Uncertain Groupings: Probabilistic combination of grouping data

Brend Wanders
University of Twente – Faculty EEMCS – Enschede, the Netherlands
b.wanders@utwente.nl

Maurice van Keulen
University of Twente – Faculty EEMCS – Enschede, the Netherlands
m.vankeulen@utwente.nl

Paul van der Vet
University of Twente – Faculty EEMCS – Enschede, the Netherlands
paul@vandervet-ca.nl

ABSTRACT

A bioinformatician has a large number of homology data sources to choose from. These data sources need to be combined before a query can be posed over the combined data. We propose a generic probabilistic approach to combining grouping data from multiple sources. Our approach incorporates an iteratively evolving view on trust, allowing the bioinformatician to express his fine-grained view on how much the data in the sources can be trusted. We evaluate our approach by combining 3 real-world biological databases and show that it scales well for realistic amounts of data and uncertainty.

1. INTRODUCTION

In the bioinformatics field, a number of databases contain homology data. Homology data consists of groupings of proteins that are expected to have the same function in different species. We use this as a real-world use case, which is further discussed in Section 1.2. In this paper we propose a technique for the combination of data sources describing groups of things.

We envision the proposed technique as part of a larger workflow in bioinformatics research. A bioinformatician has a large number of data sources to choose from. These data sources are created and cultivated by different institutes. Some of the sources are curated or partially curated, while others are automatically generated. Though bioinformaticians are knowledgeable in the field and aware of the different data sources at their disposal, they do not know the exact intricacies of each data source.

For their research, the bioinformatician wishes to query multiple data sources. Their main goal is, however, not to query these data sources; it is to extract the information they need in the form of a usable answer. So, all time spent on the integration of data sources, whether before or during querying, is time taken away from their 'core business' of investigating biological research questions.

Data sources need to be combined before a query can be posed over the combined data. Most data sources are created with a specific purpose in mind and combining them means repurposing them for something else. To combine and repurpose the data sources, the data in the sources must be understood first. Data understanding is a continuous process, with the bioinformatician's understanding of the intricacies of each data source growing over time. In this process of re-purposing the data, the bioinformatician needs to be able to express and refine his evolving opinion regarding trust into whole sources, or certain parts thereof, and then query and analyze the result of his actions to see how they reflect on the results.

Our technique is an instrument that allows the bioinformatician to express his fine-grained view on how much the data in the sources can be trusted, and query the data while taking into account that view.

1.1 Focus of this paper

The technique we propose works for categorizations and groupings of things. Such groupings are often encountered in data sources. They originate from automatic classifiers such as machine learning or data mining approaches, but also from human experts. Such data sources are not guaranteed to be correct. Measurement errors, data entry errors, or predictive heuristics may produce partially incorrect data.

For example, an administration of project teams may be incorrect if it can not keep up with people moving from team to team, get ill for possibly longer periods, etc. A solution direction for higher data quality here, would be to combine the administration with other independent data sources or other methods for determining team membership. For example, company-wide software for cooperative work (discussion boards, task boards, etc.) may be used to extract an apparent cooperation, hence team membership.

Another example is the classification of scientific articles. Libraries typically use both manual as well as automatic classification mechanisms. The correctness of the resulting classifications are affected by either the judgement of human classifiers or by the applied automatic keyword clustering algorithms. By combining multiple sources of article classifications (curated indices, automatic keyword clustering results, etc.), one may improve the overall quality of the classification.

Combining data sources that describe groupings is a challenging problem. Our goal is to automatically combine multiple sources into a single, higher quality representation of the grouping. We accomplish this with a technique for handling inconsistencies and ambiguity at various levels of granularity. This combination of data brings with it a repurposing of the data.

Given a high-level trust or resolution approach, we construct a probabilistic representation that can be stored and queried directly with current probabilistic database technology. We call this probabilistic representation, an *uncertain grouping*. We start with showing how an uncertain grouping can be constructed from a simple and rather crude trust approach like 'one-data-source-is-correct' on a real-world bioinformatics use case. We subsequently show how finer trust and resolution mechanisms can be used and that querying the constructed probabilistic database scales well.

Contributions. In this paper we present a technique for combining grouping data from multiple sources. The main contributions of this paper are:

- A generic probabilistic approach to combining grouping data in which an evolving view on trust can be iteratively incorporated.
- An experimental evaluation on a real-world bioinformatics use case.

The rest of this paper is laid out as follows: the next section discusses the real-world use case, followed by an overview of related work. Section 2 presents a formalization of our technique and on how a view on trust can evolve. Section 3 describes the experimental evaluation and discusses the results. Section 4 discusses, among other things, the complexity of the use case and the scalability of our technique. We conclude the paper with Section 5.

1.2 Use case

Our real-world use case comes from bioinformatics and concerns groups of *orthologous* proteins. Proteins in the same group are expected to have the same function(s).

The main goal of orthology is to conjecture the function of a gene or protein. Suppose we have identified a protein in disease-causing bacteria that, if silenced by a medicine, will kill the bacteria. A bioinformatician will want to make sure that the medicine will not have serious side-effects in humans. A normal procedure is to try to find orthologous proteins. If such proteins exist, they may also be targeted by the medicine, thus potentially causing side-effects.

We explain orthology, and orthologous groups, with an example featuring a fictitious paperbird taxa (see Figure 1). This example will be used throughout the paper.

The evolution of the paperbird taxa started with the Ancient Paperbird, the extinct ancestor species of the paperbird genus. Through evolution the Ancient Paperbird species split into multiple species, the three prominent ones being the Long-beaked Paperbird, the Hopping Paperbird and the Running Paperbird. The Ancient Paperbird is conjectured to have genes $K\ L\ M$. After sequencing of their genetic code, it turns out that the Long-beaked Paperbird species has genes $A\ F$, the Hopping Paperbird species has genes $B\ D\ G$, and the Running Paperbird species has $C\ E\ H$.

For the sake of the example, the functions of the different genes are known to the reader. With real taxa, the functions of genes can be ambiguous. For the paperbird species, genes A, B and C are known to influence the beak's curvature. D and E influencing the beak's length. Finally, genes F, G and H are known to influence the flexibility of the legs.

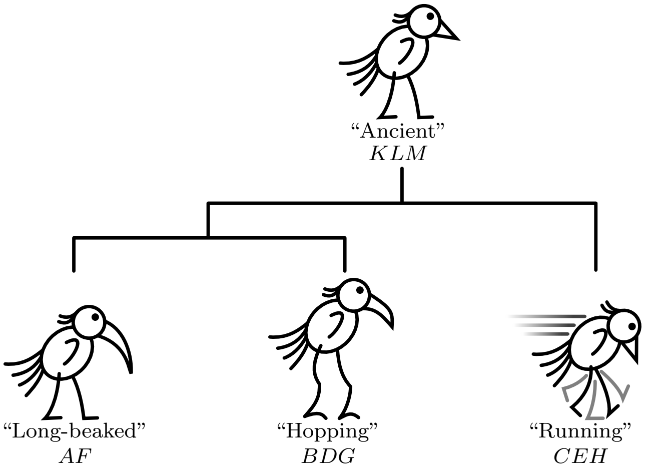


Figure 1: Paperbirds, hypothetical phylogenetic tree annotated with species names and genes.

D and E are known to govern the length of the beak. Based on this, on the similarity between the two sequences, and on the conjectured function of the beak curvature function ancestor gene L, we call D and E orthologous, with L as common ancestor. Orthology relations are ternary relations between three genes: two genes in descendant species and the common ancestor gene from which they are evolved. The common ancestor is hypothetical. An orthologous group is defined as a group of genes with orthologous relations to every other member in the group. In this case, the group DE is an orthologous group. Proteins can by analogous arguments also be called orthologs. An extended review of orthology can be found in [5].

There are various computational methods for determining orthology between genes from different species [7, 1]. These methods result in databases that contain groups of proteins or genes that are likely to be orthologous. Such databases are often made accessible to the scientific community. In our research, we aim to combine the insight into orthologous groupings contained in Homologene [10], PIRSF [15], and eggNOG [12]. An automatic combination of these sources may provide a continuously evolving representation of the current combined scientific insight into orthologous groupings of higher quality than any single heuristic could provide for other bioinformaticians to utilize.

A distinction commonly made is that between orthologous and paralogous proteins. Whereas an orthologous relation is established through speciation (the formation of a new species), paralogous relations are established through duplication. Looking back at the paperbird example, suppose that L is duplicated into L' and L'' in the Ancient Paperbird before it splits into two species. The Hopping Paperbird then features D' and D'', and the Running Paperbird features E' and E''. The relation between D' and E' is paralogous.

One of the main problems is to distinguish between orthologs and paralogs. Computational methods are scrutinized for the way they make that distinction. Databases may disagree over which genes or proteins form an orthologous group, which are paralogs, and what the hypothesized common ancestor is.

1.3 Related Work

Uncertainty forms an important aspect of data integration. Both the uncertainty created during the integration, as well as the integration of sources that contain uncertain data. [9] offers a comprehensive survey of the relevance of uncertainty management in data integration. Of special note is [8], which applies uncertain data integration in the context of biological databases by integrating heterogeneous data sources necessary for functional annotation of proteins.

Biological data sources are usually available in the form of a database. We want to have the product of the data combination available as a database as well. Probabilistic databases such as MayBMS [2] and Trio [14] allow the use of normal database techniques to apply to probabilistic data. As such, they provide a platform on which uncertain data integration can be implemented.

[6] Presents the tool ProGMAP for the comparison of orthologous protein groups from different databases. Instead of integrating protein groups, ProGMAP assists the user in comparing protein groups by providing statistical insight. Groups are compared pairwise and various visual display methods assist the user in assessing the strengths and weaknesses of each database. Our approach differs from ProGMAP in that we want to provide the user with a technique to query the combined data sources, instead of assisting the user in comparing them.

Current work in uncertain data integration is focused on entity resolution and schema integration. To the best of the authors' knowledge, no previous work using a uncertain data integration approach for the integration of classifications or groupings has been presented.

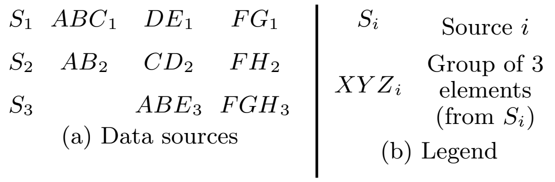
2. UNCERTAIN GROUPINGS

Different data sources offer their own view of the world: they way that data source claims is the correct way of grouping the elements. In an abstract sense, a *grouping* is a set of groups where each group is composed of elements. Without loss of generality, we view our data sources as databases storing only groups and elements, i.e., one particular grouping.

A user of data sources, such as the bioinformatician in our use case, will approach them with a critical attitude: one may be correct, certain subsets of a data source or how the data sources (dis)agree increases or decreases the confidence in its correctness, perhaps all of them are incorrect in some cases, etc. Therefore, an uncertain grouping is a grouping of elements for which the true grouping is unknown, but which faithfully represents the user's critical and fine-grained view on how much the data elements and query results can be trusted. Furthermore, the uncertain grouping should allow for scalable querying of typical queries like "Which elements are in the same group as e?"

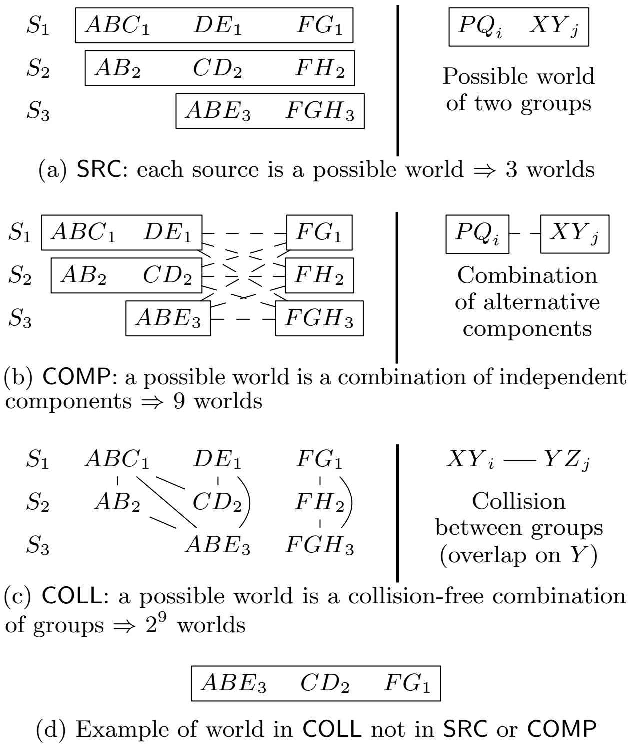
We model an uncertain grouping as probabilistic data adhering to the possible worlds model. In this model, an uncertain grouping is a compact representation of many possible groupings: the possible worlds. Probabilistic database technology is known to allow for scalable querying of an exponentially growing number of possible worlds [3]. Querying in a possible worlds model means that the query result is equivalent with evaluating the query on each possible world individually and combining those answers into one probabilistic answer.

2.1 Running example



S_1 ABC_1 DE_1 FG_1
S_2 AB_2 CD_2 FH_2
S_3 ABE_3 FGH_3
(a) Data sources
S_i Source i
XYZ_i Group of 3 elements (from S_i)
(b) Legend

Figure 2: Running example.



S_1 ABC_1 DE_1 FG_1
S_2 AB_2 CD_2 FH_2
S_3 ABE_3 FGH_3
PQ_i XY_j
Possible world of two groups
S_1 ABC_1 DE_1 FG_1
S_2 AB_2 CD_2 FH_2
S_3 ABE_3 FGH_3
PQ_i XY_j
Combination of alternative components
S_1 ABC_1 DE_1 FG_1
S_2 AB_2 CD_2 FH_2
S_3 ABE_3 FGH_3
XY_i — YZ_j
Collision between groups (overlap on Y)
ABE_3 CD_2 FG_1

(a) SRC: each source is a possible world ⇒ 3 worlds

(b) COMP: a possible world is a combination of independent components ⇒ 9 worlds

(c) COLL: a possible world is a collision-free combination of groups ⇒ 2^9 worlds

(d) Example of world in COLL not in SRC or COMP

Figure 3: Example of uncertain grouping.

Figure 2 presents three data sources, each containing two or three orthologous groups for our running example. We use the notation XYZ_i for a group of three elements, X, Y, and Z originating from source S_i. Observe that not every source is complete, for example, S_2 does not mention E. It depends on the source what this absence means:

- E is implicitly a group on its own,
- E is does not belong to any group, or
- it is unknown to which group E belongs.

2.2 Flexible trust views

From Section 1.2, we know that in our fictitious reality A B C, D E, and F G H is the correct grouping. Observe that none of the sources in Figure 2 is complete and fully correct. A bioinformatician integrating these sources, however, does not know what is the correct grouping, not even how well (s)he can trust the data. The goal to determine based on current scientific knowledge contained in the sources, what the correct grouping is, or rather, the confidence in possible groupings.

Our method of working with flexible trust views is iterative, i.e., one starts with a simple view on how the data should be integrated and trusted based on initial assumptions that may or may not be correct. By evaluating and using the integrated result, a bioinformatician gains more understanding in the data, which (s)he uses to adapt and refine the *trust view*: the assumptions and rules for data integration and trust. The reason behind this way of working is, that we believe, as we stated before, that data understanding is a continuous process, with the bioinformaticians understanding of the intricacies of each data source growing over time. With the trust view method, the bioinformatician is able to express and refine his evolving opinion regarding trust into whole sources, or certain parts thereof, and then query and analyze the result of his actions to see how they reflect on the results. In the sequel, we illustrate the method by going through three iterations, each centered around a different trust view (SRC, COMP, and COLL, respectively) and evaluate the evolving integrated data.

Suppose we would start with taking the simplistic view of 'one-data-source-is-correct', SRC for short: the belief that one source is entirely correct, but it is unknown which one. In this view, each data source is a possible world (see Figure 3(a)). There is basically one *choice*: which alternative data source is the correct one: S_1, S_2, or S_3.

Other more fine-grained views on trusting the data in the sources lead to more choices. For example, one could argue that the disputes among the sources around elements A, B, C, D, E and around F, G, H are independent of each other, hence that, say, S_1 could be correct on the component A, B, C, D, E and S_2 on F, G, H. In this view, the combination $\{ABC_1, DE_1, FH_2\}$ should be among the possible worlds (see Figure 3(b)). The general rule of this view, COMP for short, is that the independent *component*s of groups under dispute, can be freely combined to form possible worlds. In the example, the view results in two independent choices with each three alternatives resulting in $3 \times 3 = 9$ possible worlds.

To illustrate the flexibility of our approach, we present a third even more fine-grained collision-based trust view, called COLL. Two groups *collide* iff they overlap but are not equal.[1] Figure 3(c) shows the collisions between groups in our example. The idea behind the COLL-view on trust is that if two sources disagree on a group, i.e., the groups collide, only one can be correct.[2] In other words, each collision is in essence a choice. Note, however, that there are dependencies between these choices. For example, consider collisions ABC_1–AB_2 and DE_1–CD_2. If they were independent, then $2 \times 2 = 4$ combinations of groups would be possible, but the combination $\{ABC_1, CD_2\}$ violates the important grouping property that each element can only be a member of one group. Therefore, the general rule for this trust view is that all *collision-free* combinations of groups form the possible worlds. Figure 3(d) illustrates that the COLL method is indeed more fine-grained by presenting a possible world that is not considered by the SRC or COMP methods. Without any dependencies, n binary choices would generate 2^n possible worlds. In the example, the view would result in $2^9 = 512$ worlds if there would be no dependencies. With dependencies, the number of possible worlds in the example is reduced to 40 (including the empty world).

symbol	description
d, g, e	data item, group / element data item
$\mathrm{D} = \mathrm{D}_G \cup \mathrm{D}_E$	database / possible world
$\bar{\mathrm{D}} = (\dot{\mathrm{D}}, W)$	probabilistic database
$\dot{\mathrm{D}}$	compact representation (set of tuples with associated wsds)
W	world set (all possible rvas with their probabilities)
φ	world set descriptor (wsd; set of rvas)
$(r \mapsto v)$	random variable assignment (rva)
θ	valuation (set of rvas inducing set of possible worlds $\theta(\bar{\mathrm{D}})$)
$\mathrm{P}(\cdots)$	probability of possible world or rva

Table 1: Overview of notation.

Typically one would have many more considerations, sometimes rather fine-grained, that one would like to 'add' to one's trust view. For example, a bioinformatician may believe that groups CD_2 and FH_2 are extra untrustworthy, because he holds the opinion that the research group who determined those results is rather sloppy in the execution of their experiments. Or, he may have more trust in curated data, or even different levels of trust for data curated by different people or committees. Our approach can incorporate such considerations as well.

2.3 Formalization

In this section, we provide a formalization of a probabilistic database consisting of an uncertain grouping. The formalization is based on [13] which provides a generic formalization of a probabilistic database. We summarize the main concepts of [13] (DEFINITIONS) and show how it can be specialized to support uncertain groupings (SPECIALIZATIONS). Table 1 gives an overview of our notation. In Section 2.4 we subsequently show how an uncertain grouping can be constructed for a certain trust view.

DEFINITION 1 (DATABASE; DATA ITEM). *We model a 'normal'* database $\mathrm{D} \in \mathbb{P}\mathcal{D}$ *in an abstract way as a set of* data items. *Typically, a data item $d \in \mathcal{D}$ would be a tuple for a relational database or a triple for an RDF store, but in essence it can be anything.*

SPECIALIZATION 1 (ELEMENT; GROUP). *We define two special kinds of data items as disjoint subsets of $\mathcal{D}$:*

- Elements $e \in \mathcal{D}_E$*, and*
- Groups $g \in \mathcal{D}_G$*, where* $\mathcal{D}_G = \{g \mid g \subseteq \mathcal{D}_E\}$.

SPECIALIZATION 2 (DATA SOURCE). *Without loss of generality, we define a* data source *as a database* D *containing only elements and groups:* $\mathrm{D} = \mathrm{D}_G \cup \mathrm{D}_E$ *with* $\mathrm{D}_G \subseteq \mathcal{D}_G$ *and* $\mathrm{D}_E \subseteq \mathcal{D}_E$.

DEFINITION 2 (PROBABILISTIC DATABASE). *A* probabilistic database $\bar{\mathrm{D}}$ *is a database capable of handling huge volumes of data items and possible alternatives for these data items while still being able to efficiently query and update.* Possible world theory *views a probabilistic database as a set of possible databases* D_i*, also called* possible worlds*, each with a probability* $\mathrm{P}(\mathrm{D}_i)$.

Obviously, an implementation would not store the possible worlds individually, but as a compact representation capable

[1] This second condition 'not equal' is theoretically not necessary (See Section 2.4).

[2] Actually, this is a simplification: both can be incorrect. We discuss this issue in Section 4.3

of representing vast numbers of possible worlds in limited space. Possible world theory prescribes that a query Q on a compact representation should result in a compact answer representing all possible answers (equivalent with evaluating Q in each world individually).

Our compact representation is based on modeling uncertainty, the 'choices' of Section 2.2 in particular, with random events. Method SRC of the running example results in one choice: which of the three data sources is the correct one. We introduce a random variable $r \in \mathcal{R}$ with three possible assignments $(r \mapsto 1)$ representing S_1 is correct, $(r \mapsto 2)$ representing S_2 is correct, and $(r \mapsto 3)$ representing S_3 is correct.

Definition 3 (rv, rva, world set). *We call the collection of all possible random variable assignments (rvas for short) with their probabilities a* world set $W \in \mathcal{R} \rightsquigarrow \mathcal{V} \rightsquigarrow [0\,..\,1]$. *We denote with* $\mathrm{P}(r \mapsto v) = W(r)(v)$ *the probability of a rva; the probabilities of all alternatives for one random variable* $r \in \mathcal{R}$ *(rv for short) should add up to one.*

In the example, $W = \{r \mapsto \{1 \mapsto p_1, 2 \mapsto p_2, 3 \mapsto p_3\}\}$. Because all alternatives for one rv should add up to one, $p_1 + p_2 + p_3 = 1$.

Definition 4 (wsd). *Alternative data items are linked to the world set by means of world set descriptors (wsd)* φ. *A wsd is a conjunction*[3] *of rvas* $(r_i \mapsto v_i)$. *The wsd determines for which rvas, hence for which possible worlds, the data item exists.*

Definition 5 (compact representation). *The compact representation can now be defined as* $\bar{\mathrm{D}} = (\dot{\mathrm{D}}, W)$, *i.e., a set of data items each with a wsd* $\dot{\mathrm{D}}$ *and a world set* W.

In the example, there are eight groups which can be linked to the appropriate rva. See Figure 3(b) for an illustration. Note that in a concrete database, the data is normalized into three tables: **group** containing at least an identifier for each group, **element** containing all elements, and **group_element** describing which element belongs to which group. Only **group** is uncertain in this case, i.e, its tuples need to have the shown wsds φ.

Definition 6 (valuation). *'Considering a case' means that we choose a value for one or more random variables and reason about the consequences of this choice. We call such a choice a* valuation θ. *If the choice involves all the variables of the world set, the valuation is total.*

Definition 7 (possible world). *A total valuation induces a single possible world:* $\theta(\bar{\mathrm{D}}) = \{d \mid (d, \varphi) \in \dot{\mathrm{D}} \wedge \varphi(\theta)\}$, *where* $\varphi(\theta) = true$ *iff forall* $(r_i \mapsto v) \in \theta$, *there is no* $(r_i \mapsto v')$ *in* φ *such that* $v \neq v'$. *We denote with* $\mathrm{PWS}(\bar{\mathrm{D}})$ *the set of all possible worlds, and with* $\mathrm{P}(\mathrm{D})$ *the probability of a world* D.

For example, the valuation $\theta = \{r_1 \mapsto 1, r_2 \mapsto 2\}$ induces the combination $\{ABC_1, DE_1, FH_2\}$. In this way, the concept of valuation bridges the gap between the compact representation and possible world theory.

$\dot{\mathrm{D}}$

	group	φ
d_1	ABC_1	$(r_1 \mapsto 1)$
d_2	DE_1	$(r_1 \mapsto 1)$
d_3	FG_1	$(r_2 \mapsto 1)$
d_4	AB_2	$(r_1 \mapsto 2)$
d_5	CD_2	$(r_1 \mapsto 2)$
d_6	FH_2	$(r_2 \mapsto 2)$
d_7	ABE_3	$(r_1 \mapsto 3)$
d_8	FGH_3	$(r_2 \mapsto 3)$

W

rva	P	
$(r_1 \mapsto 1)$	p_1	'S_1 is correct' for component A, B, C, D, E
$(r_1 \mapsto 2)$	p_2	'S_2 is correct' for component A, B, C, D, E
$(r_1 \mapsto 3)$	p_3	'S_3 is correct' for component A, B, C, D, E
$(r_2 \mapsto 1)$	p_4	'S_1 is correct' for component F, G, H
$(r_2 \mapsto 2)$	p_5	'S_2 is correct' for component F, G, H
$(r_2 \mapsto 3)$	p_6	'S_3 is correct' for component F, G, H

Figure 4: Probabilistic database representation $\bar{\mathrm{D}} = (\dot{\mathrm{D}}, W)$ for the uncertain grouping constructed under trust view COMP (see Figure 3(b)).

Queries can be evaluated directly on the compact representation to obtain a compact representation of all possible answers. For example, the query "which elements are in the same group as A?" can be evaluated by selecting all groups containing A, which results in three tuples d_1, d_4, and d_7. Observe that these tuples are mutually exclusive, because their wsds contain an rva for r_1 with different values.

From this compact representation, one can derive different kinds of answers to the query, such as, the answer in the most likely world, the most likely answer (not necessarily the same, because different worlds may agree on an answer, hence the probability of that answer is the sum of the probabilities of the worlds that agree on that answer), or the second most likely answer. For numerical queries, one can derive the minimum, maximum, expected value, standard deviation, etc. In this example, we may derive that C and E are only in the same group as A if the respective group exists, i.e., under valuations $\{(r_1 \mapsto 1)\}$ and $\{(r_1 \mapsto 3)\}$, respectively. Therefore, C is homologous with A with a probability of p_1 and E is homologous with A with a probability of p_3. Observe that B is in the same group as A in all three tuples, hence it is homologous with A with a probability of $p_1 + p_2 + p_3 = 1$.

We like to emphasize that the above is a summary of the main concepts of [13] which provides a generic formalization of a probabilistic database. In addition to summarizing, we have also shown how the formalization can be specialized to support uncertain groupings. For a more detailed presentation of the generic formalization, we refer to [13].

2.4 Trust views revisited

We argue that trust can be modelled in terms of choices that can be formalized with random events, which in turn can be represented in a probabilistic database with random variables and annotating tuples with world set descriptors composed of random variable assignments. In this section, we like to emphasize the flexibility of the approach.

Consider for example the probabilistic database constructed according to trust view COLL. Observe how the 9 collisions result in 9 random variables in a straightforward way. Fur-

[3] Theoretically an arbitrary propositional formula with $\wedge$, $\vee$, and $\neg$ is possible, but here a simple conjunction suffices.

$\dot{\mathrm{D}}$

	group	φ
d_1	ABC_1	$(r_1 \mapsto 1) \wedge (r_2 \mapsto 1) \wedge (r_3 \mapsto 1)$
d_2	DE_1	$(r_5 \mapsto 1) \wedge (r_6 \mapsto 1)$
d_3	FG_1	$(r_7 \mapsto 1) \wedge (r_8 \mapsto 1)$
d_4	AB_2	$(r_1 \mapsto 2) \wedge (r_4 \mapsto 1)$
d_5	CD_2	$(r_2 \mapsto 2) \wedge (r_5 \mapsto 1)$
d_6	FH_2	$(r_7 \mapsto 2) \wedge (r_9 \mapsto 1)$
d_7	ABE_3	$(r_3 \mapsto 2) \wedge (r_4 \mapsto 2) \wedge (r_6 \mapsto 2)$
d_8	FGH_3	$(r_8 \mapsto 2) \wedge (r_9 \mapsto 2)$

W

rva	P	
$(r_1 \mapsto 1)$	p_1	'S_1 is correct' for collision ABC_1–AB_2
$(r_1 \mapsto 2)$	p_2	'S_2 is correct' for collision ABC_1–AB_2
$(r_2 \mapsto 1)$	p_3	'S_1 is correct' for collision ABC_1–CD_2
$(r_2 \mapsto 2)$	p_4	'S_2 is correct' for collision ABC_1–CD_2
$(r_3 \mapsto 1)$	p_5	'S_1 is correct' for collision ABC_1–ABE_3
$(r_3 \mapsto 2)$	p_6	'S_3 is correct' for collision ABC_1–ABE_3
$(r_4 \mapsto 1)$	p_7	'S_2 is correct' for collision AB_2–ABE_3
$(r_4 \mapsto 2)$	p_8	'S_3 is correct' for collision AB_2–ABE_3
$(r_5 \mapsto 1)$	p_9	'S_1 is correct' for collision DE_1–CD_2
$(r_5 \mapsto 2)$	p_{10}	'S_2 is correct' for collision DE_1–CD_2
$(r_6 \mapsto 1)$	p_{11}	'S_1 is correct' for collision DE_1–ABE_3
$(r_6 \mapsto 2)$	p_{12}	'S_3 is correct' for collision DE_1–ABE_3
$(r_7 \mapsto 1)$	p_{13}	'S_1 is correct' for collision FG_1–FH_2
$(r_7 \mapsto 2)$	p_{14}	'S_2 is correct' for collision FG_1–FH_2
$(r_8 \mapsto 1)$	p_{15}	'S_1 is correct' for collision FG_1–FGH_3
$(r_8 \mapsto 2)$	p_{16}	'S_3 is correct' for collision FG_1–FGH_3
$(r_9 \mapsto 1)$	p_{17}	'S_2 is correct' for collision FH_2–FGH_3
$(r_9 \mapsto 2)$	p_{18}	'S_3 is correct' for collision FH_2–FGH_3

Figure 5: Probabilistic database representation $\bar{\mathrm{D}} = (\dot{\mathrm{D}}, W)$ for the uncertain grouping constructed under trust view COLL (see Figure 3(c)).

thermore, the concept of collision-freeness is represented in the world set descriptors. For example, tuple ABC_1 can only exist if all collisions in which it is involved fall in its favour. It is important to understand that a query result contains all possible answers, each with a probability as a measure for the trustworthiness of the answer, essentially the combined probability of all worlds that agree on that answer. Note that the way we modelled COLL has as a consequence that all total valuations that would lead to a world with one or more collisions, in fact induce an empty database as possible world. One could, for example, normalize the probabilities of query answers with $1 - \mathrm{P}(\emptyset)$, which is the combined probability of all collision-free combinations.

Observe also how such an intricate trust view as COLL, does not produce more tuples in the **group** table, only the world set grows because of the higher number of choices, and the world set descriptors become larger because of the need to faithfully represent the dependencies between the existence of tuples caused by the collision-freeness condition. Nevertheless, this is only more data. We show in Section 3 that this does not cause scalability problems even in a voluminous real-world case such as homology.

Finally, we would like to emphasize that the process of discovering trust issues and imposing the associated consideration on the data but refining one's trust view, is an iterative process. We claim that such considerations can be imposed on the data by introducing more random variables and adding rvas to the wsds of the appropriate tuples. Recall, for example, the issue of the sloppy research group of Section 2.2. Here, one new random variable is introduced and a rva is added to the wsd of all tuples of this research group. After such a refinement, the bioinformatician obtains a database that can be directly queried so that he can examine its consequences. He thus iteratively refines his trust view until the data faithfully expresses his opinions as well as the result of any query or analysis run on this data.

3. EVALUATION

The experiments are based on a test database created from three actual homology databases and two query classes derived from queries commonly executed on homology databases.

3.1 Experimental Setup

For the evaluation, we constructed a test set of homology data from the Homologene (release 67, [10]), PIRSF (release 2012_03, [15]), and eggNOG (release 3.0, [12]) biological databases. The groupings from each of these databases were loaded into a single database for the construction of trust views and querying. Where necessary database-specific accession numbers were converted to UniProt accession numbers. This ensures that identical proteins in different groups are correctly referenced.

Commonly executed queries can be split up into two query classes, each class corresponding to a common question:

1. 'What protein is homologous with X?' with X from known proteins. This is the 'single' class.

2. 'Are X and Y homologues?' with X and Y from known proteins. This is the 'pair' class.

Based on these two classes we generate query suites for use in the evaluation.

The first query suite, which is used exclusively to determine average query times over all trust views, contains 1000 single queries and 1000 pair queries based on proteins sampled from the combined database. The sampled pairs are all guaranteed to have a homologous relation.

The second query suite, used for all further experiments, contains 100 single queries, and 200 pair queries. The singled queries were generated by sampling 100 proteins from the known proteins in the combined databases. The 200 pair queries were generated by sampling 100 pairs of proteins that have a homologous relation, and 100 pairs that are known to have no relation.

Random variable assignments for the trust views SRC, COMP and COLL were generated based analysis of the combined database. Uniform distribution is used to assign probabilities to the assignments.

We have implemented our technique on top of MayBMS. Because of building on top of existing software, we accept some technical limitations inherent in these systems. Overcoming these limitations is not the focus of our work. A note on the limitations can be found in Appendix A. Due to the technical limitations we can represent at most 500 rvas. Any rvas above 500 were discarded. Additional trust views based on COLL were generated with world set descriptors of sizes 450, 400, . . . , 100, 50. These trust views are referred to as COLLN, with N being the size of the world set descriptor, without size indication COLL500 is meant.

The experiments were conducted on an Intel i7 x86-64bit with 7.7GB ram running Linux 3.2.0. Compilation was done with gcc 4.6.3.

3.2 Experiments

3.2.1 *Mean query times*

The first experiment is conducted using the first query suite. The experiment process is as follows: each query in the query suite is repeated 10 times, the first time measurement is discarded to reduce the impact of caching on the measurements. The mean query time of each executed query is calculated based on the 9 measurements. The mean query times are used to determine the mean query time per trust view and the standard deviation of mean query time, both in milliseconds. This process was used for each of the three trust views SRC, COMP and COLL:

SRC mean: 18.627, std.dev.: 26.864

COMP mean: 19.061, std.dev.: 27.569

COLL mean: 23488.197, std.dev.: 93184.375

3.2.2 *World Set Descriptor size*

The second experiment is conducted to determine the impact of world set descriptor (wsd) size on query execution time. The second query suite is used, together with the trust views COLL50, COLL100 . . . COLL450, COLL500. The experiment process is as follows: each query in the query suite is repeated 10 times, the first measurement is discarded. The mean query time per query are calculated based on the 9 time measurements.

Figure 6(a) shows the mean query times over all 'single' queries and the mean times of each separate measurement. Figure 6(b) shows the same for 'pair' queries.

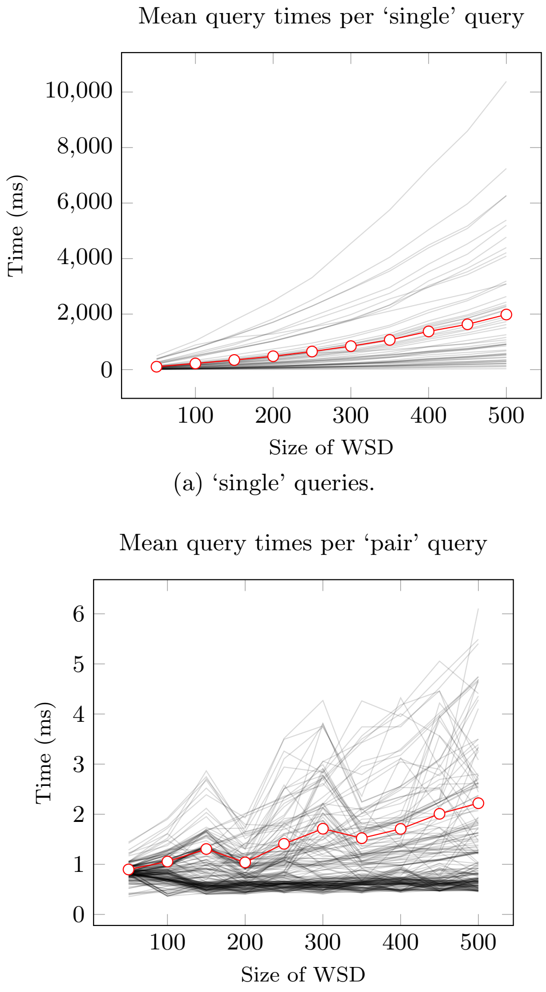
Compared to the time taken by the 'single' queries, all 'pair' queries are orders of magnitude faster due to the smaller amount of uncertainty per query result. The two drops in Figure 6(b) (at COLL200 and COLL350) are most likely due to favourable alignment of data in memory.

3.2.3 *Third Experiment*

The third, and final experiment, is conducted to measure the impact of the number of wsds and rvas on the query time. A counting function is used to count the number of wsds used to answer the query, and the number of unique rvas that were encountered while answering the the query. The counting function is applied to all queries from the 'single' and 'pair' suite for all trust views COLL50, . . . , COLL500.

Figure 7 shows the number of unique rvas plotted against the mean time of the query, results from all trust views are displayed. Figure 8 shows the unique number of rvas, the number of wsds used and the mean time of the query.

As can be seen in Figure 8 the framework handles real-world uncertainty very well. The larger part of the queries is handled within 2 seconds. The slower queries are slow due to a combination of a large number of unique random variable assignments and a large number of world set descriptors. Based on the mean query times from the first experiment, showing that only the trust view with a large amount of uncertainty takes time, and the measurements in the last experiment, we can conclude that the slowest factor is the exact confidence computation, not the modelling of the framework.



(a) 'single' queries.

(b) 'pair' queries.

Figure 6: Mean query time (in white-red) and distinct query times (in gray) for (a) 'single' and (b) 'pair' queries.

3.3 Discussion

In our experiments we use wsd size as an artificial bound on the amount of uncertainty. Both SRC and COMP feature only a single rva per group, and are therefore effectively equivalent with regards to execution speed. Due to technical constraints COLL has a maximum of 500 rvas per group. This is not a hinder for the evaluation, since by scaling down the size of the wsd we can simulate a data set with less uncertainty.

Our implementation uses a representation of wsds different from that of MayBMS (see Appendix A for more details). We measured the impact of converting this representation during the actual querying, and during the generation of the trust view. Queries involving small wsds were sped up if the conversion was done during the query, while queries involving large wsds were slowed down. In absolute terms, both the speedup and the slowdown were of little impact.

During the experiments, we encountered three measurements that qualified as outliers. Two outliers occurred during the measurements of the 'pair' queries. As the experiments

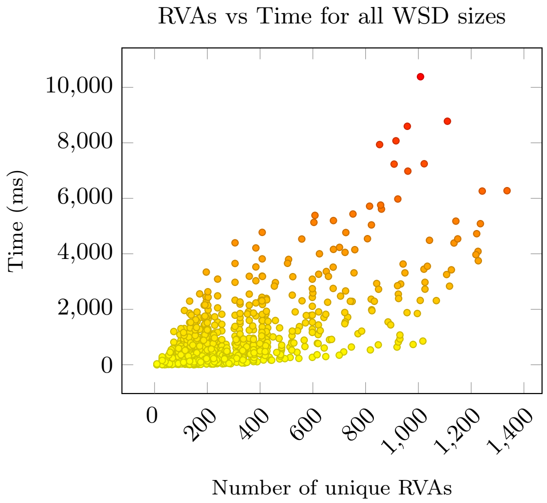


Figure 7: Unique rvas against mean time for all 'single' queries.

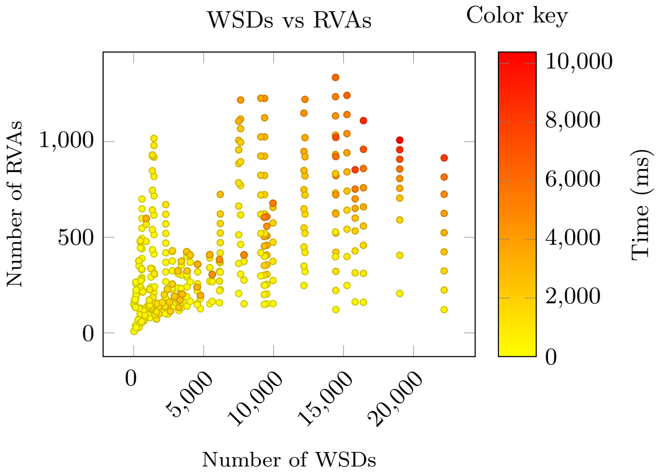


Figure 8: Number of wsds against unique rvas for all 'single' queries.

were conducted on a normal workstation, we strongly suspect that another program interfered with the query execution. One outlier occurred during the measurements of the 'single' queries, specifically the measurements for protein F6ZHU6 (a UniProt identifier). This protein is related to muscle activity. It is a member of a large number of orthologous groups, the cause of which is further discussed in Section 4.1.

While conducting the experiments, a small number of queries did not finish. We suspect the method we use to interface with MayBMS to be the cause. Because our implementation is intended as a research prototype we have not spent significant effort on finding the cause, as it is not scientifically relevant.

4. DISCUSSION

4.1 Complexity from practice: the use case revisited

An unsuspecting bioinformatician him/herself would perhaps, just like us, initially also assume that groups within one source are non-overlapping. For homology databases, one discovers that this is not true. According to bioinformatician A. Kuzniar whom we consulted about this issue: "the reason is that orthologous groups are nested as the orthology relations are defined based on a phylogenetic tree. Depending on how far you go back in time to infer these relations, e.g., for mammals (subset) vs. vertebrates (superset), there will be a different level of granularity in the orthologous groups. The overlap is between a superset and its subsets. However, things get more complicated when one also considers gene fusion events (hybrids) where two distinct genes in one species are fused together into a single gene in another species. In this instance, the tree model is inadequate and therefore one needs to resolve to a graph (network) model, see also [7]."

We have ignored these issues in our experiments as they are not relevant to our research questions. The way the issue has been encountered in our own research is, however, a nice illustration for data understanding being a continuous process happening concurrently with the re-purposing, combination, and analysis of data from multiple sources. A next step in the refinement of the trust view could be the proper incorporation of this discovery.

4.2 Scalability and confidence precision

The scalability of our technique is explained in two parts. The first part is normal relational data, this scales as well as can be expected from a relational database. We do not generate additional normal data, so the amount of tuples is equivalent to the union of tuples of the separate data sources. All overhead, both in terms of space and in terms of computation time, is generated by the random variable assignments. Normal queries are handled purely by the RDBMS and only the uncertainty adds computational overhead.

We currently use the exact confidence computation implemented by MayBMS and described in [4]. The COLL trust view generates one random variable assignment per collision. In this paper we only take the first 500 collisions into account due to technical reasons. We have observed groups that would generate as much as 17885 random variable assignments.

Because of this the exact confidence computation has to deal with extremely small chances. Further work needs to be done to see whether approximate confidence computation, such as in [11], can be done over large amounts of random variable assignments.

4.3 'Tunnel-vision': an answer to the open vs. closed world dilemma

Consider, for example, source S_1 and the fact that it doesn't mention H. Should this be interpreted (closed world assumption) as a statement that H is not orthologous to any protein, in particular, F and G? Or (open world assumption) that S_1 doesn't make a statement at all about H, i.e., it might be orthologous to any protein?

Considering only sources S_1 and S_2 — note that S_2 doesn't mention G — one could hold the view that it is possible for G and H to be orthologous as both are possibly orthologous to F according to the respective sources. There is, however, no possible world in the uncertain grouping of S_1 and S_2 where G and H are in the same group using any of the trust view methods presented. Hence, the trust views of Section 2.2 all follow a closed world assumption.

The universe of discourse here is the domain of all proteins. Assuming that this domain is finite, one could theoretically construct a trust view following an open world assumption by adding group tuples for all combinations of proteins and associating them with the appropriate wsds. In practice, this is of course infeasible due to sheer data explosion. Nevertheless, the idea can be applied in a restricted form: the world is assumed to be open only to the combined domain of the integrated sources, i.e., $\mathrm{D}_E^1 \cup \mathrm{D}_E^2$. We call this the 'tunnel-vision' world assumption as one doesn't view the world of the sources to be completely closed, also not completely open, but open/closed to the 'target world'.

In our example of combining S_1 and S_2, the combined domain of elements is $\mathrm{D}_E = \{A, \ldots, H\}$. A tunnel-vision view can be achieved by adding possible group tuples to S_1 that include H and possible group tuples to S_2 that include E and/or G. Using either of the trust view methods, an uncertain grouping is established that includes the possibility that G and H are orthologous at the expense of a limited number of tuples and only one rva per unmentioned element per source. Since the performance bottleneck of probabilistic databases does not reside in the query evaluation itself, but in the probability computation with growing wsds, a tunnel-view is expected to be feasible in practice.

4.4 Graph representation and optimization

During our research, we explored alternative representations based on graph theory. The investigated graph-based representation is one in which each orthology relation is represented as an edge, and each protein as a vertex. Although a translation can be made from a groupings representation to a graph representation, the translation from graph representation to groupings representation was found to be problematic. Questions like 'What other members are there in the groups containing protein X?' require clique-finding or a less precise form of clustering, which were found to be computationally undesirable.

This did lead us to an interesting venue for optimizing the COLL trust view: if a set of collisions forms a clique, that is if all groups are mutually exclusive with each other, these dependencies can be expressed with a single random variable. So any clique of n collision relations (which requires the introduction of n random variables and $2n$ random variable assignments) can be reduced to a single random variable and n random variable assignments.

This reduction does not change the semantics of the involved dependencies. It can be applied selectively on any number of cliques without creating an inconsistent state, allowing the optimization to be executed incrementally during idle time.

5. CONCLUSIONS

Motivated by a real-world use case we propose a generic technique to combine multiple groupings.

Homology data consists of groupings of proteins. The proteins in a group are expected to have the same function in different species. Homology data is relevant when, for example, a medicine is being developed and the potential for side-effects has to be determined. We combine 3 different biological databases containing homology data. We introduced this real-world use case of homology in Section 1.2.

Data understanding is a continuous process happening concurrently with the re-purposing, combination, and analysis of data from multiple sources. To allow querying over this combined data we employ a probabilistic approach to the handling of conflicting data sources. During the process of data combination an evolving view on trust can be iteratively incorporated. This is exemplified in this paper by three trust views (SRC, COMP, COLL).

We show, through experimental evaluation, that our proposed technique scales well. Our evaluation is based on realistic amounts of data obtained form the combination of 3 biological databases, yielding 776 thousand groups with a total of 14 million members and 2.8 million random variables. The experiments are conducted using typical queries for the use case.

Our technique allows the bioinformatician to focus on the semantics of the data sources, instead of on the technical details of integration. Integration choices can be modelled through the assignment of random variables, instead of through directly changing the data itself – allowing the bioinformatician to take a step back and look at the bigger picture, instead of worrying about each integration detail.

6. ACKNOWLEDGEMENTS

We would like to thank the late Tjeerd Boerman for his work on the use case and his initial concept of groupings. We would also like to thank Arnold Kuzniar for his insights and feedback on our use of biological databases and Ivor Wanders for his reviewing and editing assistance.

7. REFERENCES

[1] A. Altenhoff and C. Dessimoz. Phylogenetic and functional assessment of orthologs inference projects and methods. *PLoS Computational Biology*, 5:e1000262, 2009.

[2] L. Antova, T. Jansen, C. Koch, and D. Olteanu. Fast and simple relational processing of uncertain data. In *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*, pages 983–992. IEEE, 2008.

[3] L. Antova, C. Koch, and D. Olteanu. $10^{(10^6)}$ worlds and beyond: Efficient representation and processing of incomplete information. *The VLDB Journal*, 18(5):1021–1040, Oct. 2009.

[4] C. Koch and D. Olteanu. Conditioning probabilistic databases. *Proceedings of the VLDB Endowment*, 1(1):313–325, 2008.

[5] E. Koonin. Orthologs, paralogs, and evolutionary genomics. *Annual Review of Genetics*, 39:309–338, 2005.

[6] A. Kuzniar, K. Lin, Y. He, H. Nijveen, S. Pongor, and J. A. M. Leunissen. Progmap: an integrated annotation resource for protein orthology. *Nucleic Acids Research*, 37(suppl 2):W428–W434, 2009.

[7] A. Kuzniar, R. van Ham, S. Pongor, and J. Leunissen. The quest for orthologs: finding the corresponding gene across genomes. *Trends in Genetics*, 24:539–551, 2008.

[8] B. Louie, L. Detwiler, N. Dalvi, R. Shaker, P. Tarczy-Hornoch, and D. Suciu. Incorporating uncertainty metrics into a general-purpose data integration system. In *Scientific and Statistical*

Database Management, 2007. SSBDM '07. 19th International Conference on, pages 19–19, July 2007.

[9] M. Magnani and D. Montesi. A survey on uncertainty management in data integration. *J. Data and Information Quality*, 2(1):5:1–5:33, July 2010.

[10] NCBI Resource Coordinators. Database resources of the national center for biotechnology information. *Nucleic Acids Research*, 41(D1):D8–D20, 2013.

[11] D. Olteanu, J. Huang, and C. Koch. Approximate confidence computation in probabilistic databases. In *Data Engineering (ICDE), 2010 IEEE 26th International Conference on*, pages 145–156. IEEE, 2010.

[12] S. Powell, D. Szklarczyk, K. Trachana, A. Roth, M. Kuhn, J. Muller, R. Arnold, T. Rattei, I. Letunic, T. Doerks, et al. eggNOG v3. 0: orthologous groups covering 1133 organisms at 41 different taxonomic ranges. *Nucleic Acids Research*, 40, 2011.

[13] M. van Keulen. Managing uncertainty: The road towards better data interoperability. *IT - Information Technology*, 54(3):138–146, May 2012.

[14] J. Widom. Trio: A system for integrated management of data, accuracy, and lineage. Technical Report 2004-40, Stanford InfoLab, August 2004.

[15] C. H. Wu, A. Nikolskaya, H. Huang, L.-S. L. Yeh, D. A. Natale, C. R. Vinayaka, Z.-Z. Hu, R. Mazumder, S. Kumar, P. Kourtesis, R. S. Ledley, B. E. Suzek, L. Arminski, Y. Chen, J. Zhang, J. L. Cardenas, S. Chung, J. Castro-Alvear, G. Dinkov, and W. C. Barker. Pirsf: family classification system at the protein information resource. *Nucleic Acids Research*, 32(suppl 1):D112–D114, 2004.

APPENDIX

A. LIMITATIONS OF MAYBMS AND POSTGRESQL

We ran into several technical limitations of PostgreSQL and MayBMS. PostgreSQL tables are limited to 250-1600 columns, according to the manual. This means that the limit on expressing random variables using MayBMS' 3-column system is 83-533 without actual data and one less random variable for each three columns of data. So, with 2 columns used up by other data, we can support at most 532 random variables.

Furthermore, MayBMS' confidence computation aggregates are implemented through PostgreSQL and PostgreSQL can not pass more than 100 arguments to a function. This limits the number of random variables to 33.

To overcome the problem of not having more than 100 arguments to a function, we wrote our own representation of random variable assignments that is functionally equivalent to MayBMS' representation but allowed us to represent up to the limit of 532 random variable assignments. We did so by taking advantage of the PostgreSQL ability to use arrays as a column type, combined with our own implementation of a RVA base type to represent rvas and to use as the elements of the array. Our implementation uses a custom aggregation function to feed our representation to the MayBMS functions for confidence computation.