



## **Acknowledgments**

Shawn Bowers

Bertram Ludaescher

Ruth Ann Bertsch

Melora Svoboda

# **Inferring Phylogenetic Trees with Kepler: An Introduction to Collection-Oriented Workflows**

**Timothy M. McPhillips**

UC Davis Genome Center

President, Natural Diversity Discovery Project

# Overview

- Interdisciplinary applications of Kepler.
- Nested collections in scientific computing.
- Collection-oriented workflows.
- Inferring phylogenetic trees (demo).

# Kepler is a multidisciplinary project

## Kepler project voting members

Ilkay Altintas	<i>SDM, NLADR, Resurgence, EOL, ROADNet, GEON, CIPRes</i>
Chad Berkley	<i>SEEK</i>
Shawn Bowers	<i>SEEK</i>
Matthew Brooke	<i>SEEK</i>
Christopher Brooks	<i>Ptolemy II</i>
Zhengang Cheng	<i>SDM</i>
Tobin Fricke	<i>ROADNet</i>
Daniel Higgins	<i>SEEK</i>
Efrat Jaeger	<i>GEON</i>
Matt Jones	<i>SEEK</i>
Werner Krebs	<i>EOL</i>
Edward A. Lee	<i>Ptolemy II</i>
Bertram Ludaescher	<i>SEEK, SDM, GEON, ROADNet</i>
Timothy McPhillips	<i>NDDP</i>
Steve Mock	<i>Informnet</i>
Stephen Neuendorffer	<i>Ptolemy II</i>
Rod Spears	<i>SEEK</i>
Wibke Sudholt	<i>Resurgence</i>
Jing Tao	<i>SEEK</i>
Mladen Vouk	<i>SDM</i>
Xiaowen Xin	<i>SDM</i>
Yang Zhao	<i>Ptolemy II</i>



Science Environment for  
Ecological Knowledge



Natural Diversity  
Discovery Project



Ptolemy II



Real-time  
Observatories  
Applications  
and Data  
Management  
Network



Encyclopedia  
of Life



The Geosciences Network

Cyberinfrastructure  
for the Geosciences

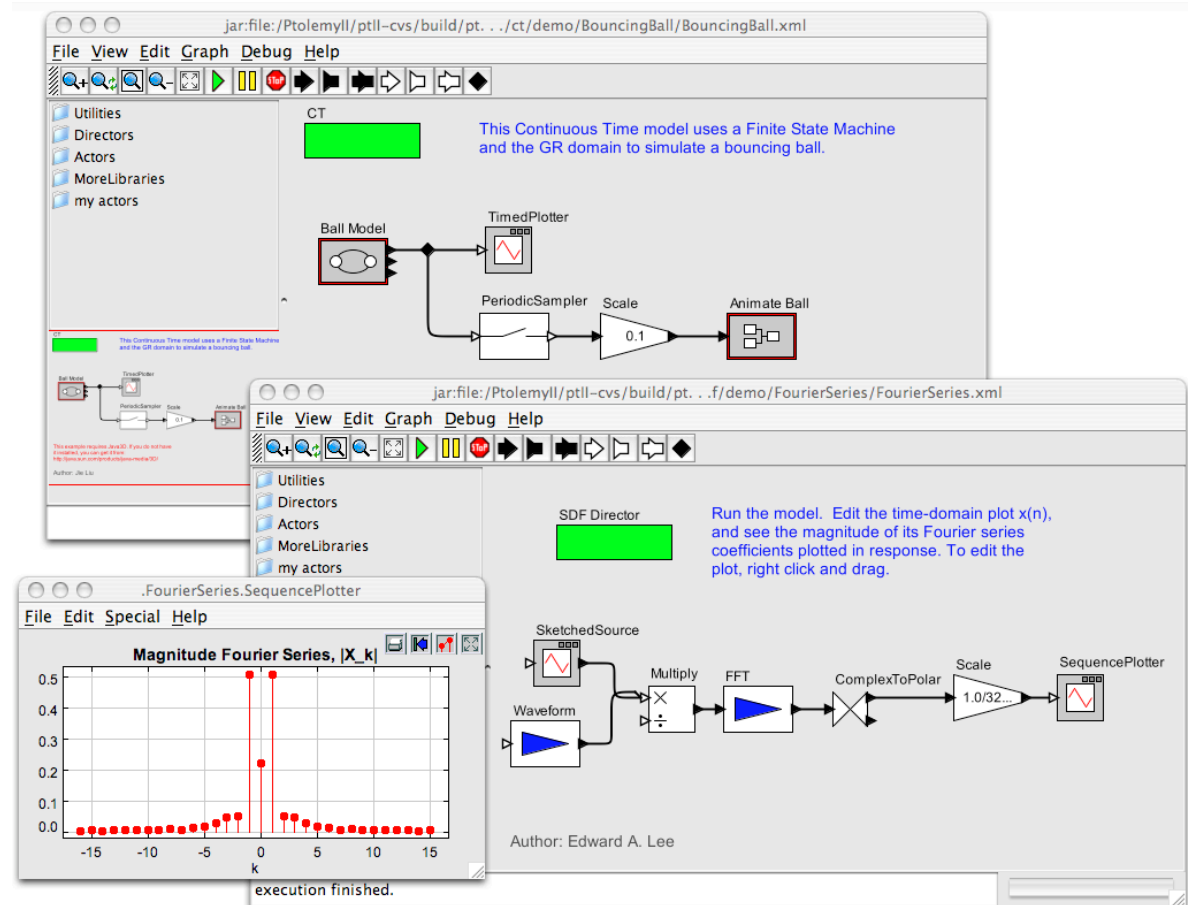


SDM Center/Scientific Process Automation

<b>CIPRes</b>	<a href="http://www.phylo.org">www.phylo.org</a>
<b>EOL</b>	<a href="http://eol.sdsc.edu">eol.sdsc.edu</a>
<b>GEON</b>	<a href="http://www.geongrid.org">www.geongrid.org</a>
<b>Kepler</b>	<a href="http://www.kepler-project.org">www.kepler-project.org</a>
<b>NDDP</b>	<a href="http://www.nddp.org">www.nddp.org</a>
<b>Ptolemy II</b>	<a href="http://ptolemy.eecs.berkeley.edu/ptolemyII">ptolemy.eecs.berkeley.edu/ptolemyII</a>
<b>ROADNet</b>	<a href="http://roadnet.ucsd.edu">roadnet.ucsd.edu</a>
<b>SEEK</b>	<a href="http://seek.ecoinformatics.org">seek.ecoinformatics.org</a>
<b>SciDAC</b>	<a href="http://www-casc.llnl.gov/sdm">www-casc.llnl.gov/sdm</a>

# Support for multiple computing models enables diverse applications

- Kepler inherits the notion of directors from Ptolemy.
- *Directors* define the semantics of component interaction and overall execution.
- Multiple computing models, and ability to add new models, enables Kepler to support a broad range of scientific disciplines
- Ability to mix computing models in a single workflow facilitates multidisciplinary efforts.



# Actor-orientation of Kepler facilitates collaboration in multiple dimensions

- Between researchers in a particular field (e.g., two ecologists sharing actors and workflows).
- Between researchers in different fields (e.g., ecologists using actors developed by systematists).
- Between experts (in a particular methodology) and less experienced researchers (e.g., an expert systematist sharing her experience with others by providing pre-configured workflows).
- Between researchers with different levels of expertise in programming (e.g., a systematist skilled at Java programming providing new actors to the community).
- Between researchers in the natural and information sciences (e.g., a software engineer developing new actors requested by ecologists).

# Natural Diversity Discovery Project

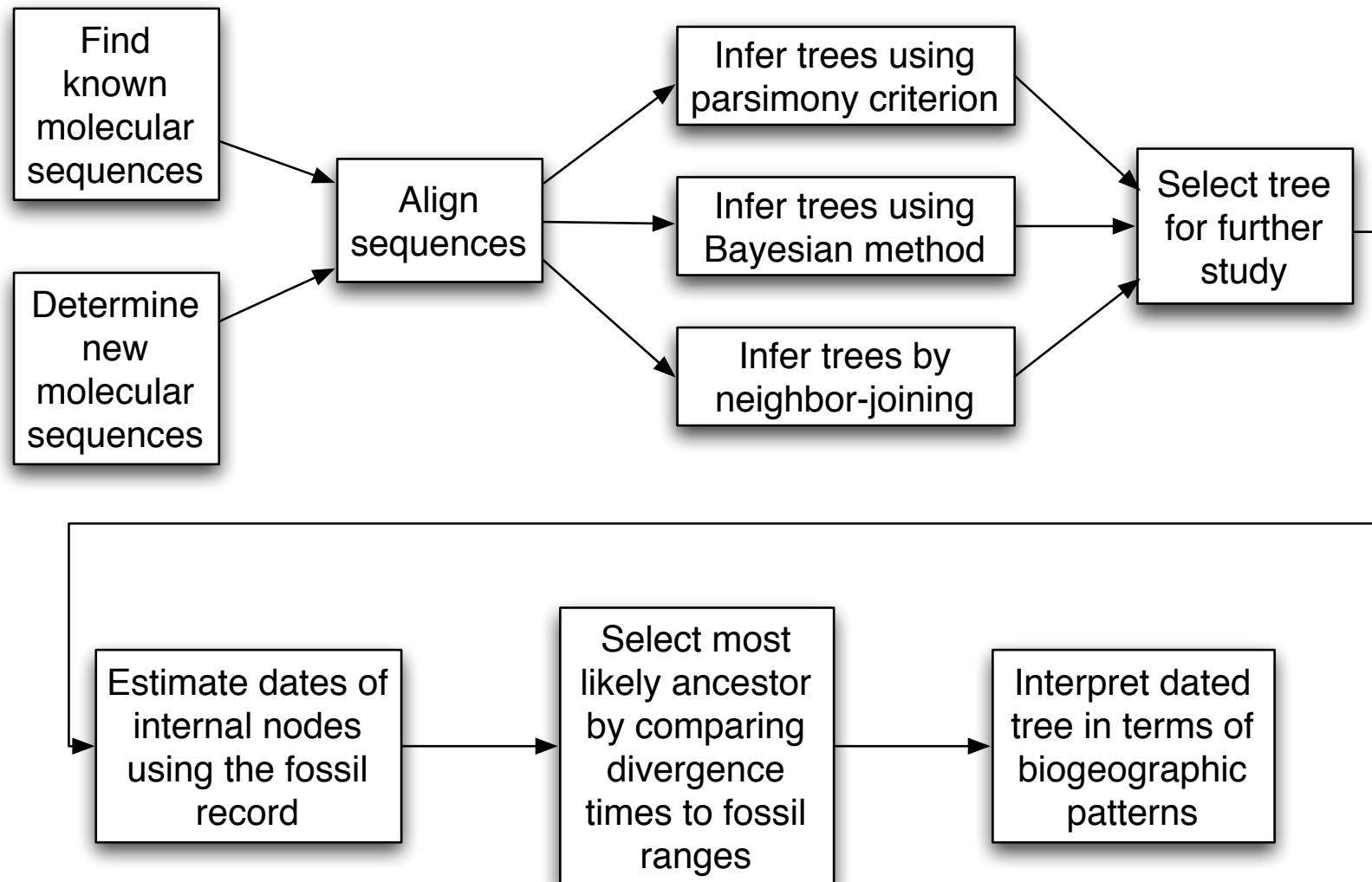
*resources for exploring and understanding the diversity of life*

- The NDDP aims to help the public understand scientific explanations for the diversity of life by enabling non-experts to discover the implications of the latest scientific data and methods for themselves.
- We plan to enable the public to reproduce the results of studies carried out by professional researchers and reported in news media, magazines and journals.
- Our web-based tools will allow the public to infer phylogenies using morphological data, molecular sequences, and genome-level features.
- They will use the fossil record and other information to correlate these phylogenies with events in Earth history.

# Why use phylogenetics to educate the public about science?

- Recent methodological developments have yielded a variety of approaches for inferring and interpreting phylogenetic trees.
- Genome sequencing efforts have lead to a dramatic increase in phylogenetically informative data for an increasing number of taxa, and this data is publicly accessible via the Internet. Efforts are underway towards assembling the entire Tree of Life.
- Rapidly improving technology, increasing computing resources, and global access to these resources make the project computationally feasible.
- **Making sense of the Tree of Life in light of Earth history requires a multidisciplinary approach that illustrates the coherence of scientific theories in a wide range of fields.**

# Example: When did the modern amphibians originate and who were their closest Paleozoic ancestors?



Zhang, et al (2005). Mitogenomic Perspectives on the Origin and Phylogeny of Living Amphibians. Syst. Biol. 54:391-400.

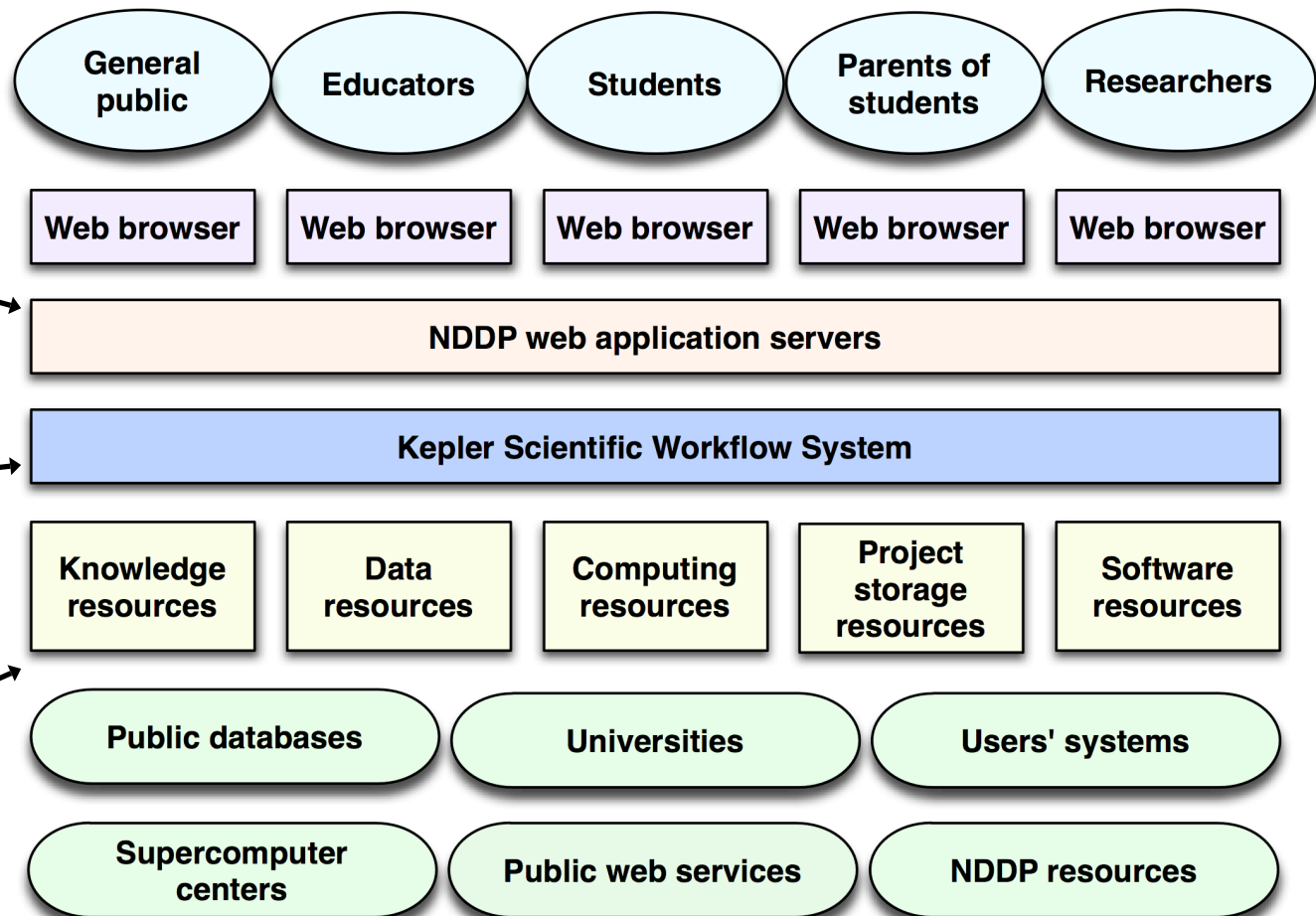


# A virtual laboratory for exploring and understanding the diversity of life

Easy-to-use interface provided to the general public and the educational community as a web application.

Flexible scientific workflow system provided to researchers as a desktop application.

Public, academic, and government-funded resources made available to the public through the NDDP.



## Kepler will...

- **Integrate a variety of resources for researchers and the public.**
- **Automate the scientific workflows behind the NDDP tools.**
- **Enable the public to reproduce the results of professional researchers.**

## Nested data collections are common in scientific research

- Many scientific workflows operate on multiple pieces of data related to each other in scientifically meaningful ways.
- Nested data collections can maintain these relationships between different pieces of data.

# Many scientific data sets are naturally represented as nested collections

```

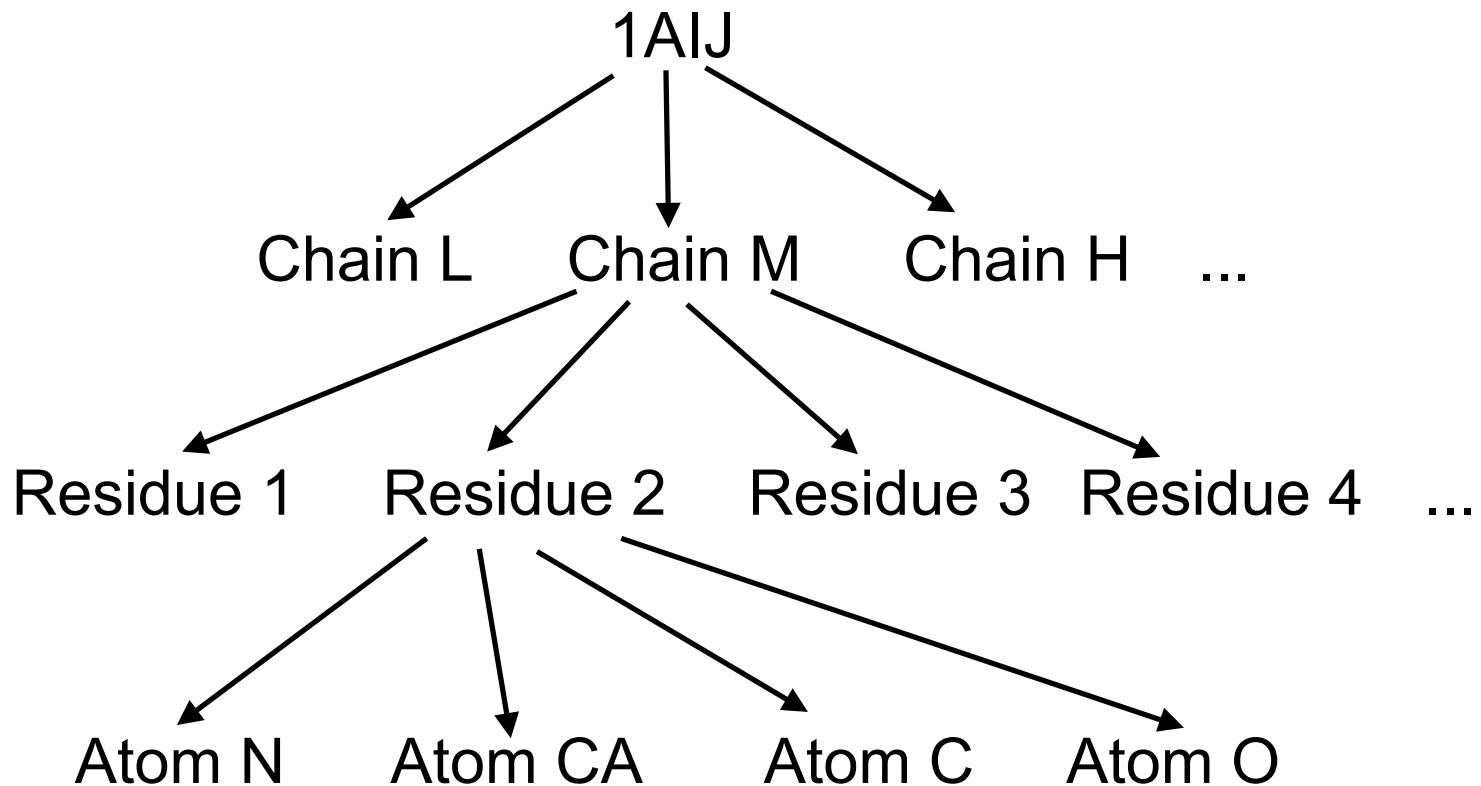
:
ATOM  2225  CG  ASN  L  280      58.188  80.207  56.085  1.00  42.70    C
ATOM  2226  OD1 ASN  L  280      57.052  80.273  55.622  1.00  39.93    O
ATOM  2227  ND2 ASN  L  280      59.051  81.100  55.627  1.00  37.74    N
ATOM  2228  N   GLY  L  281      58.254  77.606  60.504  1.00  51.78    N
ATOM  2229  CA  GLY  L  281      58.506  76.337  61.166  1.00  57.51    C
ATOM  2230  C   GLY  L  281      57.923  76.307  62.577  1.00  62.72    C
ATOM  2231  O   GLY  L  281      57.798  75.234  63.160  1.00  67.50    O
ATOM  2232  OXT GLY  L  281      57.625  77.356  63.139  1.00  65.61    O
TER   2233           GLY  L  281
ATOM  2234  N   ALA  M   1      81.104  96.260  13.652  1.00  92.72    N
ATOM  2235  CA  ALA  M   1      80.015  97.008  13.062  1.00  92.60    C
ATOM  2236  C   ALA  M   1      80.331  98.386  13.640  1.00  90.68    C
ATOM  2237  O   ALA  M   1      81.530  98.693  13.704  1.00  88.32    O
ATOM  2238  CB  ALA  M   1      78.678  96.475  13.579  1.00  90.83    C
ATOM  2239  N   GLU  M   2      79.371  99.216  14.049  1.00  86.28    N
ATOM  2240  CA  GLU  M   2      79.694 100.459  14.727  1.00  79.94    C
ATOM  2241  C   GLU  M   2      80.119 100.109  16.150  1.00  74.61    C
ATOM  2242  O   GLU  M   2      80.008  98.953  16.590  1.00  75.59    O
ATOM  2243  CB  GLU  M   2      78.490 101.390  14.820  1.00  79.43    C
:

```

# A PDB (protein data bank) file is implicitly nested

- A PDB file is a flat table of records with one ATOM record per atom in a protein structure.
- However, based on their knowledge of protein structure, structural biologists interpret this file as if it were structured in the following way:
  - Each PDB file contains one or more proteins.
  - Each protein is composed of one or more protein chains.
  - Each protein chain is composed of many amino acids (“residues”).
  - Each residue is composed of a number of atoms.

The nested structure of a PDB file  
can be represented as a tree



# Spotting the structure in a PDB file

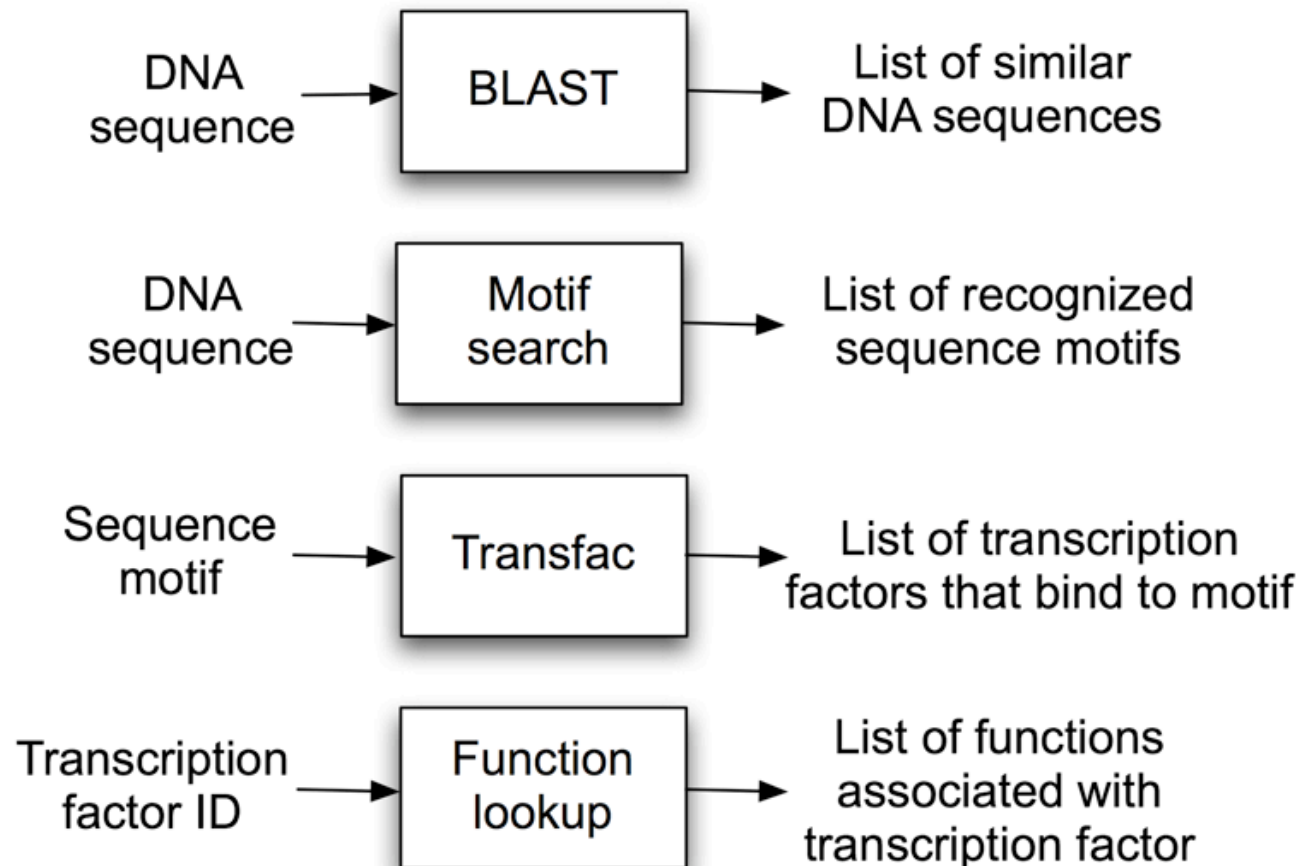
		Residue type	Chain ID	Residue ID						Atom type
ATOM	2225	CG	ASN L	280	58.188	80.207	56.085	1.00	42.70	C
ATOM	2226	OD1	ASN L	280	57.052	80.273	55.622	1.00	39.93	O
ATOM	2227	ND2	ASN L	280	59.051	81.100	55.627	1.00	37.74	N
ATOM	2228	N	GLY L	281	58.254	77.606	60.504	1.00	51.78	N
ATOM	2229	CA	GLY L	281	58.506	76.337	61.166	1.00	57.51	C
ATOM	2230	C	GLY L	281	57.923	76.307	62.577	1.00	62.72	C
ATOM	2231	O	GLY L	281	57.798	75.234	63.160	1.00	67.50	O
ATOM	2232	OXT	GLY L	281	57.625	77.356	63.139	1.00	65.61	O
TER	2233		GLY L	281						
ATOM	2234	N	ALA M	1	81.104	96.260	13.652	1.00	92.72	N
ATOM	2235	CA	ALA M	1	80.015	97.008	13.062	1.00	92.60	C
ATOM	2236	C	ALA M	1	80.331	98.386	13.640	1.00	90.68	C
ATOM	2237	O	ALA M	1	81.530	98.693	13.704	1.00	88.32	O
ATOM	2238	CB	ALA M	1	78.678	96.475	13.579	1.00	90.83	C
ATOM	2239	N	GLU M	2	79.371	99.216	14.049	1.00	86.28	N

Three tip-offs that this data is implicitly nested in structure are (1) the residue ID groups atoms in the same amino acid, (2) the residue IDs are unique only within a particular chain, and (3) a special TER record is used to terminate a chain.

# Explicitly representing the nested structure of a PDB file (in XML)

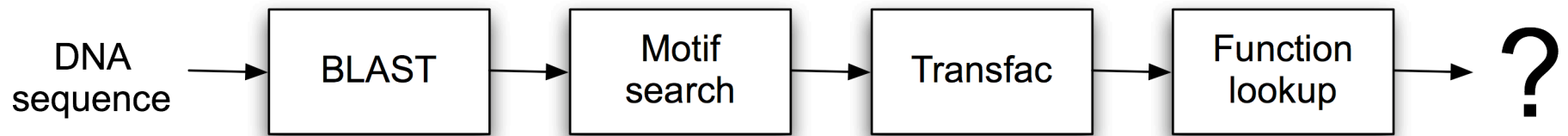
```
<Chain id="L">
  :
  <Residue id="280" type="ASN">
    :
    <Atom name="CG" type="C" x="58.188" y="80.207" ..... />
    <Atom name="OD1" type="O" x="57.052" y="80.273" ..... />
    <Atom name="ND2" type="N" x="59.051" y="81.100" ..... />
  </Residue>
  <Residue id="281" type="GLY">
    <Atom name="N" type="N" x="58.254" y="77.606" ..... />
    <Atom name="CA" type="C" x="58.506" y="76.337" ..... />
    <Atom name="C" type="C" x="57.923" y="76.307" ..... />
    <Atom name="O" type="O" x="57.798" y="75.304" ..... />
    <Atom name="OXT" type="O" x="57.625" y="77.356" ..... />
  </Residue>
</Chain>
<Chain id="M">
  <Residue id="1" type="ALA">
    <Atom name="N" type="N" x="81.104" y="96.260" ..... />
    <Atom name="CA" type="C" x="80.015" y="97.008" ..... />
    :
```

# Some steps in scientific workflows naturally generate lists of results



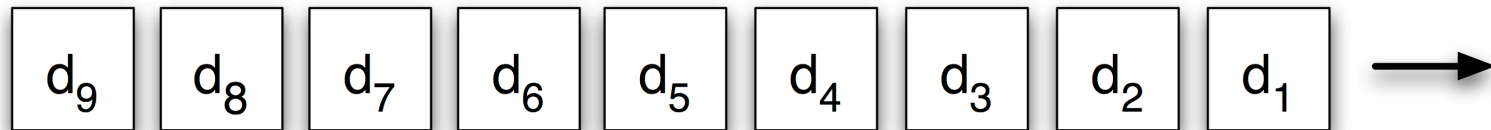


# How do you compose a workflow to run each of these steps in series?

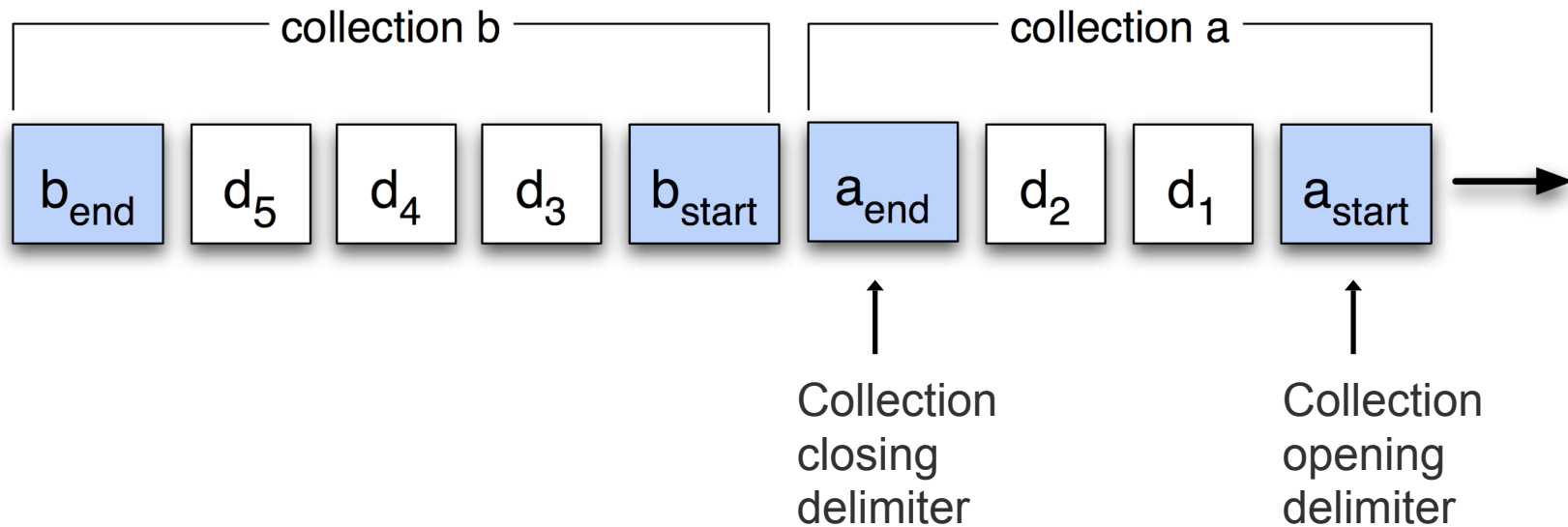


- **Problem:** while each step above generates a list, it operates on one input item at a time.
- This means that the data flowing through this workflow becomes more nested with each step.
- Conventional Kepler workflows can become very complex (i.e., require many wires, special actors for managing data, etc) when dealing with deeply nested data.
- **Solution:** Collection-oriented workflows operate naturally on data collections nested for any of these reasons.

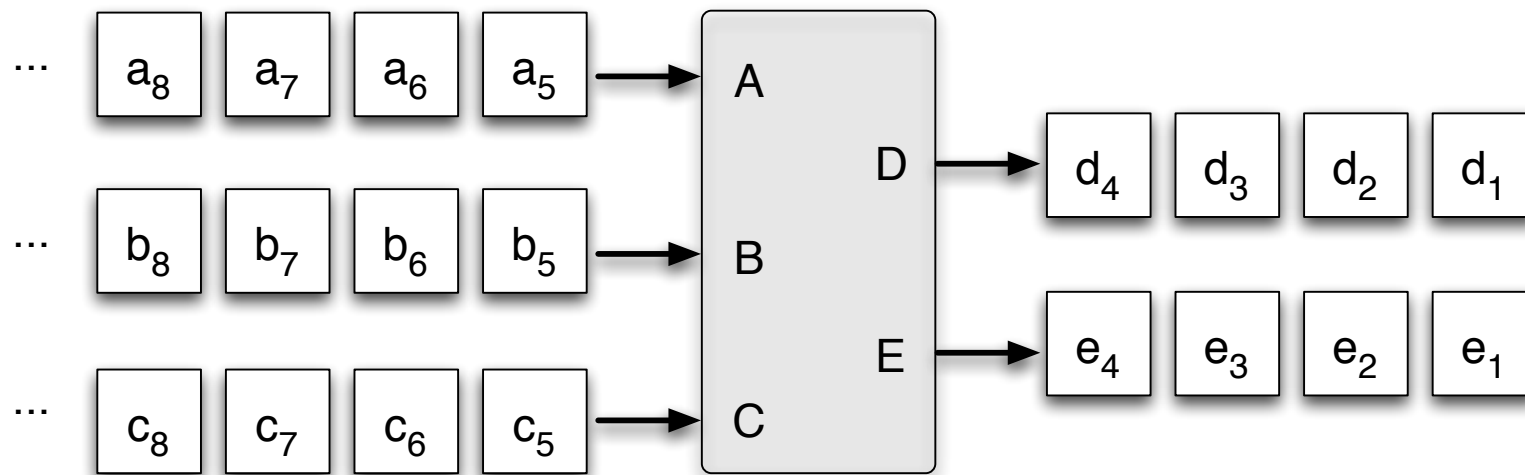
## Flow of tokens in conventional workflows



## Flow of tokens in collection-oriented workflows

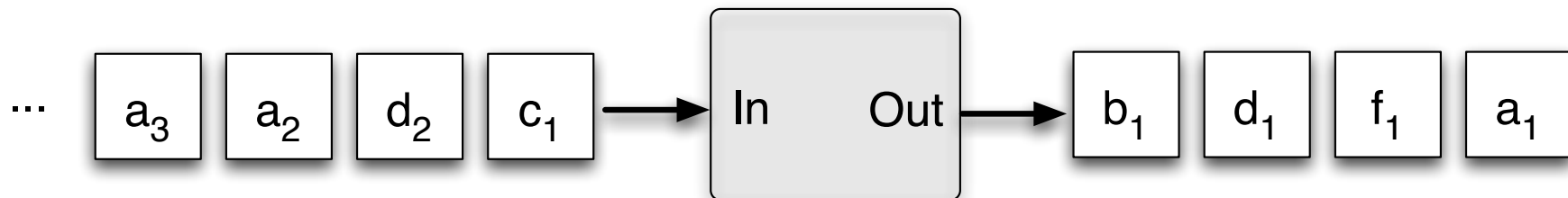


Conventional actors often have multiple input and output ports with distinct types



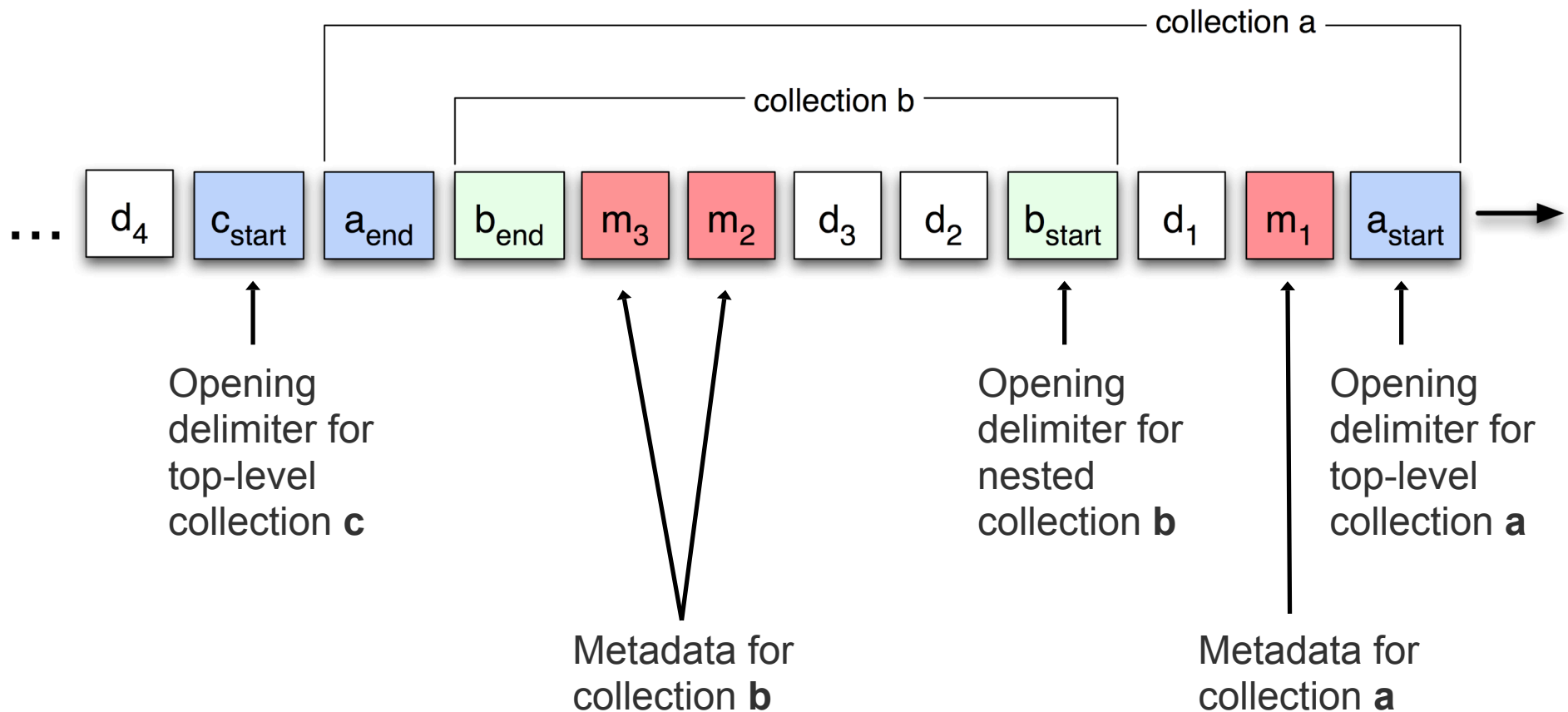
- Note that tokens are typically destroyed and new ones created at each step in a conventional workflow.

In contrast, most collection-oriented actors have just one input and one output port



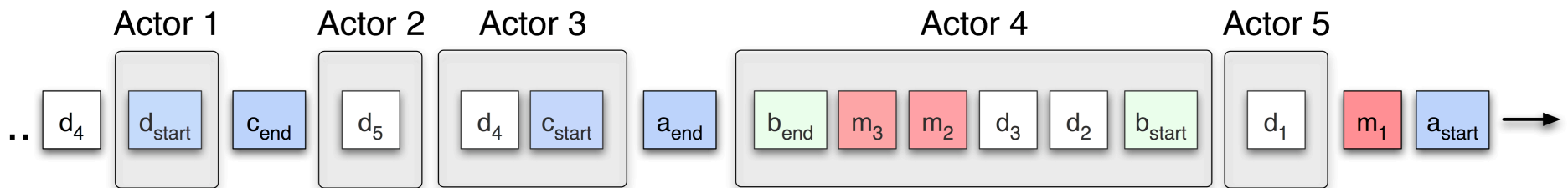
- Collection-oriented workflows are composed by simply stringing together collection-oriented actors (“coactors”).
- Any type of data may be sent to any coactor.
- Coactors do the work of selecting relevant tokens from their input.
- This makes it easy to compose collection-oriented workflows (at least, I think so).

# Collections may contain data tokens, metadata tokens, and other collections



# Collection-oriented workflows enable efficient computing by “pipelining” data

- Coactors may process one token at a time or operate on entire collections at once.



Actors 4 and 5 are processing the contents of collection **a** at the same time.

- Coactors may specify what types of collections and data they process.

- Pipelining is completely automatic.

Actor 4 processes entire collections (of a particular type) at one time.

Actor 5 processes one data token at a time.

# Demonstration