Exploring PSI-MI XML Collections Using DescribeX

Reza Samavi, Mariano Consens, Shahan Khatchadourian, Thodoros Topaloglou

Information Engineering Center, Department of Mechanical and Industrial Engineering, University of Toronto, Toronto, Ontario, M5S 3G8, Canada

Summary

PSI-MI has been endorsed by the protein informatics community as a standard XML data exchange format for protein-protein interaction datasets. While many public databases support the standard, there is a degree of heterogeneity in the way the proposed XML schema is interpreted and instantiated by different data providers. Analysis of schema instantiation in large collections of XML data is a challenging task that is unsupported by existing tools.

In this study we use DescribeX, a novel visualization technique of (semi-)structured XML formats, to quantitatively and qualitatively analyze PSI-MI XML collections at the instance level with the goal of gaining insights about schema usage and to study specific questions such as: adequacy of controlled vocabularies, detection of common instance patterns, and evolution of different data collections. Our analysis shows DescribeX enhances understanding the instance-level structure of PSI-MI data sources and is a useful tool for standards designers, software developers, and PSI-MI data providers.

1 Introduction

In the last decade, XML [20] has emerged as a standardized flexible markup language format that is widely used in the bioinformatics research community. Data specified in XML are tree structures properly nested using pairs of markup tags. Data providers (e.g., [9], [11], [14]) use XML as a common format to make bioinformatics databases publicly available on the World Wide Web. While XML provides flexibility for data providers to define their own attributes, it is also responsible for heterogeneity in data from different research groups. As described in the HUPO 2004 perspective report [8], despite being well-documented, the databases produced by data providers were not synchronized with each other and their data formats were incompatible.

The protein informatics community has come together to develop a common data exchange format for protein-protein interaction (PPI) data, with the goal of producing a standard data product and integrating PPI datasets. The proteomics standards initiatives (PSI) has developed an XML-based format for exchanging PPI, called Proteomics Standards Initiative Molecular-Interaction (PSI-MI) [8]. The schema of PSI-MI is simple and is expressed in XML schema language (XSD) [21]. PSI-MI [1] has been endorsed as the de-facto model for making data available by many popular community molecular interaction databases such as BIND [9], BioGrid [10], DIP [11], HPRD [12], IntAct [13], MINT [14], and OPHID [15].

A sustainable standard requires overseeing, monitoring, and understanding of its strengths, weaknesses, and its usage over time. In the case of PSI-MI, the XML schema has been overseen by the HUPO Proteomics Standards Initiative via a series of workshops over the last three years [4], [5], [6]. The workshop reports suggest various improvements such as: evolving and enhancing the use of controlled vocabularies, overseeing the usage of attributes under the *attributeList* element, and defining tools that support streaming in order to read large datasets. Missing in the HUPO-PSI reports is the monitoring of standards compliance by

different data sources which requires understanding how the schema is being used. The schema usage reveals the actual structure of a collection at the instance level, element usage frequency, and general patterns of usage. Understanding schema usage helps the user community answer questions which are not possible by knowledge of the schema alone or by data browsing. For instance, a standards designer would like to know how frequently optional nodes are used for *proteinInteractor;* or what is the most popular substructure of *attributeList* in a collection. These kinds of analyses in large data collections (the size of the smallest collection in PSI-MI format exceeds 40MB) are non trivial, as they require summarization, and are not supported by conventional XML tools.

Quantitative schema usage is more complex than traditional schema analysis tools which validate data documents against the schema or generate XSDs from XML files. Many structural issues and potential improvements of a schema standard, like PSI-MI, are rooted in the actual distribution of data instances. For example, frequent use of an optional element encourages turning it into a mandatory element. More importantly, a specific pattern of data usage can be indicative of the comprehensiveness of a data source, or evidence of data quality factors. Investigation of such issues is more complicated than validating a document against the schema. We need to know, not only if the presence of certain elements follows the directives of the schema, but also what is under an element in terms of instance distribution and sub-element structure. The only way to address these issues is to explore schema usage at the instance level then feed the results back to the schema level.

In this study we report on the application of DescribeX to explore five public community PPI data collections, and gain insights of their usage of the PSI-MI standard. Although there are many ways one can explore these data sources using DescribeX, here we focus on three particular tasks that are specific to PSI-MI and comprehensiveness of the sources. The three tasks are: (a) the degree and variability of optional attributes of the PSI-MI standard that different data sources instantiate, (b) the compliance level of different data sources with the standard's guideline for controlled vocabulary usage, and (c) the frequency that select patterns of attributes appear in a source.

Our work makes two types of contributions. First, we show that the visualization techniques and quantitative schema usage analysis supported by DescribeX can be used to better understand a collection. Second, we generate specific insights into PSI-MI data standard usage by different data providers in the spatial dimension (attribute usage), or, for a given data collection, its evolution over time. This work offers a new approach and tool to the Proteomics Standard Initiative for managing, monitoring, and growing the PSI-MI standard.

2 Background

In this section we provide an overview of the main principles of DescribeX, an overview of the PPI data sources which are explored in Section 3, and expand on the methods and the metrics used for the visual exploration of their PSI-MI collections.

2.1 DescribeX

DTDs and XML Schemas are prescriptive and are used to validate documents for conformance to a given structure. On the other hand, summaries are descriptive in that they show the actual structure of data contained in a document collection. Summaries are usable in a broad class of applications, and can be constructed even when DTDs and XML Schemas are not present.

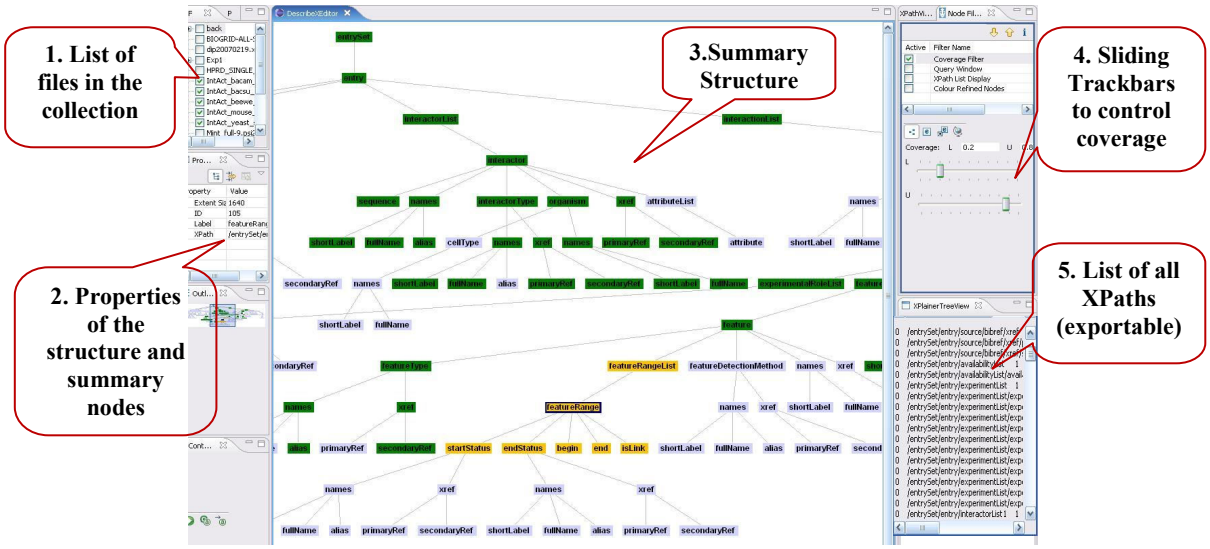

Figure 1: A screenshot of the DescribeX console

DescribeX is a tool for generating, visualizing, and exploring arbitrarily complex heterogeneous structural summaries of large collections of XML documents [22]. XML structural summaries are labeled graphs which use XPath expressions to partition the structural patterns present in XML document instances. While structural summaries were introduced to help understand the structure present in semi-structured data collections [24], DescribeX summaries are a novel technique employing (Axis) Path Regular Expressions (AxPRE) [16] in partitioning the summary extents. In DescribeX, an incoming summary is generated by partitioning elements in the collection having similar incoming paths, i.e. nodes sharing a common XPath expression starting from the root, or in other words, the parent axis relationship. Each summary node is represented by a unique summary identifier and a corresponding *extent*, the set of XPath expressions captured by its AxPRE. A local node refinement repartitions its set of extents based on a different AxPRE expression. For example, a node refinement from a parent to a parent-child AxPRE requires distinguishing XPaths based not only on their parent relationship but their child relationships as well. DescribeX summaries are unique by allowing heterogeneous summary types localized to individual nodes; they have also been shown to be robust in capturing all existing summaries. A refinement example is illustrated in Section 3.3.

Another important feature of DescribeX that is relevant to our analysis is coverage. Coverage allows users to interactively control the visibility of the most popular labels (nodes) in the collection. A coverage setting of 0 allows the display of the elements with the highest popularity. As the coverage value increases, then progressively less popular nodes become visible [22]. Popularity of a node is measured based on the *extent size*. Precisely, popularity is measured based on the extent size of the node relative to the total number of extents present within the summary graph. DescribeX also supports specifying lower bound coverage, so one can freeze certain elements in a particular coverage and then compare the appearance of other elements as the upper bound progresses, and additional kinds of coverage: node size, extent size, logarithm of extent size, and simulated browse. In our experiments, however, we consider only coverage by extent size. DescribeX is in the process of being made open source and is currently available upon request from the University of Toronto[1].

[1] Department of Mechanical and Industrial Engineering, Mariano P. Consens, consens@mie.utoronto.ca

2.2 Data Sources

In our study we explore the PSI-MI Version 2.5 schema instantiation of the PPI datasets listed in Table 1.

Name	Description	Number of Interaction	Collection Size	Ref.
BioGrid	BioGrid is a curated set of physical and genetic interactions.	186302	209 MB	[10]
DIP	Database of Interacting Protein (DIP™) catalogs experimentally determined proteins interactions.	56048	96 MB	[11]
HPRD	Human Protein Reference Database (HPRD) contains literature mined and curated interactions.	37581	62 MB	[12]
IntAct	IntAct is a freely available, open source database system for protein interaction data. Interactions are derived from the literature or direct user submission.	138931	1.39 GB	[13]
Mint	MINT is based on IntAct data model and focuses on experimentally verified protein interactions mined from the scientific literature by expert curators.	103808	1.41 GB	[14]

Table 1: Description of Protein Interaction Data Sources

In this study, we focus on the application of instance level schema analysis for these collections rather than making judgment on entire collection. Therefore, in some cases we used part of the data source for our experiments.

2.3 PSI-MI Exploration

While all of the above sources are PSI-MI compliant, the actual structure of these collections exhibit variations due to factors such as: usage of optional elements, inclusion of extra attributes, and usage of controlled vocabularies (CV). Meanwhile, different groups of users including software developers [21], [2], standard designers, and scientific users need to understand the variation in order to either pose a query against a data source or find a best match amongst data from different sources. Existing XML instance analysis tools such as Altova XMLSpy® or Stylus Studio® either cannot answer the user's questions due to maximum file size limitations, or require an extensive effort in order to express and evaluate basic user questions. For instance, an XPath query may be created from examining an XSD schema but may not retrieve any data because the path does not actually appear in the collection.

Grey et al. [19] advocate *visual exploration* as an approach to gain insights from large scientific datasets, encouraging the use of DescribeX with PSI-MI XML collections to align the needs of the previously-mentioned user groups. DescribeX supports parsing large collections of XML-based datasets and viewing their corresponding AxPRE-based summary structure, all in a matter of a few minutes. For example, creating an incoming summary for a collection of about 100MB size took 2 minutes on a conventional PC (a Pentium 1.8GHZ with 1GB RAM). Furthermore, by interactively changing the coverage which is available in DescribeX, a user can find the popularity of an element in a collection, and in turn, the pattern of usage for any part of the tree. Last but not least, the *refining* capability of DescribeX makes dynamic summary structures available which is highly important in posing XPath queries based on paths that already exist in the collection.

We define two more metrics, instance-oriented *breadth* and *depth* [18] that can be measured by dynamic summaries generated by DescribeX. Breadth of a node is defined as the number of leaf nodes for a particular parent node. Since summary labels in the tree are calculated based on *extent size* (i.e., #instances), we call this instance-oriented breadth. The *instance-oriented depth* is defined as the number of nodes in the longest branch from the root of the

subtree to the leaf. Similarly, because summary labels in the subtree are calculated based on extent size (i.e., #instances), we call this instance-oriented depth [3], [7]. These two metrics help understand the growth of a substructure when the coverage changes in DescribeX.

3 Results and Discussion

Here we discuss the results of the visual exploration and analysis of the PSI-MI collections of Table 1, using DescribeX. Although there are many ways one can explore a data source using DescribeX, we limit our investigation to three particular tasks specific to PSI-MI and comprehensiveness of the sources.

3.1 Optional vs. mandatory elements in the PSI-MI XML standard

In the PSI-MI XML schema, elements are either optional or mandatory. The PSI-MI designers limit the number of mandatory elements to a minimum in order to encourage the data providers to make flexible use of the standard. A complementary community effort, MIMIx [23], proposes minimum information requirements for capturing molecular interaction data in public databases; interestingly, that part of the minimum required information is among the optional elements of the PSI-MI XML standard. Therefore, from a user standpoint it is quite desirable to gain insight into a collection before doing thorough analysis and writing complex queries to extract information that may or may not be present. Validating the schema in such a scenario does not help because the schema is valid even without incorporating any optional elements. To pursue his/her goal a developer may generate the XSD file from the current collection to find out whether an optional element exists in the collection or not but the problem with this approach is that the presence of an element, even in one instance, is enough to be included in the generated schema, preventing achieving a meaningful conclusion. Having said that, even to reach this point requires successfully reading of the collection by a conventional XML viewer. Unfortunately, our experience shows that none of the large size PPI collections (e.g. DIP, HPRD, BioGrid) can be opened by existing XML tools on a conventional PC.

Given one or more PSI-MI XML sources, a user may understand the data collections through two different kinds of analyses. The first is to analyze the collection by itself, in order to explore the presence of certain elements. The second is to compare two or more collections in order to choose the more suitable one. Below are two example questions, motivated by the MIMIx report [23], that we study using DescribeX.

- What is the pattern of use for optional nodes in a specific part of the schema (for instance, the *interactionList* subtree) versus total nodes in the collection? Using a higher number of instantiated optional elements in a collection intuitively correlates with the higher quality of the data source.
- How often are optional elements including *biologicalRole*, *experimentalRole*, and *confidence* used in the subtree under *proteinInteraction*, i.e., path *entrySet/entry/interacti-onList/interaction/participantList/participant,* relative to optional elements used in the subtree under *interaction*, i.e., path *entrySet/entry/interactionList/interaction?*

Without DescribeX, an answer to these questions requires writing and posing a large number of XPath queries and aggregating the results. With DescribeX there is no need to write XPath queries. DescribeX provides visual cues about each collection, creates a dynamic summary graph, and reports extent sizes. For example, in Figure 1, we view the substructure of different parts of the summary graph which changes as we alter the coverage value using a slider. Furthermore, we can visualize differences at two different coverage levels. The effect

of an increase in coverage is the appearance of new, less popular paths (grey nodes), relative to the most popular paths (green nodes) displayed at the previous coverage level.

In order to answer the first question, we use DescribeX to process the structure of the five data collections. For each, we increase the coverage at 10% intervals and measure the following four parameters:

- **#T.Node**: the total number of the nodes (elements) that appear in a coverage interval.
- **#O.Node**: the number of optional elements in a selected subtree (*entrySet/entry /interactionList/interaction/participantList/participant)* in a coverage interval.
- **Breadth**: the instance-oriented breadth of optional nodes for the selected subtree.
- **Depth**: the instance-oriented depth of the optional nodes for the selected subtree.

We then see how paths appear as coverage increases (node popularity decreases), and at each step, we characterize the extent size of a node. These measurements are shown in Table 2.

Coverage(%) / Collections		10	20	30	40	50	60	70	80	90	100	# of instances	Sample Size
BioGrid	# T. Node	6	10	12	12	15	17	19	22	24	53	167571	209 MB
	# O. Node			1	1	1	1	1	1	1	1		
	Breadth			1	1	1	1	1	1	1	1		
	Depth			5	5	5	5	5	5	5	5		
DIP	# T. Node	10	12	14	17	21	23	25	29	36	56	19265	96 MB
	# O. Node												
	Breadth												
	Depth												
HPRD	# T. Node	7	10	12	14	16	18	24	24	29	50	35290	62 MB
	# O. Node												
	Breadth												
	Depth												
IntAct	# T. Node	11	16	20	25	39	49	60	77	92	212	4413	47.5 MB
	# O. Node	-	5	9	14	20	20	20	36	44	72		
	Breadth	-	3	4	7	9	9	9	16	23	41		
	Depth	-	7	7	8	9	9	9	11	11	11		
MINT	# T. Node	8	19	27	40	49	57	68	77	93	168	3339	46.6 MB
	# O. Node		4	12	22	31	39	50	54	54	73		
	Breadth		2	6	10	13	17	25	29	31	43		
	Depth		6	7	9	11	11	11	11	11	11		

Table 2: Number of Total Nodes, and selected Optional Nodes in different collections

Table 2 suggests that some sources have a very uniform and basic schema instantiation (DIP, HPRD, BioGrid) as is visible due to the lack of variability of the four parameters used, while other sources have a rich and extensive instantiation (IntAct, MINT). Below we present a detailed interpretation of Table 2.

Figure 2, illustrates the total number of nodes in each interval. For example, at 80% coverage the most popular paths in MINT and IntAct collections have 77 nodes, while other collections have less than 30 nodes for the same coverage. This can be interpreted that these two collections provide almost twice the information about each protein interaction. On the other hand, the increase of coverage from 90% to 100% drastically changes the number of nodes in IntAct and MINT while it has a lesser effect on the other four collections. The latter indicates that these two collections have many sparsely instantiated elements, i.e., a few interactions have much more information than others. Comparison between IntAct and MINT shows that the total number of nodes at 100% coverage for IntAct is 26% more than MINT (212 vs. 168), while for coverage values from 20% to 80%, the total number of nodes for MINT is always

more than that of IntAct. Based on this sample, we can interpret that MINT provides extensive information for more instances than IntAct.

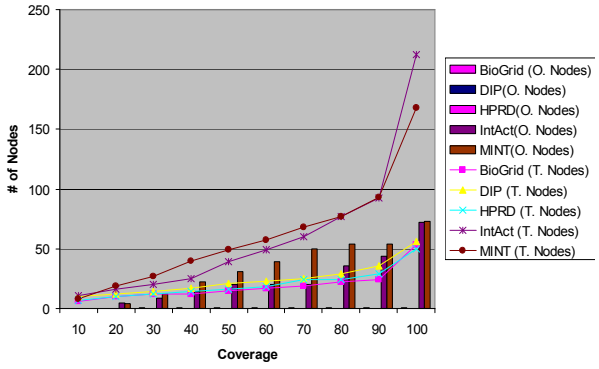

Figure 2: Total number of nodes vs. number of optional nodes for Protein Participants

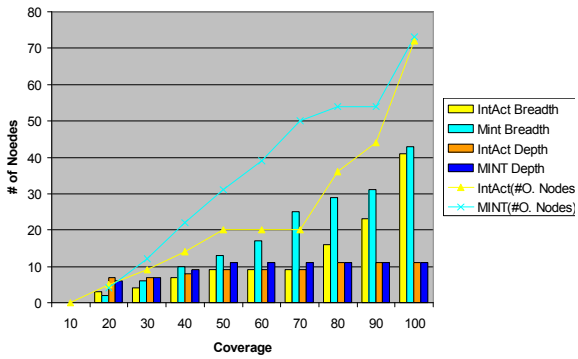

Figure 3: Comparison of Change in Optional Nodes of ProteinParticipants in IntAct and MINT

Figure 2 also displays the number of optional nodes for the *proteinParticipant* subtree compared to the total number of nodes for each collection. The result shows that the IntAct and MINT use optional nodes much more often than the other collections. The number of optional nodes in MINT between 20 and 90% coverage is higher than IntAct, but at 100% coverage, both have the same number. This suggests that the use of optional nodes is generally more popular in MINT than IntAct.

In Figure 3, we see changes in optional nodes for *proteinParticipants* in IntAct and MINT with instance-level breadth and depth. These two metrics help to understand if, for instance, optional nodes are used to provide more nested type information for one element (higher depth) or variety of information for different elements in a select subtree (higher breadth). These two metrics for IntAct and MINT indicate that although the breadth of MINT collection in all coverage intervals is higher than MINT, the depth of the two collections is similar.

Collections \ Coverage(%)		10	20	30	40	50	60	70	80	90	100	# of Instances	File Size
BioGrid	#O.Node(1)			1	1	1	1	1	1	1	1	167571	209 MB
	Breadth(1)			1	1	1	1	1	1	1	1		
	Depth(1)			5	5	5	5	5	5	5	5		
	#O.Node(2)				1	4	4	6	10	11	11		
	Breadth(2)				1	3	3	4	5	6	6		
	Depth(2)				5	5	5	5	5	5	5		
IntAct	#O.Node(1)	-	5	9	14	20	20	20	36	44	72	4413	47.5 MB
	Breadth(1)	-	3	4	7	9	9	9	16	23	41		
	Depth(1)	-	7	7	8	9	9	9	11	11	11		
	#O.Node(2)					3	3	10	11	14	29		
	Breadth(2)					1	1	5	6	8	15		
	Depth(2)					5	5	5	5	5	7		
MINT	#O.Node(1)		4	12	22	31	39	50	54	54	73	3339	46.6 MB
	Breadth(1)		2	6	10	13	17	25	29	31	43		
	Depth(1)		6	7	9	11	11	11	11	11	11		
	#O.Node(2)									7	16		
	Breadth(2)									3	10		
	Depth(2)									5	5		

Table 3: Usage of two different set of Optional Nodes in three collections

In Table 3 we compare the usage of two sets of optional elements in two parts of the schema. O. Nodes(1) shows the number of optional elements for *proteinParticipant* (path *entrySet/entry/interactionList/interaction/participantList/participant*) while O.Nodes(2) shows the number of optional elements for *proteinInteraction* (path *entrySet/entry/interactionList/interaction*). The comparison of use of optional elements in the two subparts of the collection, for IntAct only, suggests that while the usage of the first group of optional elements is highly popular compared to the second group, the pattern of depth and breadth for both sets is similar.

Until now we showed that visual cues and information extracted by DescribeX provides higher level of understanding on schema usage from the instance level. In the next section we continue exploring other features of PSI-MI schema using DescribeX.

3.2 Usage of Controlled Vocabulary and *attributeList* element

PSI-MI makes extensive use of controlled vocabularies (CV), which are viewed as an essential part of encoding molecular interactions in interoperable manner [23]. The CVs used in PSI-MI are not static; they will be maintained and updated by the HUPO PSI workgroup based on the user requirements and new experimental methodologies [4], [5], [6]. Data providers may adopt the new vocabularies at different points in time. This situation implies that, not only could the usage of a schema in one collection be different from other collections, it is also possible that a single provider may offer data sets using different versions, while still complying with the PSI-MI standard. Such cases lead to insufficient understanding of the evolution of a collections structure, and complicate the expression of meaningful XPath patterns.

Although an exhaustive investigation of CV usage in PSI-MI is beyond the scope of this work, here we examine cases that data providers decide to use extra attributes by extending the *attributeList* element. We remind the reader that a common way in PSI-MI to record a source of a CV or reference to an external database is via an *xref* element. Therefore, the extension of the *attributeList* element next to an *xref* element is interpreted as a potential indication of extension for the CV or inadequacy of information provided by an external database. This type of analysis can be of help to the HUPO workgroup that oversees the standard, in order to either introduce more reliable sources of CV for that particular element in the schema or make changes to the required attributes. We use the following example questions to illustrate how DescribeX is applied in understanding CV usage:

- How often is *attributeList* present next to the *xref/primaryRef* in any part of the schema?
- How often is *secondaryRef* used in addition to the *primaryRef* for *proteinInteractor* and *proteinInteraction*? This metric can be interpreted in different ways. For instance it could be the signal of rich presence of different external CVs. As such, the metric can be informative to an analyst of the standard.

For the first question, we process each collection to find at what coverage point an *attributeList* appears in the summary graph. DescribeX's current version provides frequency of each node in the collection (window #2 in Fig.1) as the summary is being created and can be easily extended to include other metrics. For instance, Figure 4 shows a screen shot from DescribeX using the MINT collection. Here *attributeList* appears under path *entrySet/entry/interactorList/Interactor* very early (at 10% coverage). By clicking on *attributeList*, DescribeX displays the extent size for this element which is 3804, very close to the number of *interactor* (3822) elements. This implies that 99% of interactor instances need additional annotation in order to be described. This could be an indication of inadequacy in the external database which is used by the data provider. In the same collection for the *experimentDescription* in *entrySet/entry/interactionList/interaction/experimentList* path, with

extent size 752, *attributeList* appears slightly before 100% coverage with extent size 183. This is a suggestion that in 75% of experiments the referenced CV in *xref* provides enough information. In other word, only in 25% of experiments is the CV insufficient, and for those instances, extra attributes are required.

One important aspect of studying *attributeList* in PPI collections is to understand the frequency of using additional attributes by providers. For example, we may want to know how frequently a particular attribute such as *name* or *size* is used under *attributeList*. The current version of DescribeX creates summaries of elements but does not expand the tree to cover the usage pattern at the *attribute* level. Therefore, we use the extent size of the *attribute* element to estimate the number of required attributes. For instance, the extent size of the *attribute* element in *Interactor* path for MINT is 26186. If we divide this number by the number of *interactor* instances (3822) we realize that on average each instance needs 6.9 extra attributes which are not explicitly present in the current PSI-MI standard. Although the name and individual frequency of these attributes are not identified, the total number is still highly informative (i.e., suggesting an extension of the standard with a group of new attributes).

We evaluate the second question similarly to the first, except this time we focus on the extent size information provided by DescribeX in order to show usage of *primaryRef* and *secondaryRef* in different collections for the elements *.../proteinInteractor/xref/* and *.../proteinInteractor/interactorType/xref/*. Figure 5 shows a DescribeX screenshot of a summary structure for the *proteinInteractor* subtree. Red circles show whenever we have an element *names*, the element *xref* is presented to show the source for the database or CV. The extent size of *primaryRef* and *secondaryRef* compared to the extent size for *interactor* and *interactorType* are shown in Table 4.

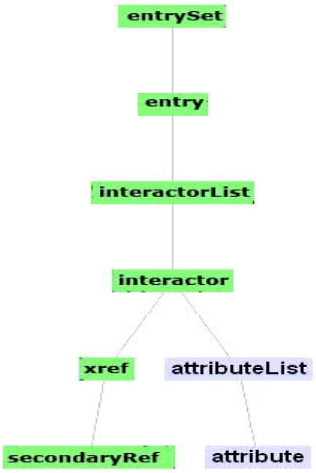

Figure 4: MINT collection at 10% coverage

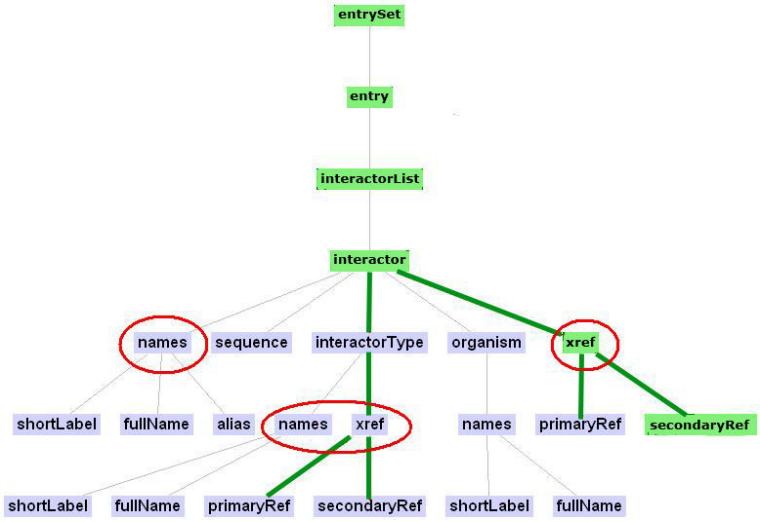

Figure 5: Typical *proteinInteractor* substructure in summary graph produced by DescribeX

The results in Table 4 show that all collections meet the mandatory requirement to provide *primaryRef* for each instance of *name* (100% for all *primaryRef*). However the pattern of usage for *secondaryRef* is quite different. For instance, on average DIP introduces 1.5 *secondoryRef* per *proteinInteractor* name, while BioGrid presents 26.7 and MINT 12.4. This can be interpreted as comprehensiveness in introducing a source of synonyms by BioGrid and MINT compared to the others.

XPaths / Collection:element		# of instances for .../ProteinInteractor	/interactorType/xref/		/xref/		# of instances for PI
			# of Inst.	%	# of inst.	%	
BioGrid	PrimaryRef	23263	-	0	23263	100	1676571
	SecondaryRef		-	0	620509	2667	
DIP	PrimaryRef	19265	19265	100	19265	100	55964
	SecondaryRef		-	0	29644	154	
HPRD	PrimaryRef	9020	9020	100	9020	100	35290
	SecondaryRef		-	0	22893	254	
IntAct	PrimaryRef	4927	4927	100	4927	100	4413
	SecondaryRef		9835	200	43757	888	
MINT	PrimaryRef	3822	3822	100	3822	100	3255
	SecondaryRef		7644	200	47277	1237	

Table 4: Usage of *PrimaryRef* vs. *SecondaryRef* for two different Paths

3.3 Finding Similar Substructures in a Collection

Usefulness of summarization in query discovery is evaluated in a number of studies such as [17], [24]. In this section we illustrate how the explorative nature of DescribeX helps in posing efficient queries by identifying XPaths which are actually present in a collection. For example, in the current PSIMI2.5 XML format the path *entry/entryset/interactionList/ interaction/participantList/participant/featureList/feature* has a subtree with 21 different paths to its leaves. A developer interested in finding similar substructures for node *feature* in a collection without knowing the existence of particular paths at the instance level will have more than $3X10^9$ possible queries due to the possible combinations of child elements.

DescribeX provides a descriptive view of the collection to help the developer find the XPath queries that capture paths currently present in the collection. Figure 6 shows a partial view of the summary graph for the IntAct collection for the element *feature* in DescribeX. Although this graph is quite helpful in understanding the actual structure at the instance level compared to the information captured from schema itself, it cannot provide enough information to reduce the exponential number of possible queries for this node. This happens, because until now the summary graph only represents common parents, while we need to know what is happening below the element, not just above. The local node refinements in this example represent a partitioning of extents from an incoming summary to an incoming and outgoing summary, thus taking into account both the structure from the root to the refined element as well as its substructure. This can also be interpreted using the XML axes relationships by refining the node from a parent relationship to a parent-child relationship.

In Figure 7 we show the same tree where elements: *featureRange*; *startSatatus* and *endStatus*; *xref* and *names*; are iteratively refined. The first iteration, a node refinement of *featureRange*, splits it into five elements, implying existence of five different substructures. In the second iteration, refinement of *startStatus* and *endStatus* splits each into three elements, and the last iteration affecting elements *names* and *xref* splits them into three and two elements, respectively. The last iteration reveals the reason for different substructures: while all 1631 extents of *featureRange* have *shortLabel* for their *startStatus* name, 947 extents among them provide *fullname* for the *startStatus* as well, and 47 extents among them besides *shortLabel* and *fullName* provide *alias* as extra information as well. Therefore, we have three different substructures for element *name* in *.../featureRangeList/featureRange/startStatus/names* (indicated with nodes circled in red in Figure 7). Comparing this path in Figure 6 and 7 shows how refinement helps to explore similar substructures.

The refinement procedure shows that out of millions of possible XPaths, only 36 are present in the collection, drastically reducing the number of XPath queries. Furthermore, the extent

size shown beside each element in Figure 7 helps the developer to focus on the more frequent XPath when required. For example the *featureRange* in the far left side of Figure 7 shows that *featureRange* with the given particular substructure has only one instance in the collection.

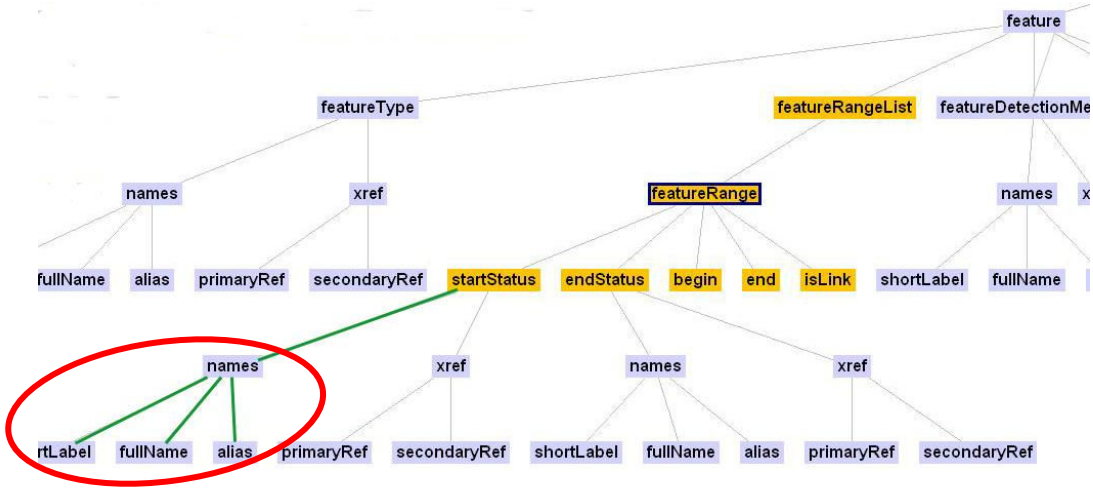

Figure 6: Partial view of DescribeX, showing incoming summary structure for *featureRange*

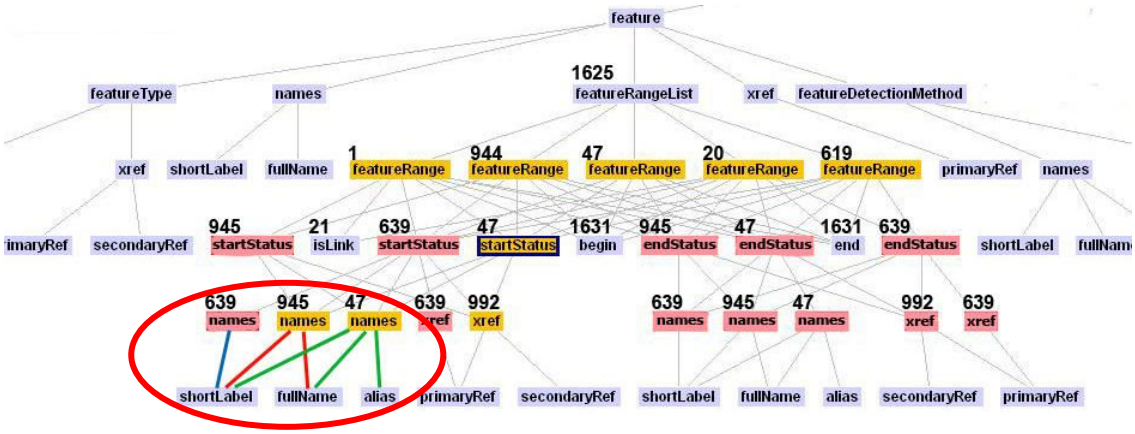

Figure 7: Summary refinement for *featureRange*, *startStatus*, *endStatus*, *names*, and *xref*

4 Conclusion

In this paper, we introduced a new approach to quantitative and qualitative analysis of PSI-MI XML collections using a novel framework called DescribeX. Our motivation and goal was to better understand the schema at an instance level. We showed how DescribeX helps to process and visualize large-scale collections of XML data. We also showed how DescribeX helps to make conclusive analysis of instance data to study specific schema-level issues such as attribute usage, adequacy of a controlled vocabulary and analysis of common substructures. An important difference of DescribeX relative to other schema analysis techniques is the generating of summary graphs from instance data answer important questions such as what structures are present in a collection, at what numbers and in what combinations. We demonstrated how this functionality is useful to standard designers, software developers and data providers of PSI-MI data collections. We have chosen to apply DescribeX on PSI-MI collections because they are important and well-known in the bioinformatics community. As part of future works, DescribeX can be applied to gain insights of other XML based data sources and dialects of XML based standards (such as BSML) that are popular in bioinformatics. The tool providers are extending DescribeX to include attribute level analysis and provide further schema analysis metrics.

Our work makes two types of contributions. First, we introduce new visualization and quantitative schema usage analysis techniques to explore community based XML collections. It this paper we focus on PSI-MI collections, but these techniques will benefit other types of XML collections. Second, we gained specific insights into the PSI-MI data standard usage by different data providers. This work offers a new approach and tool to the Proteomics Standard Initiative for managing, monitoring, and growing the PSI-MI standard.

5 References

[1] PSI-MI XSD [online]. http://psidev.sourceforge.net/mi/rel25/doc/. 12th June 2007.
[2] E G. Cerami, G D. Bader, B E. Gross, and C. Sander. cPath: open source software for collecting, storing, and querying biological pathways. BMC Bioinformatics,7:497, 2006.
[3] J. Visser. Structure metrics for XML Schema. Proceedings of XATA 2006.
[4] S. Orchard, et al. Further steps in standardization, Report of the 2nd annual Proteomics Standard Initiative Spring Workshop. Proteomics 5:3552-3555. 17-20 April 2005.
[5] S. Orchard, et al. Proteomics and Beyond, Report of the 3rd Annual Spring Workshop of the HUPO-PSI. Proteomics 6(16):4439-4443. 21-23 April 2006.
[6] S. Orchard, et al. Entering the Implementation Era, Report on the HUPO-PSI Fall workshop. Proteomics;7(3):337-9. 25-27 September 2006.
[7] R. Lammel, et al. Analysis of XML schema usage. Conference Proceedings XML 2005.
[8] H. Hermjakob et al.The HUPO PSI MI format. Nature Biotechnology,22,177-183, 2004.
[9] C. Alfarano, et al. The Biomolecular Interaction Network Database and related Tools. 2005 update. Nucleic Acids Research, 33(Database issue), D418-424, 2005.
[10] C. Stark, et al., BioGRID: A general repository for interaction datasets. Nucleic Acids Research, 34 (Database issue): D535-9, 2006.
[11] L. Salwinski, et al. The DB of Interacting Proteins. Nucleic Acids Research, 32 (Database issue): D449-451, 2004.
[12] S. Peri, et al. Dev. of human protein reference database as an initial platform for approaching systems biology in humans. Genome Research. 13:2363-2371, 2003.
[13] H. Hermjakob,et al. IntAct: an open source MI DB. Nucleic Acids Research, 32(Database issue): D452-455, 2004.
[14] A. Chatraryamontri, et al. MINT: the Molecular INTeraction database. Nucleic Acids Research, doi: 10.1093/nar/gkl950, 2006.
[15] K.R. Brown,et al. Online Predicted Human Interaction DB. Bioinformatics 21(9):2076-82, 2005.
[16] M. P. Consens, F. Rizzolo, and A. A. Vaisman. Exploring the (Semi-)Structure of XML Web Collections, Technical Report, University of Toronto - DCS, 2007. http://www.cs.toronto.edu/~consens/describex/.
[17] C. Yu, and H. V. Jagadish. Schema summarization. Proceedings of the 32nd International conference on VLDB, pages 319–330, 2006.
[18] G.J. Bex, et al. DTDs versus XML Schema: A Practical Study. WebDB, 79-84, 2004.
[19] J. Gray, et al. Scientific Data Mgmt. in the Coming Decade. SIGMOD Rec. 34(4), 2005.
[20] W3C- The WWW Consortium [online].http://www.w3.org/XML/. 12th June 2007.
[21] R. Aragues, D. Jaeggi, and B. Oliva. PIANA: protein interactions and network analysis. Bioinformatics, 22(8): 1015 – 1017, April 15, 2006.
[22] M. Ali, M. Consens, and F. Rizzolo. Visualizing Structural patterns in web Collections. WWW 2007.
[23] S. Orchard, et al. The Minimum Information required for reporting a Molecular Interaction Experiment (MIMIx). Proteomics 6(16):4439-4443, 2006.
[24] R. Goldman and J. Widom. DataGuides: Enabling Query Formulation and Optimization in Semistructured Databases. VLDB, 1997.