**Prediction of Protein Interaction Dynamics by Graph Neural Networks**

Mehdi Koşaca, Melis Oktayoğlu, Burcu Özden, Ahmet Süreyya Rifaioğlu, Tunca Doğan\*, Ezgi Karaca\*

### Link to check:

### https://www.abhishaike.com/p/an-argument-for-integrating-molecular

### ABSTRACT

TO BE ADDED

### 

### INTRODUCTION

Understanding the dynamics of protein-protein interactions (PPIs) is essential for comprehending the functioning of biological processes, facilitating drug discovery, regulating cellular functions, and uncovering the biological mechanism of diseases. [[ref](https://doi.org/10.1016/j.sbi.2008.03.003),[ref](https://doi.org/10.1146/annurev-biochem-060614-034142),[ref](https://doi.org/10.1038/nchembio.232)]. Encoding PPI dynamics, especially for flexible or disordered segments, remains a challenge, although we have static structures validated through CASP and CAPRI. Approaches to address the challenges of PPI dynamics include experimental and computational techniques. Nuclear magnetic resonance (NMR) spectroscopy is widely used to explore the dynamics of small proteins from picosecond to millisecond timescale. Other experimental techniques (cryo-electron microscopy and X-ray crystallography) are insufficient to clarify the protein motions, however, the conformational distribution present in the density map can be used to catch possible motions [[ref](https://doi.org/10.1016/j.sbi.2023.102736), [ref](https://doi.org/10.1038/s42256-020-00290-y)]. Despite the challenges and lots of labor of experimental techniques, Molecular dynamic (MD) simulations are still more favored to explore the conformational changes and motion of the interface [[ref](https://pubs.acs.org/doi/10.1021/acs.chemrev.0c00534)].

To address the complexities in studying PPIs and MD, machine learning (ML) methods are increasingly necessary [[ref](https://doi.org/10.1007/s12038-019-9909-z)]. ML algorithms can quickly process large datasets, find patterns, and predict interactions that traditional methods can't [[ref](https://www.ijsr.net/archive/v9i1/ART20203995.pdf)]. Recent advances in deep learning methods have catalyzed many revolutions in the field of computational structural biology [[ref](https://doi.org/10.1016/j.patter.2020.100142) [ref](https://wires.onlinelibrary.wiley.com/doi/full/10.1002/wcms.1618)]. Graph Neural Networks (GNNs) could be a great alternative to analyze PPIs since they are designed to work with relational data. GNNs represent proteins as nodes and their interactions as edges in a graph, which matches the way proteins interact. They use message passing to share information between connected nodes, helping to capture the relationships between proteins. This makes GNNs good at learning complex interaction patterns and making accurate predictions about protein functions and interactions [[ref](https://ieeexplore.ieee.org/document/9046288#)]. In 2022, Réau et al [[ref](https://academic.oup.com/bioinformatics/article/39/1/btac759/6845451)] developed DeepRank-GNN algorithm that is used to learn problem-specific interaction patterns. It converts the atoms and residues of the protein into nodes and the covalent bonds and contacts into edges. It outperforms traditional docking methods in predicting the binding affinity and identifying critical interaction sites. ScanNet [[ref](https://doi.org/10.1038/s41592-022-01490-7)], is another GNN algorithm for predicting protein binding sites. It provides a probability for each amino acid being part of a binding site. ScanNet builds representations of proteins by considering both atomic and amino acid levels, learning the patterns in their spatial arrangements. This helps the model accurately identify which amino acids are likely involved in binding sites. These algorithms successfully used GNN architecture in characterizing the PPIs. However, proteins are dynamic macromolecules and despite the growing array of deep learning methods in computational biology, PPI dynamics are still explored by traditional force field techniques.

In this study, we aim to analyze time-based PPI dynamics with deep learning. We selected recurrent neural networks (RNNs) due to their ability to effectively capture and summarize the PPI dynamic sequences by generating local and global representations. [[ref](https://pubmed.ncbi.nlm.nih.gov/31113301/)]. By adjusting the parameters of an existing application, we implemented an approach called RNN-MD, which utilizes past interaction data from MD simulation and interfacea output (<https://github.com/JoaoRodrigues/interfacea>)[[ref](https://zenodo.org/record/3516439#.ZBQTnOxBxqs)] of the PPI to predict future interactions. For our analysis, we selected MD simulation data (100 ns) of the crystal structure of tissue factor (TF) in complex with the humanized Fab D3h4 (Fab) (PDB ID: 1JPS) due to its unique antigen-antibody binding strategy [[ref](https://doi.org/10.1021/acs.jctc.1c00336), [ref](https://doi.org/10.1006/jmbi.2001.5036)]. The TF:Fab interface consists of highly polar three heavy (H) and three light (L) patches as well as two hydrophobic patches (P), with H2 playing a crucial role in binding with TF. Then, we divided the interaction data into training, validation and test as such 80-10-10 ns, 50-25-25 ns, 25-37.5-37.5 ns split rules to compare the model performances in different training sizes. We also performed hyperparameter optimization on these splits and evaluated their performances by recall, precision, FPR, F1 score and Matthews Correlation Coefficient (MCC) score over common (≥50% frequency) and uncommon (<50% frequency) interactions. Notably, the maximum MCC values obtained for the common interactions were 0.63, 0.56, and 0.60, respectively. Our results demonstrate that RNN-MD is capable of accurately predicting future common interactions, even when trained on a limited dataset of historical time-series interaction data.

***Brief outline for the rest of the Introduction***

For such an acceleration ML methods are needed

GNN could be a great alternative here, why? List some of the methods

Expanding on the applicability of GNNs in PPI characterization, we explore the use of dynamic GNN in predicting time-based PPI dynamics

We choose RNN because it is capable of summarizing past event sequences by generating local and global representations, where the global representation retains the past graph information**.**

Introduction of the case

This case was selected because antigen antibody interactions have a unique binding strategy from others.

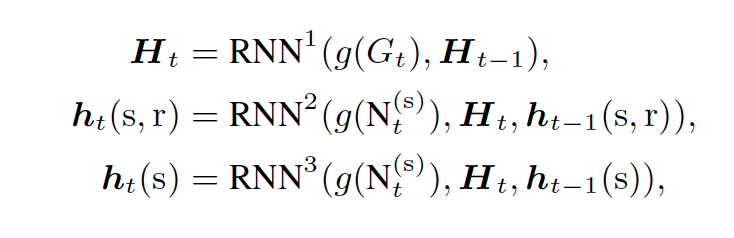
### METHODS

**Recurrent Neural Network for Molecular Dynamics model (RNN-MD)**

The implemented model we selected for our problem, RE-Net (Recurrent Event Network) (ref), is proposed for modeling temporal knowledge graphs (TKGs). The main challenge RE-Net addresses is two fold: sequential prediction of future events and concurrent events happening within a specific time frame which is specifically suited to our problem of prediction of interacting residues in the MD. In our study, each node in the GNN represents each interacting residue or atom, and the edges represent the type of interaction as hydrophobic, ionic, or hydrogen bonds.

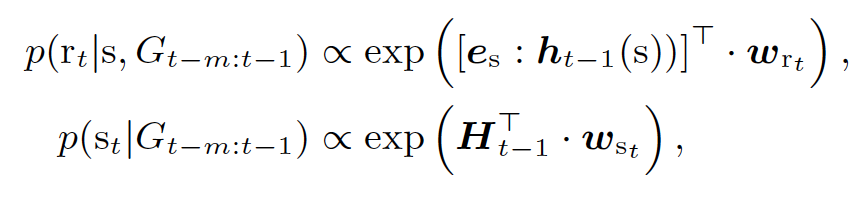
To tackle this prediction challenge, RE-Net uses autoregression to model the joint probability distribution of all events in a TKG. It breaks down event probabilities, residue/atom interactions in our context, into a combination of conditional distributions that depend on prior interaction events and graph structure, enabling the prediction of subject-object-relation triplets by sequentially sampling graphs.

The architecture of RE-Net consists of three key components. Firstly, there is a recurrent event encoder, implemented as a Recurrent Neural Network (RNN), to summarize past event sequences. It generates local (h) and global (H) representations, where the global representation retains the past graph information. The encoder recurrently utilizes the previous global and local representations, along with the aggregated global graph structure, g(Gt), and the aggregated local graph of a specific subject, g(Nt(s)) to encode the new global and local representations. These representations are then used to define the probabilities of interactions of objects and subjects, and the type of their relations using also learned static embedding vectors (es, er).

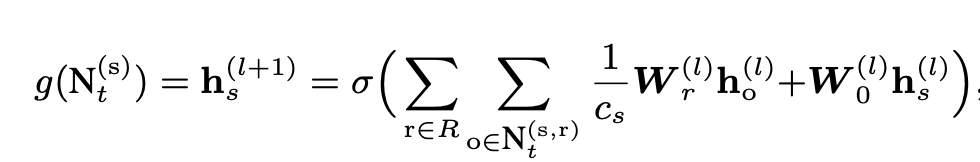


Text

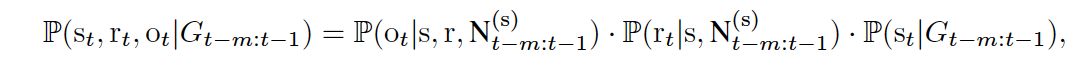
Description automatically generated with medium confidence



Secondly, RE-Net employs a neighborhood aggregator, utilizing the Multi-Relational Graph Convolutional Network (RGCN) aggregator function. This function operates on node features and the graph structure to generate new aggregated (g) node representations. It summarizes information from neighbors that are steps away from the original node in the graph structure, allowing the network to capture the complex relationship between nodes. This is especially important for our problem as distant neighbors in the atomic or residue interaction GNN can impact other interactions.



Finally, the architecture includes a multilayer-perceptron (MLP) decoder to estimate the interaction probabilities, using the joint probability of the probabilities calculated by the encoder.



To fit the model to our problem, we switched the num\_k parameter of the model, which describes the cutoff value for the top-k candidates because of the nature of our problem, the stereochemistry and the spatial arrangement of the residues limits the number of top candidates possible, making the top candidates significantly less than natural language problems like the original model intends to solve. Again, due to the constraints our problem poses, we also picked smaller batch sizes than the original model (with 1024 batch size) as we have significantly less data points. The model was tested in WIKI, ICEWS18, GDELT, and ICEWS14 datasets with all having data points in the factor of hundred thousands, whereas our dataset had about 6 thousand data points.

​​**Hyperparameter optimization and comparison of prediction performances**

Hyperparameter optimization was performed for interaction prediction of protein complexes by RE-Net. To do so, we aim to get maximum prediction performance by training certain simulation data. We changed the following RE-Net parameters to be optimized: dropout, learning rate, maximum epoch number, number of hidden, and batch size. We tried the parameters with the following values: dropout 0.1, 0.2, number of hidden 200, 300, 400, learning rate 1e-5, 1e-4, 1e-2, maximum epoch number 25, 50, 100, 150, and batch size 64. We tried the parameters one by one in pre-train, train, and test. Equation 1 shows the default parameters of the RE-Net for pre-train, train, and test steps.

**pretrain:** --gpu 0 --dropout 0.5 --n-hidden 100 --lr 1e-3 --max-epochs 30 --batch-size 128

**train:** --gpu 0 --dropout 0.5 --n-hidden 100 --lr 1e-3 --max-epochs 10 --batch-size 128 --num-k 5

**test:** --gpu 0 --n-hidden 100 --num-k 5

**Equation 1.** Default RE-Net parameters as they were used in this work.

Then, we compare the prediction performances of hyperparameters with recall, precision, FPR, F1 score, and MCC scores.

Recall (Equation 2.), also known as sensitivity or true positive rate, is calculated by dividing the number of true positives by the sum of true positives and false negatives.t assesses the accuracy of correctly predicting the ground truth within a test set, with a potential range from a maximum of 1 to a minimum of 0. A high recall indicates that the algorithm effectively captures most of the true positive interactions, minimizing false negatives.

**Equation 2.** Mathematical representation of recall. It is calculated as the ratio of true positive to the sum of true positive and false negative.

Precision (Equation 3.) quantifies the accuracy of positive predictions made by the algorithm, calculated as the number of true positives divided by the sum of true positives and false negatives with a potential range from a maximum of 1 to a minimum of 0. It is crucial for assessing the reliability of predicted interactions, as high precision indicates a low rate of false positives.

**Equation 3.** Mathematical representation of precision. It is calculated as the ratio of true positive to the sum of true positive and false positive.

False Positive Rate (FPR) (Equation 4.) represents the proportion of actual negative instances that are incorrectly predicted as positive by the algorithm. It is calculated as the number of false positives divided by the sum of false positives and true negatives with a potential range from a maximum of 1 to a minimum of 0. FPR is important for evaluating the algorithm's specificity and measuring the rate of false alarms in predicting interactions.

**Equation 4.** Mathematical representation of the FPR metric. It is calculated as the ratio of false positive to the sum of false positive and true negative.

The F1 score (Equation 5.) is the harmonic mean of precision and recall, providing a balanced measure of a model's performance. It is calculated as 2 times the product of precision and recall divided by the sum of precision and recall with a potential range from a maximum of 1 to a minimum of 0. The F1 score considers both false positives and false negatives, offering a comprehensive assessment of the algorithm's overall predictive accuracy.

**Equation 5.** Mathematical representation of F1 score.

MCC (Equation 6.) is a correlation coefficient that considers true and false positives and negatives. MCC is particularly useful for imbalanced datasets and provides a balanced measure of prediction performance. The MCC score ranges between -1 and +1, where +1 indicates a perfect prediction. 0 indicates a random prediction, and -1 indicates a total disagreement between the prediction and actual labels.

The next step involves summing all the numbers in the result matrix and determining the lengths that are greater than zero. Finally, the metric computes the division of the sum matrix by the length. The formula for this metric is provided below.

**Moleculer Dynamics Data**

We used molecular dynamics simulation data of a crystal structure of tissue factor (TF) in complex with the humanized Fab D3h4 (Fab) with 1JPS PDB ID. The 100 ns dynamic trajectory of this complex was retrieved from Jandova *et al.* [<https://doi.org/10.1021/acs.jctc.1c00336>]. The authors simulated 1JPS was simulated by using GROMACS simulation package version 2019 with CHARMM36m forcefield. We extracted a snapshot from the trajectory at every 500 ps, leading to the generation of 200 time-dependent coordinates for 1JPS.

The 1JPS structure contained three chains consisting of two antibody chains (heavy and light), and TF. We merged heavy and light chains of the MD ensemble (100 ns, 200 frames) into a single chain to be able to analyze Fab as a monomeric entity. For doing so, not to have overlapping residue numbering, we renumbered Fab’s heavy chain residues by adding 1000 to the original residue numbering. Then, for each snapshot, the non covalent interaction properties were calculated with the interfacea python package [[ref](https://zenodo.org/record/3516439#.ZBQTnOxBxqs)]. This library takes a complex pdb coordinate as an input and calculates the inter-chain hydrophobic, hydrogen bond, and salt bridge interactions. We converted the pairwise interaction data obtained from interfacea into a new format, where we only print relation and timesteps, such as “19 1 38 1” (First and third numbers represent the interacting residues located at different chains, second number represent the interaction type i.e. hydrophobic, and the last one denotes the time stamps.). Finally, we divided the converted simulation data according to time subsets as training, validation, and test sets. These followed three different split ratios, including 80-10-10 ns, 50-25-25 ns and 25-37.5-37.5 ns.

### 

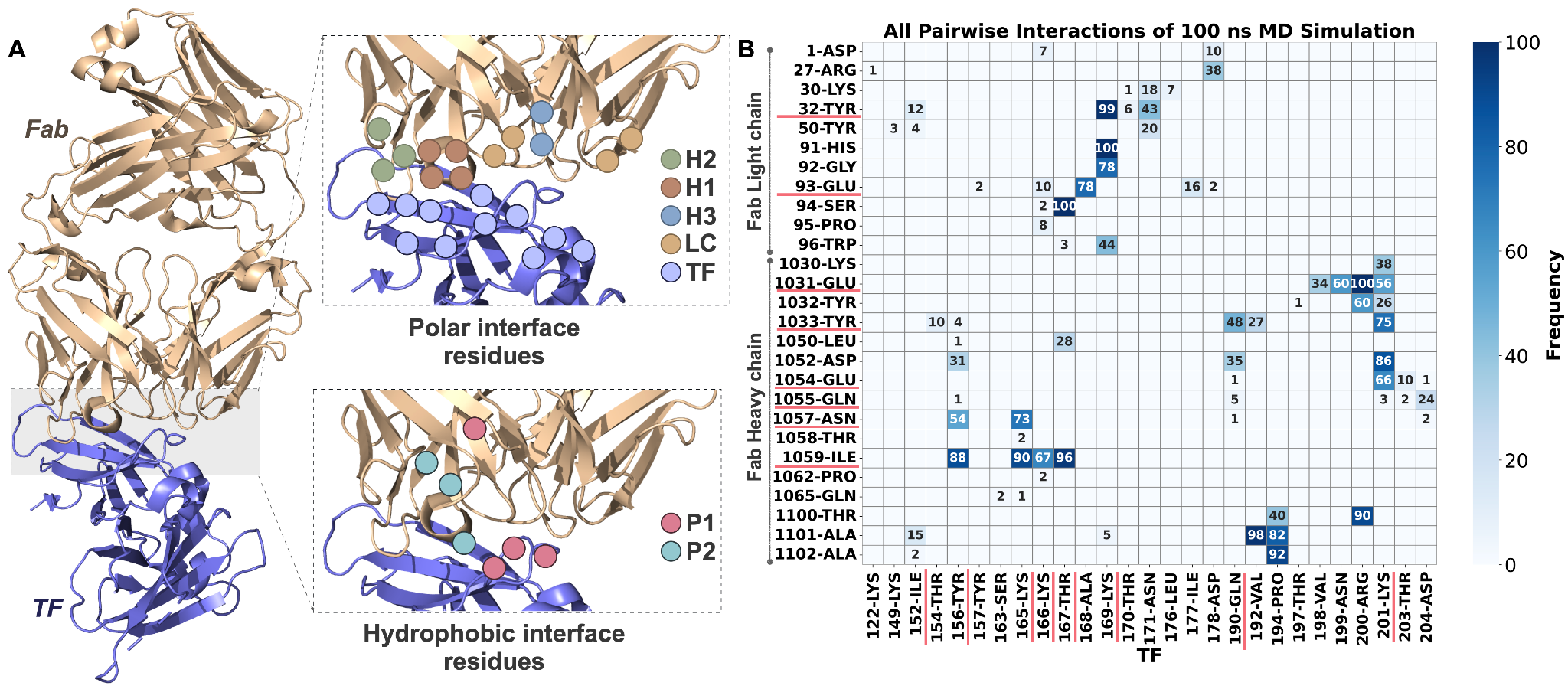
### RESULTS

**The static and dynamic interaction profiles of the TF:Fab complex**

TF is a membrane-anchoring protein. It is an essential cofactor of serine protease factor VIIa (F.VIIa) [https://pubmed.ncbi.nlm.nih.gov/9198138/]. Upon TF’s interaction with the soluble F.VIIa, the blood coagulation pathway is activated. Thus, this interaction plays an active role in thrombotic disorders. As a potential way to inhibit the formation of this interaction, an engineering humanized Fab (D3h44) was proposed. 1JPS represents the high-resolution structure of this TF:Fab interaction (<https://doi.org/10.1006/jmbi.2001.5036>).

The interaction between TF and Fab is predominantly polar, driven mainly by hydrogen bonding (Figure Sx) [<https://doi.org/10.1006/jmbi.2001.5036>]. On the Fab, polar interactions are concentrated on H2, with the involvement of 1052-Asp, 1055-Gln, and 1057-Asn residues (Figure 1A). Over the H1 of Fab, the secondary polar patch encompasses 1030-Lys, 1031-Glu, 1032-Tyr, and 1033-Tyr, while the smaller H3 patch involves 1100-Thr and 1101-Ala. The contribution of the light chain is limited to 32-Tyr, 50-Tyr, 92-Gly, and 94-Ser residues. On the TF side, 154-Thr, 156-Tyr, 167-Thr, 169-Lys, 171-Asn, 190-Gln, 197-Thr, 199-Asn, 200-Arg, 201-Lys, and 204-Asp stand out as the critical polar amino acids for binding (Figure 1A). Besides the large polar interaction site, there are two small hydrophobic patches across the TF:Fab surface. The first patch (P1), centered around 1097-Ala and extending towards 192-Val, 152-Ile, and 194-Pro of TF, is noteworthy. The second (P2), smaller patch comprises 1050-Leu, 1058-Ile, and 167-Thr (Figure 1A).

To predict the dynamic interaction profiles of TF:Fab, we used a 100 ns long MD simulation of this complex (Methods, <https://doi.org/10.1021/acs.jctc.1c00336>). Here, we calculated all the TF:Fab interactions observed throughout the simulation in every 0.5 ns (corresponding to 200 frames). The frequency of all interfacial contacts is demonstrated as a heatmap in Figure 1B, which revealed that all the critical interface residues from the crystal structure were retained throughout the simulation except for the ones made by 1097-Ala. To efficiently analyze the interaction network presented by the heatmap, we concentrated on the “well-connected” amino acids. An interfacial residue was considered to be well-connected, if it forms contacts with at least four for other residues on the partner chain (Figure 1B). On Fab; 32-Tyr, 93-Glu, 1031-Glu, 1033-Tyr, 1054-Glu, 1055-Gln, 1057-Asn, 1059-Ile, 1101-Ala, and on TF; 152-Ile, 156-Tyr, 165-Lys, 166-Lys, 167-Thr, 169-Lys, 190-Gln, and 201-Lys on TF fit to this criterion. On the Fab side, these interactions define H2 as the most interaction-dense spot, where the least amount of interactions was formed by L2 (Figure S1, S2, and S3). We expect these residues to be present as the main interaction points in our predictions.



**Figure 1. A. Structural depiction of the TF:Fab complex, with a zoomed view of its polar and hydrophobic interface residues.** TF is presented in salmon cartoons while Fab is demonstrated in wheat cartoons. The interface amino acids, forming the polar and hydrophobic patches are displayed as the inset. H1, H2, and H3 illustrate the heavy chain patches, LC denotes the light chain, TF represents the tissue factor, P1 and P2 signify the hydrophobic patches. **B. The frequency of all pairwise interactions observed during 100 ns MD simulation (ground truth).** Underlined interactions highlight the well-connected amino acids of Fab and TF.

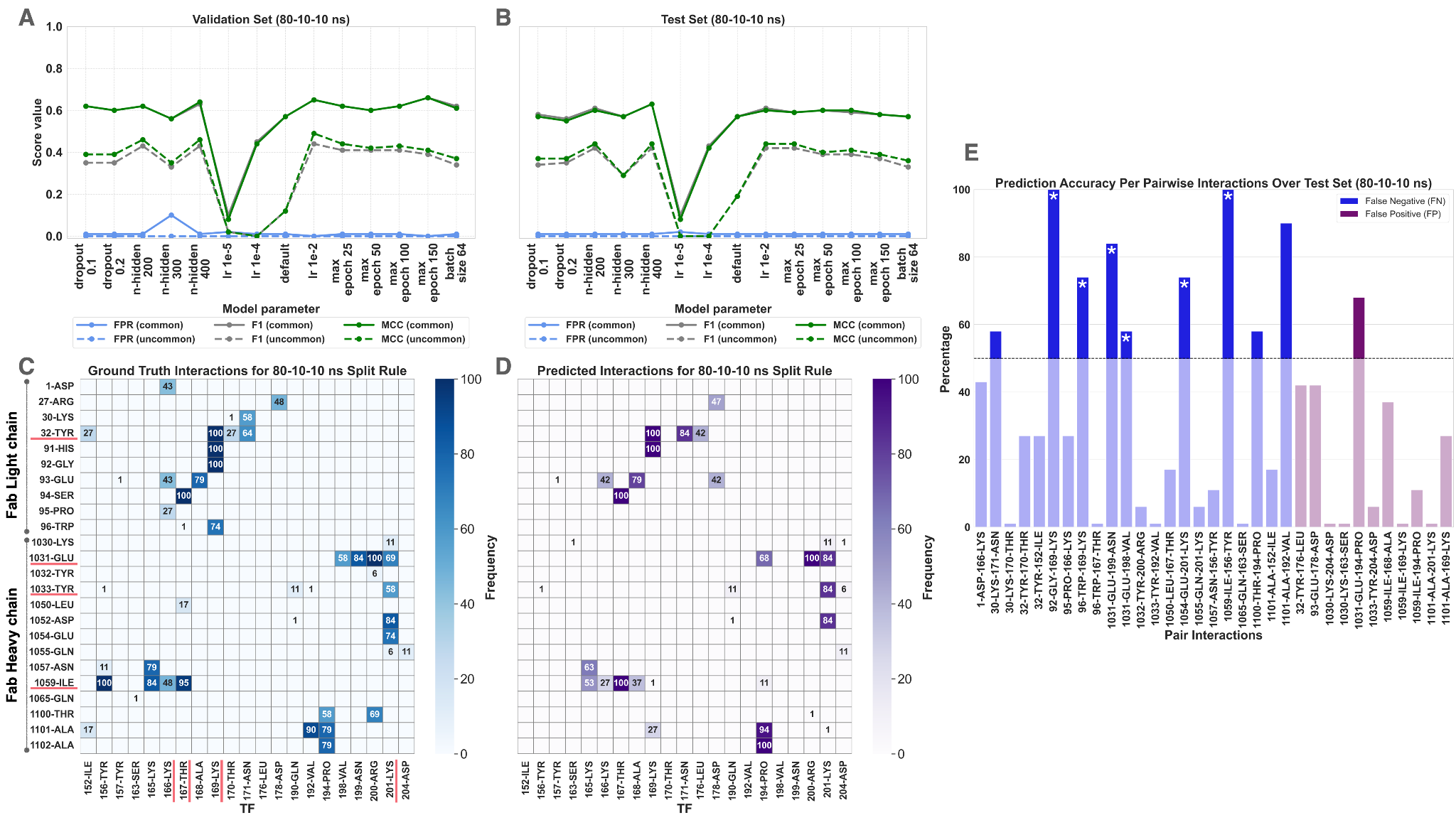
**RNN-MD aims at reproducing the dynamic interaction profile of TF:Fab complex**

We chose an RNN approach to reproduce the dynamic interaction profile of TF:Fab observed within its 100 ns long MD simulation (RNN-MD). To train our RNN-MD algorithm, instead of using the actual coordinates, we fed the pairwise inter-protein interactions calculated for each frame (200 frames, leading to 24 pairwise interactions per time point on average, see Methods). Considering three different time cut-offs, this interaction list was divided into training, validation, and test subsets, using three different split rules: 80-10-10 ns, 50-25-25 ns, and 25-37.5-37.5 ns. We applied our RNN-MD model on each split ratio, where we performed hyperparameter optimization to identify the best parameter set for reproducing the dynamic interaction profiles. For this, we modified RNN-MD’s dropout (parameter range: 0.1, 0.2, 0.5), number of hidden layers (100, 200, 300, 400), learning rate ( lr 1e-2, 1e-3, 1e-4, 1e-5), max epoch (10, 25, 50, 100, 150), and batch size (64, 128) (default values underlined). The performance of each hyperparameter was measured by False Positive Rate (FPR), F1, and Matthews Correlation Coefficient (MCC). FPR, F1, and MCC were calculated over common (≥50% frequency) and uncommon (<50% frequency) interactions.

**RNN-MD could reproduce most of the common interactions of the last 10 ns**

We initially trained our RNN-MD model with 80 ns dynamic interaction profiles to predict the final 10 ns contact set. Within this perspective, over various hyperparameters, we observed similar trends for FPR, F1, and MCC over validation and test sets (Figure 2A,B). Over the test set, for common interactions, FPR always remains low (ranging between 0.00 and 0.01). This indicates that RNN-MD predicted most of the ground truth interactions accurately. MCC values, on the other hand, span a wide accuracy extent, ranging from 0.08 to 0.63. The lowest MCC value was observed for low learning rates, i.e., 1e-5 and 1e-4, while the highest MCC was achieved when we set the number of hidden layers to 400. High MCC scores were obtained also for maximum epoch numbers 50 and 100. The same performance evaluations were observed for uncommon interactions. F1 behaved similarly to MCC. In the light of these findings, we chose 400 number of hidden layers as our best hyperparameter.

In the ground truth test set, we had 24 common and 22 uncommon interactions, where 62.5% of the common and 59.0% of the uncommon ones could be reproduced by RNN-MD (Figure 2C,D). The mispredicted common interactions belong to 1031-Glu, 1054-Glu, 1059-Ile, and 1101-Ala residues on Fab; 156-Tyr, 169-Lys, and 201-Lys residues on TF. Among these, the false negative interactions predicted for the well-connected amino acids correspond to 1031-Glu:199-Asn, 1031-Glu:198-Val, 1054-Glu:201-Lys, 1059-Ile:156-Tyr, 92-Gly:169-Lys, 96-Trp:169-Lys (Figure 2E). It is promising to see that RNN-MD had only one false positive common interaction (1031-Glu-194-Pro), explaining the low FPR rates.

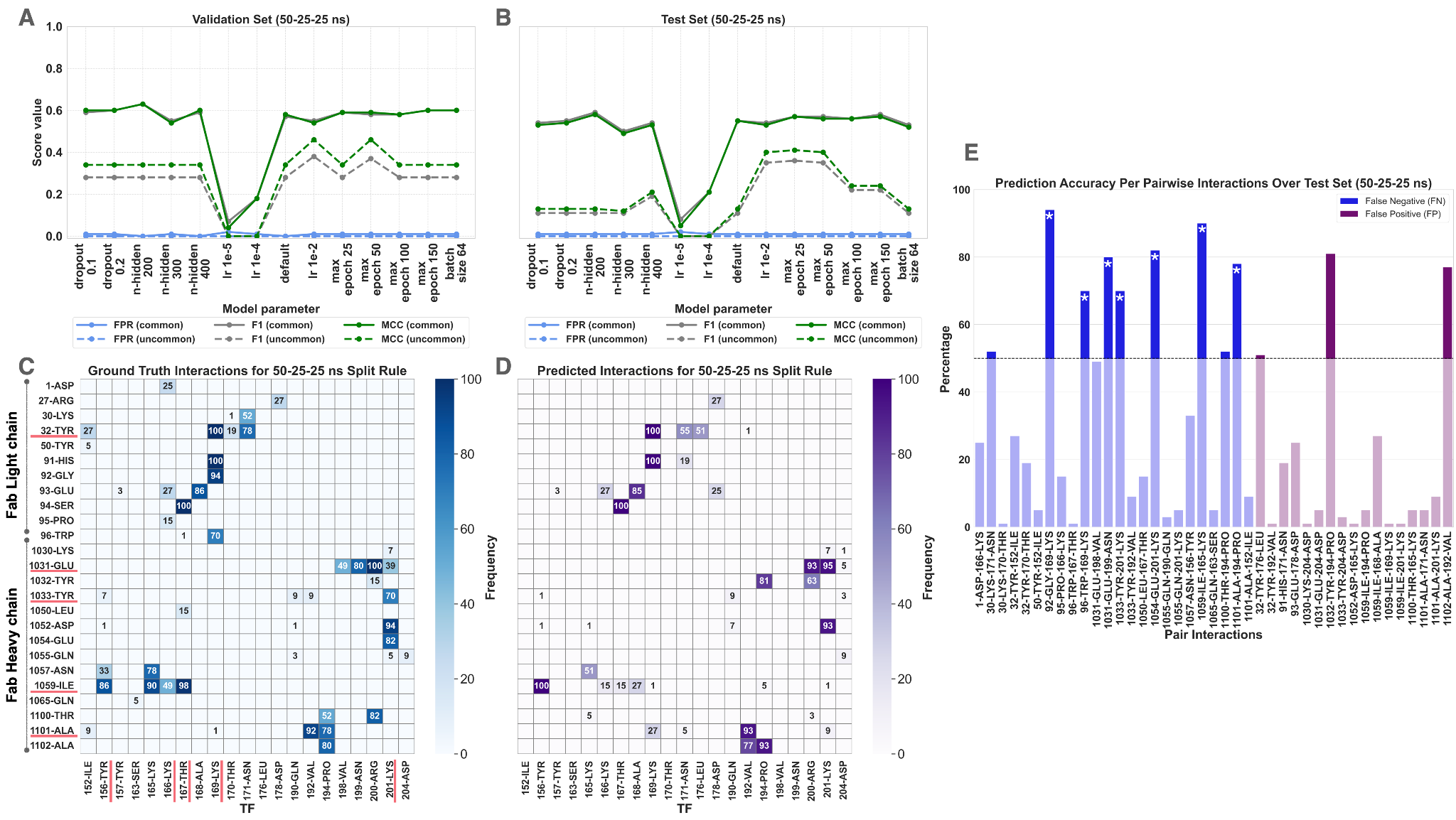


**Figure 2. The evaluation of FPR, F1, and MCC scores calculated for different RNN-MD hyperparameters over A. validation and B. test sets for the 80-10-10 split rule**. The default RNN-MD hyperparameter values are 0.5 for dropout, 100 for number of hidden layers, 1e-3 for learning rate (lr), 10 for max epoch, and 128 for batch size. RNN-MD training was performed over the first 80 ns where the validation was performed over the next 10 ns. The test set covered the final 10 ns. Thick lines correspond to common interactions and the dashed ones correspond to uncommon ones. **C**. **The frequency of ground truth and D. predicted interactions over the test set.** The prediction set was calculated with the number of hidden = 400, where the rest of the parameters were left to their default values. Underlined interactions represent the well-connected residues on Fab and TF. **E. False negative and false positive interaction frequencies over the test set.** The blue bars correspond to the ground truth interactions that cannot be predicted by RNN-MD (false negatives), while the purple ones represent the wrongly produced ones (false positives). The uncommon interaction cutoff was presented with a dashed line.

**Reduction in the training simulation time from 80 ns to 50 ns did not decrease the prediction performance significantly**

To challenge RNN-MD in predicting the dynamic interaction profiles of TF:Fab, we decreased the training simulation time to 50 ns, while increasing the test simulation time to 25 ns. In this scenario, too, FPR, F1, and MCC behaved similarly at different hyperparameters over validation (25 ns) and test sets (Figures 3A,B). For this split set, MCC ranged between 0.05 and 0.58. Over the common interactions, the highest MCC was achieved when the number of hidden was set to 200 and the maximum epoch was set to 25. Over the uncommon interactions, 25 maximum epoch still led to a high MCC (0.41) while 200 number of hidden resulted in a significantly smaller one (0.13). As F1 followed the same trend, 25 maximum epoch was chosen as RNN-MD’s optimal hyperparameter.

In the 25 ns long test set, there were 22 common and 27 uncommon interactions. RNN-MD could predict 59.0% of the common interactions and 44.4% of the uncommon ones. The mispredicted common interactions come from 1031-Glu, 1033-Tyr, 1054-Glu, 1059-Ile, and 1101-Ala residues on Fab; 165-Lys, 169-Lys and 201-Lys residues on TF. Seven of such interactions belong to well connected amino acids (1031-Glu:199-Asn, 1033-Tyr:201-Lys, 1054-Glu:201-Lys, 1059-Ile:165-Lys, 1101-Ala:194-Pro, 92-Gly:169-Lys, and 96-Trp:169-Lys, Figure 2E). With the increasing prediction challenge, RNN-MD produced three false positive common interactions (1032-Tyr:194-Pro, 1102-Ala:192-Val, 32-Tyr:178-Asp).

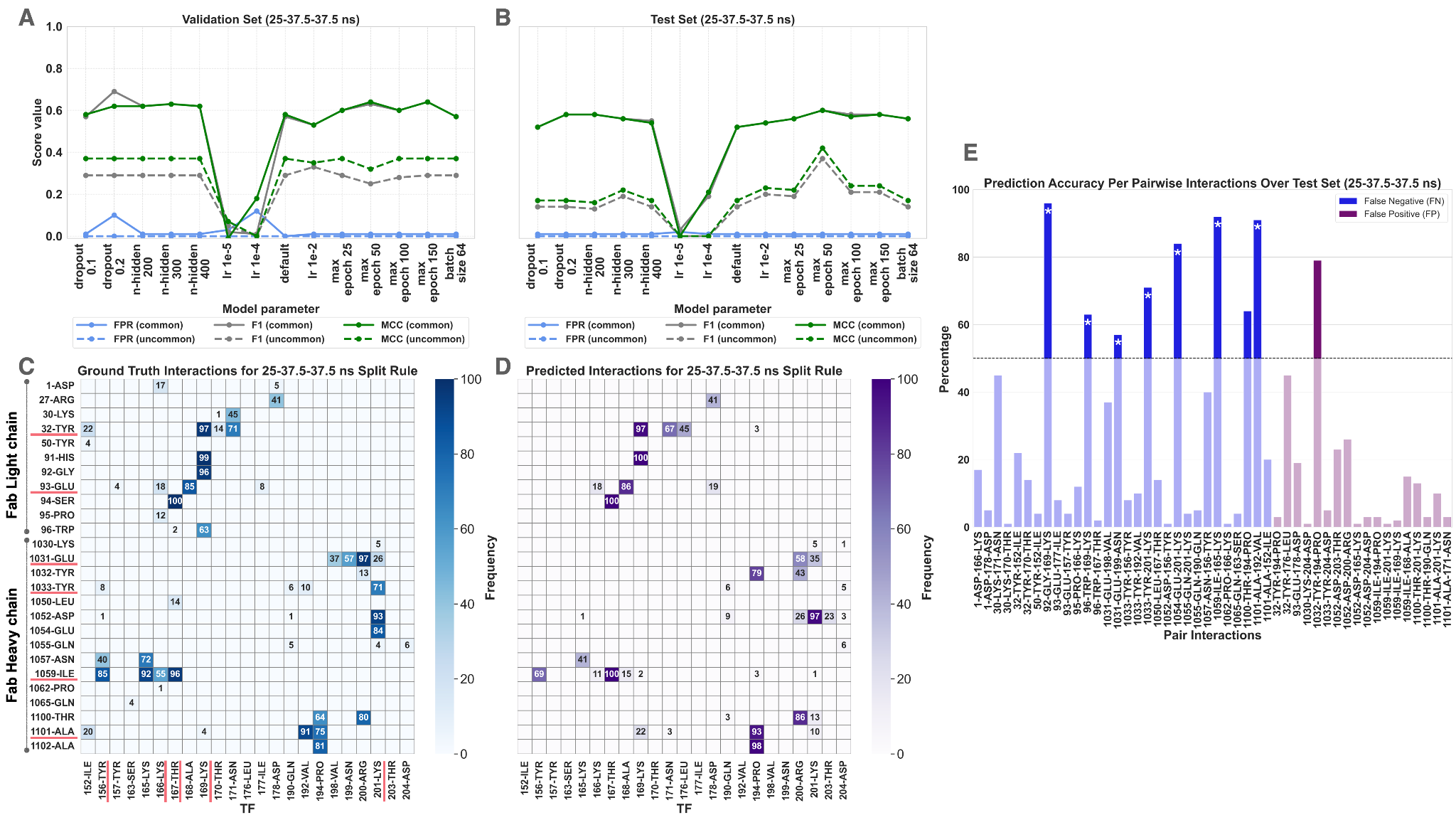


**Figure 3. The evaluation of FPR, F1, and MCC scores calculated for different RNN-MD hyperparameters with over A. validation and B. test sets for the 50-25-25 split rule**. RNN-MD training was performed over the first 50 ns where the validation was performed over the next 25 ns. The test set covered the final 25 ns. The thick lines correspond to common interactions and the dashed ones correspond to uncommon ones. **C. The frequency of ground truth and D. predicted interactions over the test set.** The predicted interactions were calculated with RNN-MD with the maximum epoch 25. Underlined interactions represent the well-connected residues of Fab and TF. **E. False negative and false positive interaction frequencies over the test set.** The blue bars correspond to the ground truth interactions that cannot be predicted (false negatives), while the purple ones represent the ones wrongly produced (false positives). The uncommon interaction cutoff was presented with a dashed line.

**The prediction performance for common and uncommon interactions significantly differs when training time was decreased to 25 ns**

Finally, we trained the RNN-MD model using 25 ns dynamic interaction profiles to predict the final 37.5 ns interaction set. In this context, we observed steady trends in FPR, F1, and MCC across various hyperparameters on both the validation and test sets (Figures 4A, B). MCC extended between 0.00 and 0.60. For common interactions, the highest MCC and F1 scores were obtained at maximum epoch 50. The same performance evaluations were observed for uncommon interactions as well. Therefore, maximum epoch 50 was chosen as our optimal hyperparameter.

In the 37.5 ns long test set, we had 22 common and 31 uncommon interactions. Among these, RNN-MD could predict 63.0% of the common and 29.0% of the uncommon interactions (Figure 4C, D), leading to a significant performance difference over these sets. The mispredicted common interactions come from 1031-Glu, 1033-Tyr, 1054-Glu, 1059-Ile, and 1101-Ala residues on Fab; 156-Tyr, 169-Lys and 201-Lys residues on TF. Seven of such interactions belong to well connected amino acids (1031-Glu:199-Asn, 1033-Tyr:201-Lys, 1054-Glu:201-Lys, 1059-Ile:165-Lys, 1101-Ala:192-Val, 92-Gly:169-Lys and 96-Trp:169-Lys, Figure 3E). RNN-MD false positively predicted only one common interaction (1032-Tyr:194-Pro).



**Figure 4. The evaluation of FPR, F1, and MCC scores calculated for different RNN-MD hyperparameters with over A. validation and B. test sets for 25-37.5-37.5 split rule**. RNN-MD training was performed over the first 25 ns where the validation was performed over the next 37.5 ns. The test set covered the final 37.5 ns. The thick lines correspond to common interactions and the dashed ones correspond to uncommon interactions. **C. The frequency of ground truth and D. predicted interactions over the test set.** The predicted interactions were calculated with RNN-MD with the maximum epoch 50. Underlined interactions present the well-connected residues of the Fab and TF. **E. False negative and false positive interaction frequencies over the test set.** The blue bars correspond to the ground truth interactions that cannot be predicted (false negatives), while the purple ones represent the ones wrongly produced (false positives). The uncommon interaction cutoff was presented with a dashed line.

**Training RNN-MD with 50 ns or 25 ns simulation time leads to a similar performance**

Across three split sets, due to varying test simulation times, common and uncommon interaction lists differ. To assess RNN-MD’s performance on the same basis over the complete data set, we proposed a novel assessment metric, called normalized offset (NO). For a given split set, NO was obtained by calculating the normalized absolute difference between the ground truth and prediction interaction matrices, as presented in panels C and D of the Figures 2, 3, 4 (see Methods). The calculation yielded NO values of 6.70, 15.07, and 16.64, for the split rules 80-10-10 ns, 50-25-25 ns, 25-37.5-37.5 ns, respectively. A low NO score signifies high accuracy, which was achieved, as expected in the first split rule. The NO scores for 50-25-25 ns and 25-37.5-37.5 ns were found to be nearly identical, suggesting that, even with a 25 ns training set, RNN-MD achieved a similar level of prediction accuracy.

**DISCUSSION**

***Important points to discuss:***

Considering the missing common interactions, we note that A1032-C194 in false positives, 1031-Glu-199-Asn, 1054-Glu-201-Lys, 1059-Ile-165-Lys, 92-Gly-169-Lys, and 96-Trp-169-Lys in false negatives were assigned commonly in all split rules.

Common and uncommon is co-assed with NO

Interestingly, the maximum MCC values achieved for the common interactions were comparable for all split sets (0.63, 0.56, 0.60). The minimum MCC values were always obtained at low learning rates.

**DISCARDED SECTIONS**

Deep learning applications in the field of MD have been limited to small molecules, ions, peptides, and single protein chains

General overview:

https://link.springer.com/chapter/10.1007/978-3-030-40245-7\_16

Ref talks: https://slideslive.com/38969375/designing-molecular-models-with-machine-learning-and-experimental-data?ref=recommended

https://slideslive.com/38941462/machine-learning-of-coarsegrained-molecular-force-fields

Learning MD

https://pubs.acs.org/doi/10.1021/acs.jctc.0c01343

https://www.annualreviews.org/doi/10.1146/annurev-physchem-042018-052331

https://www.nature.com/articles/s41467-020-18959-8?fbclid=IwAR2ieoXoTQ5CeAistGEPGxFC6Gmelt-5OnvezsAH1wMboBeb\_OPivWWtBAg

QM/MM application:

https://pubs.acs.org/doi/10.1021/acs.jctc.0c01112 (learns QM)

Generating different protein conformations with AI methods (learning from MD)

https://www.frontiersin.org/articles/10.3389/fmolb.2021.781635/full (there should be another earlier structure paper on this)

Simulate Time-integrated Coarse-grained Molecular Dynamics with Geometric Machine Learning

https://onlinelibrary.wiley.com/doi/full/10.1002/prot.26068

ENM, AF, MD https://pubs.acs.org/doi/10.1021/acs.jctc.2c01027#

Clustering and analysis of MD traj:

https://www.pnas.org/doi/10.1073/pnas.1818411116#sec-3

https://github.com/DeepDriveMD

Generating parallel sim coordinates by parallel ML learning?

https://par.nsf.gov/servlets/purl/10188151

Taking collective variables from ML to guide MD?

https://d0192efdd77670f7012d3fece2c7c0fc720c897f.vetisonline.com/10.1063%2F5.0030931

https://iopscience.iop.org/article/10.1088/2632-2153/ac3de0

Other relevant publications (not yet classified as above ones)

https://aip.scitation.org/doi/full/10.1063/5.0083060

https://arxiv.org/abs/2106.13277

https://proceedings.neurips.cc/paper/2021/hash/a45a1d12ee0fb7f1f872ab91da18f899-Abstract.html

There is no approach on predicting the time-based evolution of protein interactions (essential to biological function). We aim to address this gap.

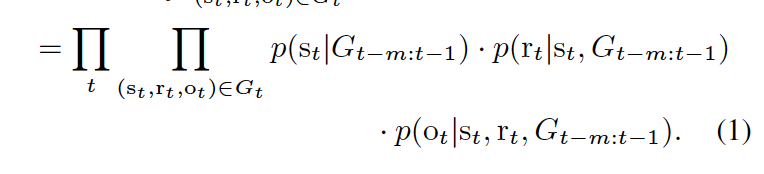
**RE-Net (Recurrent Event Network)**

Problem Description:

Temporal knowledge graphs (TKGs) are a powerful tool for capturing and representing the dynamic and evolving nature of real-life knowledge. They differ from traditional knowledge graphs as they not only model the static structure of the entities and their relations, but also incorporate the temporal dimension of how that knowledge changes over time. TKGs have two modes of inference: interpolation and extrapolation. Interpolation predicts within a given time range in a TKG, while extrapolation, a less explored problem, predicts beyond the maximum time in the TKG. Recent methods cannot make sequential predictions about future events without prior knowledge of preceding events and do not tackle concurrent events happening within a specific time frame.

RE-Net Method:

In our study, RE-Net models the joint probability distribution of all protein interactions in a TKG using autoregression with breaking each interaction probability down into a sequence of conditional distributions where each event, a protein interaction, is dependent on previous interactions in representing the PPI graph of a given time frame of MD. In the GNN, a node represents each interacting residue or atom depending on the user’s choice. Finally, future interaction prediction as triplets of subject, object (two protein interaction partners) and relation (interaction type as hydrophobic, ionic or hydrogen) is done by sequentially sampling graphs over time as a multistep inference scheme.Text, letter

Description automatically generated

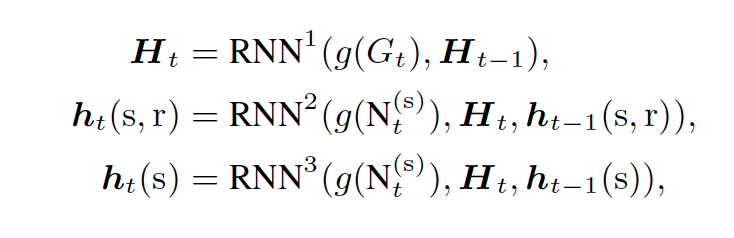
Icon

Description automatically generated

The RE-Net architecture consists of a **recurrent event encoder** which is a Recurrent Neural Network (RNN) that summarizes sequences of past interaction pairs ensuring temporal dependency, a **neighborhood aggregator** that integrates concurrent interaction information which preserves graph structural dependency, and finally, a **multilayer-perceptron (MLP)** **decoder** that defines the joint probability of the next interaction event of MD.

* **Recurrent Event Encoder and Neighborhood aggregators**

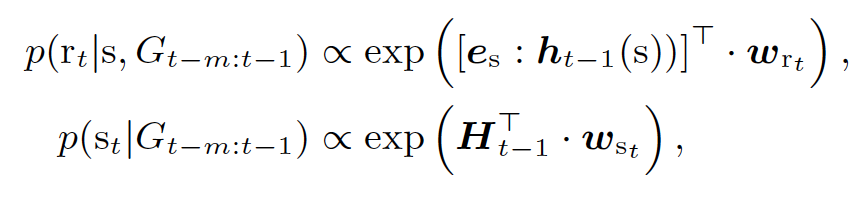
To parametrize the interaction probabilities, the encoder generates local (h) and global (H) representations with the latter conserving the past PPI graph information. In doing this it uses the previous global and local representations as well as the aggregated global PPI graph structure which incorporates all protein interaction events of MD g(Gt) and aggregated local graph of a specific subject s denoted g(Nt(s)).



These dynamic local and global representations combined with learnable static embedding vectors (es, er) are used to define the probabilities of objects, subjects, and relations.

Text

Description automatically generated with medium confidence



Chart

Description automatically generated

For the aggregator function *g*, RE-Net uses the Multi-Relational Graph (RGCN) Aggregator. It operates on the node features and the PPI graph structure to generate new node representations in the PPI GNN. The aggregator takes the node's own feature and the features of its neighbors to compute a new representation, allowing the network to capture the relationship between nodes in the graph. This pooling operation is different from other traditional pooling methods as it incorporates information from multiple relationships and multiple steps away neighbors in the graph structure.

Finally for inference, RE-Net performs multi-step inference by picking a sample after the last graph time stamp t, as t+1 until it reaches the final desired timestamp t + Δt and uses this new graph for future inferences.

