Why Data Citation is a Computational Problem

Susan B. Davidson

University of Pennsylvania

Work partially supported by
NSF IIS 1302212, NSF ACI 1547360
NIH 3-U01-EB-020954-02S1
The power of abstraction
   And how it has helped with two of my favorite problems in bioinformatics

New problem: data citation

Bigger picture: Data Science
The power of abstraction

- The “right” abstraction is key to developing solutions to many practical problems.
  - Data Integration
  - Provenance
  - ....
  - Data Citation

- Developing the right abstraction requires close collaboration between end-users, systems builders, and theoreticians.
Databases meets bioinformatics

"Genomics is the next moon landing." (1992)
The 97-megabase genomic sequence of the nematode Caenorhabditis elegans reveals over 19,000 genes. More than 40 percent of the predicted protein products find significant matches in other organisms. There is a variety of repeated sequences, both local and dispersed. The distinctive distribution of some repeats and highly conserved genes provides evidence for a regional organization of the chromosomes.
Report of the Invitational DOE Workshop on
Genome Informatics, 26–27 April 1993

The Office of Health and Environmental Research of the Department of Energy (OHER/DOE) sponsored a meeting in Baltimore on 26–27 April of a group of experts to assess the state of current OHER-related bioinformatics efforts and to offer advice on planning and coordinating future activities. OHER has a considerable interest in bioinformatics, in large part, because of the DOE Human Genome and the Structural Biology programs.

TOPICS DISCUSSED IN DOE INFORMATICS SUMMIT

26–27 APRIL 1993, BALTIMORE, MD

"Until a fully relationalized sequence database is available, none of the queries in this appendix can be answered.”
Why would they say that?

- Needed to pose set-oriented queries against multiple, heterogeneous databases, files, and software packages.
  - Most integration work at the time was based on the relational model
  - Embedded links in files: Clicking doesn’t scale!

- Needed in-depth understanding of what data sources were available and what information they contained.
We were able to answer the “unanswerable queries” within about a month using our data integration system, Kleisli.

Kleisli used a complex-object model of data, language based on comprehension syntax, and optimizations that went beyond relational systems.

Limsoon Wong
Kyle Hart, Jonathan Crabtree,…
Leonid Libkin, Dan Suciu,…

BioGuideSRS (Cohen-Boulakia)
The Q Query System (Ives)…?
Example 2: Provenance

Public data sources

Bioinformatics protocols

Alignments
- ClustalW
- PAUPS
- Bootstrap
- Phillips
...

How this result has been generated?

Which sequences have been used to produce this result?

Which data are really important to keep?

Biologist’s workspace

Which sequences have been used to produce this result?
Different types of provenance

- “Coarse-grained” workflow provenance
  - Kepler (Ludaescher et al.), Pegasus (Deelman, Gil et al.), Taverna (Goble, Oinn et al.), Vistrails (Freire et al.), ...

- “Fine-grained” database style provenance
  - Why and Where: Buneman, Khanna, Tan
  - Provenance Semirings: Tannen, Green, Karvouranakis
  - Trio: Widom, Cui, Weiner et al

- Event logs (ordering, timing, causality matters)
  - Provenance-Aware Storage Systems: Seltzer et al
  - Secure Network Provenance: Zhou, Loo, Haeberlen
The problem with provenance...
Continuing challenges...

- Combining difference types of provenance
- Tools to query, explore, and understand provenance
- Summarizing provenance
- Approximating provenance
- ...

…
Outline

- The power of abstraction

- **New problem: data citation**
  - State of the art
  - Citations for general queries
  - Building a citation system

- Bigger picture: Data Science
Data Citation
Publication is changing

- Information is increasing published on the web.
- Much of this information is in **curated databases** – crowd- or expert-sourced data.
- These datasets are complex, structured, and evolving, and contributors need to be acknowledged.
Large number of organizations are involved, and standards are emerging: Datacite, DataONE, GEOSS, D-Lib Alliance, DCC, COPDES, Force-11, AGU, ESIP, DCMI, CODATA, ICSTI, IASSIST, ICSU…

- **Force 11**: “Data citations should be accorded the same importance in the scholarly record as citations of other research objects, such as publications.”

- **DataCite**: “We believe that you should cite data in just the same way that you can cite other sources of information, such as articles and books.”

- **Amsterdam Manifesto**: “Data should be considered citable products of research.”
Our manifesto...

- Principles and standards for data citation are unlikely to be used unless the process of extracting information is coupled with that of providing a citation for it.

- We need to **automatically generate citations** as the data is extracted.

- Data citation is a **computational problem**.

Buneman, Davidson, Frew: Why data citation is a computational problem. *Commun. ACM 59(9): 50-57 (2016)*
What is a (conventional) citation?

- A collection of “snippets” of information: authors, title, date, etc. and some kind of access mechanism
- Not exactly provenance
- Self contained, immutable (to within some choice of format)
- Needed for a variety of reasons: kudos, currency, authority, recognition, access…

Buneman, Davidson, Frew: Why data citation is a computational problem, Commun. ACM, 59(9): 50—57 (2016)
Citation goes beyond DOIs


  **Einstein:**
  Does the inertia of a body depend on its energy content?

- *Nature, 171,* 737-738

  **Watson and Crick:**
  Molecular Structure of Nucleic Acids;
  a structure for deoxyribose nucleic acid
State of the art in data citation ...
Example 1: eagle-i

- A “resource discovery” tool built to facilitate translational science research.

- Developed by a consortium of universities under NIH funding, headed by Harvard.
  - Penn is a member.

- End users: researchers who wish to share information about research resources (Core Facilities, iPS cell lines, software resources).

- Data is stored and distributed as RDF files (graph database).

- Resources have a “Cite this resource” button!
# Significance Tester for the Accumulation of Reads

**Algorithmic software component**

<table>
<thead>
<tr>
<th>Software Description</th>
<th>STAR was developed to identify regions enriched for a histone modification based on ChIP-Seq evidence, by identifying regions with a significant accumulation of reads.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software Additional Name</td>
<td>STAR</td>
</tr>
<tr>
<td>Used by</td>
<td><a href="#">Computational Biology and Informatics Laboratory</a></td>
</tr>
<tr>
<td>Contact</td>
<td><a href="#">Grant, Gregory R., Ph.D.</a></td>
</tr>
<tr>
<td>Related Technique</td>
<td>ChIP-seq assay</td>
</tr>
<tr>
<td>Software purpose</td>
<td>DNA modification site prediction objective</td>
</tr>
</tbody>
</table>
Significance Tester for the Accumulation of Reads

Algorithmic software component

eagle-i ID for this resource:
http://eagle-i.itmat.upenn.edu/i/0000013d-8d96-57e1-2ed7-105480000000

Click here for citation examples and more information.

Contact
Grant, Gregory K., Ph.D.

Related Technique
ChiP-seq assay
Citing an eagle-i Resource

Citing eagle-i resources is an easy way to give credit.

The formats suggested below provide the minimum information necessary to identify and credit the resource provider, and are designed to provide a traceable, durable, and unambiguous reference for the resource being cited. These suggestions can and should be used along with those from other resource identifiers (i.e. Antibody Registry ID, Addgene, DSHB, RRID) or from the journal publishing your work.

Note that for all types, the names of Core Facilities or other ambiguously named organizations should be followed by the name of the affiliated eagle-i institution in order to disambiguate them (e.g. Flow Cytometry Core, Montana State University vs. Flow Cytometry Core, Dartmouth College).

Citation Guidelines

Although only the most commonly cited types are listed below, the same rules can be used to cite any eagle-i resource.
Automating citation in eagle-i

IUPHAR Guide to Pharmacology is a database of information about drug targets, and the prescription medicines and experimental drugs that act on them.

Information is presented to users through a hierarchy of web views, with an underlying relational implementation.

Targets are arranged into groups called “families.”

Contents of the database are generated by hundreds of experts who, in small groups, contribute to portions of the database. Thus the authorship depends on what part of the database is being cited.
Citations in IUPHAR

- Citation to the IUPHAR database as a whole (the root) is a traditional paper written by the main curators (owners) of the database.

- Each IUPHAR Family and Family Introduction page has an independent citation.
  - Information about a Family is managed by a set of curators, which may be different for each family.
  - The detailed Family Introduction page is written by a set of contributors, which may be different from the curators of the Family.
To cite this family introduction, please use:

Miller, Drucker, Bataille, Chan, Delagrange, Göke, Mayo, Thorens, Hills.
Glucagon receptor family.
Accessed on 08/05/2017.
Why not just hard code citations?

- Citations vary with what part of the database is being cited.
  - There are a very large number of “parts” of a database.

- In the future, IUPHAR would like to enable general queries
  - Queries may combine “parts” in different ways.

- We cannot expect to create a citation for each possible query result.

- Citations should be lifted up to schema-level specifications so they can be reasoned about.
Given a database $D$ and a query $Q$, generate an appropriate citation.

Citations for general queries
The citation generation problem

- It is common for database owners to supply citations for some parts (views) of the database, $V_1 \ldots V_n$. 

- So the problem becomes: Given a query $Q$, can it be rewritten using the views? That is, is there a $Q_i$ such that

  $$\forall D \in S. \ Q(D) = Q_i(V_{i_1}(D), \ldots, V_{i_k}(D))$$

- If so, the citations for $V_{i_1}, \ldots, V_{i_k}$ could be used to create a citation for $Q$. 

The problem of answering queries using views has been well studied and is generally hard – but in our context may be tractable.

“Parameterized” views

- In IUPHAR there are views for Family and Family Introduction pages, parameterized by FID, and views for Target pages, parameterized by FID, TID

“Binding adornments”:
Rajaraman, Sagiv, Ullman:
Answering Queries Using Templates With Binding Patterns. PODS 1995.

Also used in the context of Access Control:
Rizvi, Mendelzon, Sudarshan, Roy:
Extending Query Rewriting Techniques for Fine-Grained Access Control. SIGMOD 2004
Effect of parameters

Parameterized views define a family of views, one for each value of the parameter.

<table>
<thead>
<tr>
<th>FID</th>
<th>FName</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Glucagen receptor...</td>
<td>GPCR</td>
</tr>
<tr>
<td>2</td>
<td>CLR (calcitonin receptor-like receptor) ...</td>
<td>GPCR</td>
</tr>
<tr>
<td>3</td>
<td>Peptidases and proteinases...</td>
<td>Kinase</td>
</tr>
<tr>
<td>4</td>
<td>A multifunctional molecule, adenosine...</td>
<td>Kinase</td>
</tr>
<tr>
<td>5</td>
<td>Chromatin modifying enzymes...</td>
<td>Kinase</td>
</tr>
</tbody>
</table>

\[ \lambda F. V1(F, N, Ty) :- Family(F, N, Ty) \]

\[ V4(F, N, Ty) :- Family(F, N, Ty) \]

“Instantiated views”: V1(F, N, Ty)(1), V1(F, N, Ty)(2), ..., V1(F, N, Ty)(5)
To specify a citation, there are three components:

- **View definition**: specifies what is being cited
- **Citation query**: specifies what snippets of information to include
- **Citation function**: specifies how to construct the snippets of information

We call this triple a **citation view**.

For now, we will focus on the **view definition**, which is expressed in Datalog.

“Universal” across different types of databases (e.g. relational, XML, RDF...)

Simplifies reasoning over queries and views
IUPHAR: Citation views

Schema:
- Family(FID, FName, Type)
- FamilyIntro(FID, Text)
- Person(PID, PName, Affiliation)
- FC(FID, PID)
- FIC(FID, PID)

View definitions:
- $\lambda F. \text{V1}(F, N, Ty) :- \text{Family}(F, N, Ty)$
- $\lambda F. \text{V2}(F, Tx) :- \text{FamilyIntro}(F, Tx)$

Citation queries:
- $\lambda F. \text{C}_1(F, PN) :- \text{Family}(F, N, Ty), \text{FC}(F, P), \text{Person}(P, PN)$
- $\lambda F. \text{C}_2(F, PN) :- \text{FamilyIntro}(F, Tx), \text{FIC}(F, P), \text{Person}(P, PN)$
Generating citations

- If the query matches a view definition, we can use the associated citation query and function.
- But what if it doesn’t?
  - Nothing matches the query
  - A set of view definitions are used to rewrite the query
  - More than one set of view definitions can be used to rewrite the query
What is a “good” citation?

- Contains appropriate snippets of information
  - E.g. as suggested by DataCite Schema
- Allows the data as it appeared at time of citation to be retrieved
  - Query and timestamp

- Concise
- Specific

◆ Our approach enables the DBA to specify the tradeoff between conciseness and specificity.
A query is another Datalog expression (unparameterized).

\[ Q_1(F, N, Ty) :\neg \text{Family}(F, N, Ty), F = 1 \]

This can be rewritten using \( V1 \)

\[ Q'_1(F, N, Ty) :\neg V1(F, N, Ty)(1) \]

We can then construct a citation to \( Q \) in terms of the citation for \( V1(F, N, Ty) \)(“1”).
Consider another input query

$Q_2(F, N, Y) :\text{\texttt{- Family}(F, N, Ty)}$

This can be rewritten using $V1$

$Q_2'(F, N, Y) :\text{\texttt{- V1}(F, N, Ty)}$

Now we must use all instantiations of $V1$ to construct a citation to $Q$

$V1(F, N, Ty)(1), V1(F, N, Ty)(2), \ldots, V1(F, N, Ty)(5)$
Consider the following query, with another view V4

\[ Q_2(F, N, Y) :\text{Family}(F, N, Ty), \text{Ty}=\text{“GPCR”} \]

This can be rewritten using V1 or V4 (alternate use)

\[ Q_2'(F, N, Ty) : V1(F, N, Ty), \text{Ty}=\text{“GPCR”} \]
\[ Q_2''(F, N, Ty) : V4(F, N, Ty), \text{Ty}=\text{“GPCR”} \]

We can then construct a citation to Q in terms of the citations V1(F, N, Ty)(1), V1(F, N, Ty)(2) or V4(F,N,Ty)
Another query:

$$Q_1(F, N, Ty, Tx) :- \text{Family}(F, N, Ty), \text{FamilyIntro}(F, Tx), F = 1$$

This can be rewritten using V1 and V2 (joint use)

$$Q'_1(F, N, Ty, Tx) :- V1(F, N, Ty)(1), V2(F, Tx)(1)$$

We can then construct a citation to Q in terms of the citations for $V1(F, N, Ty)(1)$ and $V2(F, Tx)(1)$. 

**Schema:**

- Family(FID, FName, Type)
- FamilyIntro(FID, Text)

**View definitions:**

- $$\lambda F. V1(F, N, Ty) :- \text{Family}(F, N, Ty)$$
- $$\lambda F. V2(F, Tx) :- \text{FamilyIntro}(F, Tx)$$
Citation views are a type of annotation on tuples.

Provenance is a form of annotation on tuples, which is well understood while being carried through queries.

- Green, Karvounarakis, Tannen: Provenance Semirings, PODS 2007:

  - Joint use: joins of tuples
  - Alternate use: unions and projections of tuples

Can we use these ideas to understand how citation “annotations” on tuples are combined in general queries?
Given a (conjunctive) query, we rewrite it to a set of minimal equivalent queries that contain at least one citation view.

Let the set of queries obtained in this way be \( \{Q_1, ..., Q_n\} \)

Each \( Q_i \) contains a set of citation views \( \{V_{i1}, ..., V_{imi}\} \). The **joint** use (*) of their citations constructs a citation for \( Q_i \), \( C(Q_i) \).

\[
C(Q_i) = C(V_{i1}) \ast ... \ast C(V_{imi})
\]

The **alternate** use (+) of each \( C(Q_i) \) constructs a citation for \( Q \), \( C(Q) \).

\[
C(Q) = C(Q_1) + ... + C(Q_n)
\]
Interpreting * and +

- **Joint** use of citations: \( C(V_{i1}) \times \ldots \times C(V_{imi}) \)
  - * could be union or some sort of join
  - E.g. in example 4, \( V1 \) and \( V2 \) were jointly used: \( V1(F,N,Ty)("F123") \times V2(F,Tx)("F123") \)

- **Alternate** use of citations: \( C(Q_1) + \ldots + C(Q_n) \)
  - + could be union or min (wrt some ordering on views)
  - E.g. in example 3, both the parameterized and unparameterized views on Family matched \( (V1(F,N,Ty)(1), V1(F,N,Ty)(2)) + V4 \)

- **Joint and alternate use are “policies” specified by the DBA**
Example of output citation

**View definition:**
\[ \lambda F. \ V_1(F, N, Ty) \ :- \ Family(F, N, Ty) \]

**Citation query:**
\[ \lambda F. \ C_{V_1}(F, PN) \ :- \ Family(F, N, Ty), \ FC(F, P), \ Person(P, PN) \]

**Q_1(F, N, Ty) :- Family(F, N, Ty), F = 1**

**Q_1'(F, N, Ty) :- V_1(F, N, Ty)(1)**

<table>
<thead>
<tr>
<th>FID</th>
<th>FName</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Glucagen...</td>
<td>GPCR</td>
</tr>
</tbody>
</table>

**Citation:**
Miller, Drucker, Bataille, Chan, Delagrange, Göke, Mayo, Thorens, Hills. Glucagon receptor family. Accessed on 08/05/2017. IUPHAR/BPS Guide to PHARMACOLOGY, Family(F, N, Ty), F = 1
Example, with * as “join”

View definitions:

\[ \lambda F. V1(F, N, Ty) : - Family(F, N, Ty) \]
\[ \lambda F. V2(F, Tx) : - FamilyIntro(F, Tx) \]

Citation queries:

\[ \lambda F. C_{V1}(F, PN) : - Family(F, N, Ty), FC(F, P), Person(P, PN) \]
\[ \lambda F. C_{V2}(F, PN) : - FamilyIntro(F, Tx), FIC(F, P), Person(P, PN) \]

FID | FName       | Type | Text                  |
--- | ----------- |------|-----------------------|
1   | Glucagen... | GPCR | Glucagon regulates... |

\[ Q_1(F, N, Ty, Tx) : - Family(F, N, Ty), FamilyIntro(F, Tx), F= 1 \]

\[ Q_1'(F, N, Ty, Tx) : - V1(F, N, Ty)(1), V2(F, Tx)(1) \]

Citation:

Miller, Drucker, Bataille, Chan, Delagrange, Göke, Mayo, Thorens, Hills.
Glucagon receptor family, introduction.

Miller, Drucker, Bataille, Chan, Delagrange, Göke, Mayo, Thorens, Hills.
Glucagon receptor family.

Accessed on 08/05/2017.
IUPHAR/BPS Guide to PHARMACOLOGY, Family(F, N, Ty), FamilyIntro(F, Tx), F= 1
Reaction of the DBA...
“Partitioning” views

- In current practice, citation views are simple
  - Project-select views of a single relation

- It is easily shown that if the views “partition” a relation then there is a single maximal rewriting using the views.

- And the implementation is much simpler...
Building a citation system
Database owners need to be able to specify citation views for the database – schema level information.

Database users ("authors") need to have citations "served up" as they extract data through queries.

Dereferencing the citation should bring back the data to "readers" as of the time it was cited.

Alawini, Davidson, Hu, Wu: Automating Data Citation in CiteDB. VLDB 2017 (Demo paper).
Citation architecture

Versioning system

Curated DB

Citation Views

Policies

Query Rewriting

Citation Generator

Query

Citation Dereferencing

DBA

define

define

Cited data

Author

Data (result set)

citation queries
citation snippets
views used for rewriting query

applicable policies

Reader

citation
dereferencing
Computational challenges

- Schema-level versus instance level?
  - Should we store the citations as annotations on tuples, or should we reason at the schema level and then calculate the citation?

- Given an expected query workload, what are the “best” citation views?
  - And are the necessary snippets of citation information in the schema?

- The number of alternative uses of citation views can be large.
  - Are there efficient algorithms to find the “best” according to some metric of quality (e.g. involving the number of views, the specificity of views, or related to a view hierarchy)?
Take home message

- If we want people to cite the data they use, we need to make it easy for them to do so.

- We must also make it easy for people who publish data to specify how their data should be cited.

- For many applications, there is a notion of "parameterized views" to which citations can be attached.

- Joint and alternate use semantics are "policies" to be specified by the DBA.
Outline

- The power of abstraction
- New problem: data citation
- Bigger picture: Data Science
The Tsunami of Data Science...
Role of computing research in DS

- Report by CRA’s Committee on Data Science
  - Lise Getoor (Chair), David Culler, Eric de Sturler, David Ebert, Mike Franklin, and H. V. Jagadish

- Topics:
  - Models for data representation, acquisition, storage and access.
  - Large scale system and algorithms.
  - Learning with biased, incomplete and heterogeneous data.
  - User interaction: with data and models.
  - Ethical use: privacy, fairness, transparency
My personal perspective...

**Data Science** =
(CS+Stat+Math) ∩ (Science | Economics | Sociology | Business | Law | …)

- (Data Management) ∩ (Machine Learning)
- “Data Engineering” akin to “Software Engineering”
  - Collecting, cleaning and organizing data sets is reported to take nearly 80% of a data scientist’s time yet is the least enjoyable part of their job
- “Why Analysis” of Algorithms
- Ethical data management
Thanks to my collaborators
And to our funders...

NSF IIS 1302212,
NSF ACI 1547360,
...

NIH 3-U01-EB-020954-02S1
Questions?