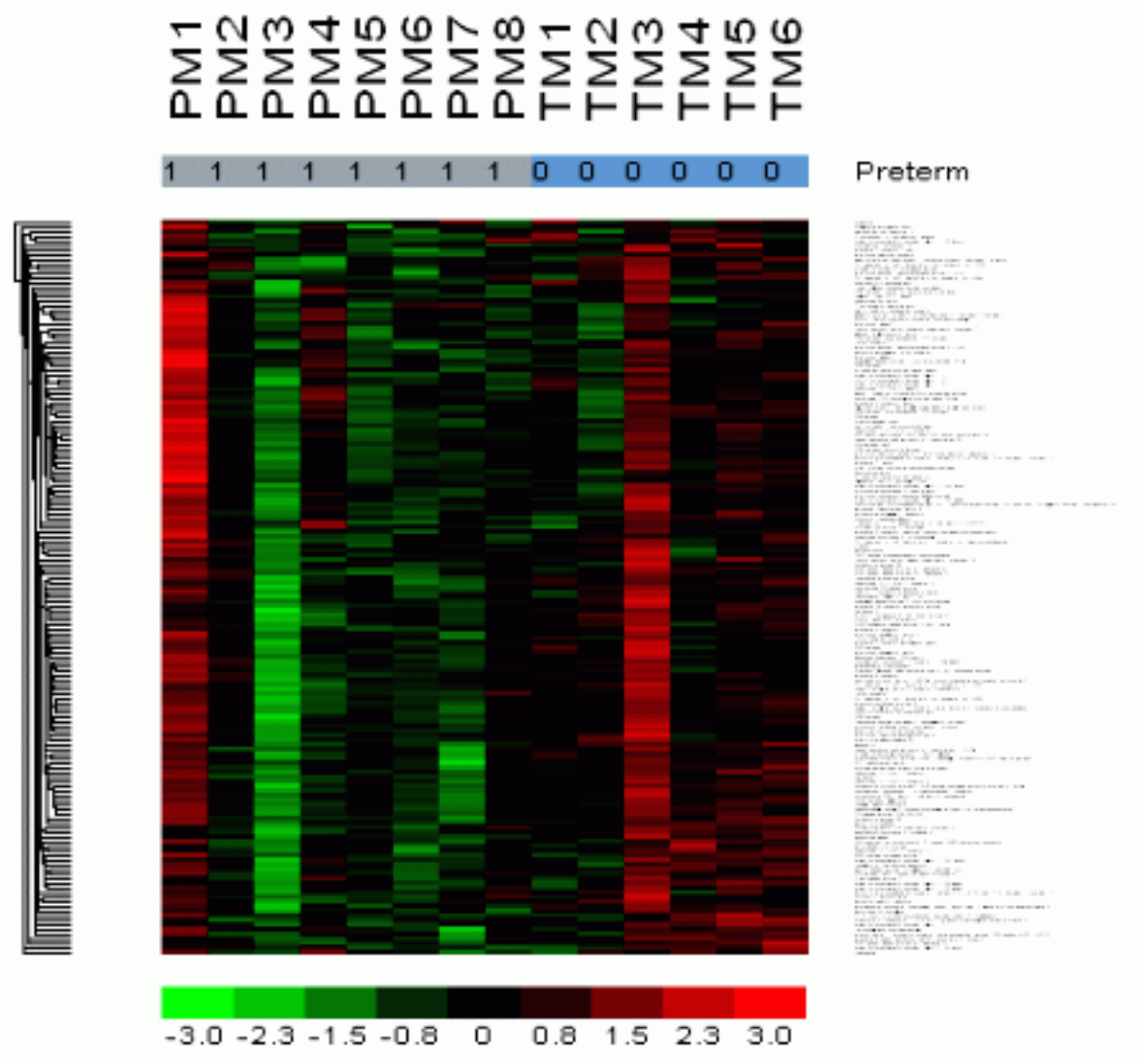
Introduction to the Semantic Web for Bioinfomatics

Ken Baclawski
Northeastern University

The Problems

- The dramatic increase of bioinformatics data available in web-based systems and databases calls for novel processing methods.
- The high degree of complexity and heterogeneity of bioinformatics data and analysis requires integration methods.
- Information must be processed by a sequence of tools that often use different formats and data semantics.



Example of a complex data format

Flat File Records

Consider the following records in a flat file:

011500 18.66 0 0 62 46.271020111 25.220010
011500 26.93 0 1 63 68.951521001 32.651010
020100 33.95 1 0 65 92.532041101 18.930110
020100 17.38 0 0 67 50.351111100 42.160001

What do they mean?

Metadata

- The explanation of what data means is called metadata or “data about data.”
- For a flat file or database the metadata is called the *schema*.

```
NAME     LENGTH FORMAT            LABEL
instudy 6      MMDDYY      Date of randomization into
study
bmi      8      Num         Body Mass Index.
obesity 3      0=No 1=Yes Obesity (30.0 <= BMI)
ovrwt    8      0=No 1=Yes Overweight (25 <= BMI < 30)
Height   3      Num         Height (inches)
Wtkgs    8      Num         Weight (kilograms)
Weight   3      Num         Weight (pounds)
```

XML Data is Self-Describing

<Interview RandomizationDate="2000-01-15" BMI="18.66" Height="62"... />
<Interview RandomizationDate="2000-01-15" BMI="26.93" Height="63"... />
<Interview RandomizationDate="2000-02-01" BMI="33.95" Height="65"... />
<Interview RandomizationDate="2000-02-01" BMI="17.38" Height="67"... />

```
<ATTLIST Interview
      RandomizationDate  CDATA #REQUIRED
      BMI                CDATA #IMPLIED
      Height             CDATA #REQUIRED
>
```

Attribute Types

- Attributes generally contain a specific kind of data such as numbers, dates and codes.
- XML does not include any capability for specifying kinds of data like these.
- XML Schema (XSD) allows one to specify data structures and data types.
- The syntax for XSD differs from that for DTDs, but it is easy to convert from DTD to XSD using the dtd2xsd.pl Perl script.

XSD Basic Types

string Arbitrary text without embedded elements.

decimal A decimal number of any length and precision.

integer An integer of any length. This is a special case of decimal. There are many special cases of integer, such as positiveInteger and nonNegativeInteger.

date A Gregorian calendar date.

time An instant of time during the day, for example, 10:00.

dateTime A date and a time instance during that date.

duration A duration of time.

gYear A Gregorian year.

gYearMonth A Gregorian year and month in that year.

boolean Either true or false.

anyURI A web resource.

Element Hierarchy

- XML elements can contain other elements.
- An XML document is a hierarchy of elements.
- But what does the hierarchy mean?

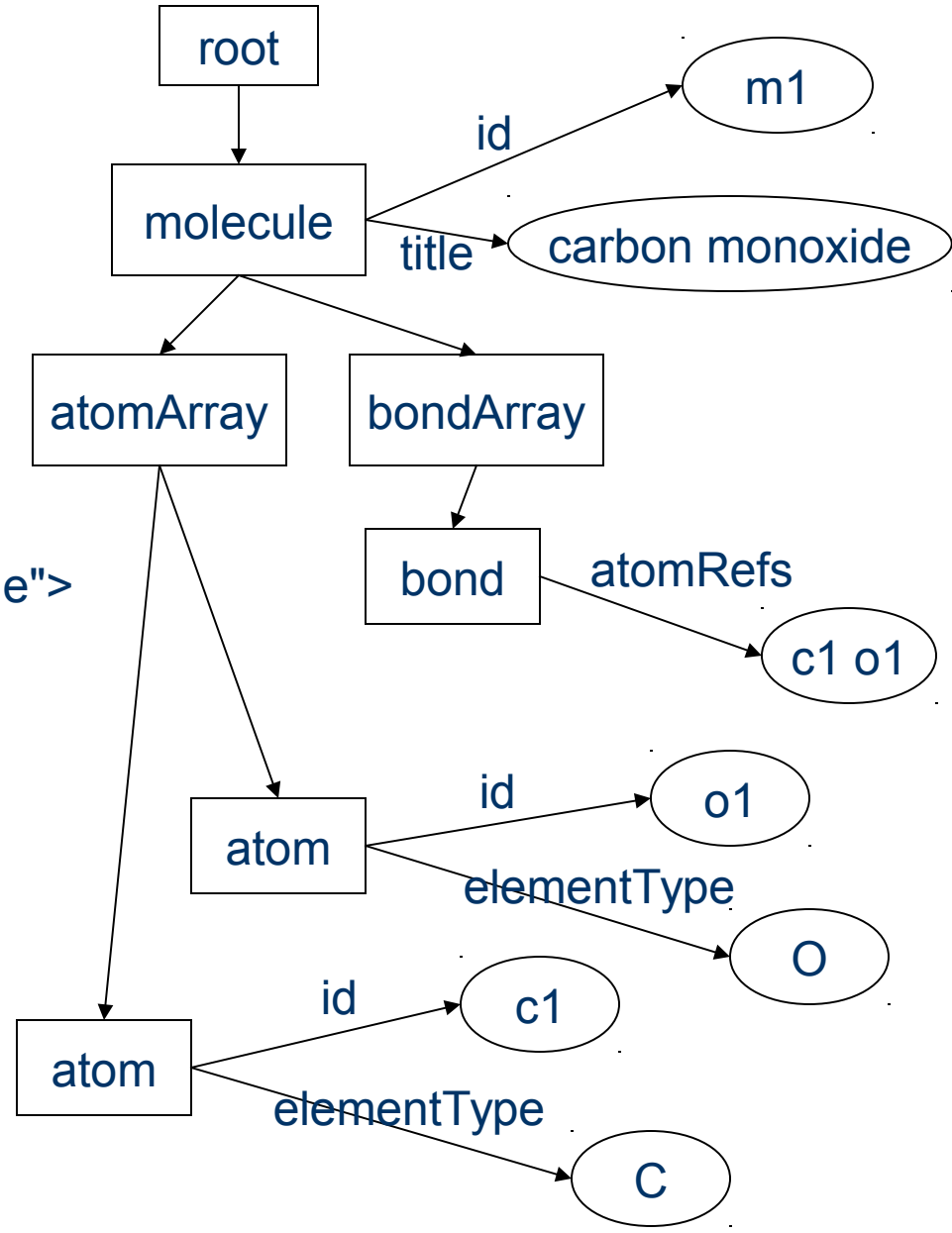
```
<bioml>
  <organism name="Homo sapiens (human)">
    <chromosome name="Chromosome 11" number="11">
      <locus name="HUMINS locus">
        <reference name="Sequence databases">
          <db_entry name="Genbank sequence" entry="v00565" format="GENBANK"/>
          <db_entry name="EMBL sequence" format="EMBL" entry="V00565"/>
        </reference>
        <gene name="Insulin gene">
          <dna name="Complete HUMINS sequence" start="1" end="4992">
          1 ctcgaggggc ctagacattg ccctccagag agagcaccca acaccctcca ggcttgaccg
          ...
          </dna>
          <ddomain name="flanking domain" start="1" end="2185"/>
          <ddomain name="polymorphic domain" start="1340" end="1823"/>
          <ddomain name="Signal peptide" start="2424" end="2495"/>
          <exon name="Exon 1" start="2186" end="2227"/>
          <intron name="Intron 1" start="2228" end="2406"/>
        </gene>
      </locus>
      <locus>
      ...
      </locus>
    </chromosome>
  </organism>
</bioml>
```

Formal Semantics

- Semantics is primarily concerned with *sameness*. It determines that two entities are the same in spite of appearing to be different.
- Number semantics: 5.1, 5.10 and 05.1 are all the same number.
- DNA sequence semantics: cctggacct is the same as CCTGGACCT.
- XML document semantics is defined by infosets.

XML infoset for carbon monoxide

```
<molecule id="m1" title="carbon monoxide">
  <atomArray>
    <atom id="c1" elementType="C"/>
    <atom id="o1" elementType="O"/>
  </atomArray>
  <bondArray>
     <bond atomRefs="c1 o1"/>
  </bondArray>
</molecule>
```



The Resource Description Framework

- RDF is a language for representing information about resources in the web.
- While RDF is expressed in XML, it has different semantics.
- RDF decouples information from the document where it is asserted. This has many advantages for data integration and interoperability.

RDF Semantics

- All relationships are explicit and labeled with a property resource.
- The distinction in XML between attribute and containment is dropped, but the containment relationship must be labeled on a separate level. This is called *striping.*

```
...
    <locus name="HUMINS locus">
      <contains>
        <gene name="Insulin gene">
          <isStoredIn>
            <db_entry name="Genbank sequence" entry="v00565"
                      format="GENBANK"/>
            <db_entry name="EMBL sequence" format="EMBL"
                      entry="V00565"/>
          </isStoredIn>
          <isCitedBy>
            <db_entry name="Insulin gene sequence" format="MEDLINE"
                      entry="80120725"/>
            <db_entry name="Insulin mRNA sequence" format="MEDLINE"
                      entry="80236313"/>
            <db_entry name="Localization to Chromosome 11" format="MEDLINE"
                      entry="93364428"/>
          </isCitedBy>
          <hasSequence>
            <dna name="Complete HUMINS sequence" start="1" end="4992">
              1 ctcgaggggc ctagacattg ccctccagag agagcaccca acaccctcca ggcttgaccg
              ...
            </dna>
          </hasSequence>
        </gene>
      </contains>
    </locus>
...
```

XSD vs. RDF

- XML semantics is based on infosets
- Meaning of hierarchy is implicit
- Support for data structures and types
- Data is contextual: element and document

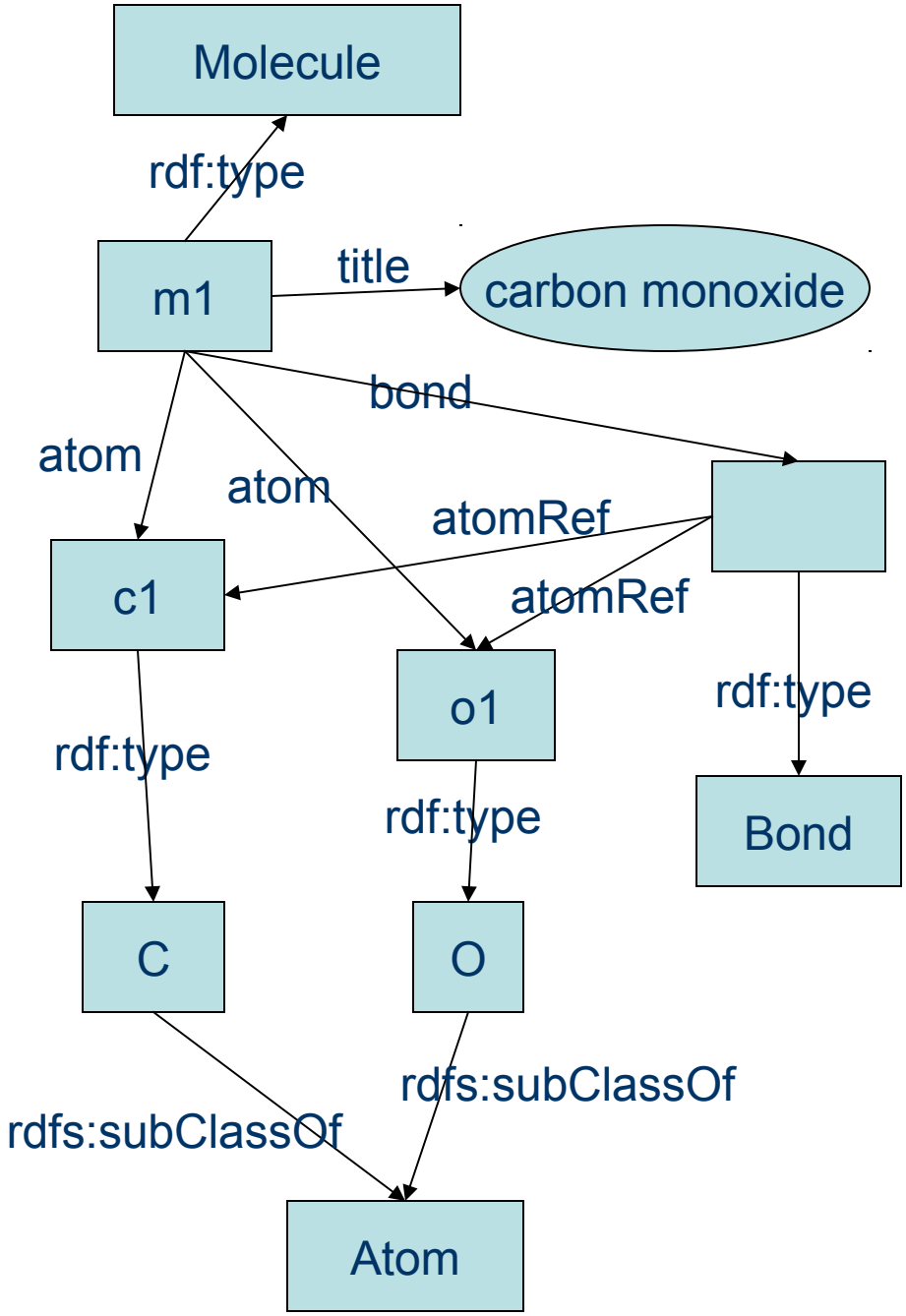
- RDF semantics is based on graphs
- All relationships are explicit (self-describing)
- Uses only XSD basic data types
- Data is decoupled from any context

XML vs. RDF Terminology

XML	RDF
Element Type	Class
Element Instance	Resource
Data attribute	DatatypeProperty
Reference attribute	ObjectProperty
Containment	Property

RDF graph for carbon monoxide

```
<Molecule rdf:id="m1"
      title="carbon monoxide">
 <atom>
  <C rdf:id="c1"/>
  <O rdf:id="o1"/>
 </atom>
 <bond>
  <Bond>
   <atomRef rdf:resource="c1"/>
   <atomRef rdf:resource="o1"/>
  </Bond
 </bond>
</Molecule>
```



RDF Triples

- RDF graphs consist of edges called *triples* because they have three components: subject, predicate and object.
- The semantics of RDF is determined by the set of triples that are explicitly asserted or inferred.
- In the chemical example, some of the triples are:
  - (m1, rdf:type, cml:Molecule)
  - (m1, cml:title, “carbon monoxide”)
  - (m1, cml:atom, c1)
  - (m1, cml:atom, o1)
- Properties are many-to-many relationships.

Web Ontology Language

- OWL classes can be *constructed* from other classes.
- Resources can be can be declared (or inferred) to be the *same*.
- Class constructors and resource equivalence are useful for interoperability.
- Properties can be constrained to be
  - Functional (many-to-one)
  - Inverse functional (database key)

Class Construction

- Concepts are generally defined in terms of other concepts. For example:

  The iridocorneal endothelial syndrome (ICE) is a disease characterized by corneal endothelium proliferation and migration, iris atrophy, corneal oedema and/or pigmentary iris nevi.

- ICE-Syndrome class is the intersection of:
  - The set of all diseases
  - The set of things that have at least one of the four symptoms

```
<owl:Class rdf:ID="ICE-Syndrome">
  <owl:intersectionOf parseType="Collection">
    <owl:Class rdf:about="#Disease"/>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#has-symptom"/>
      <owl:someValuesFrom>
        <owl:Class rdf:ID="ICE-Symptoms">
          <owl:oneOf parseType="Collection">
            <Symptom name="corneal endothelium proliferation and migration"/>
            <Symptom name="iris atrophy"/>
            <Symptom name="corneal oedema"/>
            <Symptom name="pigmentary iris nevi"/>
          </owl:oneOf>
        </owl:Class>
      </owl:someValuesFrom>
    </owl:Restriction>
  </owl:intersectionOf>
</owl:Class>
```

Example of Class Construction

OWL Semantics

- An OWL ontology defines a theory of the world. States of the world that are consistent with the theory are called *interpretations* of the theory.
- A fact that is true in every model is said to be *entailed* by the theory. OWL semantics is defined by entailment.
- By contrast relational database semantics is defined by *constraints*.

Open vs. Closed Worlds

- OWL assumes an *open world*, while databases assume a *closed world*.
- The advantage of the open world assumption is that it is more compatible with the web where one need not know all of the facts, and new facts are continually being added.
- The disadvantage is that operations (such as queries) are much more computationally complex.

The Semantic Web and Uncertainty

- There are many sources of uncertainty, such as measurements, unmodeled variables, and subjectivity.
- The Semantic Web is based on formal logic for which one can only assert facts that are unambiguously certain.
- The *Bayesian Web* is a proposal to add reasoning about certainty to the Semantic Web.
- The basis for the Bayesian Web is the concept of a Bayesian network.

Bayesian Web facilities

- Common interchange format
- Ability to refer to common variables (diseases, drugs, ...)
- Context specification
- Authentication and trust
- Open hierarchy of probability distribution types
- Component based construction of BNs
- BN inference engines
- Meta-analysis services

Bayesian Web Capabilities

- Use a BN developed by another group as easily as navigating from one Web page to another.
- Perform stochastic inference using information from one source and a BN from another.
- Combine BNs from the same or different sources.
- Reconcile and validate BNs.

Ontology Issues 1

- What is the most appropriate language?
  - XML, RDF, OWL (Lite, DL, Full)
  - The choice depends on the requirements
- Ontology design
  - Classes, properties and rules
- What tools are appropriate?
  - Design tools, rule engines, theorem provers
- Reuse vs. interoperation

Ontology Issues 2

- Coping with complexity
  - Worst cases can be very complex
  - In practice, processing is efficient
- Validation
  - Correctness, formal consistency
- Maintenance
  - Requirements and circumstances change

To Learn More

For more information, see K. Baclawski and T. Niu, *Ontologies for Bioinformatics*, MIT Press, October, 2005.

The website the book is ontobio.org.

A longer version of this talk is available at CSB2005 Tutorial.

Data fusion is covered in meta-analysis.