Bioinformatics, 37(20), 2021, 3473–3479
doi: 10.1093/bioinformatics/btab350
Advance Access Publication Date: 13 May 2021
Original Paper

OXFORD

Structural bioinformatics

In silico prediction of *in vitro* protein liquid–liquid phase separation experiments outcomes with multi-head neural attention

Daniele Raimondi [1,†], Gabriele Orlando [2,†], Emiel Michiels[2], Donya Pakravan[3,4], Anna Bratek-Skicki[5], Ludo Van Den Bosch[3,4], Yves Moreau[1,*], Frederic Rousseau[2,*] and Joost Schymkowitz[2,*]

[1]ESAT-STADIUS, KU Leuven, 3001 Leuven, Belgium, [2]SWITCH Lab, Department of Cellular and Molecular Medicine, KU Leuven, 3001 Leuven, Belgium, [3]Department of Neurosciences, LBI, KU Leuven, 3001 Leuven, Belgium, [4]VIB, Center for Brain and Disease Research, Laboratory of Neurobiology, 3000 Leuven, Belgium and [5]VIB-VUB Center for Structural Biology (CSB), 1050 Brussels, Belgium

*To whom correspondence should be addressed.
†The authors wish it to be known that these authors contributed equally.

Associate Editor: Yann Ponty
Received on November 5, 2020; revised on April 15, 2021; editorial decision on April 29, 2021; accepted on May 10, 2021

Abstract

Motivation: Proteins able to undergo liquid–liquid phase separation (LLPS) *in vivo* and *in vitro* are drawing a lot of interest, due to their functional relevance for cell life. Nevertheless, the proteome-scale experimental screening of these proteins seems unfeasible, because besides being expensive and time-consuming, LLPS is heavily influenced by multiple environmental conditions such as concentration, pH and temperature, thus requiring a combinatorial number of experiments for each protein.

Results: To overcome this problem, we propose a neural network model able to predict the LLPS behavior of proteins given specified experimental conditions, effectively predicting the outcome of *in vitro* experiments. Our model can be used to rapidly screen proteins and experimental conditions searching for LLPS, thus reducing the search space that needs to be covered experimentally. We experimentally validate Droppler's prediction on the TAR DNA-binding protein in different experimental conditions, showing the consistency of its predictions.

Availability and implementation: A python implementation of Droppler is available at https://bitbucket.org/grog-drinker/droppler

Contact: yves.moreau@kuleuven.be or joost.schymkowitz@kuleuven.be or frederic.rousseau@kuleuven.vib.be

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Recently, much effort has been devoted to the study of proteins able to undergo a phase transition called liquid–liquid phase separation (LLPS) (Orlando *et al.*, 2019; Wang *et al.*, 2018) under certain biophysical conditions or thanks to the presence of macromolecules such as RNA (Weber and Brangwynne, 2012). In LLPS, the proteins undergo reversible demixing, thus forming suspended condensates similar to droplets (Banani *et al.*, 2017), that, in eukaryotic cells, might act as membraneless organelles (Feric *et al.*, 2016; Nott *et al.*, 2015). Recent studies showed that the formation of these organelles might have an important role in various biological processes (Banani *et al.*, 2017; Shin and Brangwynne, 2017), such as RNA metabolism (Uversky, 2017).

Although the biophysical characteristics of the protein sequences are important determinants for the LLPS behavior, such as cation–π interactions between Tyr and Arg residues (Wang *et al.*, 2018), the presence of RNA-recognition motifs (RRMs) and prion-like domain (PLD) separated by spacer regions (Orlando *et al.*, 2019; Wang *et al.*, 2018), experimental and environmental conditions such as temperature, pH and salt concentration play a crucial role in modulating LLPS behavior (Li *et al.*, 2020; Wang *et al.*, 2018).

In our previous work (Orlando *et al.*, 2019), we developed an unsupervised probabilistic model for the detection of sequences with LLPS behavior similar to the one observed in FUS-like proteins (Wang *et al.*, 2018). Our method, called PSPer, was able to prioritize and identify proteins forming membraneless organelles in eukaryotic cells and to determine which mutations were more likely to alter the LLPS propensity.

PSPer was based on empirical rules defined for a small class of proteins (FUS-like family) (Wang *et al.*, 2018) because very little data about other LLPS proteins were available at the time.

© The Author(s) 2021. Published by Oxford University Press. All rights reserved. For permissions, please e-mail: journals.permissions@oup.com

Downloaded from https://academic.oup.com/bioinformatics/article/37/20/3473/6275261 by INACTIVE user on 29 October 2024

Other sequence-based LLPS predictors have also been developed afterwards, using machine learning (ML) classifiers such as Random Forest (Saar *et al.*, 2020) or support vector machines (SVMs) (Sun *et al.*, 2019). A deeper analysis of the current state of the art LLPS predictors from sequence only is available in Vernon and Forman-Kay (2019).

Recently, more data related to LLPS proteins have been gathered in publicly available databases (Mészáros *et al.*, 2020; You *et al.*, 2020). In particular, thanks to the publication of LLPSDB (Li *et al.*, 2020), which is a database collecting in a standardized way the outcomes and the details of LLPS experiments, it has become possible to take another step towards improving the *in silico* prediction of the LLPS behavior, modeling for the first time the interaction between protein sequence and experimental conditions. From this resource we could indeed gather enough *in vitro* experimental data to train a *supervised* model, called Droppler, able to predict *in silico* the outcomes of such experiments. From the ML perspective, our model is the first to *jointly* model the role of the protein sequence and the experimental conditions towards the prediction of LLPS.

Droppler is an end-to-end neural network (NN) model which uses a novel multi-head neural attention mechanism, a class of ML algorithms that have already been proving their strength in other prediction tasks (Raimondi *et al.*, 2020). The network is trained to predict, for each input protein sequence, its likelihood to undergo LLPS given a user-specified set of experimental conditions such as temperature, salt and protein concentration, presence of crowding agent and pH. From the ML methodological standpoint, this end-to-end neural attention architecture allows us to (i) natively deal with variable-length input sequences (which is usually a non-trivial task for conventional ML approaches) and (ii) provide a way to interpret the prediction, thus shedding light on the molecular mechanisms driving the LLPS.

To the best of our knowledge, Droppler is the first model designed to predict the likelihood of proteins to undergo LLPS given a specific set of experimental conditions because existing approaches (Orlando *et al.*, 2019; Saar *et al.*, 2020; Sun *et al.*, 2019; Vernon and Forman-Kay, 2019) focus only on the protein sequence and predict some sort of *average* LLPS propensity or in very specific experimental conditions. For instance, in Saar *et al.* (2020) the authors only predict the LLPS propensity of a protein in 'nearly physiological conditions'. Thanks to the explicit modeling of the experimental conditions, Droppler could be used instead to directly prioritize the *in vitro experiments* that should be performed for (i) the discovery of novel LLPS proteins and (ii) to efficiently explore the experimental conditions space in order to find the physico-chemical settings that maximize the likelihood of LLPS for each specific protein sequence. Droppler predictions are instantaneous and thus large sets of combinations of pH, temperature, salt and protein concentrations can be explored in a very short time.

While the task of predicting *in silico* the outcome of LLPS experiments is challenging, as indicated by the low area under the receiver operating characteristic curve (AUC), we show that the relatively high Sensitivity and Precision of Droppler allows it to screen the (protein, experimental conditions) space identifying 75% of the experiments in which LLPS occurs, while discarding 50% of the negative experiments. Thanks to a relatively high precision, 70% of the times Droppler indicates that LLPS is likely to happen, its prediction is indeed correct. To show that Droppler can be used to explore *in silico* the experimental conditions landscapes of specific proteins, identifying the pH, salt and protein concentrations or temperature regions in which LLPS is likely to occur, we experimentally validated its prediction on the terminal domain of TAR DNA-binding protein 43 (TDP-43_LCD), showing that the protein behaves consistently with Droppler's predictions in the region of experimental conditions space explored.

Moreover, the interpretable nature of the neural attention allowed us to investigate where the NN *focuses* its attentions in order to compute its predictions, and indeed we show that the PLD described in Wang *et al.* (2018) is consistently attended by our model, indicating that the learned patterns indeed correlate with known biophysical aspects related to LLPS. Additionally, our tool works from single sequence, without using any evolutionary information as input. This grants both fast predictions and the absence of selection biases described in Orlando *et al.* (2016).

Droppler's code is freely available at https://bitbucket.org/grogdrinker/droppler and the multi-head architecture that we propose could be applied to many protein bioinformatics problems.

2 Materials and methods

2.1 Datasets

We trained and tested our model on the data extracted from the recently published LLPSDB (Li *et al.*, 2020) database, which contains a manually curated list of proteins undergoing LLPS *in vitro* and the correspondent experimental conditions. LLPSDB contains in total 1182 experiments on 273 proteins, but before performing our analysis we applied various consistency filters to the data, ending up with 896 experiments on 366 unique sequences obtained from 137 proteins. This is due to the fact that many experiments were indeed performed on mutated proteins.

In our filtering, we first considered only experiments involving just one protein at a time (instead of pairs of proteins or proteins plus RNA). Then we filtered out experiments for which we could not (i) properly parse the experimental conditions values or (ii) convert the measure units or (iii) some experimental conditions values were missing. Moreover, since some experiments were reporting extreme conditions, we decided to remove the outliers, defined as the top and bottom 10% of the distribution of each experimental condition taken into consideration. The selected experiments used as training samples are available from our git repository: https://bitbucket.org/grogdrinker/droppler.

2.2 Processing of the experimental conditions

The annotation of experimental conditions may have varying degrees of consistency. For example, different measure units are used to indicate salt concentrations, temperature ranges and protein concentrations. We parsed and uniformed these values, defining a standardized encoding for each characteristic.

For what concerns the salt concentration, since different types of salts are mentioned (NaCl, KCl, MgCl), we described them by using the ionic strength. When only ranges of salt concentrations were indicated, we took the mean value.

Temperatures are often represented as ranges (e.g. $<40°C$, 0–20°C, >281 K) instead of precise values. We first uniformed them to Celsius and then we decided to represent the ranges as 10 bins from 0 to 100°C (e.g. $<40°C$ becomes $[1, 1, 1, 1, 0, 0, 0, 0, 0, 0]$).

From the description of the buffer used in each experiment we extracted the pH values and the presence of crowding agent (such as PEG, Dextran or Ficoll), which we encoded as 1 or 0 values (present or absent).

These experimental conditions are thus concatenated into a numeric vector, which is used as input to Droppler, alongside the target protein sequence. The conditions considered are temperature (10 dimensions), protein concentration (one dimension), ionic strength (one dimension), presence of crowding agent (one dimension) and buffer pH (one dimension), for a total of 14 dimensions.

2.3 Multi-head neural attention for protein sequences

Droppler is an NN that uses the multi-head neural attention mechanism to predict the outcome of LLPS experiments. The NN model takes as input one protein sequence and a specific set of experimental conditions and predicts the likelihood of the input protein to show *in vitro* LLPS behavior given the specified experimental conditions.

Although it is a very common necessity in bioinformatics, feeding variable-length sequences to ML methods is still a non-trivial task because most of the conventional ML algorithms are designed to take fixed-length input vectors (e.g. SVM, Random Forest). While various forms of quantization or compression of the input

Downloaded from https://academic.oup.com/bioinformatics/article/37/20/3473/6275261 by INACTIVE user on 29 October 2024

have been used in literature (Clark and Radivojac, 2014; Raimondi *et al.*, 2018; Xiao *et al.*, 2018), to translate the variable-length inputs into fixed-size encoding, methods that can natively take as input variable-length sequences are still uncommon.

In the following, we describe the details of this architecture, which is also shown in Figure 1.

2.3.1 Feeding the experimental condition to the NN

As described in Section 2.2, the experimental conditions are encoded as 14-dimensional vectors. Since each of these features has its own specific range of values, we scaled each one by using the scikit-learn (Pedregosa *et al.*, 2011) minMaxScaler. As shown in the rightmost branch in Figure 1, this input vector of condition is then processed by a one-layer feed-forward NN, with dropout ($P = 0.2$) and Tanh activation, transforming it into a 10-dimensional vector that will be used later in the NN.

2.3.2 Feeding the target protein sequence to the NN

As shown in the left branch of Figure 1 the target protein sequence is initially treated separately from the experimental conditions. First, we encoded the amino acid sequence with a 20-dimensional trainable embedding, obtaining a $20 \times L$ dense matrix encoding each input protein, where L is the protein length. This matrix is used as input for the neural attention architecture (left branch in Fig. 1), whose final output is then joined to the processed condition vector.

Similar to the SKADE model presented in Raimondi *et al.* (2020), the branch of the NN that processes the protein sequence of Droppler is composed by two almost identical sub-networks, the attention (A) and the predictor (P) (see Fig. 1). Both A and P take the $20 \times L$ sequence embedding and feed it into a two layers bi-directional gated recurrent unit (GRU) network with 10 hidden neurons and dropout ($P = 0.2$) between each layer. The final output of the GRU is thus a $(10 + 10) \times L$ output describing the entire sequence. This matrix is then processed by a feed-forward network (yellow and orange dots in Fig. 1) that *slides* on the $20 \times L$ output of the GRU, taking as input each 20-dimensional column and transforming it into a 10-dimensional column, thus producing a $10 \times L$ tensor, as shown in Figure 1.

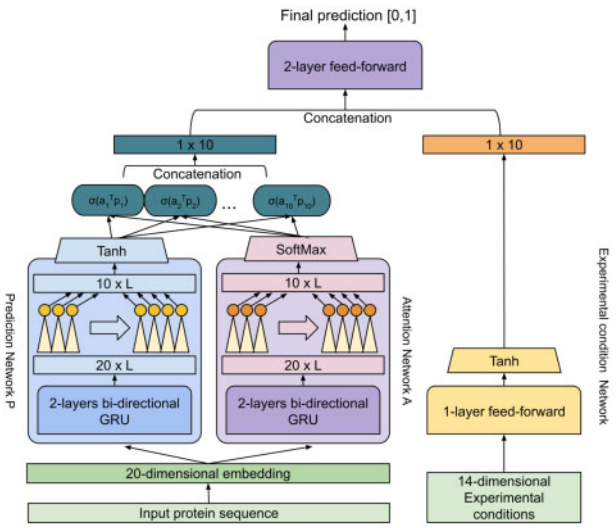

Fig. 1. Figure showing the architecture of Droppler. The protein sequence is translated into a 20-dimensional embedding and then passed to the predictor network P and the attention network A. Each of these sub-networks contain two layers of bi-directional GRU network, followed by a feed-forward NN. The predictor ends with a Tanh activation, while the attention network has a SoftMax activation. Finally, the 1×10 outputs of the sub-network processing the sequence (left block) and the one processing the experimental conditions (right block) are concatenated and fed to the final feed-forward module, which computes the final probability-like prediction

Both the A and P sub-networks are completely symmetrical up to this point. The only difference (see Fig. 1) is that the A sub-network applies a SoftMax activation to each of the 10 rows of the $10 \times L$ output tensor, which will act as 10 *heads* in the multi-head attention mechanism (Vaswani *et al.*, 2017). The P network applies the same non-linear activation used so far, the Tanh.

The A and P sub-networks are then joined with a series of row-wise dot products between the $10 \times L$ tensor produced by P and the $10 \times L$ SoftMaxed tensor produced by A, obtaining a 10×1 vector. Each of these dimensions contain the values produced by the predictor P network *mediated* by the SoftMaxed values produced by the multi-head attention computed by the attention network A. In other words, the P sub-network generates a residue-based prediction, while A a residue-based weight used to distribute a certain *attention budget* (SoftMax values sum to 1) over the most relevant residues in the predictions generated by P. The attention is thus acts as a filter able to ignore the less relevant sequence positions, and the SoftMax forces the NN to select a limited number of them. The dot product of these two vector produces the final prediction of each head of the multi-head architecture, which produces the 1×10 vector summarizing the information contained in the protein sequence. With this procedure, we are able to obtain a fixed-length vector that encodes a protein sequence of arbitrary length.

2.3.3 Joining sequence and experimental condition information

Finally, the 10-dimensional vector containing the results of the multi-head attention applied on the input sequence are concatenated with the 10-dimensional vector representing the experimental conditions (see Fig. 1) and sent through the final two-layer feed-forward NN with 10 hidden neurons and Tanh activation that produce the final prediction, with a final Sigmoid activation that provides a probability-like score. Supplementary Figure S1 shows a diagram of the whole pipeline of Droppler.

2.3.4 Training parameters

The NN described here has been implemented in pyTorch (Paszke *et al.*, 2017) and contains 8891 trainable parameters. It has been trained for 50 epochs with Adam optimizer with batch size of 101, L2 regularization with $\lambda = 1 \times 10^{-5}$ and learning rate of 0.01.

2.4 Computational validation procedure and performance evaluation

To evaluate the performances of Droppler we performed a 5-fold stratified cross-validation. When creating each train-test split, we ensured that proteins in the training set shared <20% of sequence identity (SI) at 90% of coverage with proteins in the test set, using BLASTCLUST (Altschul *et al.*, 1990).

We evaluated the performance of Droppler by computing the classical metrics to evaluate the quality of a binary classification, such as Sensitivity, Specificity (Spe), balanced accuracy, Precision, Matthews correlation coefficient, AUC and area under the precision recall curve.

2.5 Experimental evaluation of phase separation in TDP-43_LCD

For the optical density measurements unlabeled TDP-43_LCD (stock concentration of 40 μM in a 20 mM MES buffer pH5) was mixed 1:1 with each buffer condition (50 mM buffers) and immediately transferred to a 96-well plate (Corning, half area, non-binding plate) and OD600 measurements were performed with a FluoOmega plate reader (BMG LABTECH). At least four independent repeats were performed. For the imaging experiments, purified protein stored in MES buffer was combined with tagged TDP-43_LCD. Alexa Fluor™ 488 NHS Ester was used to tag TDP-43_LCD, which was mixed with unlabeled protein at a 1:50 ratio. The protein was added at a 1:1 ratio with buffer 1 or buffer 2 at a final concentration of 20 μM. The sample left to incubate for 10 min before images were obtained using the Nikon A1R Eclipse Ti (Japan).

Downloaded from https://academic.oup.com/bioinformatics/article/37/20/3473/6275261 by INACTIVE user on 29 October 2024

3 Results

3.1 Droppler predicts the outcome of LLPS experiments

Droppler is an end-to-end NN model that takes as inputs (i) a protein sequence *S* and (ii) a list of experimental conditions and predicts the likelihood of *S* to undergo LLPS given the choice of experimental conditions.

To train and test it we used the 896 LLPS experiments on 137 proteins (corresponding to 366 unique sequences due to mutations) that we extracted from LLPSDB (Li *et al.*, 2020), which is a recently published database containing the details and outcomes of LLPS experiments collected from literature (see Section 2 for more details).

Each of those 896 experiments are annotated with experimental conditions such as temperature, the pH and salt concentration of the buffer, the presence of crowding agent and the protein concentration. We manually cleaned the data, we uniformed the measurement units and we translated them into a notation suitable for ML applications. For example, we transformed the salt concentration into ionic strength and we encoded the temperature as a binary 10-dimensional vector, in which each dimension represents a 10°C degree range. This allowed us to consider ranges of temperatures (e.g. <40°C, 50–60°C) as input. The total range of temperatures considered by Droppler goes from 0 to 99°C.

To evaluate the performance of Droppler, we ran a 5-fold stratified cross-validation. We used BLASTCLUST to ensure that the proteins in each cross-validation set shared <20% SI at 90% coverage with proteins in other sets, thus ensuring an unbiased evaluation of the performance. The final performance scores have been obtained by concatenating the cross-validation results and computing the metrics described in Section 2.

Supplementary Table S1 show Droppler's performance. Supplementary Figure S2 shows the corresponding ROC curve and the confusion matrix. From the relatively low AUC of 0.64, we can see that the task of predicting LLPS behavior of proteins given their sequence and the experimental conditions is quite hard. On the other hand, the Sensitivity of 0.75 indicates that Droppler is able to detect 75% of the positive cases, while discarding half of the negatives (Spe = 0.49). The precision score is also quite high (0.69), meaning that roughly 70% of the times Droppler predicts that a protein undergoes LLPS under certain conditions, the prediction is indeed correct. This means that our method can be used to prioritize experiments that are likely to provide positive results among all the possible experiments that can be performed, thus likely improve the efficiency of discovery of novel LLPS proteins or conditions sets.

3.2 There is no clear correlation between experimental conditions and LLPS

In Figure 2, we analyze the relation between the experimental conditions and the LLPS label and the Droppler predictions over the entire dataset. In this plot, we compare the distribution of the conditions considered by Droppler (pH, crowding agent, salt and protein concentration) with respect to the positive and negative labels (LLPS positive, LLPS negative) and with respect to Droppler predictions (positive preds, negative preds), trying to detect a possible correlation or linear dependence. This plot shows that the conditions by themselves do not correlate with the LLPS propensity of the proteins. The Pearson correlation between each condition and the LLPS labels is always ≤ 0.07. Moreover, we see that not even Droppler predictions linearly correlate with these conditions, indicating that (i) the interaction between experimental conditions and protein sequence is indeed much more complex and (ii) also the modeling performed by Droppler does not rely on univariate correlations.

3.3 Droppler can be used to explore the LLPS experimental conditions landscape

Droppler predictions are nearly instantaneous, meaning that it can be used to perform a large scale *in silico* exploration of the experimental conditions space and find the combinations that are more likely to allow the protein *S* to form LLPS condensates. Due to the emergent nature of protein folding and of the LLPS behavior, the landscape of the LLPS likelihood as a function of the protein sequence plus the five experimental conditions considered is highly complex and thus requires an highly non-linear model such as the multi-head attention architecture (see Section 2).

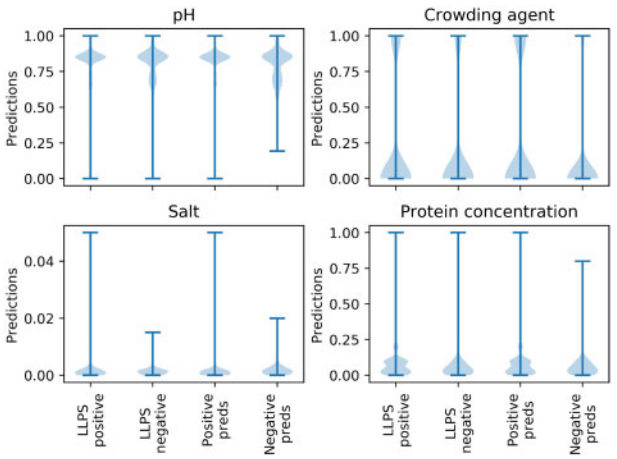

Fig. 2. Plot showing the distribution of the target conditions used by Droppler (pH, crowding agent, salt and protein concentration) with respect to the positive and negative labels (LLPS positive, LLPS negative) and with respect to Droppler predictions (positive preds, negative preds)

Figure 3A and B show how the predicted LLPS likelihood for the heterogeneous nuclear ribonucleoprotein A1 (Uniprot ID: P09651) changes according to the variation of protein concentration and, respectively, the pH (Fig. 3B) and the ionic strength of the buffer (Fig. 3A). The other experimental conditions (temperature, presence of crowding agent) are kept constant. The red and blue dots represent the real outcomes of experiments performed on P09651 in the same experimental conditions, and have been extracted from the LLPSDB database (Li *et al.*, 2020). The red dots indicate experiments successfully showing LLPS behavior, while the blue dots indicate that no phase transition was registered at those combinations of conditions.

3.4 Analysis of the temperature change in the *in silico* experiments

Temperature is another factor that influences the LLPS behavior (Ambadipudi *et al.*, 2017; Molliex *et al.*, 2015; Nott *et al.*, 2015). Droppler encodes the temperature information in a binary 10-dimensional vector, thus allowing ranges of temperature. Each dimension represents a bin of 10 degrees, and Droppler has a range of 0–99°C.

By running Droppler, we can investigate the relationship between temperature, protein concentration and LLPS behavior *in silico*, exploring the entire landscape determined by the variation of these two experimental conditions. In Figure 3C, we show this LLPS landscape for the ATP-dependent RNA helicase laf-1 protein (D0PV95). The blue and red dots are the experiments annotated in LLPSDB as LLPS000008. Red dots indicate sets of conditions at which D0PV95 undergoes LLPS, while blue dots are negative experimental outcomes. We can see that the LLPS likelihood predicted by Droppler mirrors the experimental results, since it assigns lower (darker) colors in the regions with blue dots and lighter colors where most of the positive experiments (red dots) are located.

3.5 Experimental validation on human TAR DNA-binding protein 43 shows results consistent with the predictions

We experimentally validated the ability of Droppler to predict the LLPS behavior of the terminal domain of TAR DNA-binding protein 43 (TDP-43_LCD) given varying experimental conditions. To do so,

Downloaded from https://academic.oup.com/bioinformatics/article/37/20/3473/6275261 by INACTIVE user on 29 October 2024

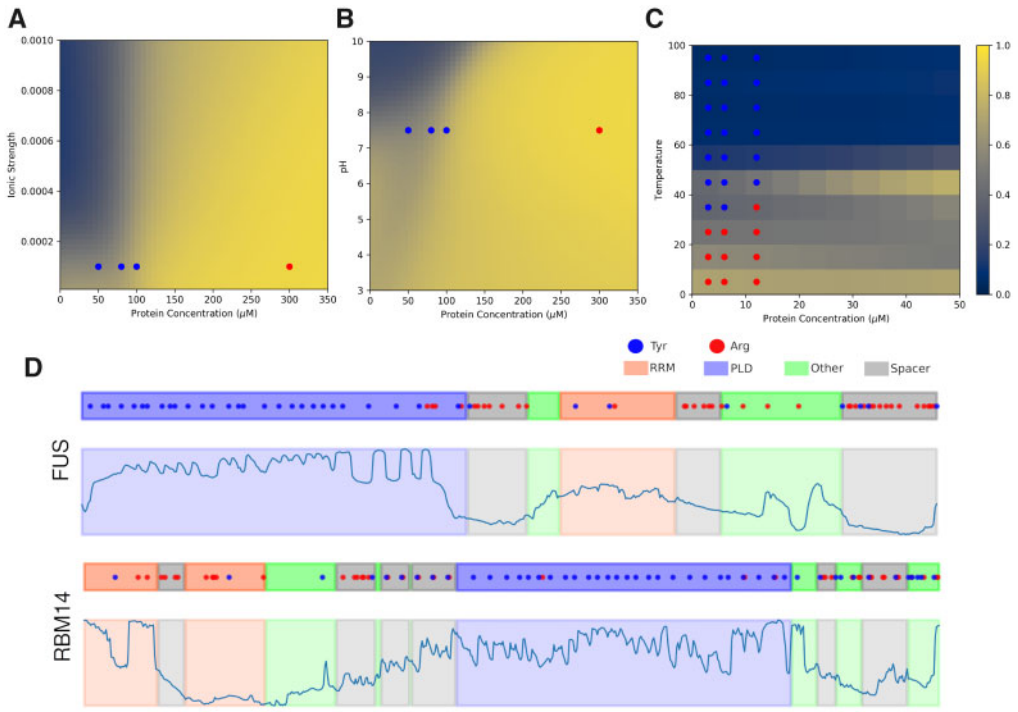

Fig. 3. Computational evaluation of the NN outputs. (**A–C**) The effects of different experimental conditions on the predicted LLPS likelihood for heterogeneous nuclear ribonucleoprotein A1 (**A** and **B**) and ATP-dependent RNA helicase laf-1 (**C**). (**A**) How the combination of pH and protein concentration influences the predicted LLPS likelihood, (**B**) the ionic strength versus protein concentration likelihood landscape and (**C**) the temperature versus protein concentration. Red dots represent actual experiments extracted from LLPSDB (Li *et al.*, 2020) that showed the formation of LLPS, while the blue ones experiments that did not show the presence of LLPS. (**D**) The output of the attention branch obtained from the application of Droppler to FUS and RBM14. The colored regions represent the domain annotation as predicted by PSPer: blue: PLD, red: RRM, gray: spacer, green: other. The blue and red dots represent the tyrosines and arginine, respectively

we used Droppler to screen the salt concentration and pH conditions that would, respectively, minimize and maximize the LLPS propensity of TDP-43_LCD. Temperature and concentration have been kept constant (25°C and 20 μM, respectively). The condition at which Droppler predicted the highest LLPS probability is high pH (>10) and no salt, while the conditions that minimize the LLPS probability are low pH (<4) and salt concentration of 0.018 M. To verify these predictions, we expressed the protein and tested how the two buffers modified its behavior. Figure 4 shows the results of the experiments: panels C and D are the images obtained with transmission microscopy for condition 1 (low predicted LLPS probability) and 2 (high predicted LLPS probability). Panels A and B show the images obtained with fluorescent microscopy for condition 1 and 2, respectively, and panel E shows the difference in absorbance at OD600.

From the four microscopy images (Fig. 4A–D) it can be noticed that buffer number 2 makes TDP-43_LCD form a cloud of small droplets, while the first one only generates few larger and sparse droplets. The usage of buffer 2 results therefore in the detection of diffuse fluorescence when observed with a fluorescence microscopy. The average number of droplets observed per image acquired (taking the mean of three observations as true value) is 63.7 and 200.0 for condition 1 and 2, respectively. The measure of the absorbance at OD600 is consistent with what observed at the microscopy, showing a much larger absorbance at in condition 2.

It is important to notice that TDP-43_LCD is known to undergo LLPS, so the relevance of Droppler's prediction is that it is able to identify regions in the experimental conditions space in which its LLPS capability is reduced.

3.6 The neural attention detects protein regions that are crucial for LLPS

Droppler is an NN model that uses a multi-head neural attention architecture to process the input sequences (Raimondi *et al.*, 2020; Vaswani *et al.*, 2017). This means that, as shown in Figure 1, the part of its architecture devoted to sequence processing is composed by two sub-networks, called P and A, which are, respectively, tasked to extract predictive values from the sequence (P) and determine to which protein regions the network should *focus* its attention (A). One interesting aspect of neural attention architectures is that they allow a certain level of understanding of their inner decision process, since just by looking at where the NN is focusing its *attention* we can identify the regions that have been deemed most relevant for the prediction.

From Droppler, we thus extracted the per-residue attention values produced while predicting the RNA-binding protein FUS (P35637) and the RNA-binding protein 14 (Q96PK6), whose biophysical properties in relation to its LLPS behavior have been experimentally investigated in great detail in Wang *et al.* (2018). We decided to use these two case studies because in the aforementioned paper there is an in-depth analysis of the features and relative mutations that enhance or reduce their capability of undergo to LLPS. Moreover, they are involved in many cellular processes (Yamaguchi and Takanashi, 2016) and they are known to form liquid droplets *in vitro*. Their LLPS behavior is determined by the interaction of regions with different biophysical and structural characteristics (Orlando *et al.*, 2019; Wang *et al.*, 2018) and these features are also common to certain number of evolutionarily unrelated proteins, called the FUS-like family (Orlando *et al.*, 2019; Wang *et al.*, 2018).

The FUS-like proteins have a peculiar organization (Orlando *et al.*, 2019), which consists in two main elements: a large Tyr-rich disordered domain, called PLD, and one or more RNA-binding motifs (RRMs). PLD and RRM are separated by shorter Arg-rich linker regions called Spacers (Orlando *et al.*, 2019; Wang *et al.*, 2018).

The upper panel of Figure 3D shows the attention profile extracted from Droppler for the human FUS protein (P35637), along with the annotations of PLD, RRMs and Spacers regions on P35637 annotated with Orlando *et al.* (2019) in order to normalize the attention values and allow an easier visualization, we applied a

Downloaded from https://academic.oup.com/bioinformatics/article/37/20/3473/6275261 by INACTIVE user on 29 October 2024

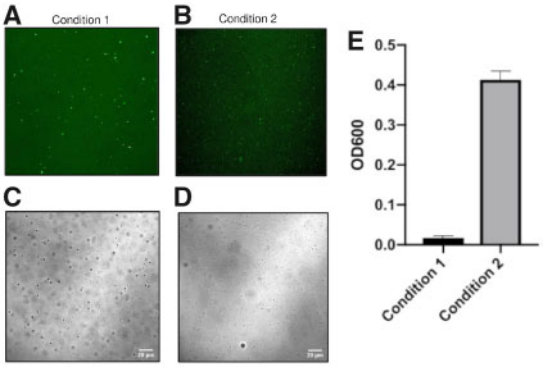

Fig. 4. Experimental evaluation of the effect of different buffers on the behavior of TDP-43_LCD: (**C** and **D**) images obtained with transmission microscope for conditions 1 (low LLPS probability) and 2 (high LLPS probability), respectively. (**A** and **B**) Images obtained with fluorescence microscope for conditions 1 and 2, respectively. (**E**) The difference in absorbance at OD600 for the two conditions

quantile normalization to the attention scores (Pedregosa *et al.*, 2011). We can see that Droppler focuses most of its attention on the PLD domain (blue), that is indeed crucial for the LLPS behavior of P35637, due to the cation–π interactions occurring between Tyr and Arg side-chains (Vernon *et al.*, 2018; Wang *et al.*, 2018). The NN focuses also on the RRM region (red), which is indeed also very important (Orlando *et al.*, 2019; Wang *et al.*, 2018) and in two smaller peaks in the uncharacterized 'Other' annotations.

In the lower half of Figure 3D we plotted the attention profile for the RNA-binding protein 14 (Q96PK6). Similarly to the previous example, we see that Droppler focuses most of its attention on the central PLD region (blue), but also on the first RRM domain (red), with smaller peaks in two of the Other regions (green).

These two examples show that Droppler focuses its attention on regions that are known to be essential for the LLPS of the protein (Wang *et al.*, 2018). In particular, it learned to recognize the PLD and RRM domains, without being explicitly taught of their relevance. It is important to highlight that Droppler is completely agnostic about these features. In other words, we do not define them in the encoding scheme, but the NN learns them automatically from the data.

Further interpretation studies on more complex models based on larger amounts of data could indeed provide insights on the NN's take about the molecular processes behind LLPS in different proteins.

4 Discussion

4.1 Droppler solves a different prediction problem with respect to existing methods

Droppler is, to the best of our knowledge, the first LLPS predictor able to predict *in silico* the outcome of *in vitro* LLPS experiments given different proteins and experimental conditions. The growing interest in LLPS has been rapidly followed by the development of computational methods able to prioritize proteins that are likely to undergo LLPS (Orlando *et al.*, 2019; Saar *et al.*, 2020; Sun *et al.*, 2019; Vernon and Forman-Kay, 2019), in an effort towards helping reducing the search space that must be experimentally visited in order to discover new LLPS proteins. These methods nevertheless perform their predictions while considering only the target protein sequence, thus ignoring environmental conditions such as temperature, pH, protein concentration, which are crucial for the LLPS behavior (Saar *et al.*, 2020; Sun *et al.*, 2019). This implementation choice was due to the fact that, until very recently, only extremely limited data regarding LLPS proteins were available. For example, this scarcity of data forced us to adopt an unsupervised, rule-based model for our previous attempt at modeling LLPS behavior (Orlando *et al.*, 2019).

With the publication of the LLPSDB database (Li *et al.*, 2020), which collects in a standardized way the outcome and the details of hundreds of LLPS experiments, we could finally relax the *sequence only* limitation of the previous attempts at modeling the LLPS behavior, including also the experimental conditions (in the form of temperature, pH, protein concentration, crowding agent) in our model. We thus trained a supervised model with the goal of replicating *in silico* the outcome of the hundreds of *in vitro* LLPS experiments that have been published so far. Droppler thus addresses a different prediction problem with respect to the existing LLPS predictors, which focus only on the protein sequence. Given a protein P, Droppler is indeed trained to predict it as LLPS if the associated experimental conditions are suitable, or non-LLPS if the conditions do not allow the phase transition. Figure 3C show an example of this behavior, where the same protein (laf-1) shows a temperature-driven LLPS capability. Due to the different nature of the prediction problem solved by Droppler and existing LLPS predictors, it is thus not possible to provide a fair comparison between them, because they are conceptually different: the existing tools cannot differentiate between the LLPS behavior of laf-1 above 40°C from laf-1 below 30°C, which, with the remaining conditions fixed, respectively, impair or allow LLPS. Our tool therefore expands the boundaries of the state of the art models, including experimental conditions as variables of the model.

4.2 Predicting the outcome of *in vitro* LLPS experiments is *hard*

The possibility to screen the optimal experimental conditions for the emergence of the LLPS behavior with Droppler could significantly reduce both the time and resources required to generate new experimental data, thus boosting the discovery of new LLPS proteins.

Nevertheless, truly understanding the LLPS molecular mechanisms and driving forces is an extremely complex task and the currently available data are insufficient to model all its facets, and this is mirrored by the relatively low prediction performances of Droppler. Notwithstanding these difficulties, we believe it is important to start these attempts at modeling the LLPS behavior including the experimental conditions, as a building block for future approaches.

In our view, the major aspects that currently impair the quality of the prediction are the following. First, the available experimental data are very sparse. It is indeed still rare to find experiments that are performed in similar settings, providing a more or less 'continuous' landscape for a certain experimental condition. The NN is thus forced very often to impute the missing data. We expect the quality of the ML model to increase with the amount of available data. Second, the available experimental conditions are sometimes inconsistent or not specific enough. An example of this are the 'ranges' of temperature (>40°C, 0–20°C, >281 K) that are sometimes reported instead of specific values. This clearly reduces the amount of information available for the model on the 'temperature' dimension, which might carry instead crucial information when it comes to predict the LLPS behavior. Third, there is likely a bias towards positive results (successful LLPS experiments), because there are way more studies that report conditions in which LLPS occurs, due to the fact that negative results (that are would be extremely valuable for machine leaning purposes) are generally considered 'less interesting' by the scientific community. Finally, the problem presents severe intrinsic difficulties, because determining which proteins are able to undergo LLPS given certain environmental condition heavily depends on folding-related structural properties of the target proteins and thus these aspects have to be modeled from the input sequence, which is a non-trivial task. Moreover, the relation of these properties with the environmental conditions is highly non-linear. For example, some proteins increase their LLPS propensity with temperature, some decrease it and some have a range of temperatures in which they undergo LLPS. Temperature is only one of the conditions we consider, and also the others (pH, salt concentration, protein concentration) are likely to show similar behaviors, making

Downloaded from https://academic.oup.com/bioinformatics/article/37/20/3473/6275261 by INACTIVE user on 29 October 2024

this problem non-linear over a complex multi-dimensional landscape.

Nevertheless, even if including the experimental conditions causes the prediction problem to become more complex with respect to considering only the protein sequence, we believe that this is the most realistic way to address the LLPS prediction, because for each protein that is able to undergo LLPS, we can imagine a set of protein concentration, salt concentration, temperature at which the LLPS does not occur.

Even if the results obtained with Droppler are promising, this is just a first attempt to discover the biophysical rules that stay behind LLPS. The relatively low performances of the model highlight the necessity of additional efforts in the subject, especially for proteins and experimental settings in which our model fails to provide a reliable prediction. In order to facilitate this, in Supplementary Table S2 we provide a list of the experiments for which the prediction are completely wrong (difference between the ground truth and the prediction >0.9).

4.3 The neural attention architecture is suitable for sequence-based prediction tasks

Another novelty in the approach we presented in this article is the multi-head attention NN architecture for protein sequences. Performing inference on protein sequences is a common task in bioinformatics, but most of the existing ML methods (e.g. SVM, Random Forest) are not able to natively deal with variable-length input sequences, because they expect a fixed-size input. People circumvent this issue using, for example, various forms of quantization of the sequences (Clark and Radivojac, 2014; Leslie *et al.*, 2001; Yang *et al.*, 2016) or compression (Raimondi *et al.*, 2018), but these methods almost always imply either (i) the loss of the intrinsic *sequential* nature of the input or (ii) a loss of information due to the compression needed to *shrink* arbitrary length proteins into a fixed-size feature vector (Raimondi *et al.*, 2018).

The main advantages of our neural attention are that it can directly take as input sequences of any length (i) without requiring pre-processing (thus making it an end-to-end approach), (ii) without loss of information due to compression and (iii) while preserving the information encoded in the sequential ordering of the amino acids. Second, even though the NN used is arbitrarily complex and highly non-linear, the neural attention mechanism provides an elegant tool for the interpretation of the predictions, because attention makes straightforward to look at the regions of the input sequence that have been used to compute the predictions, as shown for the Q96PK6 and P35637 proteins in Figure 3.

Another peculiarity is that the end-to-end information flow within Droppler is as straightforward as possible. For example, we encoded each protein sequence into a 20-dimensional trainable embedding, thus letting the NN optimize the numeric description of each amino acid specifically for the task at hand, avoiding preprocessing the sequence by encoding it into a feature vector using predicted or pre-computed features such as biophysical propensity scales, which have been shown to be sub-optimal for many similar bioinformatics tasks (Raimondi *et al.*, 2019).

Finally, the GPU-ready pyTorch (Paszke *et al.*, 2017) implementation allows Droppler to predict thousands of sequences and conditions in seconds, making it useful to rapidly screen both the mutational and the experimental LLPS landscape of proteins.

Funding

D.R. is funded from an FWO post-doctoral fellowship. The Switch Laboratory was supported by grants from the Flanders institute for biotechnology (VIB), the University of Leuven and the Flanders Funds for Scientific Research Flanders (FWO) (grant numbers I011220N, I011620N, G0H1716N).

Conflict of Interest: none declared.

References

Altschul,S.F. *et al.* (1990) Basic local alignment search tool. *J. Mol. Biol.*, **215**, 403–410.

Ambadipudi,S. *et al.* (2017) Liquid–liquid phase separation of the microtubule-binding repeats of the Alzheimer-related protein tau. *Nat. Commun.*, **8**, 275.

Banani,S.F. *et al.* (2017) Biomolecular condensates: organizers of cellular biochemistry. *Nat. Rev. Mol. Cell Biol.*, **18**, 285–298.

Clark,W.T. and Radivojac,P. (2014) Vector quantization kernels for the classification of protein sequences and structures. In: *Biocomputing 2014*, World Scientific, pp. 316–327.

Feric,M. *et al.* (2016) Coexisting liquid phases underlie nucleolar subcompartments. *Cell*, **165**, 1686–1697.

Leslie,C. *et al.* (2001) The spectrum kernel: a string kernel for SVM protein classification. In: *Biocomputing 2002*, World Scientific, pp. 564–575.

Li,Q. *et al.* (2020) LLPSDB: a database of proteins undergoing liquid–liquid phase separation in vitro. *Nucleic Acids Res.*, **48**, D320–D327.

Mészáros,B. *et al.* (2020) PhaSePro: the database of proteins driving liquid–liquid phase separation. *Nucleic Acids Res.*, **48**, D360–D367.

Molliex,A. *et al.* (2015) Phase separation by low complexity domains promotes stress granule assembly and drives pathological fibrillization. *Cell*, **163**, 123–133.

Nott,T.J. *et al.* (2015) Phase transition of a disordered nuage protein generates environmentally responsive membraneless organelles. *Mol. Cell*, **57**, 936–947.

Orlando,G. *et al.* (2016) Observation selection bias in contact prediction and its implications for structural bioinformatics. *Sci. Rep.*, **6**, 36679.

Orlando,G. *et al.* (2019) Computational identification of prion-like RNA-binding proteins that form liquid phase-separated condensates. *Bioinformatics*, **35**, 4617–4623.

Paszke,A. *et al.* (2017) *Automatic Differentiation in PyTorch*. In: Advances in Neural Information Processing Systems 32, Curran Associates, Inc, pp. 8024–8035.

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

Raimondi,D. *et al.* (2018) Ultra-fast global homology detection with discrete cosine transform and dynamic time warping. *Bioinformatics*, **34**, 3118–3125.

Raimondi,D. *et al.* (2019) Exploring the limitations of biophysical propensity scales coupled with machine learning for protein sequence analysis. *Sci. Rep.*, **9**, 1–11.

Raimondi,D. *et al.* (2020) Insight into the protein solubility driving forces with neural attention. *PLoS Comput. Biol.*, **16**, e1007722.

Saar,K.L. *et al.* (2020) Machine learning models for predicting protein condensate formation from sequence determinants and embeddings. *bioRxiv*.

Shin,Y. and Brangwynne,C.P. (2017) Liquid phase condensation in cell physiology and disease. *Science*, **357**, eaaf4382.

Sun,T. *et al.* (2019) Prediction of liquid-liquid phase separation proteins using machine learning. SSRN, 3515387.

Uversky,V.N. (2017) Protein intrinsic disorder-based liquid–liquid phase transitions in biological systems: complex coacervates and membrane-less organelles. *Adv. Colloid Interface Sci.*, **239**, 97–114.

Vaswani,A. *et al.* (2017) Attention is all you need. In: *Advances in Neural Information Processing Systems*, pp. 5998–6008.

Vernon,R.M. and Forman-Kay,J.D. (2019) First-generation predictors of biological protein phase separation. *Curr. Opin. Struct. Biol.*, **58**, 88–96.

Vernon,R.M. *et al.* (2018) Pi-pi contacts are an overlooked protein feature relevant to phase separation. *Elife*, **7**, e31486.

Wang,J. *et al.* (2018) A molecular grammar governing the driving forces for phase separation of prion-like RNA binding proteins. *Cell*, **174**, 688–699.

Weber,S.C. and Brangwynne,C.P. (2012) Getting RNA and protein in phase. *Cell*, **149**, 1188–1191.

Xiao,M. *et al.* (2018) K-mer counting: memory-efficient strategy, parallel computing and field of application for bioinformatics. In: *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, IEEE, pp. 2561–2567.

Yamaguchi,A. and Takanashi,K. (2016) FUS interacts with nuclear matrix-associated protein SAFB1 as well as Matrin3 to regulate splicing and ligand-mediated transcription. *Sci. Rep.*, **6**, 35195.

Yang,H. *et al.* (2016) Identification of secretory proteins in mycobacterium tuberculosis using pseudo amino acid composition. *Biomed. Res. Int.*, **2016**, 5413903

You,K. *et al.* (2020) PhaSepDB: a database of liquid–liquid phase separation related proteins. *Nucleic Acids Res.*, **48**, D354–D359.

Downloaded from https://academic.oup.com/bioinformatics/article/37/20/3473/6275261 by INACTIVE user on 29 October 2024