PLOS ONE

RESEARCH ARTICLE

Protein transfer learning improves identification of heat shock protein families

Seonwoo Min[1], **HyunGi Kim**[1], **Byunghan Lee**[2]*, **Sungroh Yoon**[1,3]*

1 Department of Electrical and Computer Engineering, Seoul National University, Seoul, South Korea, **2** Department of Electronic and IT Media Engineering, Seoul National University of Science and Technology, Seoul, South Korea, **3** Department of Biological Sciences, Interdisciplinary Program in Bioinformatics, Interdisciplinary Program in Artificial Intelligence, ASRI, INMC, and Institute of Engineering Research, Seoul National University, Seoul, South Korea

* bhlee@seoultech.ac.kr (BL); sryoon@snu.ac.kr (SY)

OPEN ACCESS

Citation: Min S, Kim H, Lee B, Yoon S (2021) Protein transfer learning improves identification of heat shock protein families. PLoS ONE 16(5): e0251865. https://doi.org/10.1371/journal.pone.0251865

Editor: Thippa Reddy Gadekallu, Vellore Institute of Technology: VIT University, INDIA

Received: March 26, 2021

Accepted: May 4, 2021

Published: May 18, 2021

Data Availability Statement: All the data are available at our repository (https://github.com/mswzeus/DeeperHSP).

Funding: This research was supported by the National Research Foundation (NRF) of Korea grants funded by the Ministry of Science and ICT (2018R1A2B3001628 (S.Y.), 2014M3C9A3063541 (S.Y.), 2019R1G1A1003253 (B.L.)), the Ministry of Agriculture, Food and Rural Affairs (918013-4 (S.Y.)), and the Brain Korea 21 Plus Project in 2021 (S.Y.). The funders had no role in study design,

Abstract

Heat shock proteins (HSPs) play a pivotal role as molecular chaperones against unfavorable conditions. Although HSPs are of great importance, their computational identification remains a significant challenge. Previous studies have two major limitations. First, they relied heavily on amino acid composition features, which inevitably limited their prediction performance. Second, their prediction performance was overestimated because of the independent two-stage evaluations and train-test data redundancy. To overcome these limitations, we introduce two novel deep learning algorithms: (1) time-efficient DeepHSP and (2) high-performance DeeperHSP. We propose a convolutional neural network (CNN)-based DeepHSP that classifies both non-HSPs and six HSP families simultaneously. It outperforms state-of-the-art algorithms, despite taking 14–15 times less time for both training and inference. We further improve the performance of DeepHSP by taking advantage of protein transfer learning. While DeepHSP is trained on raw protein sequences, DeeperHSP is trained on top of pre-trained protein representations. Therefore, DeeperHSP remarkably outperforms state-of-the-art algorithms increasing F1 scores in both cross-validation and independent test experiments by 20% and 10%, respectively. We envision that the proposed algorithms can provide a proteome-wide prediction of HSPs and help in various downstream analyses for pathology and clinical research.

Introduction

Heat shock proteins (HSPs) are stress-induced proteins that are highly conserved across organisms ranging from bacteria to humans [1]. HSPs participate in several cellular processes such as intercellular transportation and signal pathway modulation. Most importantly, HSPs play a pivotal role as molecular chaperones against unfavorable conditions, such as elevated temperature and inflammation [2]. They prevent irreversible aggregation of denatured proteins and assist protein folding for functional conformation. Because the dysfunction of HSPs may lead to fatal illness (*e.g.*, neurodegenerative disorders, cardiovascular diseases, and

data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

cancers), their identification has been an important problem in pathology and clinical research [3].

According to core functions and molecular weights [4], HSPs can be categorized into six major families: HSP20 (small HSPs), HSP40 (DnaJ proteins), HSP60 (GroEL proteins), HSP70 (DnaK proteins), HSP90 (HptG proteins), and HSP100 (Clp proteins). Traditional methods rely on nuclear magnetic resonance spectroscopy to identify HSP families [5]. However, as an exponential number of proteins are becoming available, time-consuming and resource-intensive processes of experimental annotation have become a serious disadvantage.

Therefore, numerous computational methods have been developed to identify HSP families (Table 1). They commonly used a support vector machine (SVM) classifier trained on a variety of sequence composition features, *e.g.*, pseudo amino acid composition (PAAC), dipeptide composition (DPC), and spaced-DPC (SDPC). While early methods [6, 7] only focused on classifying HSP sequences into one of the six HSP families, PredHSP [8] and ir-HSP [9] proposed two-stage algorithms to cope with non-HSP input sequences as well. They are based on sequence composition feature extraction, followed by two trained SVM classifiers. In the first stage, they used an SVM model to discriminate HSP sequences from non-HSP sequences. In the second stage, they used another SVM model to classify those predicted as HSPs into one of the six families. The main difference between the algorithms lies in the type of extracted features.

Previous studies have provided high-throughput methods for identifying HSP families. However, they had two major limitations. First, they relied heavily on the sequence composition features, focusing only on dipeptide statistics. Because they cannot capture more complex high-level information, it limited the performance of the previous algorithms. Second, their prediction performance was overestimated owing to biased experiments. During cross-validation, the second SVM was evaluated independently without considering the first SVM. This resulted in a higher number of true positives, although some of them already have been misclassified as non-HSPs in the first stage. Moreover, during additional tests, it was not ensured that the additional datasets were independent. We found that there are numerous sequences similar to those in the training dataset. The data redundancy also inevitably caused overrated evaluations.

With the advancement of deep learning, several studies have proposed deep learning models for bioinformatics [10]. As conventional machine learning models heavily rely on extracted features, machine learning researchers often focus on designing effective features for various tasks [11–13]. In contrast, deep learning models eliminate the laborious feature engineering and use deep neural networks to learn hierarchical representations from data. They showed that deep learning models, trained with a substantial amount of labeled data, can achieve state-of-the-art performance in various problems such as CRISPR activity and microRNA target prediction. [14, 15].

Transfer learning is an important cornerstone of deep learning. For example, in natural language processing, word representations are pre-trained using a huge amount of unlabeled text [16, 17]. The learned information can be transferred to a wide range of tasks by training task-

Table 1. Summary of related works.

Method	Feature	Model	Non-HSP	HSP Families
iHSP-PseRAAC [6]	PAAC	SVM	X	O
Ahmad *et al.* [7]	DPC	SVM	X	O
PredHSP [8]	DPC	SVM	O	O
ir-HSP [9]	SDPC	SVM	O	O

https://doi.org/10.1371/journal.pone.0251865.t001

Table 2. Pre-trained protein language models.

	Model	Parameter (M)	Dimensions	Proteins (M)
UniRep	RNN	18	1,900	24
PLUS-RNN	RNN	59	2,048	15
SeqVec	RNN	94	1,024	33
ProtXLNet	TFM	409	1,024	216
ProtBERT	TFM	421	1,024	2,122
ESM	TFM	669	1,280	27

https://doi.org/10.1371/journal.pone.0251865.t002

specific models on top of the pre-trained word representations. The crux of transfer learning is how to pre-train representations. Several studies have proposed language model (LM)-based approaches that can exploit unlabeled data [16, 17]. Given a sentence, they train an LM such that a randomly masked word is predicted from representations of other words.

Similarly, a variety of studies have proposed LM-based approaches for protein transfer learning [18–22]. As evolutionary pressure constrains naturally occurring proteins to maintain indispensable functions, they could obtain implicit information underlying protein sequences even without any experimental annotations. Taking advantage of a large number of unlabeled protein sequences, it was demonstrated that pre-trained protein representations convey biochemical, structural, and evolutionary information. Therefore, pre-trained representations can help improve model performance in various protein biology tasks [23].

The key differences among the previous studies originate from two factors: (1) LM architecture and (2) the number of proteins used for pre-training (Table 2). In terms of the LM architecture, UniRep, PLUS-RNN, and SeqVec use recurrent neural networks (RNNs); ProtXLNet, ProtBERT, and ESM use transformers (TFMs). RNN-based models require less resources for both pre-training and producing representations. Although TFM-based models require significantly more resources, they are better at capturing long-term dependencies within proteins and can provide more informative representations [24]. The number of unlabeled proteins used in each study varied considerably. The LMs with more parameters were usually pre-trained with a larger number of proteins. Exceptionally, while ESM used the largest protein LM, it was pre-trained with a relatively small number of proteins. This can be attributed to its high-diversity dataset, which contains only representative proteins from clusters based on sequence identity [22].

In this work, we introduce two novel deep learning algorithms for the identification of HSP families. First, we propose time-efficient DeepHSP based on a convolutional neural network (CNN). It leverages (1) the representation learning capability of deep learning and (2) a one-stage algorithm trained to classify both non-HSPs and the six HSP families simultaneously. It outperforms state-of-the-art algorithms, despite taking 14–15 times less time for both training and inference. We further improve DeepHSP by taking advantage of protein transfer learning. We train the CNN model on top of pre-trained protein representations instead of the raw sequences used for DeepHSP. We denote the resulting model as DeeperHSP considering that the representations are obtained from a pre-trained deep neural network. We demonstrate that high-performance DeeperHSP remarkably outperforms state-of-the-art algorithms in both cross-validation and independent test experiments, increasing F1 score by 20% and 10%, respectively.

In summary, the contributions of our paper are as follows:

- We introduce time-efficient DeepHSP and high-performance DeeperHSP for the computational identification of HSP families.

- DeepHSP outperforms state-of-the-art algorithms, despite taking 14–15 times less time for both training and inference.
- Incorporating pre-trained protein representations and a CNN model, DeeperHSP remarkably outperforms state-of-the-art algorithms both in cross-validation and independent test experiments, increasing the F1 scores by 20% and 10%, respectively.
- All the data, codes, and pre-trained models are available at https://github.com/mswzeus/DeeperHSP.

Materials and methods

DeepHSP

We propose time-efficient DeepHSP which categorizes a protein sequence into seven classes: non-HSP and the six major HSP families (Fig 1). Hereafter, we explain the CNN-based model step-by-step with an input protein sequence of variable-length L denoted as

$$S = (s_1, \ldots, s_L), \quad s_i \in \{20 \text{ standard amino acids}\}.$$

Given a protein sequence S, DeepHSP first uses one-hot encoding to convert it into $\mathbf{X} \in \mathbb{R}^{L \times 20}$, a sequence of 20-dimensional vectors:

$$\mathbf{X} = \langle \mathbf{x}_1, \ldots, \mathbf{x}_L \rangle, \quad \mathbf{x}_i = \text{OneHot}(s_i),$$

such that all the elements in $\mathbf{x}_i$ are set to zero, except the element corresponding to s_i, that is set to one.

Subsequently, the convolution and max-pooling layers compute hidden representations, $\mathbf{H} \in \mathbb{R}^{500}$, from the encoded input matrix:

$$\mathbf{H} = \text{MaxPool}(\text{Conv}(\mathbf{X})).$$

The convolution layer uses $d_c = 500$ filters of length $l_c = 5$ followed by a rectified linear unit activation function. The filters can be regarded as position-weighted matrices similar to those used in traditional analyses [10]. They are convolved along protein sequences and trained to identify discriminative motifs. The global max-pooling layer computes the maximum value of

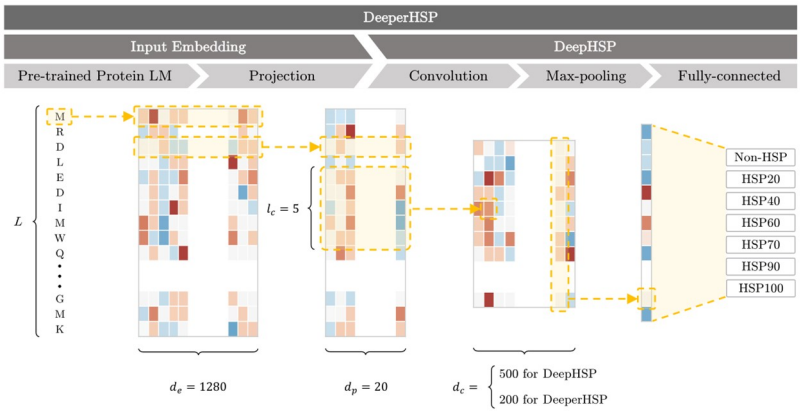

Fig 1. Overview of DeepHSP and DeeperHSP.

https://doi.org/10.1371/journal.pone.0251865.g001

the output of each filter. This helps us to obtain a fixed-length representation vector from the variable-length input sequence.

Finally, the fully-connected (FC) layer computes the outputs **P** from the representations:

$$\mathbf{P} = \mathrm{FC}(\mathbf{H}), \quad \mathbf{P} = (p_1, \ldots, p_7)$$

where p_c denotes the probability of each class c of the input sequence, and $\sum_{c=1}^{7} p_c = 1$. The FC layer uses dropout regularization and a softmax activation function. The former randomly zeroes some of the input vectors to help avoid overfitting. The latter normalizes the output vector so it can be interpreted as a probability distribution.

DeeperHSP

The main limitation of DeepHSP originates from the one-hot encoding. It can only identify the amino acid in each position and cannot provide any other information. To tackle this problem, we propose high-performance DeeperHSP, which takes advantage of protein transfer learning. DeeperHSP embeds an input sequence using a pre-trained protein LM and a projection layer (Fig 1).

Given a protein sequence S, DeeperHSP uses a pre-trained protein LM to convert it into a sequence of d_e-dimensional vectors, $\mathbf{E} \in \mathbb{R}^{L \times d_e}$:

$$\mathbf{E} = \langle \mathbf{e}_1, \ldots, \mathbf{e}_L \rangle, \quad \mathbf{E} = \mathrm{ProteinLM}(S).$$

In contrast to one-hot encoding, which independently converts each amino acid, the protein LM computes representations as a function of the entire sequence. By leveraging a large number of unlabeled proteins through pre-training, they provide biochemical, structural, and evolutionary information that can help us to identify HSP families. Among a variety of pre-trained protein LMs, we used the largest ESM [22], which produces vectors of dimension d_e = 1280. The effects of different protein LMs are presented in the ablation studies.

The size of the representations, d_e, is more than a couple of thousand dimensions. This may significantly increase the number of parameters in the following model. Therefore, DeeperHSP uses a projection layer to further embed **E** into vectors $\mathbf{Z} \in \mathbb{R}^{L \times d_p}$, where d_p = 20:

$$\mathbf{Z} = \langle \mathbf{z}_1, \ldots, \mathbf{z}_L \rangle, \quad \mathbf{z}_i = \mathrm{Proj}(\mathbf{e}_i),$$

where it independently embeds $\mathbf{e}_i$ into $\mathbf{z}_i$ with shared weights across different positions. The projection layer minimizes the additional number of parameters required for DeeperHSP.

Finally, DeeperHSP uses a CNN on top of the embedded input matrix. It utilizes the same CNN architecture as DeepHSP except that (1) its input X is replaced with Z and (2) its convolution layer uses d_c = 200 filters. The latter is to make the number of parameters of DeeperHSP (47K) similar to that of DeepHSP (54K). It helps us to clearly examine the effectiveness of the pre-trained representations used for DeeperHSP.

Training of DeepHSP and DeeperHSP

For the training of both DeepHSP and DeeperHSP, we use class-weighted cross-entropy objective function defined as

$$\mathcal{L} = -\sum_{c=1}^{7} w_c \cdot y_c \log(p_c),$$

$$fw_c = \sqrt{\frac{\max(\text{number of samples in each class})}{\text{number of samples in class } c}},$$

where $y_c \in \{0, 1\}$ denotes the label of each class for the input. Because training datasets are highly class-imbalanced, we use the class weights w_c to manually scale the training loss for each class.

We trained the models for 20 epochs using the Adam optimizer [25] with a mini-batch size of 100, a learning rate of 0.0004, and a dropout probability of 0.4. Note that for DeeperHSP, we left the pre-trained LM intact and only trained the projection layer and the CNN model. We used PyTorch [26] and Bio_Embeddings [27] libraries for model implementations and to obtain pre-trained representations, respectively.

Results

Datasets

Cross-validation dataset. For cross-validation experiments, we utilized the same dataset used in previous studies [8, 9]. Non-HSP sequences were randomly selected without homologous proteins from SwissProt [28]. HSP sequences were derived from HSPIR [4]. Thereafter, the proteins with $\geq$ 40% pairwise sequence similarity within the same family were removed using CD-HIT [29]. Finally, the non-HSP and HSP sequences containing non-standard amino acids were filtered out to obtain a cross-validation dataset (Table 3).

Independent test dataset. Although previous studies used additional test datasets, they did not ensure that those were independent from the cross-validation dataset [8, 9]. Therefore, we curated a new independent test dataset to evaluate the generalization performance (Table 3). We randomly sampled non-HSP sequences from Pfam [30] and collected manually verified HSP sequences from three data sources, *i.e.*, HGNC [31], RICE [32, 33], and InterPro [34]. Most importantly, we used CD-HIT [29] to remove homogeneous proteins such that no two proteins from the cross-validation and test datasets have 40% or more pairwise sequence similarity within the same class. We filtered out about 80% of the 3,911 curated sequences and obtained an independent test dataset of 680 sequences.

Table 3. Summary of cross-validation and independent test datasets.

Class	Cross-Validation Dataset	Independent Test Dataset
Non-HSP	9,965	500
HSP20	354	12
HSP40	1,257	52
HSP60	159	8
HSP70	278	53
HSP90	52	35
HSP100	81	20

https://doi.org/10.1371/journal.pone.0251865.t003

Feature extraction-based baselines

We compared the performance of DeepHSP and DeeperHSP with those of two state-of-the-art algorithms: PredHSP [8] and ir-HSP [9]. Because their codes are not publicly available, we re-implemented them using the Scikit-learn library [35]. First, we extracted DPC and SDPC features for PredHSP and ir-HSP, respectively. Then, we trained the radial basis function kernel SVM models. We selected SVM hyperparameters with the best performance among 144 configurations: 12 points of regularization penalty C and kernel coefficient γ that were evenly spaced between 10^3 and 10^{-3}.

To set competitive baselines, we made some modifications during the re-implementations. While PredHSP and ir-HSP are two-stage algorithms, we converted them into one-stage algorithms that classify both non-HSPs and the six HSP families simultaneously. We found that the latter performs better by utilizing all class information in a single integrated model. In addition, for ir-HSP, we removed random forest (RF)-based feature selection and used all 1600-dimensional SDPC features. We discovered that feature selection did not improve the classification performance. We report the performance of the modified baselines for the following cross-validation and independent results. The performance comparisons between the original and modified baselines are provided in the ablation studies.

Cross-validation results

We evaluated the classification performance of PredHSP, ir-HSP, DeepHSP, and DeeperHSP using five-fold cross-validation. We used eight evaluation metrics: accuracy, F1 score, precision, recall, specificity, MCC, AUC-ROC, and AUC-PR. Because all the evaluation metrics, except for accuracy, are defined for binary classification, we used unweighted averages of the scores computed for each class.

First, we compared the overall classification performance (Table 4). The results show that the proposed DeepHSP and DeeperHSP significantly outperformed the state-of-the-art algorithms. The gap between ir-HSP and DeepHSP verifies the importance of deep learning. DeepHSP was able to learn discriminative representations that could not be captured using the sequence composition features. The performance improvement obtained by DeeperHSP demonstrates the effectiveness of the pre-trained protein representations. They provide a wealth of information learned from a large number of unlabeled protein sequences. By incorporating the pre-trained representations and the CNN model, DeeperHSP outperformed the previous algorithms in terms of all the evaluation metrics, notably increasing the F1 score by 20%.

Next, we compared their class-wise classification performance in terms of the F1 score (Table 5). The results show similar performance improvement trends. DeepHSP outperformed the previous algorithms for most classes. However, it did not perform well for the classification of HSP90, where the least number of training samples are available. This indicates the difficulty of training deep neural networks from scratch without sufficient data. In contrast, DeeperHSP

Table 4. Comparison of overall classification performance using 5-fold cross-validation.

Model	Accuracy	F1 Score	Precision	Recall	Specificity	MCC	AUC-ROC	AUC-PR
PredHSP †	0.9128	0.6839	0.9044	0.5856	0.9409	0.6686	0.9496	0.7725
ir-HSP †	0.9483	0.8276	0.9437	0.7611	0.9678	0.8147	0.9725	0.8651
DeepHSP	0.9682	0.8613	0.9617	0.7984	0.9778	0.8554	0.9779	0.8931
DeeperHSP	**0.9927**	**0.9693**	**0.9745**	**0.9666**	**0.9957**	**0.9664**	**0.9947**	**0.9667**

† Modified version for performance improvement.

https://doi.org/10.1371/journal.pone.0251865.t004

Table 5. Comparison of class-wise classification performance in terms of F1 score using 5-fold cross-validation.

Model	Non-HSP	HSP20	HSP40	HSP60	HSP70	HSP90	HSP100	Average
PredHSP †	0.9502	0.6353	0.7573	0.4181	0.6159	0.6558	0.7544	0.6839
ir-HSP †	0.9698	0.8190	0.8696	0.6175	0.8165	0.8315	0.8692	0.8276
DeepHSP	0.9812	0.8554	0.9540	0.7157	0.8149	0.8079	0.9001	0.8613
DeeperHSP	**0.9956**	**0.9873**	**0.9847**	**0.9607**	**0.9648**	**0.9152**	**0.9768**	**0.9693**

† Modified version for performance improvement.

https://doi.org/10.1371/journal.pone.0251865.t005

provided the best classification performance for all classes. We can conclude that the pre-trained protein representations could help stabilize the training of the CNN model, particularly for classes with limited training data.

Finally, we examined the latent representations of DeepHSP and DeeperHSP. We used the t-distributed stochastic neighbor embedding (t-SNE) visualizations [36] with representations obtained from their penultimate layers (Fig 2). The latent representations of DeeperHSP are more clearly clustered into different groups according to their classes. By comparing the results, we confirmed the superiority of DeeperHSP over DeepHSP.

Independent test results

We used an independent test dataset to evaluate the different algorithms. The results show that the proposed models consistently outperformed the previous algorithms (Table 6). In particular, compared to ir-HSP, DeeperHSP increased F1 score by 10%. Considering that DeeperHSP has less parameters than DeepHSP, it is remarkable that the pre-trained representations could improve the performance of the CNN model.

We also present the confusion matrices of the ir-HSP and DeeperHSP predictions for the independent test dataset (Fig 3). We can observe that DeeperHSP provides better classification performance. It correctly classified the majority of the HSP100 samples, where ir-HSP did not perform satisfactorily. Meanwhile, the confusion matrices of ir-HSP and DeeperHSP exhibited similar misclassification patterns. In particular, among the 21 samples misclassified by DeeperHSP, 18 samples were also misclassified into the same incorrect classes by ir-HSP. This

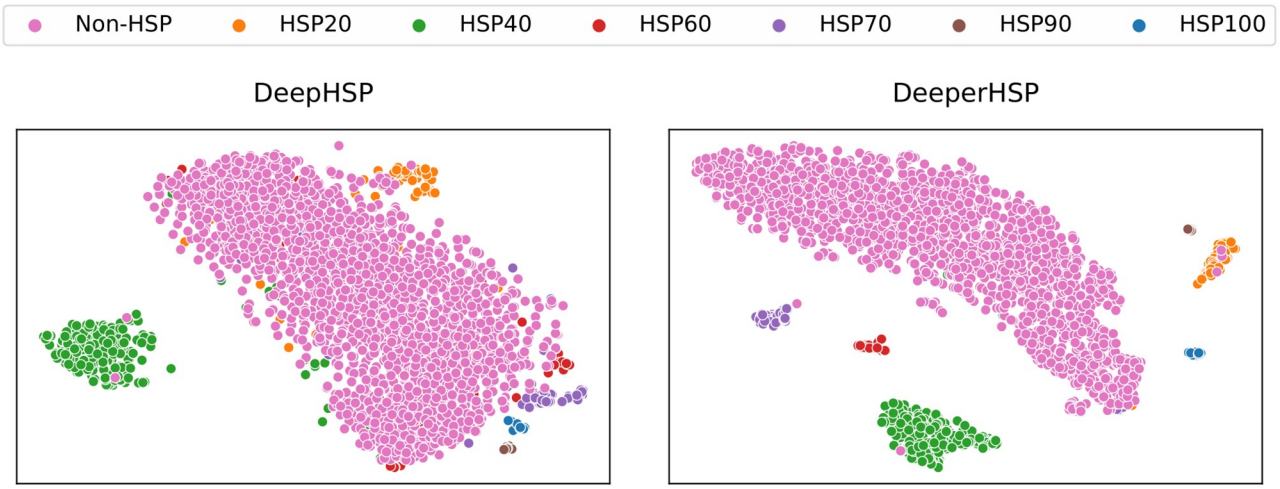

Fig 2. t-SNE plot of the latent representations of DeepHSP and DeeperHSP for the cross-validation dataset.

https://doi.org/10.1371/journal.pone.0251865.g002

Table 6. Comparison of overall classification performance using independent test.

Model	Accuracy	F1 Score	Precision	Recall	Specificity	MCC	AUC-ROC	AUC-PR
PredHSP †	0.9324	0.8017	**0.9330**	0.7643	0.9698	0.7993	0.9515	0.8154
ir-HSP †	0.9456	0.8302	0.9005	0.8092	0.9780	0.8240	0.9677	0.8400
DeepHSP	0.9500	0.8340	0.8870	0.8303	0.9805	0.8292	0.9431	0.8079
DeeperHSP	**0.9691**	**0.9126**	0.9230	**0.9077**	**0.9902**	**0.9043**	**0.9692**	**0.8747**

† Modified version for performance improvement.

https://doi.org/10.1371/journal.pone.0251865.t006

might imply that there are limitations to sequence-based identification of HSP families and additional structural information is required for performance improvement.

Running time

We compared the running time required for each algorithm. We report training time for the cross-validation dataset and inference time for the independent test dataset. Note that we used single CPU for the SVM-based algorithms and single GPU for the deep learning-based algorithms.

The results show that the proposed models have trade-off between performance and time-efficiency (Table 7). DeepHSP showed small improvement in performance but a strong advantage in time-efficiency. It was about 14–15 times faster than ir-HSP for both training and inference. On the other hand, while DeeperHSP demonstrated the best performance, it exhibited the longest running time. This was largely due to the time required for obtaining pre-trained representations. Based on these observations, we believe that the different strengths of DeepHSP and DeeperHSP can provide different options based on users' demands.

Ablation studies

Feature extraction-based baselines. For competitive baselines, we explored both one- and two-stage algorithms based on different combinations of features and classifiers. We considered six types of features [37]: amino acid composition (AAC), DPC, SDPC, PAAC,

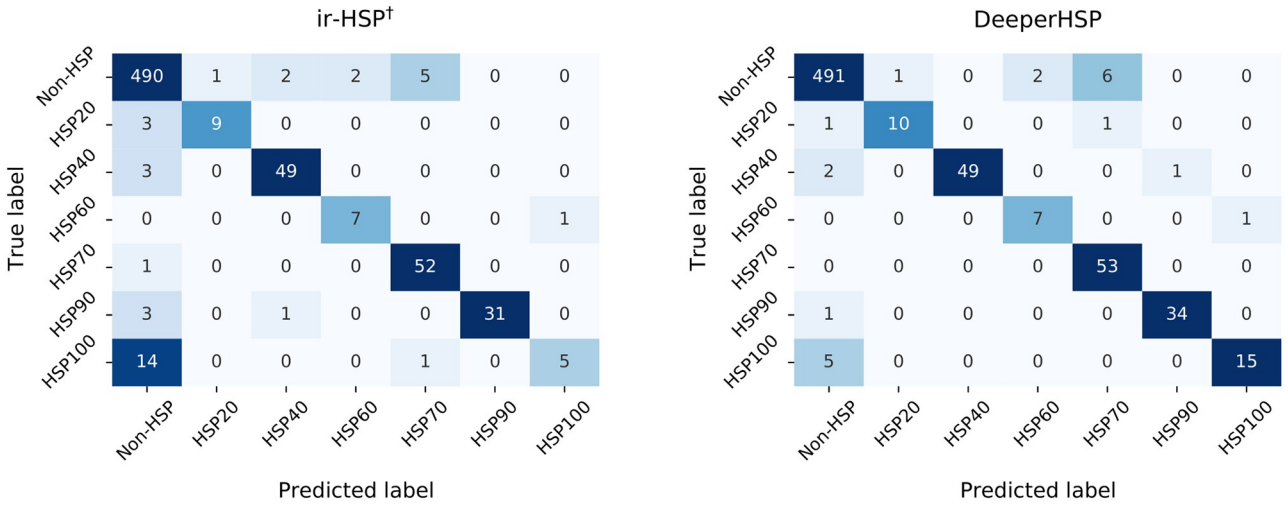

Fig 3. Confusion matrices of the modified ir-HSP and DeeperHSP predictions for the independent test dataset.

https://doi.org/10.1371/journal.pone.0251865.g003

Table 7. Comparison of running time required for each algorithm.

Model	Training (seconds)	Inference (seconds)
PredHSP †	315	5
ir-HSP †	1,265	14
DeepHSP	80	1
DeeperHSP	2,112	120

† Modified version for performance improvement.

https://doi.org/10.1371/journal.pone.0251865.t007

composition transition distribution (CTD), and Moreau-Broto auto-correlation (MBAuto). We also considered six types of classifiers [35]: XGBoost, RF, Lasso, Ridge, ElasticNet, and SVM. For each classifier, we selected its hyperparameters with the best performance among 144 configurations.

Fig 4 presents heatmaps of the F1 scores obtained from the different algorithms using five-fold cross-validation. We can observe that the one-stage algorithms performed better than the two-stage algorithms. Comparing the different combinations of features and classifiers, two of them clearly stand out. These are the modified versions of PredHSP and ir-HSP, which are based on SVM classifiers trained using DPC and SDPC features, respectively.

We further examined whether techniques used in previous works could improve the classification performance [9, 38]. We used a one-stage SVM model trained on the SDPC features as a baseline model. Then, we additionally adopted either (1) RF-based feature selection to choose a smaller number of important features or (2) the syntactic minority oversampling technique (SMOTE) [39] to deal with class imbalance. We discovered that both techniques led to lower F1 scores of 0.71 and 0.63, respectively.

Pre-trained protein representations. We explored various pre-trained protein LMs to obtain representations for DeeperHSP: UniRep, SeqVec, PLUS-RNN, ProtXLNet, ProtBERT, and ESM.

We compared the performance of DeeperHSP with different LMs using five-fold cross-validation (Fig 5). Additionally, as a baseline, we include the performance of DeepHSP in the

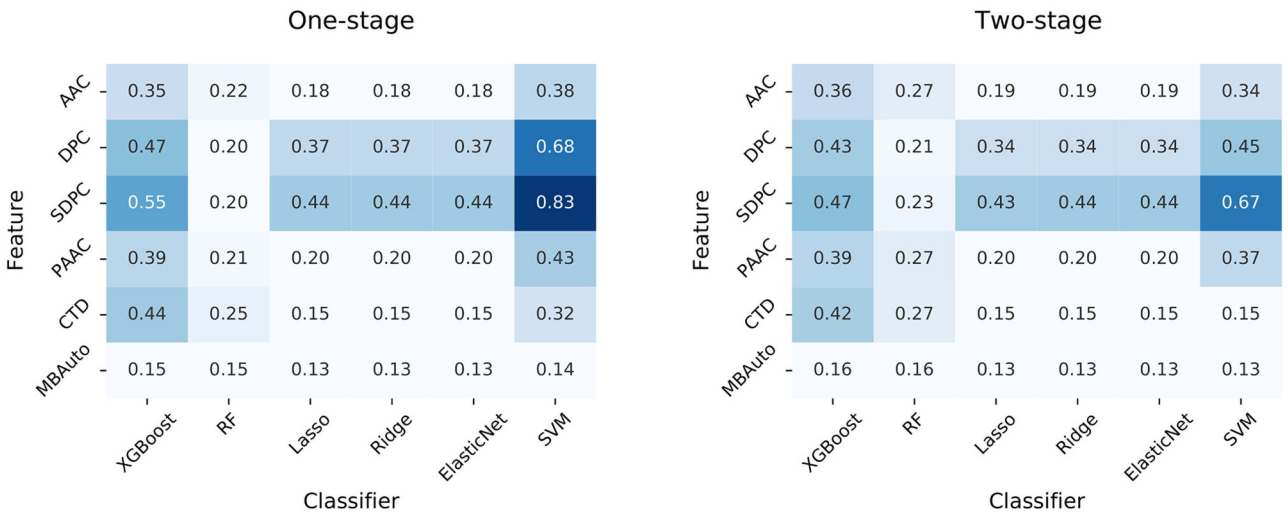

Fig 4. Heatmaps of the F1 scores obtained from baseline algorithms using 5-fold cross-validation. One-stage algorithms classify both non-HSPs and the six HSP families simultaneously. Two-stage algorithms use two models to filter out non-HSPs and classify the remaining HSPs into the six families.

https://doi.org/10.1371/journal.pone.0251865.g004

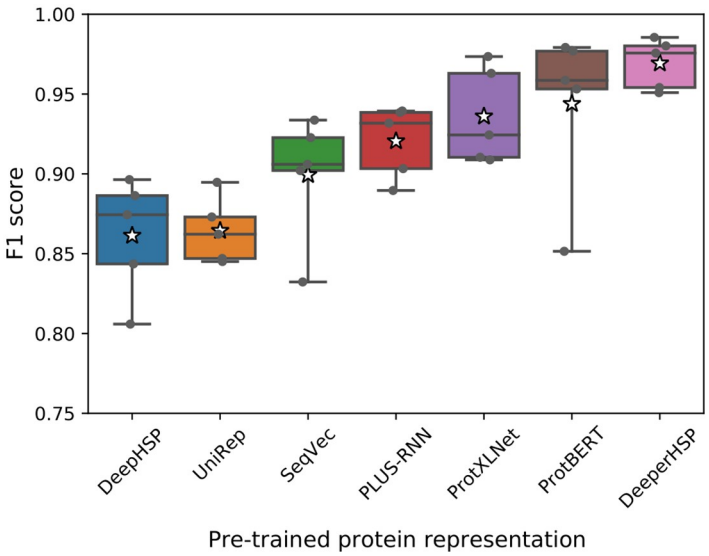

Fig 5. Boxplot of the F1 scores obtained from DeeperHSP with different pre-trained protein representations using 5-fold cross-validation.

https://doi.org/10.1371/journal.pone.0251865.g005

leftmost column. Each box denotes the quartiles of F1 scores, and the star denotes their average. The boxplot shows that all LMs improve the average classification performance compared to DeepHSP. Taking advantage of a large number of unlabeled proteins, the pre-trained protein representations provide a wealth of information that cannot be learned from one-hot encoding.

While all the pre-trained protein LMs help in the identification of HSP families, their level of performance improvement varies significantly. The small gap between OneHot (*i.e.*, DeepHSP) and UniRep indicates that a sufficient number of parameters are required to obtain a moderate increase (Table 2). LMs with more parameters generally provide more performance improvement. For example, the larger TFM-based LMs outperformed the RNN-based LMs, and the largest ESM showed the best performance. One exception is that although PLUS-RNN has fewer parameters than SeqVec, it exhibits better performance. We conjecture that this can be attributed to its additional protein-specific pre-training objective, which can better capture structural information of protein sequences than those solely pre-trained with an LM [19].

Conclusion

In this paper, we proposed two novel deep learning algorithms that classify both non-HSPs and the six HSP families simultaneously. The time-efficient DeepHSP uses a CNN model that identifies the HSP families faster and more accurately than the alternatives. Furthermore, the high-performance DeeperHSP leverages protein transfer learning to improve performance. It trains the CNN model on top of the pre-trained protein representations instead of the one-hot encoded protein sequences. Our experimental results showed that DeeperHSP remarkably outperformed the state-of-the-art algorithms. It increased F1 scores by 20% and 10% on the cross-validation and independent test datasets, respectively. We envision that the proposed algorithms can provide a proteome-wide prediction of HSPs and help various downstream analyses for pathology and clinical research.

Although the proposed algorithms have a clear advantage over the previous approaches, there are still some limitations and room for further improvement. First, there are trade-offs between the running time and classification performance. We expect that a lightweight LM would be able to greatly reduce the running time for obtaining pre-trained protein representations [40]. This will enable the development of both time-efficient and high-performance algorithms for the identification of HSP families. Second, they only focused on classifying non-HSPs and HSP families. It would be valuable to develop a more comprehensive model that can provide additional information on other protein types and functions [41]. Finally, we believe it would also be interesting to extend this work to recent research topics in machine learning such as interpretability [42, 43] and security [44–47].

Author Contributions

Conceptualization: Seonwoo Min, Byunghan Lee, Sungroh Yoon.

Data curation: Seonwoo Min.

Formal analysis: Seonwoo Min, Byunghan Lee, Sungroh Yoon.

Funding acquisition: Byunghan Lee, Sungroh Yoon.

Investigation: Seonwoo Min, Byunghan Lee, Sungroh Yoon.

Methodology: Seonwoo Min, Byunghan Lee, Sungroh Yoon.

Project administration: Byunghan Lee, Sungroh Yoon.

Software: Seonwoo Min.

Supervision: Byunghan Lee, Sungroh Yoon.

Visualization: Seonwoo Min.

Writing – original draft: Seonwoo Min, HyunGi Kim, Byunghan Lee, Sungroh Yoon.

Writing – review & editing: Seonwoo Min, HyunGi Kim, Byunghan Lee, Sungroh Yoon.

References

1. Ritossa F. A new puffing pattern induced by temperature shock and DNP in Drosophila. Experientia. 1962; 18(12):571–573. https://doi.org/10.1007/BF02172188
2. Ikwegbue PC, Masamba P, Mbatha LS, Oyinloye BE, Kappo AP. Interplay between heat shock proteins, inflammation and cancer: a potential cancer therapeutic target. American journal of cancer research. 2019; 9(2):242.
3. Jolly C, Morimoto RI. Role of the heat shock response and molecular chaperones in oncogenesis and cell death. Journal of the National Cancer Institute. 2000; 92(19):1564–1572. https://doi.org/10.1093/jnci/92.19.1564
4. Ratheesh K, N SN, S PA, Sinha D, Veedin Rajan VB, Esthaki VK, et al. HSPIR: a manually annotated heat shock protein information resource. Bioinformatics. 2012; 28(21):2853–2855. https://doi.org/10.1093/bioinformatics/bts520
5. Didenko T, Duarte AM, Karagöz GE, Rüdiger SG. Hsp90 structure and function studied by NMR spectroscopy. Biochimica et Biophysica Acta (BBA)-Molecular Cell Research. 2012; 1823(3):636–647. https://doi.org/10.1016/j.bbamcr.2011.11.009
6. Feng PM, Chen W, Lin H, Chou KC. iHSP-PseRAAAC: Identifying the heat shock protein families using pseudo reduced amino acid alphabet composition. Analytical Biochemistry. 2013; 442(1):118–125. https://doi.org/10.1016/j.ab.2013.05.024
7. Ahmad S, Kabir M, Hayat M. Identification of Heat Shock Protein families and J-protein types by incorporating Dipeptide Composition into Chou's general PseAAC. Computer methods and programs in biomedicine. 2015; 122(2):165–174. https://doi.org/10.1016/j.cmpb.2015.07.005

8. Kumar R, Kumari B, Kumar M. PredHSP: sequence based proteome-wide heat shock protein prediction and classification tool to unlock the stress biology. PloS one. 2016; 11(5):e0155872. https://doi.org/10.1371/journal.pone.0155872
9. Meher PK, Sahu TK, Gahoi S, Rao AR. ir-HSP: improved recognition of heat shock proteins, their families and sub-types based on g-spaced di-peptide features and support vector machine. Frontiers in genetics. 2018; 8:235. https://doi.org/10.3389/fgene.2017.00235
10. Min S, Lee B, Yoon S. Deep learning in bioinformatics. Briefings in bioinformatics. 2017; 18(5):851–869.
11. RM SP, Maddikunta PKR, Parimala M, Koppu S, Gadekallu TR, Chowdhary CL, et al. An effective feature engineering for DNN using hybrid PCA-GWO for intrusion detection in IoMT architecture. Computer Communications. 2020; 160:139–149. https://doi.org/10.1016/j.comcom.2020.05.048
12. Hakak S, Alazab M, Khan S, Gadekallu TR, Maddikunta PKR, Khan WZ. An ensemble machine learning approach through effective feature extraction to classify fake news. Future Generation Computer Systems. 2021; 117:47–58. https://doi.org/10.1016/j.future.2020.11.022
13. Khan RU, Zhang X, Kumar R, Sharif A, Golilarz NA, Alazab M. An adaptive multi-layer botnet detection technique using machine learning classifiers. Applied Sciences. 2019; 9(11):2375. https://doi.org/10.3390/app9112375
14. Kim HK, Min S, Song M, Jung S, Choi JW, Kim Y, et al. Deep learning improves prediction of CRISPR–Cpf1 guide RNA activity. Nature biotechnology. 2018; 36(3):239. https://doi.org/10.1038/nbt.4061 PMID: 29431740
15. Lee B, Baek J, Park S, Yoon S. deepTarget: end-to-end learning framework for microRNA target prediction using deep recurrent neural networks. In: Proceedings of the 7th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics; 2016. p. 434–442.
16. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J. Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems; 2013. p. 3111–3119.
17. Devlin J, Chang MW, Lee K, Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:181004805. 2018;.
18. Alley EC, Khimulya G, Biswas S, AlQuraishi M, Church GM. Unified rational protein engineering with sequence-based deep representation learning. Nature methods. 2019; 16(12):1315–1322. https://doi.org/10.1038/s41592-019-0598-1
19. Min S, Park S, Kim S, Choi HS, Yoon S. Pre-training of deep bidirectional protein sequence representations with structural information. arXiv preprint arXiv:191205625. 2019;.
20. Heinzinger M, Elnaggar A, Wang Y, Dallago C, Nechaev D, Matthes F, et al. Modeling aspects of the language of life through transfer-learning protein sequences. BMC bioinformatics. 2019; 20(1):1–17. https://doi.org/10.1186/s12859-019-3220-8 PMID: 31847804
21. Elnaggar A, Heinzinger M, Dallago C, Rihawi G, Wang Y, Jones L, et al. ProtTrans: Towards Cracking the Language of Life's Code Through Self-Supervised Deep Learning and High Performance Computing. arXiv preprint arXiv:200706225. 2020;.
22. Rives A, Goyal S, Meier J, Guo D, Ott M, Zitnick CL, et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. bioRxiv. 2019; p. 622803.
23. Rao R, Bhattacharya N, Thomas N, Duan Y, Chen X, Canny J, et al. Evaluating protein transfer learning with tape. Advances in Neural Information Processing Systems. 2019; 32:9689. PMID: 33390682
24. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. In: Advances in neural information processing systems; 2017. p. 5998–6008.
25. Kingma DP, Ba J. Adam: A method for stochastic optimization. arXiv preprint arXiv:14126980. 2014;.
26. Paszke A, Gross S, Chintala S, et al. Automatic Differentiation in PyTorch. NIPS Autodiff Workshop. 2017;.
27. Dallago C, Schütze K, Heinzinger M, Olenyi T, Littmann M, Lu A, et al. Learned embeddings from deep learning to visualize and predict protein sets. Under review. 2021;.
28. Boeckmann B, Bairoch A, Apweiler R, Blatter MC, Estreicher A, Gasteiger E, et al. The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003. Nucleic acids research. 2003; 31(1):365–370. https://doi.org/10.1093/nar/gkg095 PMID: 12520024
29. Li W, Godzik A. Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. Bioinformatics. 2006; 22(13):1658–1659. https://doi.org/10.1093/bioinformatics/btl158
30. Bateman A, Coin L, Durbin R, Finn RD, Hollich V, Griffiths-Jones S, et al. The Pfam protein families database. Nucleic acids research. 2004; 32(suppl_1):D138–D141. https://doi.org/10.1093/nar/gkh121 PMID: 14681378

31. Kampinga HH, Hageman J, Vos MJ, Kubota H, Tanguay RM, Bruford EA, et al. Guidelines for the nomenclature of the human heat shock proteins. Cell Stress and Chaperones. 2009; 14(1):105–111. https://doi.org/10.1007/s12192-008-0068-7 PMID: 18663603

32. Wang Y, Lin S, Song Q, Li K, Tao H, Huang J, et al. Genome-wide identification of heat shock proteins (Hsps) and Hsp interactors in rice: Hsp70s as a case study. BMC genomics. 2014; 15(1):1–15. https://doi.org/10.1186/1471-2164-15-344 PMID: 24884676

33. Sarkar NK, Kundnani P, Grover A. Functional analysis of Hsp70 superfamily proteins of rice (Oryza sativa). Cell stress and Chaperones. 2013; 18(4):427–437. https://doi.org/10.1007/s12192-012-0395-6

34. Hunter S, Apweiler R, Attwood TK, Bairoch A, Bateman A, Binns D, et al. InterPro: the integrative protein signature database. Nucleic acids research. 2009; 37(suppl_1):D211–D215. https://doi.org/10.1093/nar/gkn785 PMID: 18940856

35. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research. 2011; 12:2825–2830.

36. Van der Maaten L, Hinton G. Visualizing data using t-SNE. Journal of machine learning research. 2008; 9(11).

37. Cao DS, Xu QS, Liang YZ. propy: a tool to generate various modes of Chou's PseAAC. Bioinformatics. 2013; 29(7):960–962. https://doi.org/10.1093/bioinformatics/btt072

38. Jing XY, Li FM. Identifying Heat Shock Protein Families from Imbalanced Data by Using Combined Features. Computational and mathematical methods in medicine. 2020; 2020. https://doi.org/10.1155/2020/8894478

39. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research. 2002; 16:321–357. https://doi.org/10.1613/jair.953

40. Lan Z, Chen M, Goodman S, Gimpel K, Sharma P, Soricut R. Albert: A lite bert for self-supervised learning of language representations. arXiv preprint arXiv:190911942. 2019;.

41. Rifaioglu AS, Doğan T, Martin MJ, Cetin-Atalay R, Atalay V. DEEPred: automated protein function prediction with multi-task feed-forward deep neural networks. Scientific reports. 2019; 9(1):1–16.

42. Vig J, Madani A, Varshney LR, Xiong C, Socher R, Rajani NF. Bertology meets biology: Interpreting attention in protein language models. arXiv preprint arXiv:200615222. 2020;.

43. Kim S, Yi J, Kim E, Yoon S. Interpretation of NLP Models through Input Marginalization. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP); 2020. p. 3154–3167.

44. Song J, Zhong Q, Wang W, Su C, Tan Z, Liu Y. FPDP: Flexible privacy-preserving data publishing scheme for smart agriculture. IEEE Sensors Journal. 2020;.

45. Zhang L, Zhang Z, Wang W, Jin Z, Su Y, Chen H. Research on a Covert Communication Model Realized by Using Smart Contracts in Blockchain Environment. IEEE Systems Journal. 2021;.

46. Wang W, Huang H, Zhang L, Su C. Secure and efficient mutual authentication protocol for smart grid under blockchain. Peer-to-Peer Networking and Applications. 2020; p. 1–13.

47. Bae H, Jang J, Jung D, Jang H, Ha H, Yoon S. Security and privacy issues in deep learning. arXiv preprint arXiv:180711655. 2018;.