Uncertain Groupings: Probabilistic combination of grouping data

Brend Wanders, Maurice van Keulen, and Paul van der Vet

University of Twente – Faculty EEMCS – Enschede, the Netherlands
b.wanders@utwente.nl, m.vankeulen@utwente.nl, paul@vandervet-ca.nl

Abstract. Probabilistic approaches for data integration have much potential [7]. We view data integration as an iterative process where data understanding gradually increases as the data scientist continuously refines his view on how to deal with learned intricacies like data conflicts. This paper presents a probabilistic approach for integrating data on groupings. We focus on a bio-informatics use case concerning homology. A bio-informatician has a large number of homology data sources to choose from. To enable querying combined knowledge contained in these sources, they need to be integrated. We validate our approach by integrating three real-world biological databases on homology in three iterations.

1 Introduction

The field of bio-informatics is for an important part about combining available data sources in novel ways in a pursuit to answer new far-reaching research questions. A bio-informatician typically has a large number of data sources to choose from, created and cultivated by different research institutes. Some are curated or partially curated, while others are automatically generated based on certain biological methods.

Though bio-informaticians are knowledgeable in the field and aware of the different data sources at their disposal and methods used, they do not know the exact intricacies of each data source. Therefore, a bio-informatician typically obtains a desired integrated data set not in one attempt, but after several iterations of refinement.

Most data sources are created for a specific purpose. A bio-informatician's use typically goes beyond this foreseen use. The act of repurposing of the data, i.e., using the data for a purpose other than its intended purpose, is another source of integration complexity. For example, the quality of data in a certain attribute may be lower than required.

In short, data understanding is a continuous process, with the bio-informatician's understanding of the intricacies of data sources growing over time. It is therefore required that this evolving knowledge can be expressed and refined. We call this specification an "*integration view*". Querying and analyzing the result of a refined integration view produces more understanding which is in turn used to further refine the integration view.

In this paper, we focus on a particular bio-informatics scenario: homology. Several databases exist that contain homology data. In essence, homology data represents groups of proteins that are expected to have the same function in different species. Obtained by using different methods, the sources only partially agree on the homological relationships. Combining them allows for querying and analyzing the combined knowledge on homology.

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 integration can be iteratively refined.
- An experimental evaluation on a real-world bio-informatics use case.

The use case is further explained in Section 1.1. We then generalize the use case to the problem of integrating grouping data and elaborate on how our probabilistic integration approach addresses this problem in Section 1.2.

1.1 Use case

Our real-world use case comes from bio-informatics 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 bio-informatician 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

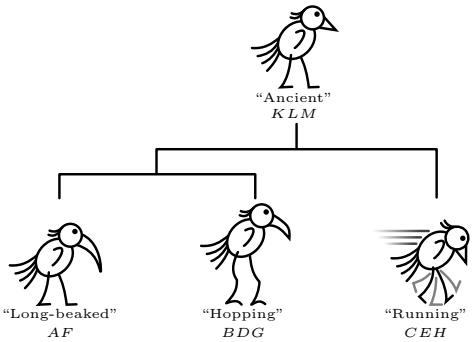


Fig. 1. Paperbirds, hypothetical phylogenetic tree annotated with species names and genes.

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.

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 [3].

There are various computational methods for determining orthology between genes from different species [5,1]. 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 [8], PIRSF [13], and eggNOG [9]. 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 bio-informaticians to utilize.

One of the main problems in homology is to distinguish between orthologs and paralogs. The distinction is beyond the scope of this paper as it does not matter for our technique.

1.2 Combining grouping data

Problem Statement We generalize the use case by viewing it as the problem of integrating data on groupings. We define a *data source* S_i as a database containing elements D_E^i and groups D_G^i where $\forall g \in \mathrm{D}_G^i : g \subseteq \mathrm{D}_E^i$. Each source holds information on different sets of proteins, i.e., the various D_E^i partially overlap. The goal is to construct a new data set with groups over $\bigcup_i \mathrm{D}_E^i$ that allows for scalable querying for questions like 'Which elements are in a group with e?' and 'Are elements e_1 and e_2 in the same group?'.

Approach We focus on an iterative probabilistic integration of the grouping data. It is based on the generic probabilistic data integration approach of [10] which constructs a *probabilistic database*. We call this representation an *uncertain grouping*. Being probabilistic, the above queries return possible answers with their likelihoods. Hence, 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. Although probabilistic data integration is an active research problem [7], there is to our knowledge no work on probabilistic integration of data on groups.

Furthermore, we view integration as an iterative process. Starting from a simple integration view such as 'one-database-source-is-entirely-correct-but-it-is-unclear-which-one', one naturally discovers the limitations of this view while using the resulting data. Subsequently, more fine-grained integration rules are specified which combine the data in a better way, deals with conflicting data in a better way, and specifies better likelihoods for certain portions of the data to be correct (trust assignment). The integration view allows for an automatic re-construction of the integration result. As long as the integration result is not good enough, the process is repeated leading to handling inconsistencies and ambiguities at ever finer levels of granularity.

Outlook 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 an integration view evolves. 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.

1.3 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. [7] offers a comprehensive survey of the relevance of uncertainty management in data integration. Of special note is [6], 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 [12] 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.

[4] 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 Probabilistic integration of grouping data

In this section, we explain our iterative probabilistic integration approach in more detail starting with a running example.

2.1 Running example

Figure 2 presents three example data sources, each containing two or three orthologous groups. 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.

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

Fig. 2. Running example.

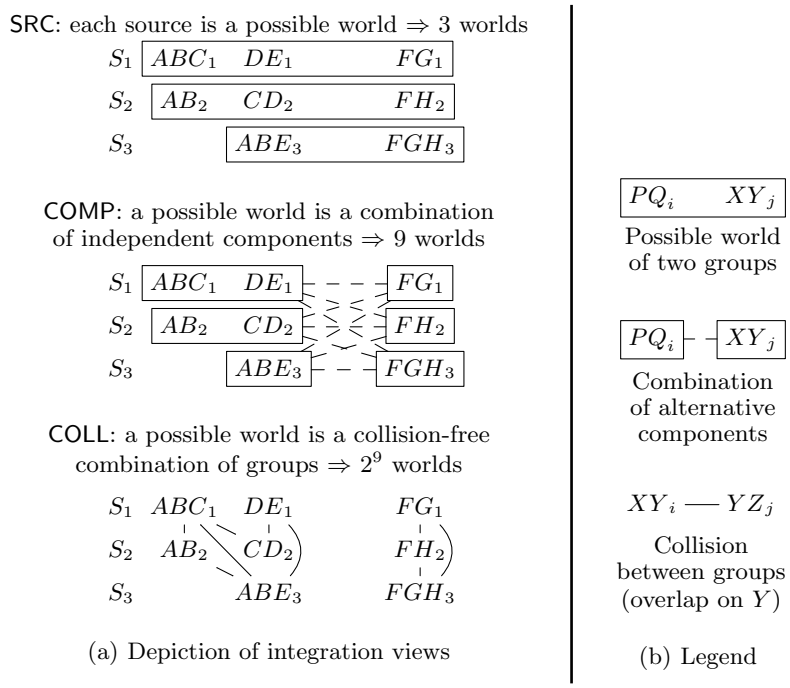
2.2 Integration views

From Section 1.1, 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 bio-informatician integrating these sources, however, does not know what is the correct grouping, not even how well (s)he can trust the data. The goal is to determine based on current scientific knowledge contained in the sources, what the correct grouping is, or rather, the confidence in possible groupings.

We model an uncertain grouping as a probabilistic database 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 [2]. 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.

Although we abstract from what an *integration view* exactly looks like, one can regard it as a set of data integration rules specifying not only how the raw data should be merged, but also which relevant alternatives exist in case of conflicts as well as what confidence to assign to certain portions of the data and such alternatives.

Our method of working with integration 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 bio-informatician gains more understanding in the data,



(a) Depiction of integration views

(b) Legend

Fig. 3. Example of uncertain grouping.

which (s)he uses to adapt and refine the integration view. The reason behind this way of working is, that we believe, as we stated before, that data understanding is a continuous process, with the bio-informaticians understanding of the intricacies of each data source growing over time. With the integration view method, the bio-informatician is able to express and refine his evolving opinion on the reliability of the data in the sources and how the data should be combined. He can 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 integration 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-entirely-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). 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 combining 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). 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 integration view, called COLL. Two groups *collide* iff they overlap but are not equal.[1] Figure 3 shows the collisions between groups in our example. The idea behind the COLL-view 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 integration view is that all *collision-free* combinations of groups form the possible worlds. One can see that the COLL method is more fine-grained by observing that $\{ABE_3, CD_2, FG_1\}$ is a possible world that is not considered by SRC nor COMP. 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).

Typically one would have many more considerations, sometimes rather fine-grained, that one would like to 'add' to one's integration view. For example, a bio-informatician 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 [10] which provides a generic formalization of a probabilistic database. We summarize the main concepts of [10] (Definitions) and show how it can be specialized to support uncertain groupings (Specializations). In Section 2.4 we subsequently show how an uncertain grouping can be constructed for a certain integration 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.[3] *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 $\mathcal{D}$:

[1] This second condition 'not equal' is theoretically not necessary (See Section 2.4).
[2] Actually, this is a simplification as both can be incorrect (see Section 4)
[3] $\mathbb{P}$ denotes a power set.

– *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 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 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 4 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.

$\dot{\mathrm{D}}$

	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

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

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.

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 groups containing A, which results in 3 tuples d_1, d_4, and d_7. Observe that these tuples are mutually exclusive, because their wsds contain different values for r_1.

From the compact representation, one can derive different kinds of answers to the query, such as, the most likely 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 [10] which provides a generic formalization of a probabilistic database. In addition, 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 [10].

$\dot{\mathrm{D}}$

	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 correct' for ABC_1–AB_2
$(r_1 \mapsto 2)$	p_2	'S_2 correct' for ABC_1–AB_2
$(r_2 \mapsto 1)$	p_3	'S_1 correct' for ABC_1–CD_2
$(r_2 \mapsto 2)$	p_4	'S_2 correct' for ABC_1–CD_2
$\vdots$		
$(r_8 \mapsto 1)$	p_{15}	'S_1 correct' for FG_1–FGH_3
$(r_8 \mapsto 2)$	p_{16}	'S_3 correct' for FG_1–FGH_3
$(r_9 \mapsto 1)$	p_{17}	'S_2 correct' for FH_2–FGH_3
$(r_9 \mapsto 2)$	p_{18}	'S_3 correct' for FH_2–FGH_3

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

2.4 Integration views revisited

We argue that integration problems such as conflicts, ambiguity, trust, etc. can all 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 integration view COLL (see Figure 5). Observe how the 9 collisions result in 9 random variables in a straightforward way. Furthermore, 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. The possible answers to a query come with a probability for the trustworthiness of the answer, essentially the combined probability of all worlds that agree on that answer. Note that our modelling of COLL induces empty databases for valuations that would lead to one or more collisions. One could normalize the probabilities of query answers with $1 - \mathrm{P}(\emptyset)$, the combined probability of all collision-free combinations.

Observe also how such an intricate integration 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 integration issues and imposing the associated consideration on the data by refining one's integration 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 can be introduced

and a rva added to the wsd of all tuples of this research group. After such a refinement, the bio-informatician obtains a database that can be directly queried so that he can examine its consequences. He thus iteratively refines his integration view until the data faithfully expresses his opinions as well as the result of any query or analysis run on this data.

3 Evaluation

Two main questions guide the evaluation: can our framework be applied in an existing probabilistic database, and if so, how well does it scale to realistic amounts of data, in particular to determine if current probabilistic database technology can cope with the amounts of uncertainty introduced by our framework. We use the probabilistic database MayBMS [2].

3.1 Experimental Setup

For the evaluation, we constructed a test set of homology data from the biological databases Homologene (release 67, [8]), PIRSF (release 2012_03, [13]), and eggNOG (release 3.0, [9]). The groupings from each were loaded into a single database for the construction of the integration 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.

Two query classes can be distinguished among commonly executed queries:

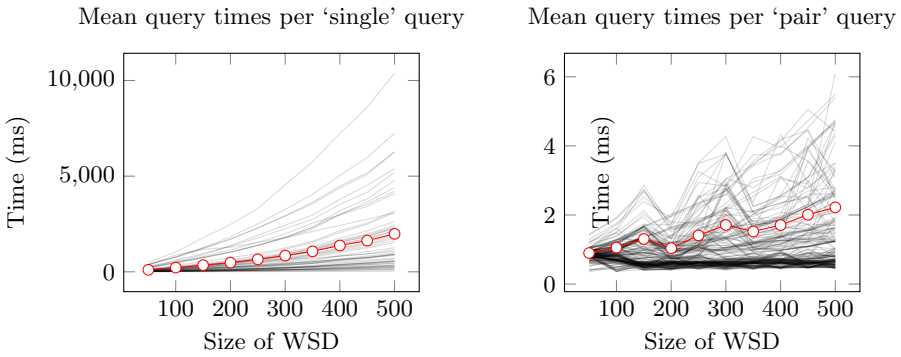
1. **single**: "Which proteins are homologous with X?" with X a known protein.
2. **pair**: "Are X and Y homologues?" with X and Y known proteins.

Based on these two classes we generate query suites based on sampling proteins from the combined database:

1. 1000 single and 1000 pair queries. All pairs are guaranteed to have a homologous relation. This suite is used to determine average query execution times for all integration views.
2. 100 single queries and 200 pair queries. For the latter, 100 queries have a homologous relation and the other 100 do not.

Random variable assignments for the integration views SRC, COMP and COLL were generated according to our integration approach. Probabilities were assigned uniformly over the rvas.

Because of experimenting with an existing system (MayBMS), we accept some technical limitations inherent in these systems. Overcoming these limitations is not the focus of our work and a note on them can be found in [11]. One of the limitations is that the wsd of a tuple can at most contain 500 rvas. Larger wsd were truncated to 500 rvas. Additional integration views based on COLL were generated with wsds of sizes 450, 400, ..., 100, 50. These integration views



(a) 'single' queries.

(b) 'pair' queries.

Fig. 6. Mean query time (in white-red) and distinct query times (in gray)

are referred to as COLLN, with N being the size of the wsd. No size indication means COLL500.

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

Experiment 1: Mean query times Based on query suite 1, each query is executed 10 times. Mean query time per integration view is calculated form the latter 9 measurements; the first is discarded to prevent adverse effects of caching.

SRC mean: 18.627 ms, std.dev.: 26.864
COMP mean: 19.061 ms, std.dev.: 27.569
COLL mean: 23488.197 ms, std.dev.: 93184.375

Preliminary results show that the amount of uncertainty of each integration view has a large impact on the mean execution time. Large standard deviations indicate large variations of query times within each integration view. The following experiments investigate the cause of this variation.

Experiment 2: World Set Descriptor size The goal here is to determine the impact of wsd size on query execution time. Query suite 2 is used on integration views COLL50, COLL100, . . . , COLL500.

Figure 6 presents the trend in mean query time with growing wsd for both query classes seperately. The 'pair' queries are orders of magnitude faster than the 'single' queries due to smaller amounts 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.

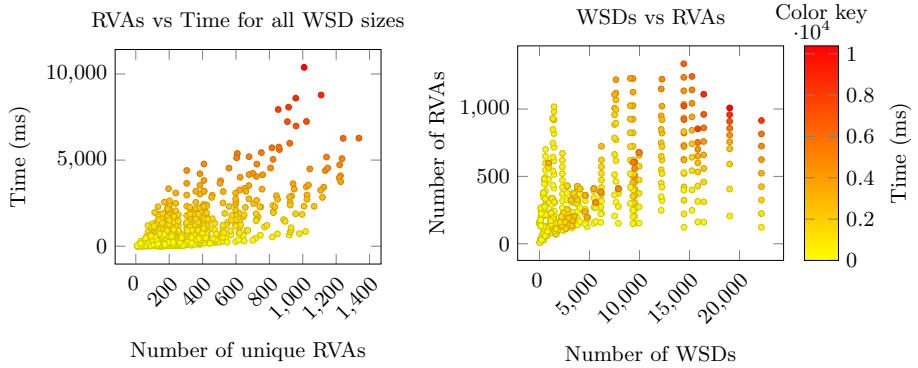


Fig. 7. Impact of number of rvas of wsds involved in query answering ('single').

Experiment 3: Numbers of wsds and rvas The goal here is to investigate the impact of the number of wsds and rvas involved in answering a query on the query time. Query suite 2 is used on integration views COLL50, . . . , COLL500.

As can be seen in Figure 7, the framework and MayBMS handle the real-world uncertainty well. For a large part, queries are executed within 2 seconds. The slower queries are slow due to a combination of a large number of unique rvas and wsds. Based on further analysis of what execution time is spent on for the integration view with a large amount of uncertainty (COLL), we conclude that most time is consumed by confidence computation.

Further remarks The wsd size is used as an artificial bound on the amount of uncertainty. Both SRC and COMP feature only a single rva hence are effectively equivalent wrt execution time. Due to technical limitations, COLL has a maximum of 500 rvas per wsd. This did not hinder the experiments, because we simulated data sets with less uncertainty by truncating the wsds to ever smaller sizes anyway.

We changed the representation of wsds in MayBMS to allow for 500 rvas per wsd, instead of being limited to normal 30. (see Appendix A of [11] for more details). Conversion of representation can be done during integration view construction or querying. Overhead of conversion during querying was shown to impact queries involving large wsds the most but still negligible.

We encountered three measurements that qualify as outliers. Two occurred for 'pair' queries with small execution times. As the experiments were conducted on a normal workstation, we strongly suspect that another program interfered with query execution. One outlier occurred during the measurements of 'single' queries, specifically for protein F6ZHU6 (a UniProt identifier). This protein is related to muscle activity and is a member of an abnormally large number of orthologous groups, the cause of which is further discussed in [11].

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

Complexity from practice. 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 this overlap is due to a subset-superset relation between the two groups.

Open world versus Closed World. 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 integration view methods presented. Hence, the integration views of Section 2.2 all follow a closed world assumption.

The technical report [11] contains a detailed discussion on both these topics, and continues with the topic of confidence precision and alternative representations of the group data.

5 Conclusions

Motivated by the real-world use case of homology, we propose a generic technique for combining groupings. Proteins in a homologous group are expected to have the same function in different species. Homology data is relevant when, e.g., a medicine is being developed and the potential for side-effects has to be determined. We combine 3 different biological databases containing homology data.

In e-science as well as business analytics, data understanding is a continuous process with the analist's understanding of the intricacies and quality of data sources growing over time. We propose a generic probabilistic approach to combining grouping data in which an evolving view on integration can be iteratively queried and refined. Such an 'integration view' models complications such as conflicts, ambiguity, and trust as probabilistic data.

Experiments show that our approach scales with existing probabilistic database technology. The evaluation is based on realistic amounts of data obtained from the combination of 3 biological databases, yielding 776 thousand groups with a total of 14 million members and 2.8 million random variables.

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.

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, 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.
3. E. Koonin. Orthologs, paralogs, and evolutionary genomics. *Annual Review of Genetics*, 39:309–338, 2005.
4. 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.
5. 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.
6. 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.
7. 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.
8. NCBI Resource Coordinators. Database resources of the national center for biotechnology information. *Nucleic Acids Research*, 41(D1):D8–D20, 2013.
9. 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.
10. M. van Keulen. Managing uncertainty: The road towards better data interoperability. *IT - Information Technology*, 54(3):138–146, May 2012.
11. B. Wanders, M. van Keulen, and P. E. van der Vet. Uncertain groupings: probabilistic combination of grouping data. Technical Report TR-CTIT-14-12, Centre for Telematics and Information Technology, University of Twente, Enschede, 2014.
12. J. Widom. Trio: A system for integrated management of data, accuracy, and lineage. Technical Report 2004-40, Stanford InfoLab, August 2004.
13. 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.