

# Estimating Mass of Harvested Asian Seabass *Lates calcarifer* from Images

Dmitry A. Konovalov<sup>1</sup>, Alzayat Saleh<sup>1</sup>, Jose A. Domingos<sup>2</sup>, Ronald D. White<sup>1</sup>, Dean R. Jerry<sup>1,2,3</sup>

<sup>1</sup>College of Science and Engineering, James Cook University, Townsville, Australia

<sup>2</sup>James Cook University Singapore, Singapore

<sup>3</sup>Centre for Sustainable Tropical Fisheries and Aquaculture, James Cook University, Townsville, Australia

Email: dmitry.konovalov@jcu.edu.aub

**How to cite this paper:** Konovalov, D.A., Saleh, A., Domingos, J.A., White, R.D. and Jerry, D.R. (2018) Estimating Mass of Harvested Asian Seabass *Lates calcarifer* from Images. *World Journal of Engineering and Technology*, 6, 15-23.

<https://doi.org/10.4236/wjet.2018.63B003>

**Received:** July 13, 2018

**Accepted:** August 6, 2018

**Published:** August 9, 2018

---

## Abstract

Total of 1072 Asian seabass or barramundi (*Lates calcarifer*) were harvested at two different locations in Queensland, Australia. Each fish was digitally photographed and weighed. A subsample of 200 images (100 from each location) were manually segmented to extract the fish-body area ( $S$  in  $\text{cm}^2$ ), excluding all fins. After scaling the segmented images to 1mm per pixel, the fish mass values ( $M$  in grams) were fitted by a single-factor model ( $M = aS^{1.5}$ ,  $a = 0.1695$ ) achieving the coefficient of determination ( $R^2$ ) and the Mean Absolute Relative Error ( $MARE$ ) of  $R^2 = 0.9819$  and  $MARE = 5.1\%$ , respectively. A segmentation Convolutional Neural Network (CNN) was trained on the 200 hand-segmented images, and then applied to the rest of the available images. The CNN predicted fish-body areas were used to fit the mass-area estimation models: the single-factor model,  $M = aS^{1.5}$ ,  $a = 0.170$ ,  $R^2 = 0.9819$ ,  $MARE = 5.1\%$ ; and the two-factor model,  $M = aS^b$ ,  $a = 0.124$ ,  $b = 0.155$ ,  $R^2 = 0.9834$ ,  $MARE = 4.5\%$ .

## Keywords

Aquaculture, Asian Seabass, Barramundi, *Lates calcarifer*, Computer Vision, Image Processing, Weight Estimation

---

## 1. Introduction

In aquaculture, the economic value of a particular fish species is primarily determined by its mass ( $M$ ). However, weight measurement usually involves manual handling, whilst length can easily be estimated from digital images through identifying the nose and tail of the fish. Therefore mathematical models were developed to estimate fish mass from its length ( $L$ ). For example, the length-mass power model,

$$M = aL^b, \quad (1)$$

was commonly used, where  $a$  and  $b$  were empirically-fitted species-dependent parameters [1] [2].

With the advances in image processing and the widespread availability of low-cost high-definition digital cameras, not only the length, but also other fish shape features could be collected automatically and used to estimate the mass. In particular, it was found that the fish image area ( $S$ ) could be used to estimate the fish mass ( $M$ ) via the linear model,

$$M = a + bS, \quad (2)$$

for grey mullet (*Mugil cephalus*), St. Peter's fish (*Sarotherodon galilaeus*) and common carp (*Cyprinus carpio*) [3]. The same area-mass linear model (Equation (2)) was confirmed to be more accurate than the length-mass power model (Equation (1)) for Jade perch (*Scortum barcoo*) [4], obtaining the coefficient of determination ( $R^2$ ) and the mean absolute relative error ( $MARE$ ) of  $R^2 = 0.99$  and  $MARE = 6\%$ , respectively. Even though the linear model (Equation (2)) appeared to perform better than Equation (1) [3] [4], Equation (2) is limited to the range of sufficiently large fish for any non-zero fitted parameter  $a$ . On the other hand, the area-mass power model,

$$M = aS^b, \quad (3)$$

does not exhibit the applicability limitations of Equation (2) and achieved the fit of  $R^2 = 0.99$  for Alaskan Pollock (*Theragra chalcogramma*) [5]. Furthermore, the fitted models had  $b \approx 1.5$  [5], which was consistent with the proportional relationships between the fish length ( $L \propto \sqrt{S}$ ), width ( $W \propto \sqrt{S}$ ) and height ( $H \propto \sqrt{S}$ ), and between the fish volume ( $V \propto LWH$ ) and fish mass ( $M$ ), obtaining

$$M = aS^{1.5}, \quad (4)$$

from  $M \propto LWH \propto S^{1.5}$ . For Atlantic salmon (*Salmo salar*), a similar area-mass power model was fitted as  $S \propto M^{0.61}$  (or  $M \propto aS^{1.64}$ ) with  $R^2 = 0.97$  by [6], and  $S \propto M^{0.629}$  (or  $M \propto S^{1.59}$ ) with  $R^2 = 0.998$  by [7].

Based on the preceding discussion, the first goal of this work was to establish the area-mass power model for the industrial scale harvesting of Asian seabass or barramundi (*Lates calcarifer*) in Queensland, Australia. The goal was successfully accomplished by fitting Equations (3) and (4), as displayed in **Figure 3**. The second goal of this study was to design a practical image-processing method to extract fish-body area while excluding the fins for enhanced accuracy and also for possible applications in industrial-scale modern selective breeding programs [8] [9]. That goal was achieved by training a segmentation neural network in Section 2.2.

## 2. Materials and Methods

### 2.1. Datasets

Two datasets were used in this study. The first was the Barra-Ruler-445 (BR445)

dataset used in [10] [11], and publically available via [12] originated from the [9] study. The second dataset was the Barra-Area-600 (BA600) dataset and released to public domain on publication of this work via [13]. In both datasets, each harvested barramundi fish (Asian seabass, *Lates calcarifer*) was digitally photographed and its weight was measured and recorded against the image file name. All images had a millimeter-graded ruler placed next to the fish, see **Figure 1** for examples. The weights ranged 0.2 kg - 1 kg in BR445, and 1 kg - 2.5 kg in BA600. The image scales (in millimeters per pixel) were determined manually by measuring the number of pixels between the end points of the 300 mm ruler present in each image. The BR445 image scales were checked by the automatic ruler-scaling (RS2) algorithm [11]. The BA600 images were taken from the same distance hence they had the same scale.

## 2.2. Automatic Fish-Body Segmentation

The fins of the fish can contribute significantly to the total fish image area, see typical examples in **Figure 1**. At the same time the fins' contribution to the fish mass is negligible. Therefore, ideally, only the fish-body area should be used to estimate the fish mass. For example, using the fish area without considering the fin tail was found to be more accurate when predicting the mass of Jade perch *Scortum barcoo* [4]. Furthermore, the fins are highly flexible and are more likely to change shape during harvesting, or be damaged and/or erode during the production growth cycle.

Segmentation of 200 images (100 from each dataset) into fish-body and background was done manually using the GIMP open-source software program. The

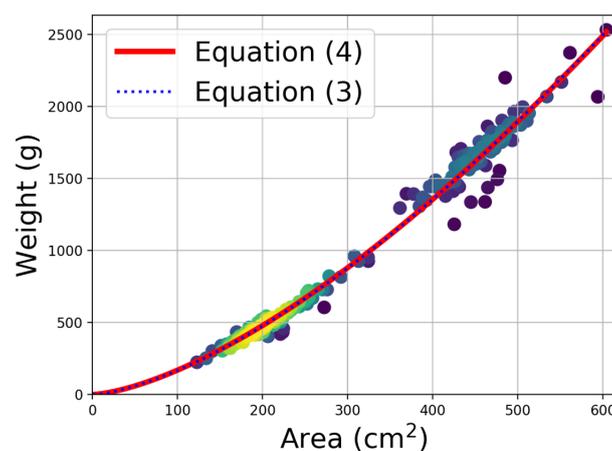


**Figure 1.** Examples of images from the BR445 (left column) and BA600 (right column) datasets.

resulting fish-body binary masks were individually scaled to have the same scale of 1 mm per pixel. In this study all custom computer programs were written in Python programming language, which was also used to calculate the fish-body pixel areas. The obtained fish areas and the corresponding measured mass values were fitted via Equation (4) and results displayed in **Figure 2**. The fit achieved highly accurate  $R^2 = 0.9819$ , and  $MARE = 5.1\%$ , which were comparable to the corresponding results obtained on other fish species [4] [5] [6] [7]. **Figure 2** clearly illustrated how the weight of the harvested Asian seabass *Lates calcarifer* could be estimated from the fish area with high accuracy. However, before such estimation method could be deployed in the aquaculture production environment, a robust automatic body-area extraction algorithm would be required, which was the focus for the rest of this section.

The recently developed semantic-segmentation Convolutional Neural Networks (CNN) [14] were highly successful in solving challenges where the segmentation of an image into per-pixel classes was required [11] [14] [15]. As discussed in the introduction, the second primary goal of this study was to design a practical Computer Vision algorithm to extract fish-body area from images. The Deep Learning neural networks [16] have revolutionized modern Machine Learning including the field of Computer Vision, and a large number of segmentation Deep Learning CNN models have been proposed. Comparing even the most popular segmentation CNN models was outside the scope of this work. Instead, the most accurate Fully Convolutional Network from [14], FCN-8s, was used. FCN-8s could be viewed as the modern baseline segmentation CNN model due to its highest citation rate out of all available segmentation CNNs (more than 4000 Google Scholar citations at the time of writing).

The FCN-8s model was implemented [17] in Python utilizing the high-level neural networks Application Programming Interface (API) Keras [18] together



**Figure 2.** Relation between the measured fish weight ( $M$  in g) and the segmented-by-hand fish-body image area ( $S$  in  $\text{cm}^2$ ) fitted by: Equation (4) as  $M = 0.1695 \times S^{1.5}$ ,  $R^2 = 0.9819$ ,  $MARE = 5.13\%$ ; and Equation (3) as  $M = 0.1622 \times S^{1.5073}$ ,  $R^2 = 0.9819$ ,  $MARE = 5.06\%$ . Higher density of data points were denoted by lighter color.

with the machine-learning Python package TensorFlow [19]. The FCN-8s model is a general features-to-segmentation *decoder* CNN, which required an image-to-features CNN *encoder*. The original FCN-8s [14] was built with the VGG16 [20] convolutional layers as the encoder. The VGG16 model within Keras was trained to recognize 1000 different ImageNet [21] object classes and commonly referred to as ImageNet-trained. The ImageNet-trained CNN models were often more accurate than randomly initialized CNN models when they were further trained to recognize new object classes [22]. Therefore the convolutional layers of the ImageNet-trained VGG16 model were used to build our version of the FCN-8s model referred at the Fish Area Segmentation (FAS) model hereafter.

The FAS model was loaded with the relevant VGG16 weights facilitating the knowledge transfer [22], where the remaining convolutional as well as de-convolutional FCN-8s layers were initialized by the uniform distribution as per [23]. Furthermore, the first two FCN-8s decoder layers had their number of neurons reduced to 512 comparing to the 4096 neurons of the original FCN-8s in [14]. Such drastic reduction was justified by the requirement to recognize and segment only the single class of objects, *i.e.* fish body. The *sigmoid* activation function was used in the last layer.

The described 200 images together with the corresponding hand-segmented body masks were used to train the FAS. The 200 image-mask pairs were randomly split 80% - 20%, where the 80% of pairs were used as the actual training set and the remaining 20% were used as the validation set to assess the training process. Since the training set had such small number of images, the encoding VGG16 layers in FAS were fixed and excluded from training. The remaining trainable weights (excluding biases) were regularized by a weight decay set to  $1 \times 10^{-4}$ . The training and validation images as well as the masks were rescaled to 1mm per pixel. Then each image-mask pair was extensively augmented for each epoch of training, *i.e.* one pass through all available training and validation images. Specifically, the python-opencv package was used to perform augmentations, where each image and if applicable the corresponding binary mask were:

- randomly rotated in the range of  $[-180, +180]$  degrees;
- randomly scaled vertically in the range of  $[0.8, 1]$  and independently horizontally within the same range;
- randomly cropped to retain  $480 \times 480$  pixels;
- each color channel was  $\pm 12.5$  range randomly shifted;
- randomly flipped horizontally and vertically;
- ImageNet color mean values were subtracted as required when working with the VGG16 model.

To assist better segmentation, the following loss function was adopted,

$$\text{loss}(Y_{gt}, Y_{pred}) = 1 - \text{dice}(Y_{gt}, Y_{pred}) + bc(Y_{gt}, Y_{pred}), \quad (5)$$

where:  $Y_{pred}$  and  $Y_{gt}$  were the predicted and ground truth (*i.e.* segmented-by-hand)  $480 \times 480$  masks;  $bc(Y_{gt}, Y_{pred})$  was the standard binary

cross-entropy; and where  $dice(Y_{gt}, Y_{pred})$  was the Dice coefficient [24] ranged between zero and 1 (for identical  $Y_{pred}$  and  $Y_{gt}$ ). Since the *sigmoid* function was used as the last activation, the per-pixel predictions  $Y_{pred}$  ranged between 0 and 1. The ground-truth  $Y_{gt}$  was per-pixel encoded as zeros for the background pixels and ones for the body pixels. The training and validation losses were averaged over all pixels and all corresponding images obtaining the total training and validation losses for each epoch.

Keras implementation of Adam [25] was used as the training optimizer. The Adam learning-rate ( $lr$ ) was set to  $lr = 0.001$ , where the rate was halved every time the total epoch validation loss did not decrease after 16 epochs. The training was done in batches of 8 images, and was aborted if the validation loss did not decrease after 32 epochs, where the validation loss was calculated from the validation set of images and masks, which were not used by the optimizer for training the FAS model. While training, the FAS model with smallest running validation loss was continuously saved. Furthermore, if the training was aborted, it was restarted (from the previously saved FAS model) two more times with the initial learning rates  $lr = 0.5 \times 10^{-3}$  and  $lr = 0.25 \times 10^{-3}$ , respectively. Note that both the validation images were also augmented by the preceding augmentation pre-processing steps in order to prevent the indirect fitting of the validation images.

### 3. Results and Discussion

Multiple training sessions with different random train/validation split produced very similar results. The FAS model and its training procedure exhibited negligible over-fitting as demonstrated by the comparable final training and validation loss values (mean of Equation (5)) of  $0.063 \pm 0.001$  and  $0.072 \pm 0.003$ , respectively. The training and validation per-pixel accuracies were  $0.9945 \pm 0.0005$  and  $0.9935 \pm 0.0005$ , respectively. The trained FAS model was applied to all available (scaled to 1mm per pixel) images including the 200 images used for training. By its design FAS could be applied to images of any size. However in practice, it was significantly faster to pad available images by zero values to fill the fixed  $640 \times 640$  shape and then feed them into FAS for prediction, where the  $640 \times 640$  square was large enough to fit all available scaled images. For each image, the prediction *heat-map* of [0, 1] range pixel values were further processed by setting values above 0.51 to ones (*i.e.* predicted as the body pixels) and the rest to zeros (*i.e.* the background pixels). The largest connected non-zero region in each image was accepted as the final fish body segmentation, and its area in pixel<sup>2</sup> (*i.e.* mm<sup>2</sup>) was calculated. Overlapping fish and/or multiple fish per image were outside the scope of this work.

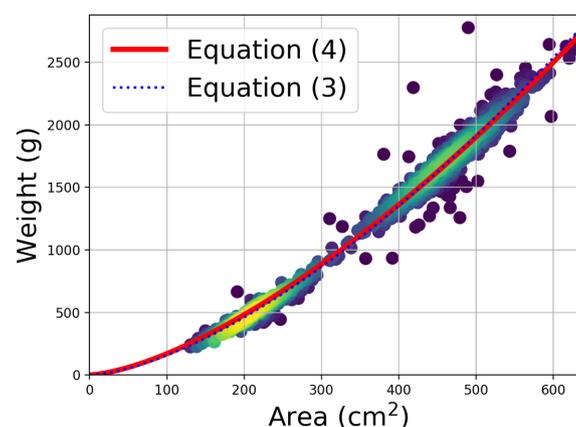
It took 2 - 3 hours to train FAS on Nvidia GTX 1080Ti GPU. However, once trained the FAS model was fast enough to process  $640 \times 640$  images at a rate of 30 images per second on the same GPU, and therefore it could even be deployed in the aquaculture production processing video feed in real time. All predicted

areas were plotted against the measured weights in **Figure 3**. The results were fitted by Equations (3) and (4) to minimize the mean squared error (MSE) between the predicted and measured weights. Quite a few points (**Figure 3**) could be viewed as outliers, *e.g.* due to human errors in the recorded weights, or due to fish having an expected odd shape due to malnourishment, disease or deformity. When the automatic image scaling method [11] was applied to the BR445 set, in the order of 1% human errors were found and corrected. Therefore it was feasible to assume that the comparable human error rate of 1% could be present in the weights values, which unfortunately could not be checked or corrected due to the fish having been sold. Therefore an important practical quality assurance recommendation naturally follows: if possible, the digital weight display should be visible in the same image together with the measuring ruler.

The difference in the Equations (3) and (4) fitting results (**Figure 3**) was open for interpretation. A better fit does not necessarily yield better predictive accuracy on future unseen samples; see detailed discussion in [26]. Therefore, Equation (4) was arguably more robust to errors since it has only one fitting parameter. Furthermore, the stability of Equation (4) was confirmed by its application to the training set of hand-segmented images (**Figure 2**) and to more than 1000 automatically segmented images (**Figure 3**), yielding essentially identical results of  $M = 0.1695 \times S^{1.5}$  and  $M = 0.170 \times S^{1.5}$ , respectively.

#### 4. Conclusion

The trained on 200 images Segmentation Convolutional Neural Network was used to automatically segment fish-body from background in all of this study's 1072 digital images of Asian seabass (barramundi, *Lates calcarifer*). The automatically extracted fish-body areas and the corresponding manually measured weights were fitted to yield highly accurate single- and two-factor mass-from-



**Figure 3.** Relation between the measured fish weight ( $M$  in g) and the automatically segmented fish-body image area ( $S$  in  $\text{cm}^2$ ) fitted as red line by Equation (4),  $M = 0.1702 \times S^{1.5}$ ,  $R^2 = 0.9828$ ,  $MARE = 5.58\%$ . Dotted line is the fitted Equation (3),  $M = 0.1239 \times S^{1.55}$ ,  $R^2 = 0.9834$ ,  $MARE = 4.53\%$ . Lighter color denoted higher density of the area-weight data points.

area estimation models, see **Figure 3**. The presented automatic segmentation approach together with the previously reported automatic scaling of fish images method [11] could potentially reduce cost and time of fish mass-estimation on industrial scale.

### Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

### References

- [1] Huxley, J.S. (1924) Constant Differential Growth-Ratios and Their Significance. *Nature*, **114**, 895-896. <https://doi.org/10.1038/114895a0>
- [2] Zion, B. (2012) The Use of Computer Vision Technologies in Aquaculture—A Review. *Computers and Electronics in Agriculture*, **88**, 125-132. <https://doi.org/10.1016/j.compag.2012.07.010>
- [3] Zion, B., Shklyar, A. and Karplus, I. (1999) Sorting Fish by Computer Vision. *Computers and Electronics in Agriculture*, **23**, 175-187. [https://doi.org/10.1016/S0168-1699\(99\)00030-7](https://doi.org/10.1016/S0168-1699(99)00030-7)
- [4] Viazzi, S., Van Hoestenbergh, S., Goddeeris, B.M. and Berckmans, D. (2015) Automatic Mass Estimation of Jade Perch *Scortum Barcoo* by Computer Vision. *Aquacultural Engineering*, **64**, 42-48. <https://doi.org/10.1016/j.aquaeng.2014.11.003>
- [5] Balaban, M.O., Chombeau, M., Cirban, D. and Gumus, B. (2010) Prediction of the Weight of Alaskan Pollock Using Image Analysis. *Journal of Food Science*, **75**, E552-E556. <https://doi.org/10.1111/j.1750-3841.2010.01813.x>
- [6] Frederick, C., Brady, D.C. and Bricknell, I. (2017) Landing Strips: Model Development for Estimating Body Surface Area of Farmed Atlantic Salmon (*Salmo salar*). *Aquaculture*, **473**, 299-302. <https://doi.org/10.1016/j.aquaculture.2017.02.026>
- [7] Jaworski, A. and Wolm, J. (1992) Distribution and Structure of the Population of Sea Lice, *Lepeophtheirus Salmonis* Krøyer, on Atlantic Salmon, *Salmo salar* L., under Typical Rearing Conditions. *Aquaculture and Fisheries Management*, **23**, 577-589. <https://doi.org/10.1111/j.1365-2109.1992.tb00802.x>
- [8] Zenger, K.R., Khatkar, M.S., Jerry, D.R. and Raadsma, H.W. (2017) The Next Wave in Selective Breeding: Implementing Genomic Selection in Aquaculture. *Proceedings of the 22nd Conference of the Association for the Advancement of Animal Breeding and Genetics*, **22**, 105-112, In Press.
- [9] Domingos, J.A., Smith-Keune, C. and Jerry, D.R. (2014) Fate of Genetic Diversity Within and Between Generations and Implications for DNA Parentage Analysis in Selective Breeding of Mass Spawners: A Case Study of Commercially Farmed Barramundi, *Lates calcarifer*. *Aquaculture*, **424-425**, 174-182. <https://doi.org/10.1016/j.aquaculture.2014.01.004>
- [10] Konovalov, D.A., Domingos, J.A., Bajema, C., White, R.D. and Jerry, D.R. (2017) Ruler Detection for Automatic Scaling of Fish Images. *Proceedings of the International Conference on Advances in Image Processing*, Bangkok, ACM, New York, 90-95. <https://doi.org/10.1145/3133264.3133271>
- [11] Konovalov, D.A., Domingos, J.A., White, R.D. and Jerry, D.R. (2018) Automatic Scaling of Fish Images, *Proceedings of the 2nd International Conference on Advances in Image Processing*, Chengdu, ACM, New York, In Press.

- <https://doi.org/10.13140/RG.2.2.35572.86406>
- [12] Konovalov, D.A., Domingos, J.A. and Jerry, D.R. (2017) Barra-Ruler-445 (BR445) Dataset. <https://github.com/dmitryako/BarraRulerDataset445>
- [13] Konovalov, D.A., Domingos, J.A. and Jerry, D.R. (2018) Barra-Area-600 (BA600) Dataset. <https://github.com/dmitryako/BarraAreaDataset600>
- [14] Shelhamer, E., Long, J. and Darrell, T. (2017) Fully Convolutional Networks for Semantic Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **39**, 640-651. <https://doi.org/10.1109/TPAMI.2016.2572683>
- [15] Konovalov, D.A., Hillcoat, S., Williams, G., Birtles, R.A., Gardiner, N. and Curnock, M.I. (2018) Individual Minke Whale Recognition Using Deep Learning Convolutional Neural Networks. *Proceedings of the International Conference on Ocean Science and Technology (COST2018)*, Chengdu, 1-3 June 2018, In Press. <https://doi.org/10.13140/RG.2.2.12923.62245>
- [16] LeCun, Y., Bengio, Y. and Hinton, G. (2015) Deep Learning. *Nature*, **521**, 436-444. <https://doi.org/10.1038/nature14539>
- [17] Konovalov, D.A. (2018) Keras-TensorFlow Implementation of FCN-8s. [https://github.com/dmitryako/keras\\_fcn\\_8s](https://github.com/dmitryako/keras_fcn_8s)
- [18] Chollet, F., et al. (2015) Keras. <https://github.com/fchollet/keras>
- [19] Abadi, M., et al. (2015) TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. <http://tensorflow.org>
- [20] Simonyan, K. and Zisserman, A. (2015) Very Deep Convolutional Networks for Large-Scale Image Recognition. *The 3rd International Conference on Learning Representations (ICLR2015)*. <https://arxiv.org/abs/1409.1556>
- [21] Russakovsky, O., et al. (2015) ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, **115**, 211-252. <https://doi.org/10.1007/s11263-015-0816-y>
- [22] Oquab, M., Bottou, L., Laptev, I. and Sivic, J. (2014) Learning and Transferring Mid-Level Image Representations Using Convolutional Neural Networks. *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Columbus, 1717-1724. <https://doi.org/10.1109/CVPR.2014.222>
- [23] Glorot, X. and Bengio, Y. (2010) Understanding the Difficulty of Training Deep Feedforward Neural Networks. *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research*, **9**, 249-256. <http://proceedings.mlr.press/v9/glorot10a.html>
- [24] Dice, L.R. (1945) Measures of the Amount of Ecologic Association between Species. *Ecology*, **26**, 297-302. <https://doi.org/10.2307/1932409>
- [25] Kingma, D.P. and Ba, J. (2015) Adam: a Method for Stochastic Optimization. *The 3rd International Conference for Learning Representations*, San Diego. <http://arxiv.org/abs/1412.6980>
- [26] Konovalov, D.A., Llewellyn, L.E., Vander Heyden, Y. and Coomans, D. (2008) Robust Cross-Validation of Linear Regression QSAR Models. *Journal of Chemical Information and Modeling*, **48**, 2081-2094. <https://doi.org/10.1021/ci800209k>