An Entity Resolution Framework for Deduplicating Proteins

Lucas Lochovsky and Thodoros Topaloglou

Department of Computer Science, University of Toronto

Abstract. An important prerequisite to successfully integrating protein data is detecting duplicate records spread across different databases. In this paper, we describe a new framework for protein entity resolution, called PERF, which deduplicates protein *mentions* using a wide range of protein attributes. A *mention* refers to any recorded information about a protein, whether it is derived from a database, a high-throughput study, or literature text mining, among others. PERF can be easily extended to deduplicate protein-protein interactions (PPIs) as well. This framework translates *mentions* into instances of a reference schema to facilitate *mention* comparisons. PERF also uses "virtual attribute dependencies" to "enhance" *mentions* with additional attribute values. PERF computes a likelihood measure based upon the textual value similarity of *mention* attributes. A prototype implementation of the framework was tested, and these tests indicate that PERF can clearly separate duplicate mentions from non-duplicate *mentions*.

1 Introduction

Elucidating and cataloguing protein-protein interactions (PPIs) are important to fully understand the function and purpose of each protein in an organism's proteome. Many PPIs are now available from numerous publicly accessible databases to facilitate further research involving these interactions. Unfortunately, there are very few overlapping records between these databases [1]. Integration of this information into a single database system, however, is not straightforward, as there are many challenges to overcome in a data integration effort of this magnitude.

One particularly important data integration issue is determining which records from separate databases refer to the same actual protein [1]. This step, which is often referred to as "entity resolution" or "deduplication", is critical to ensuring that no duplicate records are present in the integrated database system. Duplicate records could be mistaken for distinct PPIs, and since these PPIs are frequently used in other analyses, quick and accurate deduplication is important to ensuring the integrity of these analyses. However, each individual PPI database usually uses its own proprietary identifier system, and therefore it is impossible to identify duplicate records by comparing identifiers. Furthermore, certain identifiers may not actually uniquely identify a single protein, but instead

A. Bairoch, S. Cohen-Boulakia, and C. Froidevaux (Eds.): DILS 2008, LNBI 5109, pp. 92–107, 2008.
© Springer-Verlag Berlin Heidelberg 2008

refer to a class of proteins [2]. Therefore, a reliable identifier with a one-to-one correspondence to proteins is necessary in order to satisfy the goals of protein entity resolution.

In this paper, we propose a new framework for performing entity resolution on protein *mentions*. A *mention* refers to any recorded information about a protein, whether it is derived from a database, a high-throughput study, or a scientific journal, among others. PPIs can be considered pairs of protein *mentions* that interact with each other. A framework for deduplicating protein *mentions* can be easily applied to deduplicating PPIs. Given two PPIs A-B and C-D, where A-B designates an interaction between protein *mentions* A and B, if A-B and C-D refer to the same PPI, then either the pair (A,C) and the pair (B,D) are the same proteins, or (A,D) and (B,C) are the same proteins.

A reliable identifier with a one-to-one correspondence to the proteins of a given species is the Amino Acid (AA) Sequence, since the primary sequence directly determines the structure and function of each protein [3]. Therefore, if a protein *mention* provides both an AA Sequence and a "Source Organism", the one protein that this *mention* refers to can be unambiguously identified. Source Organism is required since distinct proteins in different species can share the same AA Sequence. Since the similarity of the AA Sequence and the Source Organism is generally considered to be the strongest evidence that two *mentions* refer to the same protein, existing protein deduplication systems perform deduplications solely on the basis of AA Sequence and Source Organism identity [1, 2, 4]. However, for most *mentions*, one or both of these attributes may be missing, and therefore an alternate means of deduplication is required. The new framework proposed here, the Protein Entity Resolution Framework (PERF), takes two protein *mentions* as input, attempts to deduce other attributes for these *mentions*, and makes use of these attributes to determine the likelihood that the two given *mentions* refer to the same actual protein.

PERF consists of three main components:

1. **XML Reference Schema:** The PERF framework is based on an XML schema that provides a comprehensive list of *mention* attributes derived from the schemas of various popular protein databases, including NCBI, EBI, UniProt, BIND, HPRD, MINT, MIPS, IntAct, and DIP [5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17]. This Framework Schema allows *mentions* to be represented in a common format to facilitate *mention* comparisons.
2. **Virtual Attribute Dependencies (VADs):** Special rules for identifying additional *mention* attributes, called "virtual attribute dependencies" (VADs), were defined for the purpose of finding as much information as possible on each *mention* to use for the actual deduplication process.
3. **Framework Deduplication Procedure:** PERF supports a computational procedure that computes the likelihood that two given protein *mentions* refer to the same actual protein based upon the attribute values available from those *mentions*.

PERF is a modular framework that currently supports the following functions:

- *resolve*(m): This function serves as the basis for all the other functions. Given a single ambiguous *mention*, this function will resolve the protein that this *mention* refers to, if possible.
- *deduplicate*(m_1, m_2): This function uses PERF to deduplicate two protein *mentions*. PERFs calculations should be able to identify true duplicate pairs from a set of *mention* pairs.
- *deduplicate-network*(n): This function uses PERF to deduplicate a PPI network n, i.e. identify duplicate proteins and interactions in the network. This is essentially the application of *deduplicate*(m_1, m_2) to each pair of *mentions* in the network to consolidate duplicate proteins and their interactions to produce a non-redundant network.
- *compare-networks*(n_1, n_2): This function takes as input two PPI networks n_1 and n_2, both of which are internally deduplicated using *deduplicate-network*(n). This function also finds proteins in n_1 and n_2 that are the same, and thus *compare-networks*(n_1, n_2) can be used to determine the overlap between n_1 and n_2.

We implemented a prototype version of PERF that supports *resolve*(m) and *deduplicate*(m_1, m_2). We tested PERFs ability to fulfill the requirements of these functions; the test results are discussed in this paper. Although all four functions have been defined, the last two functions, *deduplicate-network*(n) and *compare-networks*(n_1, n_2), will be implemented for a future version of PERF.

The rest of this paper is organized as follows. Section 2 provides background information on protein and PPI database systems. Work previously done to tackle the PPI entity resolution problem is also discussed. Section 3 describes PERFs components in detail. Section 4 describes the testing of PERFs ability to fulfill the requirements of *resolve*(m) and *deduplicate*(m_1, m_2), and discusses the test results. Section 5 makes some concluding remarks and discusses future directions for this research.

2 Background

Many protein databases have been established to catalog all identified proteins [18]. Each of these databases relies on different sources for their records, and therefore cover very different sets of proteins. Although there is some collaboration between a few of these databases to keep each others records up-to-date, and to cross-reference corresponding records [19], most databases do not make it easy to find corresponding records in other databases. Given the exponential increase in protein data fueled by new high-throughput analyses, reliable, efficient, and automatic deduplication and integration of this data is urgently needed to properly manage this data and make sense of it.

PERF, as discussed earlier, is also applicable to the deduplication of PPIs. Many high-throughput PPI datasets have been produced in the last few years thanks to recent advances in laboratory technology [20]. These datasets are compiled from the results of high-throughput analyses. Although these analyses can

process thousands of interactions in a single run, they are also prone to particularly high false positive rates (i.e. a large number of the published interactions do not actually exist) [20]. Higher confidence can be placed in interactions that are reported in several datasets, as this represents verification of these interactions in multiple, independent experiments. Therefore, reduction of false positives provides additional motivation to find duplicates and integrate high-throughput PPI datasets.

Existing protein entity resolution systems include the International Protein Index (IPI) [1], and systems like BIOZON [2] and the Agile Protein Interaction DataAnalyzer (APID) [4] for PPI deduplication. Each of these systems, however, only deduplicate proteins and PPIs on the basis of amino acid sequence similarity. PERF, however, can also make use of other available protein attributes, in addition to amino acid sequence similarity, making PERF more versatile in deduplicating protein *mentions*.

3 Protein Entity Resolution Framework (PERF)

3.1 Mentions

PERFs inputs are protein *mentions*. Typically, we refer to actual proteins with the values of their attributes, such as "Name". *Mentions* here are collections of these values drawn from a source or sources with information pertaining to a given protein. Some sources, such as database records, contain (particularly extensive) information on a given protein. Proteins may also be discussed in certain papers, either individually or within the context of a particular group of proteins. Additionally, protein information can be drawn from the data of high-throughput elucidation experiments. Each of these sources may provide different amounts and/or different types of information, but information from each of these sources is considered a *mention* for PERFs purposes.

Formally, we define a *mention* as a list of attribute-value pairs following a nested model where attributes can contain, nested within their values, "sub-attributes" or a set of values that allow lists of attributes/values to be represented within a single attribute. This model allows several aspects of a single attribute to be represented in a *mention* as well. The general form of a *mention* is described below:

$$
\begin{aligned}
&m.name[:\, m.db\_name] := \{\\
&[p_1^1 := v_1^1]\backslash;\\
&[p_2^1 := v_2^1\backslash; v_2^2\backslash; v_2^3]\backslash;\\
&[p_3^1 := v_3^1\backslash; [p_3^2 := v_{3-2}^1]\backslash; [p_3^3 := v_{3-3}^1\backslash; v_{3-3}^2\backslash; [p_3^4 := v_{3-4}^1]]]\backslash;\\
&\vdots\\
&[p_n^1 := v_n^1 \ldots]\}
\end{aligned}
$$

Each *mention* is specified by a name, the name of the database it was derived from (if any), and a list of attributes. Each attribute can be associated with a

single value (e.g. p_1^1), a set of values (e.g. p_2^1), or a set of sub-attributes (e.g. p_3^1). The following *mention*, which describes the CCNB1 protein from the CellMap database [21], contains examples of all three types of attributes described above.

Example 1. A complete protein *mention* using the PERF input *mention* format.

ccnb1:CellMap:={
[Name:=CCNB1]\;
[Synonyms:=Cyclin B1\;G2/mitotic specific cyclin B1\;CCNB1\;CCNB]\;
[External_Links:=[PubMed:=1387877]\;[OMIM:=123836]]\;
[Complex(s):=CDC2]\;
[Physical_Interaction(s):=CDC2\;PTCH]}

3.2 The Framework Schema

A *mention* may or may not point to a single protein entity. However, *mentions* often contain attributes that can help us retrieve additional attributes that are better suited for uniquely resolving that single protein. The Framework Schema was designed to represent these attributes in a standardized format. Therefore, given an input *mention*, we first standardize it by mapping it to the Framework Schema. Then, we expand the coverage of each *mention* with "virtual attribute dependencies" (section 3.3), and finally decide if the *mention* points to a unique protein (i.e. it is unambiguous) or a group of proteins (i.e. it is ambiguous).

The Framework Schema is a predefined XML-based schema that can accommodate many common kinds of protein information. This schema allows several instances of a *mention* to be represented in a single Framework Schema record. This is accomplished by defining the top-level element to be a "Protein_Set" that can contain multiple "Protein" objects. Initially, each Framework Schema record derived from a single *mention* contains only one "Protein". However, additional "Proteins" can be added through the use of "1-to-N VADs" described in section 3.3.

Each attribute in the Framework Schema has a distinct usefulness for the entity resolution of protein *mentions*, and therefore each attribute has been assigned a "strength". This concept resembles the selectivity of attributes in relational databases: in PERF, an attribute with strength I is a key attribute, and therefore it uniquely identifies a single protein. The less useful an attribute is for narrowing down the number of possible proteins to which a *mention* may refer, the higher its strength. The strengths of select Framework Schema attributes are provided in Table 1, along with an attribute description and the domain of accepted values for that attribute. Attribute strengths were derived from experiments with database queries to determine the cardinality of the result set produced when each attribute is used as the query attribute (databases appropriate for each attribute were used for these queries). Certain attribute combinations may be more useful for unique protein identification than the individual attributes considered in isolation; these attribute combinations are listed in Table 2. The full list of Framework Schema attributes is available in [22].

Table 1. A list of select attributes defined under the Framework Schema, along with their strengths, and the domains of the values accepted for each attribute

Attribute	Description	Strength	Domain
Name(s)	Name(s) assigned to the given protein.	III-IV	Text string corresponding to one of given proteins name(s). One tag used for each distinct name.
Keywords	Short, descriptive words assigned to given protein.	IV	Terms describing key characteristics of given protein.
Database cross-references	References to database records that describe the given protein, or some characteristic of that protein.	II	Composite value with two fields: *Name*: Name of the referenced database. *ID*: Unique identifier of referred record in named database.
Amino acid (protein) sequence	The given protein's sequence of amino acids produced by transcription and translation from the corresponding gene.	II	String of amino acid one-letter codes
Source organism	The organism from which the given protein was derived.	IV	The "[genus] [species]" designation of the source organism
Free text description	Any freeform description of the given protein.	IV	Any text
NCBI Gene ID	NCBI Gene ID identifying exact locus of gene from which given protein was transcribed.	II	A valid NCBI Gene ID

Table 2. A list of the attribute combinations that have a better strength than their separate, individual attributes, and hence have been assigned a lower number as given below

Attribute Combination	Strength
(AA Sequence, Source Organism)	I
(NT Sequence, Source Organism)	II
(NCBI Gene ID, Source Organism)	II

3.3 Virtual Attribute Dependencies (VADs)

The concept of "virtual attribute dependencies" resembles that of "functional dependencies" (FDs) in relational databases [23]. In the context of PERF, we define "virtual attribute dependencies" as rules for determining additional attribute values from an external biological database, given attribute values provided with the original *mention*. For example, if a RefSeq identifier is available in a *mention* m, then the amino acid sequence can be retrieved from the protein's RefSeq record and added to m. These newly-acquired attributes help narrow down the size of the protein classes implied by ambiguous *mentions*, and therefore the new attributes have a better strength compared to the attributes they were derived from. Formally, a virtual attribute dependency is a triple (P, Q)

$\rightarrow T$, where P refers to the set of prerequisite attributes, Q is a query or web service, and T refers to the set of resultant attributes. Given a set of values for the attributes in P, Q is evaluated to produce values for the attributes in T. Therefore, VADs define a general mechanism that is applied here to the specific problem of extending the information of protein *mentions*.

The execution of a VAD for a particular set of values for P may produce one set of values for T, or may produce many sets of values for T. Therefore, there are two types of VADs: 1-to-1 VADs and 1-to-N VADs. For the 1-to-1 VADs, the values of T are added to the original *mention* by instantiating the appropriate attributes with those values. For the 1-to-N VADs, however, each resultant value set represents one possible configuration of the original *mention*. Therefore, for each resultant value set, a new Protein object must be created in the original *mention* that extends the original Protein object with the attributes and values from that set. Thus, the *mention* is extended to cover all possible proteins that the original *mention* refers to in as much detail as possible.

VADs are designed to extend/improve an instantiation of the Framework Schema. Table 3 illustrates some example VADs. These dependencies are provided in the form $(P, Q) \rightarrow T$ described above. Starting attribute strength (Start str.) indicates the strength of the prerequisite attributes, while resultant attribute strength (Res. str.) indicates the strength of the resultant attributes. The notes column describes the rationale behind each dependency, and the last column presents examples of these dependencies with actual values. Note that this list is extensible and customizable, and can be updated to meet the deduplication needs of particular data domains.

3.4 Framework Deduplication Procedure

There are three major steps to this procedure, each of which will be discussed below.

3.4.1 Mapping Protein Mentions to the Framework Schema

Recall that the Framework Schema uses attributes names that are not the same as those of the input *mentions* but are semantically equivalent to the original *mention* attributes. A mapping procedure is therefore needed for finding the Schema attributes that correspond to a given *mention's* attributes.

Let m be a *mention* and R be the Framework Schema. Also, let $S(m)$ be the schema of m. We assume that, for each attribute a_i in $S(m)$, there is exactly one matching attribute r_j in R s.t. a_i and r_j describe the same thing. The set of these attribute pairs for each attribute a_i in $S(m)$ is called the **correct mapping**. There are two ways the Framework Deduplication Procedure can infer the correct mapping between $S(m)$ and R, depending on whether or not $S(m)$ was derived from an established database schema or not. The first option involves lexical similarity comparisons between the attributes of $S(m)$ and the attributes of R. The second option involves using a lookup table to directly translate an attribute a_i in $S(m)$ into an attribute r_j in R. This works if $S(m)$ is derived from a previously established schema that has been manually matched

with the Framework Schema attributes in a one-to-one mapping. The complete description of the algorithm for inferring the correct mapping between $S(m)$ and R is available in [22].

3.4.2 Addition of Attributes to Mentions Using Virtual Attribute Dependencies

After the translation of each *mention* to a Framework Record F, the virtual attribute dependencies (VADs) in Table 3 will be used to collect additional attributes for each *mention*. Each VAD is applied sequentially, and at each step i, a Framework Record F_i is rewritten to F_{i+1}. For each VAD D_i executed on a *mention* m, the Framework Deduplication Procedure will check if all the prerequisite attributes P_i are defined in m, and if at least one of the resultant attributes T_i is not defined in m. If both of these conditions are true, then the query Q_i will be executed to produce the resultant attributes T_i to add to m. Otherwise, the next VAD will be considered, if there are any remaining VADs to consider.

3.4.3 Pairwise Matchings of Mentions

In this step of the procedure, comparisons are made between the two input mentions to determine the likelihood that they refer to the same protein. This

Table 3. A list of some of the virtual attribute dependencies (VADs) used in PERF

#	Dependency	Start str.	Res. str.	Notes	Example
1	{(Database reference), Corresponding database} → (AA Sequence, Source Organism)	II	I	All protein database records contain information on the proteins amino acid sequence, and its source organism.	{(RefSeq:= NP_660312), RefSeq} → (mmrrtlenrn . . . , Homo sapiens)
2	{(NT Sequence, Source Organism), translation service} → (AA Sequence)	II	I	The nucleotide sequence can be translated into an amino acid sequence.	{(ACGAACAGGC . . . , Homo sapiens), GlimmerHMM} → (malrvtrnsk . . .)
3	{(AA Sequence), NCBI BLASTP} ⇝ (Source Organism) (a "⇝" means the query may or may not produce resultant attribute values, see Notes column)	IV	I	If an amino acid sequence is available, but no source organism is available, the sequence can be BLASTed against a protein database, and if a strong hit is found, and the E-value of the best hit from a different organism is lower by a threshold T than the top hit, then we can deduce the Source Organism of the uniquely identified protein referenced in the given *mention*.	{(malrvtrnsk ...), NCBI BLASTP} → (Homo sapiens)

step consists of three algorithms. They are: A) Ambiguity Determination, B) Unambiguous Deduplication, and C) Ambiguous Deduplication. Each of these will be discussed below.

A) Ambiguity Determination: Like most existing protein deduplication frameworks, we assume that AA Sequence and Source Organism are the most reliable means of identifying individual proteins [1, 2, 4]. Therefore, unambiguous *mentions* have both an AA Sequence and a Source Organism defined, and ambiguous *mentions* have one or both of these attributes undefined. If both *mentions* are unambiguous, then PERF executes an Unambiguous Deduplication (described below) that directly compares the two individual proteins, and precisely determines whether or not these proteins are the same. If one or both *mentions* are ambiguous, then there is some level of uncertainty over the protein to which one or both *mentions* refer. Under these circumstances, PERF will execute an Ambiguous Deduplication (described below) that computes a likelihood measure indicating the probability that the two *mentions* refer to the same protein.

B) Unambiguous Deduplication: In an Unambiguous Deduplication, the AA Sequence and Source Organism will be directly compared to determine if the two *mentions* describe the same protein. The sequences will be compared with the BLAST2SEQ program [24], and the organisms will be compared using the Damerau-Levenshtein (DL) string edit distance [25, 26] to determine how close they are to each other. The use of a string edit distance accommodates some tolerance for simple spelling or transcriptional errors. The results of these comparisons will be compared to cutoffs to determine if the two input *mentions* refer to the same protein. In PERFs current implementation, the BLAST2SEQ cutoff is 90% sequence identity, and the DL cutoff is 5.

C) Ambiguous Deduplication: Suppose that PERF is attempting to deduplicate two input *mentions* m_1 and m_2. Let $v(a_i, m_1)$ be the set of values of attribute a_i in *mention* m_1, and let $v(a_i, m_2)$ be the set of values of attribute a_i in *mention* m_2. For each attribute a_i in $S(m_1) \cap S(m_2)$, and any pair of *mentions* m_1 and m_2, there is a maximum number of a_i values that m_1 and m_2 can have in common. This number is the *theoretical maximum similarity score* ($M(a_i, m_1, m_2)$), and is equal to $\min\{|v(a_i, m_1)|,|v(a_i, m_2)|\}$. This is the maximum number of attribute a_i values that can match between m_1 and m_2. Attributes that are defined in one *mention* but are missing from the other are not factored into this score, since *mentions* may be derived from sources with varying attribute coverage.

The *raw similarity score* $S(a_i, m_1, m_2)$ is the actual number of a_i values that m_1 and m_2 have in common. This score is determined for each attribute a_i that has a nonzero *theoretical maximum similarity score* $M(a_i, m_1, m_2)$ on m_1 and m_2. After the calculation of the *theoretical maximum similarity score* and the *raw similarity score* between m_1 and m_2 for each a_i, the sum of the *raw similarity scores* over all attributes a_i between m_1 and m_2 is divided by the sum of the *theoretical maximum similarity scores* over all attributes a_i between m_1 and m_2 to produce a final *mention percent similarity score* $\overline{P}(m_1, m_2)$:

$$\overline{P}(m_1, m_2) = \frac{\sum_{a_i} S(a_i, m_1, m_2)}{\sum_{a_i} M(a_i, m_1, m_2)} \qquad \text{for all } a_i \text{ in } S(m_1) \cap S(m_2) \qquad (1)$$

$\overline{P}(m_1, m_2)$ will be equal to 1 if all attribute values were perfect matches, and 0 if there were no matches. In general, $0 \leq \overline{P}(m_1, m_2) \leq 1$.

So far, we have assumed that all attributes are equally important to correctly deduplicating two *mentions*. However, some might be more important than others. Therefore, we introduce an attribute weight factor. The weighted variation of the *mention percent similarity score* between m_1 and m_2 will now be discussed.

Let a be the weight factor of strength I attributes, b be the weight factor of strength II attributes, c be the weight factor of strength III attributes, and d be the weight factor of strength IV attributes. In the frameworks current form, these factors are set to the following values: $a = 1000$, $b = 100$, $c = 10$, and $d = 1$. The *weighted mention percent similarity score* between m_1 and m_2 $\overline{W}(m_1, m_2)$ is similar to the *mention percent similarity score* between m_1 and m_2 $\overline{P}(m_1, m_2)$, with the exception that the *weighted raw similarity scores* and the *weighted theoretical maximum scores* are used in the summations in the numerator and denominator, respectively. ($w(a_i)$ represents the weight factor of attribute a_i)

$$\overline{W}(m_1, m_2) = \frac{\sum_{a_i} w(a_i) S(a_i, m_1, m_2)}{\sum_{a_i} w(a_i) M(a_i, m_1, m_2)} \qquad \text{for all } a_i \text{ in } S(m_1) \cap S(m_2) \qquad (2)$$

In this algorithm, both the *mention percent similarity score* $\overline{P}(m_1, m_2)$ and the *weighted mention percent similarity score* $\overline{W}(m_1, m_2)$ are computed.

4 PERF Implementation and Evaluation

4.1 Evaluation of Mention Resolution

The International Protein Index (IPI) maintains a curated database of cross-references between a wide range of other databases, including Ensembl, RefSeq, and TAIR [1]. This index can be used to identify pairs of duplicate records across different databases. Using IPI's index, five UniProt/NCBI pairs of duplicate records were arbitrarily chosen. A set of five non-duplicate pairs was also produced by taking each of the UniProt records and randomly pairing them with NCBI records (not shown). VAD #3, which defines a rule for deriving the Source Organism of a *mention* by conducting an NCBI BLAST of the *mention's* AA Sequence (section 3.3, Table 3), was tested by removing the Source Organism from each of the UniProt *mentions*. PERF was tested on these data to determine whether or not PERF can identify the correct Source Organism, and whether or not PERF can correctly identify which pairs were actual duplicates and which were non-duplicates.

Successful invocation of VAD #3 correctly identified the Source Organism for each of the UniProt *mentions*. The results of the subsequent unambiguous deduplications demonstrate that all the actual duplicates did exhibit a sequence identity of 90% or higher, while the non-duplicates exhibited significantly worse results (data not shown). Additionally, the Source Organism DL (Damerau-Levenshtein) Distance for each pair is zero, indicating that each pair's Source Organisms were perfectly identical. Therefore, PERF correctly classified each pair in the test data, and was able to fully resolve each of the UniProt *mentions*.

4.2 Evaluation of Duplicate Resolution

The International Protein Index (IPI) was used to identify pairs of duplicate records across different databases for this evaluation. The evaluation of PERF's effectiveness at deduplicating *mention* pairs involved *mentions* derived from three of the databases for which IPI maintains cross-references. These *mention* pairs are divided into two groups representing the databases from which these *mentions* were drawn:

1. **CellMap/NCBI:** Pairs in which one *mention* was drawn from the Memorial Sloan-Kettering Cancer Center's CellMap database, and one from NCBI, and
2. **Ensembl/NCBI:** Pairs in which one mention was drawn from Ensembl, and one from NCBI

Each of these groups contains 20 arbitrarily chosen pairs of duplicate records. These *mentions* comprise the body of test cases (experiments) that PERF should correctly identify as duplicates. *Mention* pairs that do not refer to the same protein (i.e. non-duplicates) were derived by randomly pairing the NCBI *mentions* in each group to the *mentions* from the other database in the same group. (e.g. in group (i), each NCBI *mention* was randomly paired with a CellMap *mention* from the same group) Therefore, each group consists of 20 examples of *mention* pairs that refer to the same protein, and a corresponding number of examples of *mention* pairs that do not refer to the same protein. Each pair was labelled with a unique identifier indicating which group it belongs to, whether it is a duplicate or non-duplicate pair, and its unique number within that group's duplicate/non-duplicate pairs. For example, the pair II-ND-3 belongs to group II, is a non-duplicate pair, and is the third pair in the set of group II non-duplicates. All pairs from these groups were scored by the PERF Attribute Value Comparison to determine if the $\overline{W}(m_1, m_2)$ score could be used to separate the duplicate pairs from the non-duplicate pairs.

Fig. 1 presents the *mention percent similarity score* $\overline{P}(m_1, m_2)$ and *weighted mention percent similarity score* $\overline{W}(m_1, m_2)$ between m_1 and m_2 for each of the duplicate *mention* pairs and non-duplicate *mention* pairs from group I. It is clear that under the current weighting scheme, most duplicate pairs' scores are increased relative to their unweighted scores, while non-duplicate pairs scores are decreased relative to their unweighted scores. For these *mentions*, the weighting scheme slightly increased the scores of the duplicate pairs, with two exceptions. First, pair I-D-8's weighted and unweighted scores are the same. The second

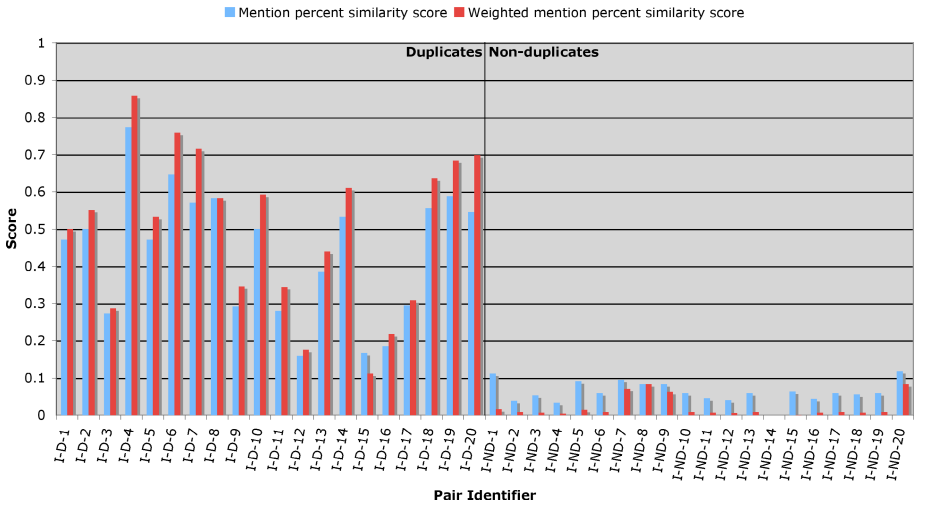

Fig. 1. Group I Results

exception is pair I-D-15, where the weighted score actually decreased relative to the unweighted score. Despite these aberrations, the scores of the duplicates are significantly higher than those of the non-duplicates. According to Fig. 1, all non-duplicate pairs' scores were drastically reduced by the weighting scheme. Therefore, overall, these weights are effective for widening the gap between actual duplicates and non-duplicates, reducing the amount of possible overlap between these two classes. Reducing this overlap is important as it reduces the number of pairs that could be mistakenly classified.

It was discovered that most of the similar attributes between duplicate *mention* pairs from this test group (i.e. between CellMap and NCBI *mentions*) were between Name attributes. Therefore, it appears that CellMap and NCBI use the same naming conventions, and Name similarity is more significant in CellMap/NCBI comparisons as a result. Consequently, the strength of Name attributes was increased when scoring these pairs.

The average $\overline{W}(m_1, m_2)$ for the duplicates was 0.497, and the average $\overline{W}(m_1, m_2)$ for the non-duplicates was 0.021. Therefore, PERF was very successful at separating true duplicates from non-duplicates. The exact score cutoff, as well as the best weighting scheme to use to separate these two classes, would be best determined by training PERF on a wider range of test data. Training could also help adjust the weighting scheme so that the weighted scores of duplicates exemplified by pairs I-D-8 and I-D-15 are increased relative to their unweighted scores.

Fig. 2 presents the $\overline{P}(m_1, m_2)$ and $\overline{W}(m_1, m_2)$ between m_1 and m_2 for each of the duplicate *mention* pairs and non-duplicate *mention* pairs from group II. Among the duplicate pairs, six pairs did not have any common attribute values, even though they actually are duplicates. (These are indicated with a Zero in

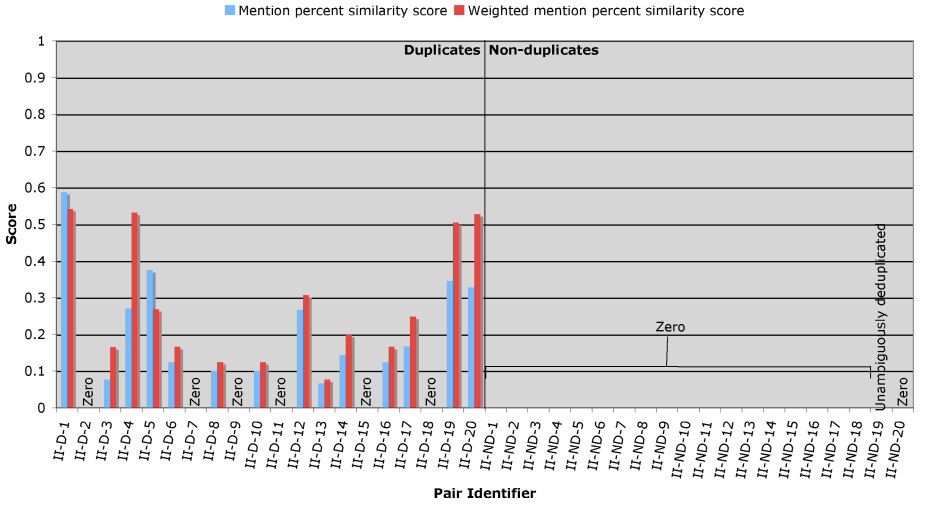

Fig. 2. Group II Results

Fig. 2) These represent duplicates that are missed, underscoring the sometimes vast differences between different databases' coverage of protein attributes. Additional attributes, possibly from the database cross-references of these *mentions*, could possibly provide attributes with similar values that PERF can identify for the purpose of establishing that these *mentions* are duplicates. PERF provides a framework where new VADs may be added to further identify new attributes. Testing with larger amounts of data in the future would help to enhance PERF capabilities in this respect.

Looking at the non-duplicate pairs, all pairs scored zero, ruling out the possibility of mistakenly classifying a non-duplicate pair as a duplicate pair. One pair, pair II-ND-19, was fully resolved by PERF, and therefore compared under the Unambiguous Deduplication Procedure described in section 3.4.3. Since the AA Sequence identity of these mentions was 24%, this pair was correctly classified as non-duplicate.

Additionally, two of the duplicate pairs from group II (pairs II-D-1 and II-D-5) have $\overline{W}(m_1, m_2)$ scores that are lower than their corresponding $\overline{P}(m_1, m_2)$ scores, much like pair I-D-15. However, as with group I, these $\overline{W}(m_1, m_2)$ scores are still adequate for distinguishing between duplicate and non-duplicate pairs. Overall, the $\overline{W}(m_1, m_2)$ scores correspond to a roughly bimodal distribution. The mean $\overline{W}(m_1, m_2)$ for the duplicates was 0.198, and the mean $\overline{W}(m_1, m_2)$ for the non-duplicates was zero, indicating that duplicates and non-duplicates are clearly separated in group II. Again, additional parameter tuning and weight training for these types of *mentions* may help produce better $\overline{W}(m_1, m_2)$ results, and help adjust the weighting scheme to increase the scores of pairs II-D-1 and II-D-5. Name attributes' strength was increased for this group as well.

The above results show that PERF is effective in deduplicating protein *mentions* by comparing a range of attributes. It is also shown that database-specific considerations are desirable for achieving a good bimodal distribution for the scores of duplicate *mention* pairs and non-duplicate *mention* pairs. *Mentions* from these databases could be used in the future as training data for PERF's attribute strengths and other database-specific parameters, allowing them to be optimized to achieve the best possible separation between duplicates and non-duplicates.

5 Conclusions and Future Work

In this paper, a new framework for deduplicating protein *mentions* was defined. Applications of this framework, PERF, to deduplicating *mention* pairs and entire networks were described. A prototype version of PERF was implemented and tested on a small set of protein *mention* pairs derived from different databases to evaluate PERF's effectiveness at fulfilling the requirements of two of the functions described earlier. These results indicate that PERF can be effective for solving the entity resolution problem for protein *mentions*. PERF forms a solid basis, grounded in techniques from database research, to address entity deduplication in biological databases.

Future plans for our work include the following developments. First, additional virtual attribute dependencies (VADs) can be produced so that there are more options available to PERF for resolving *mentions* to unique proteins. Further investigation of the Framework Schema attributes, as well as query services with which they can be used to obtain additional information, is desirable. Second, testing with larger datasets would give us more insights into increasing the effectiveness of PERF for mentions from different sources, such as published literature and high-throughput datasets. Issues specific to particular sources can also be investigated so that PERF can be better tuned for specific applications.

A third area of future development would be the creation of a better, more usable interface. Upgrading PERF to a web service will maximize its reach and enable its use by others. Fourth, there are additional steps at the end of the Framework Deduplication Procedure that could be implemented to streamline the post-deduplication user workflow. PERF could, upon completion of a deduplication, automatically consolidate two duplicate *mentions* into one, and add it to a database that serves as a repository of deduplicated *mentions*. Fifth, PERF may also have applications to the field of "data cleaning", i.e. the identification and correction of inaccurate records in a database [27]. Finally, implementations for some of the Framework Deduplication Procedure's steps could be refined to improve PERF's robustness and performance.

In conclusion, PERF forms a solid foundation for a framework for PPI deduplication. Further development of the aforementioned features, and more testing,

would broaden and enhance PERF's applicability to protein and PPI deduplication problems.

References

1. Kersey, P.J., Duarte, J., Williams, A., Karavidopoulou, Y., Birney, E., Apweiler, R.: The International Protein Index: An integrated database for proteomics experiments. Proteomics 4, 1985–1988 (2004)
2. Birkland, A., Yona, G.: BIOZON: a system for unification, management and analysis of heterogeneous biological data. BMC Bioinformatics 7(70) (2006)
3. Berg, J.M., Tymoczko, J.L., Stryer, L.: Biochemistry, 5th edn. W.H. Freeman, New York (2006)
4. Prieto, C., Rivas, J.D.L.: APID: Agile Protein Interaction DataAnalyzer. Nucleic Acids Research 34, W298–W302 (2006)
5. National Center for Biotechnology Information (NCBI), `http://www.ncbi.nlm.nih.gov`
6. Hubbard, T., Andrews, D., Caccamo, M., Cameron, G., Chen, Y., Clamp, M., Clarke, L., Coates, G., Cox, T., Cunningham, F., Curwen, V., Cutts, T., Down, T., Durbin, R., Fernandez-Suarez, X.M., Gilbert, J., Hammond, M., Herrero, J., Hotz, H., Howe, K., Iyer, V., Jekosch, K., Kahari, A., Kasprzyk, A., Keefe, D., Keenan, S., Kokocinsci, F., London, D., Longden, I., McVicker, G., Melsopp, C., Meidl, P., Potter, S., Proctor, G., Rae, M., Rios, D., Schuster, M., Searle, S., Severin, J., Slater, G., Smedley, D., Smith, J., Spooner, W., Stabenau, A., Stalker, J., Storey, R., Trevanion, S., Ureta-Vidal, A., Vogel, J., White, S., Woodwark, C., Birney, E.: Ensembl 2005. Nucleic Acids Research 33(Database issue), D447–D453 (2005)
7. The UniProt Consortium. The Universal Protein Resource (UniProt). Nucleic Acids Research 35, 193–197 (2007)
8. Bader, G.D., Betel, D., Hogue, C.V.W.: BIND: the Biomolecular Interaction Network Database. Nucleic Acids Research 31(1), 248–250 (2003)
9. Bader, G.D., Donaldson, I., Wolting, C., Ouellette, B.F., Pawson, T., Hogue, C.W.: BINDThe Biomolecular Interaction Network Database. Nucleic Acids Research 29(1), 242–245 (2001)
10. Bader, G.D., Hogue, C.V.W.: BINDa data specification for storing and describing biomolecular interactions, molecular complexes and pathways. Bioinformatics 16(5), 465–477 (2000)
11. Peri, S., Navarro, J.D., Amanchy, R., Kristiansen, T.Z., Jonnalagadda, C.K., Surendranath, V., Niranjan, V., Muthusamy, B., Gandhi, T.K., Gronborg, M., Ibarrola, N., Deshpande, N., Shanker, K., Shivashankar, H.N., Rashmi, B.P., Ramya, M.A., Zhao, Z., Chandrika, K.N., Padma, N., Harsha, H.C., Yatish, A.J., Kavitha, M.P., Menezes, M., Choudhury, D.R., Suresh, S., Ghosh, N., Saravana, R., Chandran, S., Krishna, S., Joy, M., Anand, S.K., Madavan, V., Joseph, A., Wong, G.W., Schiemann, W.P., Constantinescu, S.N., Huang, L., Khosravi-Far, R., Steen, H., Tewari, M., Ghaffari, S., Blobe, G.C., Dang, C.V., Garcia, J.G., Pevsner, J., Jensen, O.N., Roepstorff, P., Deshpande, K.S., Chinnaiyan, A.M., Hamosh, A., Chakravarti, A., Pandey, A.: Development of human protein reference database as an initial platform for approaching systems biology in humans. Genome Research 13, 2363–2371 (2003)

12. Mishra, G., Suresh, M., Kumaran, K., Kannabiran, N., Suresh, S., Bala, P., Shivkumar, K., Anuradha, N., Reddy, R., Raghavan, T.M., Menon, S., Hanumanthu, G., Gupta, M., Upendran, S., Gupta, S., Mahesh, M., Jacob, B., Matthew, P., Chatterjee, P., Arun, K.S., Sharma, S., Chandrika, K.N., Deshpande, N., Palvankar, K., Raghavnath, R., Krishnakanth, K., Karathia, H., Rekha, B., Rashmi, N.S., Vishnupriya, G., Kumar, H.G.M., Nagini, M., Kumar, G.S.S., Jose, R., Deepthi, P., Mohan, S.S., Gandhi, T.K.B., Harsha, H.C., Deshpande, K.S., Sarker, M., Prasad, T.S.K., Pandey, A.: Human Protein Reference Database - 2006 Update. Nucleic Acids Research 34, D411–D414 (2006)
13. Chatr-aryamontri, A., Ceol, A., Palazzi, L.M., Nardelli, G., Schneider, M.V., Castagnoli, L., Cesareni, G.: MINT: the Molecular INTeraction database. Nucleic Acids Research 35(Database issue), D572–D574 (2007)
14. Munich Information Center for Protein Sequences (MIPS), `http://mips.gsf.de`
15. Kerrien, S., Alam-Faruque, Y., Aranda, B., Bancarz, I., Bridge, A., Derow, C., Dimmer, E., Feuermann, M., Friedrichsen, A., Huntley, R., Kohler, C., Khadake, J., Leroy, C., Liban, A., Lieftink, C., Montecchi-Palazzi, L., Orchard, S., Risse, J., Robbe, K., Roechert, B., Thorneycroft, D., Zhang, Y., Apweiler, R., Hermjakob, H.: IntAct Open Source Resource for Molecular Interaction Data. Nucleic Acids Research 35(Database issue), D561–D565 (2007)
16. Hermjakob, H., Montecchi-Palazzi, L., Lewington, C., Mudali, S., Kerrien, S., Orchard, S., Vingron, M., Roechert, B., Roepstorff, P., Valencia, A., Margalit, H., Armstrong, J., Bairoch, A., Cesareni, G., Sherman, D., Apweiler, R.: IntAct: an open source molecular interaction database. Nucleic Acids Research 32(Database issue), D452–D455 (2004)
17. Salwinski, L., Miller, C.S., Smith, A.J., Pettit, F.K., Bowie, J.U., Eisenberg, D.: The Database of Interacting Proteins: 2004 update. NAR 32(Database issue), 449–451 (2004)
18. Apweiler, R., Bairoch, A., Wu, C.H.: Protein sequence databases. Current Opinion in Chemical Biology 8, 76–80 (2004)
19. INSDC: International Nucleotide Sequence Database Collaboration, `http://www.insdc.org`
20. Mrowka, R., Patzak, A., Herzel, H.: Is There a Bias in Proteome Research? Genome Research 11, 1971–1973 (2001)
21. The Cancer Cell Map, `http://www.cellmap.org`
22. Lochovsky, L.: An Entity Resolution Framework for Deduplicating Proteins. MSc thesis. University of Toronto (2008)
23. Lee, M.L., Ling, T.W., Low, W.L.: Designing Functional Dependencies for XML. In: Jensen, C.S., Jeffery, K.G., Pokorný, J., Šaltenis, S., Bertino, E., Böhm, K., Jarke, M. (eds.) EDBT 2002. LNCS, vol. 2287, pp. 124–141. Springer, Heidelberg (2002)
24. Tatusova, T.A., Madden, T.L.: Blast 2 sequences - a new tool for comparing protein and nucleotide sequences. FEMS Microbiol Lett. 174, 247–250 (1999)
25. Damerau, F.J.: A technique for computer detection and correction of spelling errors. Communications of the ACM 7(3), 171–176 (1964)
26. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions, and reversals. Soviet Physics Doklady 10, 707 (1966)
27. Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Morgan Kaufman, Burlington (2001)