

Supplementary information for “Comprehensive protein datasets and benchmarking for liquid-liquid phase separation studies”

Carlos Pintado-Grima, Oriol Bárcenas, Eva Arribas-Ruiz, Valentín Iglesias, Michał Burdukiewicz and Salvador Ventura

Supplementary methods

Data records’ details for the proteins collected in the study

Dataset information is collected into two independent files:

- A *tsv* file (tab-separated values) which collects, for each protein, the specific UniProt Accession number, gene name, the datasets where this protein is found, the cellular localization, the fraction of derived order and curated disorder annotated by MobiDB and the full-length protein sequence. Note that if a protein is annotated in more than one dataset, they are separated by a semicolon. Cellular location is abbreviated as: cytoplasmic (C), nuclear (N), both (CN) or none (_), e.g.:

UniProt	Gene	Datasets	GO	F. order	F. disorder	Sequence
P35637	FUS	C_D;D1;...	CN	0.027	0.964	MAS...

- A *json* file (JavaScript object notation) which collects, for each protein, the full-length sequence, two independent lists with disordered element sequences: IDRs and PrLDs, and the annotated datasets. Proteins with negative predictions or missing sequences contain empty lists, e.g.:

```
{P35637:
  {Full_seq: uniprot_full_length_seq},
  {IDRs: [seq-IDR1, seq-IDR2, ... seq-IDRn]},
  {PrLDs: [seq-PrLD]},
  {Dataset: dataset_name}
}
```

$\kappa s|s$ computations

Similarly to κ in localCIDER (Holehouse et al., 2017), we compute $\kappa s|s$ for blobs (subsequences of amino acids belonging to the protein p) of 5- and 6-residues long. For each blob, we compute sticker/spacer asymmetry (σ) (Equation 1):

$$\sigma = \frac{fsticker - fspacer}{fsticker + fspacer} \quad (1)$$

Then we compute δ , a square deviation of sticker/spacer asymmetry (Equation 2).

$$\delta = \frac{\sum_{i=1}^{Nblob} (\sigma_i - \bar{\sigma})^2}{Nblob} \quad (2)$$

Next, we relate δ to the δ_{total} . δ_{total} is computed for a theoretical protein with the same amino acid composition as protein p , where all amino acids are transposed to maximize the sticker/spacer asymmetry (Equation 3).

$$\kappa s|s = \frac{\delta}{\delta_{total}} \quad (3)$$

List of symbols:

- $fsticker$: frequency of stickers (Y, R, F) in a given blob.
- $fspacer$: frequency of spacers (G, S, Q, N) in a given blob.
- σ_i : i -th blob charge asymmetry.
- $\bar{\sigma}$: mean sticker/spacer asymmetry.
- $Nblob$: total number of residue blobs.
- δ : square deviation of sticker/spacer asymmetry.
- δ_{total} : a maximum sticker/spacer asymmetry for a given sequence, as computed by the modified version of localCIDER. Please check the κ function documentation in the original localCIDER package for the details.
- $\kappa s|s$: relative total sticker/spacer asymmetry for the IDR.

Benchmark details

The benchmark analysis focuses exclusively on available and reproducible protein-level prediction methods. Region-level predictors have been excluded to maintain a clear focus on tools that provide holistic insights at the protein level.

A total of sixteen protein-level predictors were selected for the benchmark. These included both early first-generation tools and more recent predictors.

To ensure dataset consistency, we selected proteins with lengths ranging from 100 to 1,500 amino acids, excluding those containing ambiguous or non-standard residues. To further refine the dataset and eliminate redundancy, we applied the CD-hit algorithm using a sequence identity threshold of 40%. This process resulted in a curated dataset of 2,189 unique proteins, which were used for the benchmark analysis.

To evaluate the performance of the selected LLPS predictor tools, a standardized benchmarking process was applied, mainly consisting of five parts.

1. Configuration: each tool was configured using its default or recommended settings to ensure fair comparison.
2. Data input: the same dataset was formatted and used across all tools.
3. Running the tool: where possible, standalone versions of the tools were executed, though web scrapers (via RSelenium) were developed for online-only tools.
4. Output processing: prediction outputs were processed and converted to discrete predictions using author-specified thresholds.
5. Analysis and performance metrics calculation: standard performance metrics were then calculated to assess the predictive capabilities of each tool.

Therefore, to manage the heterogeneity among all the outputs in step 4, we discretized the results by applying specific decision thresholds. By these means we aim to ensure consistency and reproducibility of the benchmark.

In Additional file 5: Table S4, we detail each predictor's name, output type, and the rationale behind the chosen threshold for discretization. The "Reason" column explains how these thresholds were determined. The term "Set by authors" indicates that the thresholds were either obtained from the literature or inferred from the tool's code or documentation. In some instances, we contacted the authors directly for clarification to ensure that the tools were used as intended. In cases where "Probability" is listed as the reason, we applied a default threshold of 0.50, as this is standard for probability-based outputs when the authors did not specify a different threshold.

Despite thorough research, we occasionally had to adopt criteria from existing benchmarks for tools like PLAAC and catGranule, where no explicit thresholds were found. We adapted methods from *Pancsa et al., 2021*, known for their fair and developer-intended benchmarking approach.

Note that for FLFB, no threshold is specified because the tool itself provides a discretized output.

An additional challenge arose with the PSAP output. Authors indicated that the tool generates a continuous probabilistic score, but they acknowledged difficulties in obtaining a distribution between 0 and 1. To clarify this feature and determine the appropriate threshold for its use, we contacted the authors but unfortunately no response was received. As a result, we could only use the PSAP output to compute AUC and AUC-PR metrics.

For the other tools that produce continuous probabilistic scores, we successfully calculated AUC and AUC-PR, followed by applying decision thresholds to convert the output into discrete binary classifications. This allowed us to calculate additional metrics such as Matthew's correlation coefficient (MCC), F1 score, false positive rate (FPR), and sensitivity.

These metrics are relevant and informative for unbalanced data, as in the case of LLPS datasets, where the phase-separating proteins are underrepresented compared to the negative proteins.

These metrics are relevant for evaluating unbalanced datasets, such as LLPS datasets, where phase-separating proteins are underrepresented compared to negative data. In such scenarios, careful selection of performance metrics is essential to ensure a fair assessment of the models' predictive performance.

However, one potential limitation of our benchmark analysis is the risk of information leakage. Since some authors do not disclose the data used for model training, there is a possibility that certain proteins in our dataset were part of their training sets, which may result in artificially inflated performance evaluations.

Supplementary Figures

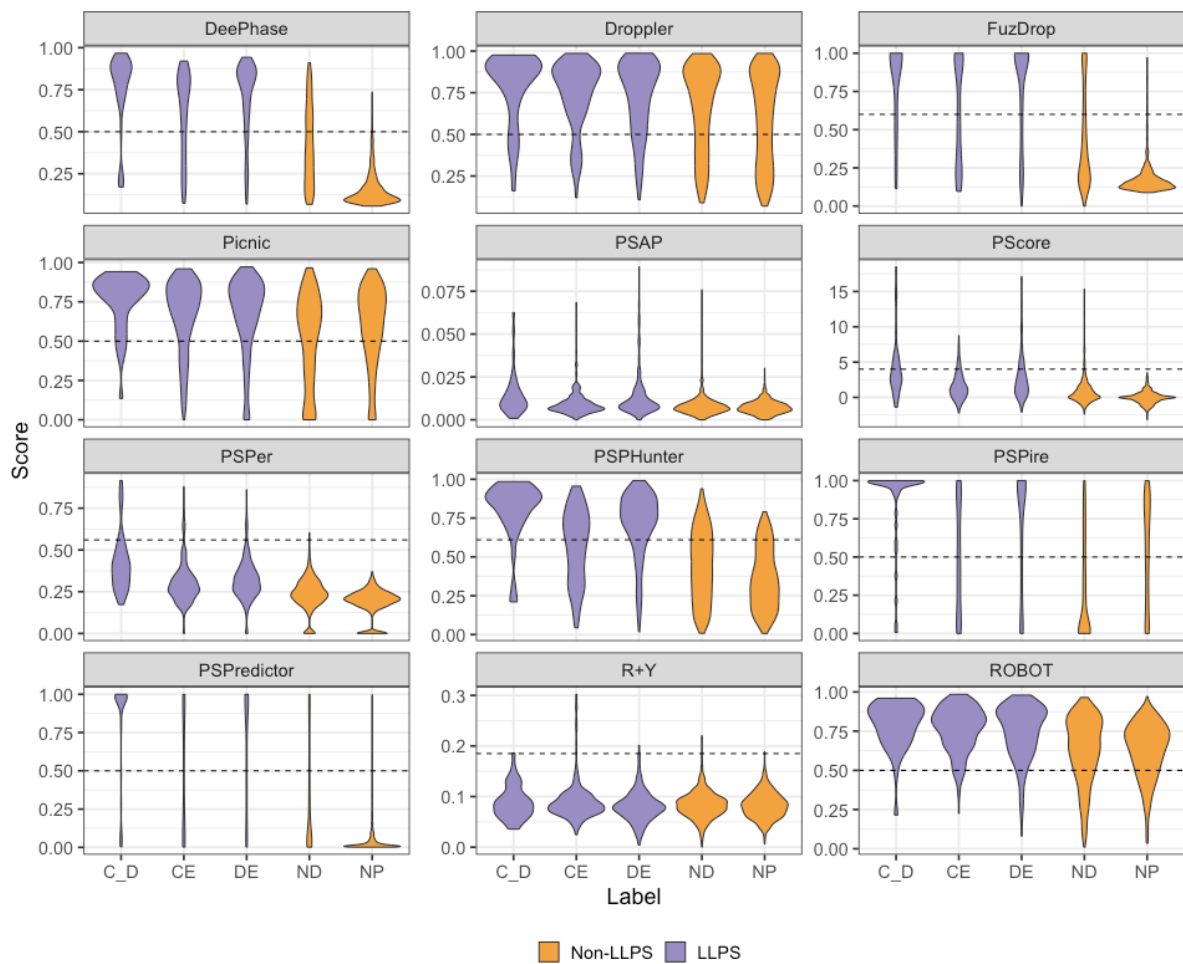


Fig. S1: Score distributions by state of the art LLPS predictors. Positive proteins (in light blue) encompass exclusive clients (CE), exclusive drivers (DE) and both clients and drivers (C_D). Negative datasets (in light red) include ordered (NP) and disordered (ND) proteins. Decision thresholds described by each tool are represented by dotted horizontal lines. Most of the tools struggle to differentiate negative proteins, especially when considering disordered entries. Also, differences between clients and drivers are not evident, although higher score variations are found in CE and DE in comparison with C_D.

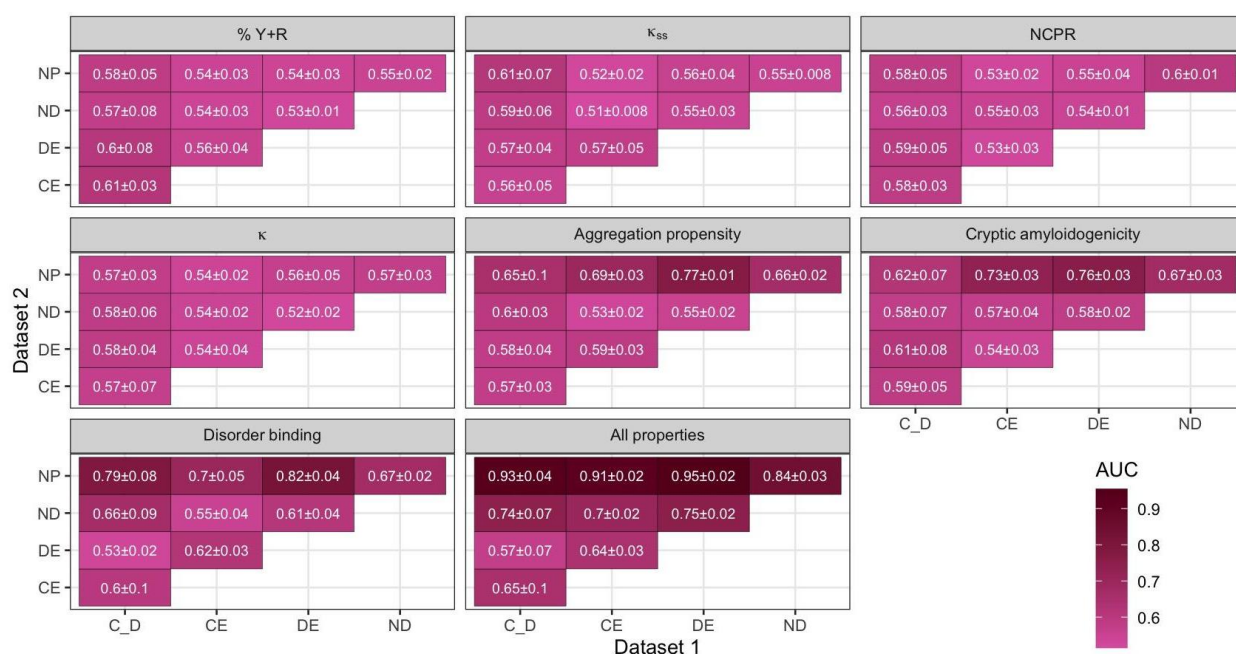


Fig. S2: Feature models generated for each property and dataset pair. Individual property models do not show a high discriminative potential. Aggregation, cryptic amyloidogenicity and disorder binding are better at distinguishing positive LLPS proteins from negative ordered proteins (NP), supporting the statistical significance observed in Figure 5. When all properties are combined into a unique model, the performance increases up to ~ 0.9 for NP and ~ 0.7 for negative disordered proteins (ND).



Fig. S3: Fraction of correctly predicted proteins for each dataset by LLPS models that provide probability output.

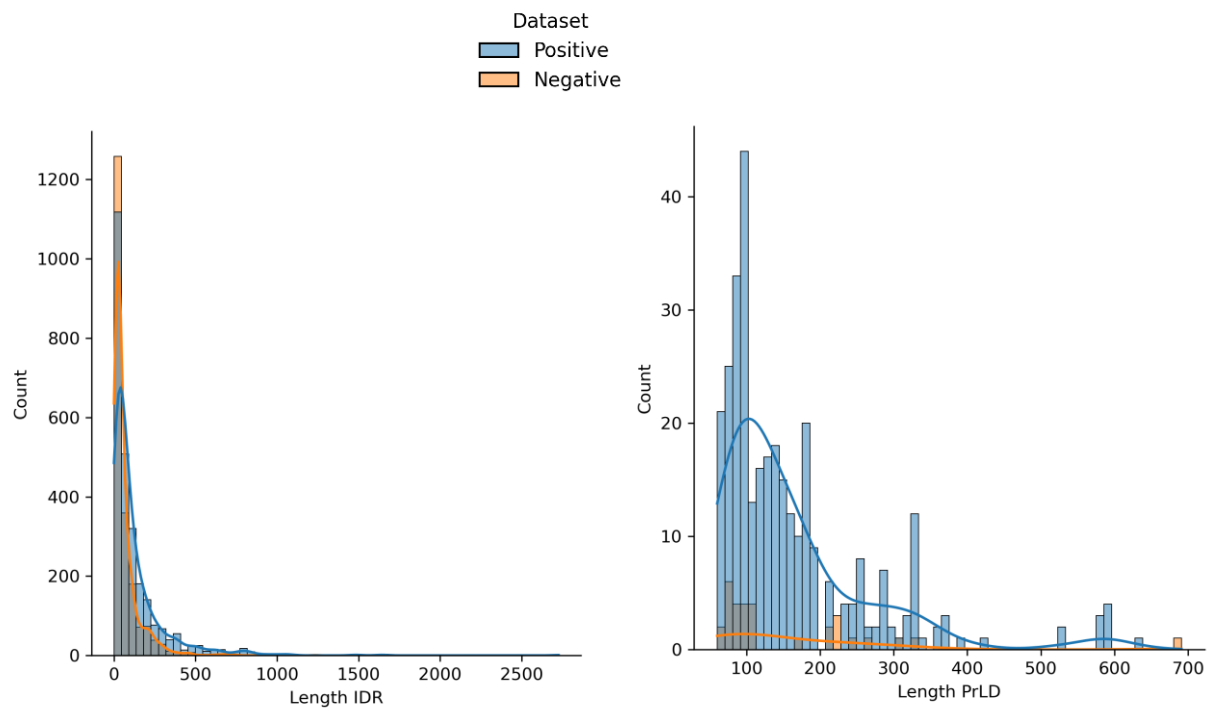


Fig. S4: Length distribution of IDRs and PrLDs in positive (D1, D2, D3, D4, D5, C1, C2; blue) and negative datasets (NP, ND; orange).