NTB_α is the number of true binders among the top-ranked candidates (e.g., α = 10%, 20% or 30%) selected by a given model, NTB_{total} is the total number of true binders for the target protein, and N_{total} is the total number of compounds in the test dataset. When $\gamma \le 1$, the Top-γ precision is calculated, and when $\gamma > 1$, the Top-γ recall is calculated. A higher EF_α indicates better performance of the model.

2) The square of Pearson correlation coefficient (r^2) metric is adopted for evaluating the performance of activity prediction from participants during the Kaggle 2012, which is defined as

$$r^2 = \frac{\left[\sum_{i=1}^{n} (y_i - \bar{y})(\hat{y}_i - \bar{y})\right]^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2 \sum_{i=1}^{n} \left(\hat{y}_i - \bar{\hat{y}}\right)^2}, \tag{2}$$

where y_i and $\hat{y}_i$ are the true and predicted bioactivity values, respectively, and $\bar{y}$ and $\bar{\hat{y}}_i$ are the mean values of the true and predicted bioactivity, respectively. n is the total number of test samples. The larger the r^2, the higher the overall accuracy of the model prediction.

3)The Root Mean Square Error (RMSE) is a widely used metric for evaluating the performance of regression models. It is defined as the square root of the average of the squared differences between the predicted and actual values as follows:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}, \tag{3}$$

where y_i and $\hat{y}_i$ are the true and predicted bioactivity values, respectively, and n is the total number of test samples. The lower the RMSE, the better the performance of the regression model.

4)The Kendall Tau-b Coefficient (τ_B) [3] is a statistic used to measure the agreement between two rankings. It is a non-parametric measure that is widely used in scientific fields such as statistics and computational biology. The τ_B statistic is calculated by counting the number of concordant pairs and discordant pairs in the two rankings, and is defined as:

$$\tau_B = \frac{C - D}{C + D}, \tag{4}$$

where C represents the number of concordant pairs and D represents the number of discordant pairs. A concordant pair refers to two items that are ranked similarly in both rankings, while a discordant pair refers to two items that are ranked differently in the two rankings. The value

of τ_B ranges from -1 to 1, where 1 indicates complete agreement, 0 indicates no agreement, and -1 indicates complete disagreement between the two rankings. This statistic is useful in a variety of applications, including comparing the performance of different machine learning algorithms, evaluating the similarity of two data sets, and identifying the degree of association between two variables.

5) The area under the receiver operating characteristic curve (AUC). The Receiver Operating Characteristic (ROC) curve is a popular method for evaluating the performance of binary classifiers. The area under the ROC curve (AUC) is a scalar value between 0 and 1, where an AUC value of 1 indicates a perfect classifier, and an AUC value of 0.5 indicates random chance.

6) Accuracy (ACC) is the ratio of the number of correct predictions to the total number of predictions. It is defined as:

$$\text{ACC} = \frac{TP + TN}{TP + TN + FP + FN}, \tag{5}$$

where TP (True Positives) are the number of instances that are correctly classified as positive, TN (True Negatives) are the number of instances that are correctly classified as negative, FP (False Positives) are the number of instances that are incorrectly classified as positive, and FN (False Negatives) are the number of instances that are incorrectly classified as negative. This metric provides an overall measure of the accuracy of a model, but it can be misleading in cases where the class distribution is imbalanced.

7) The Matthews Correlation Coefficient (MCC) is a more robust metric that takes into account both the accuracy and the imbalance of the classes. It is defined as:

$$\text{MCC} = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}, \tag{6}$$

where positive and negative classes are equally balanced, a value of 1 indicates a perfect classifier, a value of 0 indicates a random classifier, and a value of -1 indicates a classifier that performs worse than random.

8) Specificity, also known as recall, is the ratio of the number of true positive predictions to the number of actual positive instances. It is defined as:

$$\text{Sensitivity} = \frac{TP}{TP + FN}. \tag{7}$$

Specificity is a measure of the classifier's ability to identify negative instances.

9) Sensitivity, also known as the True Positive Rate (TPR), is the ratio of the number of true negative predictions to the number of actual negative instances. It is defined as:

$$\text{Specificity} = \frac{TN}{TN + FP}. \tag{8}$$

Sensitivity is a measure of the classifier's ability to identify positive instances.

References

[1] Minyi Su, Qifan Yang, Yu Du, Guoqin Feng, Zhihai Liu, Yan Li, and Renxiao Wang. Comparative assessment of scoring functions: the casf-2016 update. *Journal of chemical information and modeling*, 59(2):895–913, 2018.

[2] Yueming Yin, Haifeng Hu, Zhen Yang, Feihu Jiang, Yihe Huang, and Jiansheng Wu. Afse: towards improving model generalization of deep graph learning of ligand bioactivities targeting gpcr proteins. *Briefings in Bioinformatics*, 23(3):bbac077, 2022.

[3] Maurice G Kendall. The treatment of ties in ranking problems. *Biometrika*, 33(3):239–251, 1945.