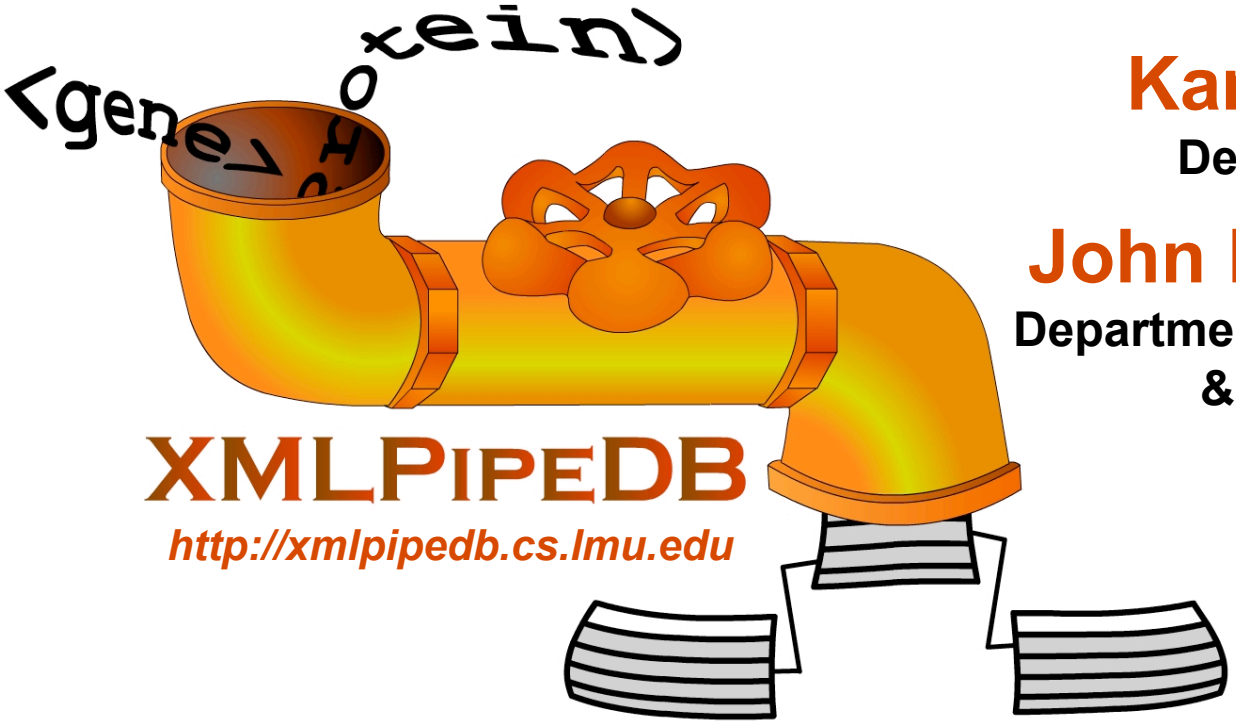


Kam D. Dahlquist
Department of Biology

John David N. Dionisio
Department of Electrical Engineering
& Computer Science

Collaborating Early and Often: Bringing Biology and Computer Science Together Through an Open Source Culture

Cub Club
November 14, 2006

LMU | LA
Loyola Marymount University

Outline

- **Process**
  - **-- open source pedagogy**
  - **-- interdisciplinary collaboration**
- **Motivation**
  - **-- flood, deluge, tsunami (!) of genomic data**
  - **-- project requirements**
- **XMLPipeDB: an open source tool chain for building relational databases from XML sources**
- **Future Directions**

Outline

- **Process**
  - **-- open source pedagogy**
  - **-- interdisciplinary collaboration**
- **Motivation**
  - **-- flood, deluge, tsunami (!) of genomic data**
  - **-- project requirements**
- **XMLPipeDB: an open source tool chain for building relational databases from XML sources**
- **Future Directions**

Recourse: An Open Source Culture in the Undergraduate Computer Science Curriculum

http://recourse.cs.lmu.edu/

Motivation: **the disconnect between undergraduate computer science training and expectations/skill sets required in industry**

Undergraduate Training	Industry Expectation
Work alone	Work in a team
"Toy" programs and algorithms	Large, modular project
Throwaway code	Code longevity (for better or worse)

Official Open Source Definition (version 1.9)

Free redistribution

Source code

Derived works

Integrity of the author's source code

No discrimination against persons or groups

No discrimination against fields of endeavor

Distribution of license

License must not be specific to a product

License must not restrict other software

License must be technology-neutral

Open Source Teaching Framework

Source Code:

- All code resides in a centralized, public repository
- As much as possible, everyone's code is visible to everyone else for code review or team fixing
- No code is thrown away, it remains available to future "generations"

Open Source Teaching Framework

Source Code:

- All code resides in a centralized, public repository
- As much as possible, everyone's code is visible to everyone else for code review or team fixing
- No code is thrown away, it remains available to future "generations"

Quality & Community:

- Documentation, inline and online
- Automated tests
- Constructive code review, beyond "does it work?"
- Long-term projects release early, release often
- Form collaborative communities among faculty, students, classes, and projects

"CourseForge"
A Hardware + Software Infrastructure for Supporting the Teaching Framework

- Certain teaching elements are impractical without some degree of automation
- "CourseForge" is currently under development
- Derived from open source software, delivered as open source software — the system will interoperate with existing open source tools
- Our course used SourceForge.net and later added a Wiki hosted by the Computer Science Department

CMSI 698: Special Studies in Bioinformatics

- **Team-taught by a biologist and a computer scientist**

CMSI 698: Special Studies in Bioinformatics

- **Team-taught by a biologist and a computer scientist**
- **Enrollment in Spring 2006:**
  - **-- eight students from Master's degree program in Computer Science**
  - **-- several coming from aerospace industry**
  - **-- none with more than college-level introductory biology**

CMSI 698: Special Studies in Bioinformatics

- **Team-taught by a biologist and a computer scientist**
- **Enrollment in Spring 2006:**
  - **-- eight students from Master's degree program in Computer Science**
  - **-- several coming from aerospace industry**
  - **-- none with more than college-level introductory biology**
- **Project-based class began development of XMLPipeDB**

CMSI 698: Special Studies in Bioinformatics

- **Team-taught by a biologist and a computer scientist**
- **Enrollment in Spring 2006:**
  - **-- eight students from Master's degree program in Computer Science**
  - **-- several coming from aerospace industry**
  - **-- none with more than college-level introductory biology**
- **Project-based class began development of XMLPipeDB**
- **XMLPipeDB development continued by four students in summer session course entitled Open Source Software Development Workshop**

CMSI 698: Special Studies in Bioinformatics

- Team-taught by a biologist and a computer scientist
- Enrollment in Spring 2006:
  - -- eight students from Master's degree program in Computer Science
  - -- several coming from aerospace industry
  - -- none with more than college-level introductory biology
- Project-based class began development of XMLPipeDB
- XMLPipeDB development continued by four students in summer session course entitled Open Source Software Development Workshop
- Both courses used the open source curricular framework embraced by the Computer Science Department

XMLPipeDB Project Management: Lessons Learned

- **Students on the project had varying levels of maturity, knowledge, and skill coming into the project**
  - **-- some naturally took on a leadership role**
  - **-- some hung back or did the minimum required to get by**

XMLPipeDB Project Management: Lessons Learned

- **Students on the project had varying levels of maturity, knowledge, and skill coming into the project**
  - **-- some naturally took on a leadership role**
  - **-- some hung back or did the minimum required to get by**

- **Needed to increase communication and sense of team**
  - **-- students preferred to interact with faculty for questions, rather than each other**
  - **-- bug trackers and developer's forum used only sporadically**
  - **-- implemented weekly reports on Wiki to increase accountability**

XMLPipeDB Project Management: Lessons Learned

- **Students on the project had varying levels of maturity, knowledge, and skill coming into the project**
  - **-- some naturally took on a leadership role**
  - **-- some hung back or did the minimum required to get by**

- **Needed to increase communication and sense of team**
  - **-- students preferred to interact with faculty for questions, rather than each other**
  - **-- bug trackers and developer's forum used only sporadically**
  - **-- implemented weekly reports on Wiki to increase accountability**

- **[SourceForge servers were frequently down during class]**

XMLPipeDB Project Management: Lessons Learned

- **Students on the project had varying levels of maturity, knowledge, and skill coming into the project**
  - **-- some naturally took on a leadership role**
  - **-- some hung back or did the minimum required to get by**

- **Needed to increase communication and sense of team**
  - **-- students preferred to interact with faculty for questions, rather than each other**
  - **-- bug trackers and developer's forum used only sporadically**
  - **-- implemented weekly reports on Wiki to increase accountability**

- **[SourceForge servers were frequently down during class]**

- **6 months from conception to product**

XMLPipeDB Project Management: Lessons Learned

- **Students on the project had varying levels of maturity, knowledge, and skill coming into the project**
  - **-- some naturally took on a leadership role**
  - **-- some hung back or did the minimum required to get by**

- **Needed to increase communication and sense of team**
  - **-- students preferred to interact with faculty for questions, rather than each other**
  - **-- bug trackers and developer's forum used only sporadically**
  - **-- implemented weekly reports on Wiki to increase accountability**

- **[SourceForge servers were frequently down during class]**

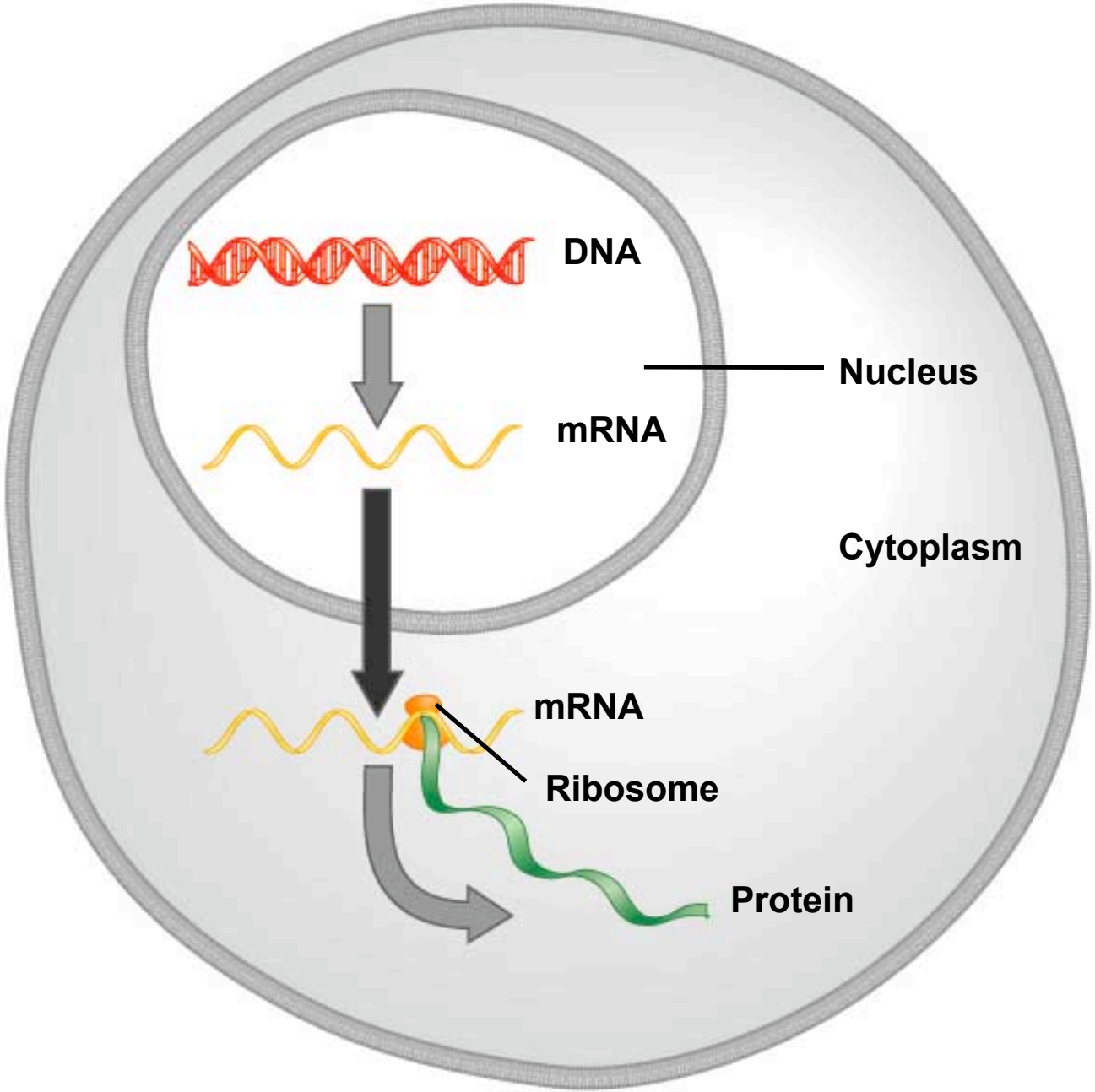
- **6 months from conception to product**

- **Even the weakest student contributed useable code**

Outline

- Process
  - -- open source pedagogy
  - -- interdisciplinary collaboration
- **Motivation**
  - -- flood, deluge, tsunami (!) of genomic data
  - -- project requirements
- XMLPipeDB: an open source tool chain for building relational databases from XML sources
- Future Directions

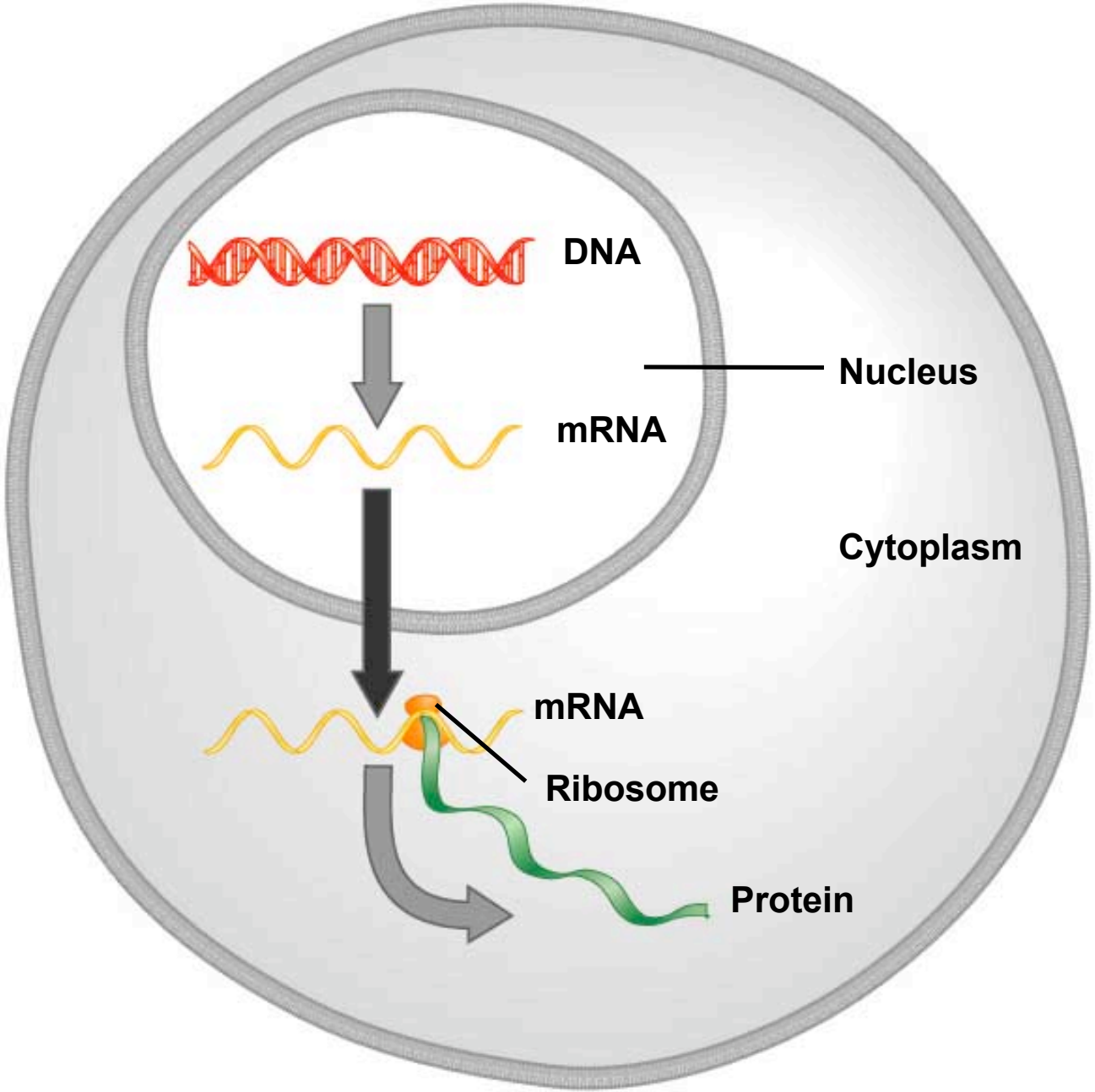
Central Dogma of Molecular Biology



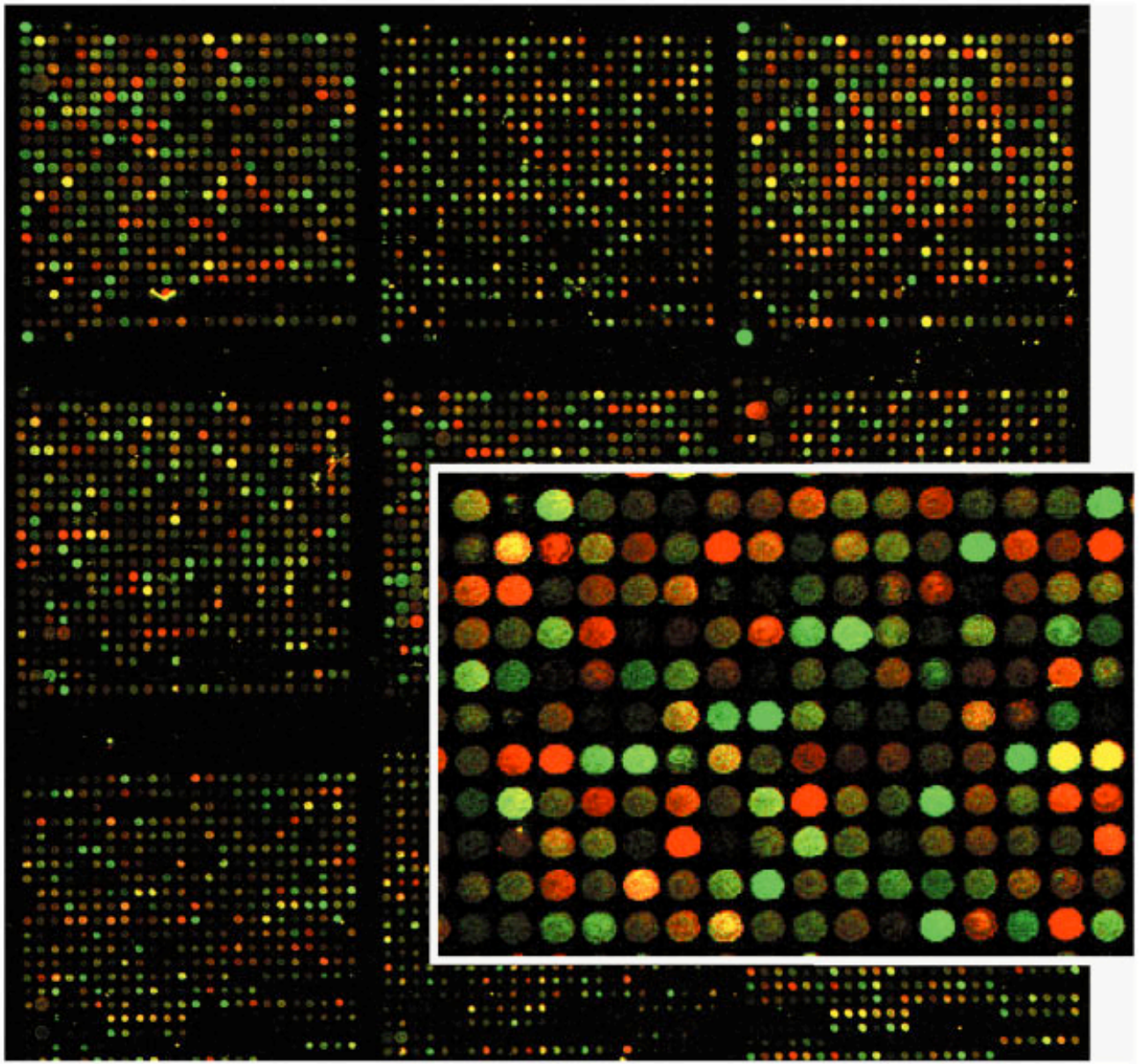
Representative Sequenced Genomes

Species	Genome size (millions of b.p.)	Estimated # of genes
Haemophilus influenzae (bacteria)	1.83	1,743
Escherichia coli (bacteria)	4.6	4,288
Saccharomyces cerevisiae (yeast)	12	6,340
Drosophila melanogaster (fruit fly)	180	13,600
Caenorhabditis elegans (worm)	97	19,000
Arabidopsis thaliana (mustard plant)	130	*26,000*
Mus musculus (mouse)	3000	*30,000*
Homo sapiens (human)	3000	22,000

Central Dogma of Molecular Biology



DNA Microarray



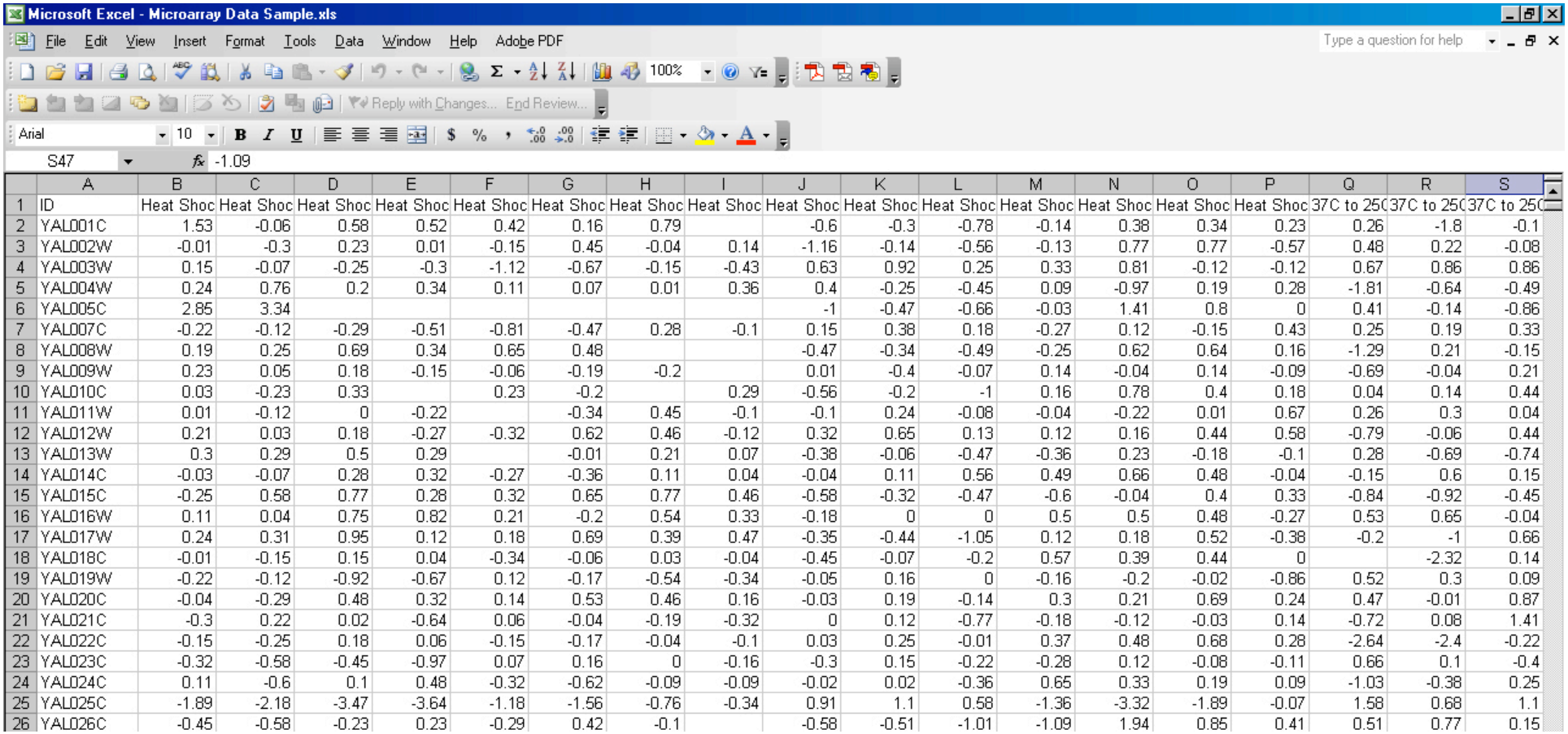
Freeman (2002)

One spot = DNA from one gene

Green = more mRNA in control sample

Red = more mRNA in experimental sample

Yellow = the same amount of mRNA in each sample



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	ID	Heat Shoc	Heat Shoc	Heat Shoc	Heat Shoc	Heat Shoc	Heat Shoc	Heat Shoc	Heat Shoc	Heat Shoc	Heat Shoc	Heat Shoc	Heat Shoc	Heat Shoc	Heat Shoc	Heat Shoc	37C to 25(	37C to 25(	37C to 25(
2	YAL001C	1.53	-0.06	0.58	0.52	0.42	0.16	0.79		-0.6	-0.3	-0.78	-0.14	0.38	0.34	0.23	0.26	-1.8	-0.1
3	YAL002W	-0.01	-0.3	0.23	0.01	-0.15	0.45	-0.04	0.14	-1.16	-0.14	-0.56	-0.13	0.77	0.77	-0.57	0.48	0.22	-0.08
4	YAL003W	0.15	-0.07	-0.25	-0.3	-1.12	-0.67	-0.15	-0.43	0.63	0.92	0.25	0.33	0.81	-0.12	-0.12	0.67	0.86	0.86
5	YAL004W	0.24	0.76	0.2	0.34	0.11	0.07	0.01	0.36	0.4	-0.25	-0.45	0.09	-0.97	0.19	0.28	-1.81	-0.64	-0.49
6	YAL005C	2.85	3.34							-1	-0.47	-0.66	-0.03	1.41	0.8	0	0.41	-0.14	-0.86
7	YAL007C	-0.22	-0.12	-0.29	-0.51	-0.81	-0.47	0.28	-0.1	0.15	0.38	0.18	-0.27	0.12	-0.15	0.43	0.25	0.19	0.33
8	YAL008W	0.19	0.25	0.69	0.34	0.65	0.48			-0.47	-0.34	-0.49	-0.25	0.62	0.64	0.16	-1.29	0.21	-0.15
9	YAL009W	0.23	0.05	0.18	-0.15	-0.06	-0.19	-0.2		0.01	-0.4	-0.07	0.14	-0.04	0.14	-0.09	-0.69	-0.04	0.21
10	YAL010C	0.03	-0.23	0.33		0.23	-0.2		0.29	-0.56	-0.2	-1	0.16	0.78	0.4	0.18	0.04	0.14	0.44
11	YAL011W	0.01	-0.12	0	-0.22		-0.34	0.45	-0.1	-0.1	0.24	-0.08	-0.04	-0.22	0.01	0.67	0.26	0.3	0.04
12	YAL012W	0.21	0.03	0.18	-0.27	-0.32	0.62	0.46	-0.12	0.32	0.65	0.13	0.12	0.16	0.44	0.58	-0.79	-0.06	0.44
13	YAL013W	0.3	0.29	0.5	0.29		-0.01	0.21	0.07	-0.38	-0.06	-0.47	-0.36	0.23	-0.18	-0.1	0.28	-0.69	-0.74
14	YAL014C	-0.03	-0.07	0.28	0.32	-0.27	-0.36	0.11	0.04	-0.04	0.11	0.56	0.49	0.66	0.48	-0.04	-0.15	0.6	0.15
15	YAL015C	-0.25	0.58	0.77	0.28	0.32	0.65	0.77	0.46	-0.58	-0.32	-0.47	-0.6	-0.04	0.4	0.33	-0.84	-0.92	-0.45
16	YAL016W	0.11	0.04	0.75	0.82	0.21	-0.2	0.54	0.33	-0.18	0	0	0.5	0.5	0.48	-0.27	0.53	0.65	-0.04
17	YAL017W	0.24	0.31	0.95	0.12	0.18	0.69	0.39	0.47	-0.35	-0.44	-1.05	0.12	0.18	0.52	-0.38	-0.2	-1	0.66
18	YAL018C	-0.01	-0.15	0.15	0.04	-0.34	-0.06	0.03	-0.04	-0.45	-0.07	-0.2	0.57	0.39	0.44	0		-2.32	0.14
19	YAL019W	-0.22	-0.12	-0.92	-0.67	0.12	-0.17	-0.54	-0.34	-0.05	0.16	0	-0.16	-0.2	-0.02	-0.86	0.52	0.3	0.09
20	YAL020C	-0.04	-0.29	0.48	0.32	0.14	0.53	0.46	0.16	-0.03	0.19	-0.14	0.3	0.21	0.69	0.24	0.47	-0.01	0.87
21	YAL021C	-0.3	0.22	0.02	-0.64	0.06	-0.04	-0.19	-0.32	0	0.12	-0.77	-0.18	-0.12	-0.03	0.14	-0.72	0.08	1.41
22	YAL022C	-0.15	-0.25	0.18	0.06	-0.15	-0.17	-0.04	-0.1	0.03	0.25	-0.01	0.37	0.48	0.68	0.28	-2.64	-2.4	-0.22
23	YAL023C	-0.32	-0.58	-0.45	-0.97	0.07	0.16	0	-0.16	-0.3	0.15	-0.22	-0.28	0.12	-0.08	-0.11	0.66	0.1	-0.4
24	YAL024C	0.11	-0.6	0.1	0.48	-0.32	-0.62	-0.09	-0.09	-0.02	0.02	-0.36	0.65	0.33	0.19	0.09	-1.03	-0.38	0.25
25	YAL025C	-1.89	-2.18	-3.47	-3.64	-1.18	-1.56	-0.76	-0.34	0.91	1.1	0.58	-1.36	-3.32	-1.89	-0.07	1.58	0.68	1.1
26	YAL026C	-0.45	-0.58	-0.23	0.23	-0.29	0.42	-0.1		-0.58	-0.51	-1.01	-1.09	1.94	0.85	0.41	0.51	0.77	0.15

And so on…

6144	YPR195C				0.6			-0.18		-0.31	-0.29	0.25	0.94	0.71	0.61	-0.17	-0.49	-0.24	-0.1
6145	YPR196W			-0.09		-0.07				0.3	-0.2	-0.04	-0.33	0.17	-0.19	0.26	0.9	-0.67	-0.7
6146	YPR197C	-0.23	-0.34	0.44	0	-0.38	-0.14	-0.29		-0.41	-0.54	-0.23	0.75	-0.31	0.19	-0.07	-0.03	-0.64	0.0
6147	YPR198W	-0.27	-0.17	0.18	-0.32	-0.03	0.54	-0.3		-0.28	-0.19	-0.23	-0.15	0.09	0.4	0.21	-0.01	-0.06	-0.3
6148	YPR199C	0.3		0.42	0.41	0.3	-0.51	0.14		-0.04	-0.09	0.1	0.64	0.29	0.07	-0.11	-0.2	-0.27	-0.2
6149	YPR200C			-0.69						0.29	0.28	-0.32	-0.15	0.12	-0.23	-0.01	0.25	1.01	2.0
6150	YPR201W					0.44				-0.33	-0.62	-0.64	0.82	0.53	0.34	-0.05	-0.3	-0.56	0.3
6151	YPR202W	-0.92	-1.84	-1.06	-1.03	-0.76	0.45	-0.4		-0.19	0.08	-0.15	-0.78	-0.43	-0.25	0.02	-0.03	-0.07	-0.2
6152	YPR203W	-0.58	-1.4	-1.06	-0.67	-0.69	-0.2	-0.27	-0.56	-0.22	-0.21	-0.12	0.16	-0.27	-0.12	-0.15	-0.07	-0.6	-0.2
6153	YPR204W	-0.49	-0.22	-0.01	-0.04	0.21	0.68	0.1		-0.16	0.04	-0.34	-0.68	0.2	0.14	0.12	-0.29	-0.4	-0.2
6154																			

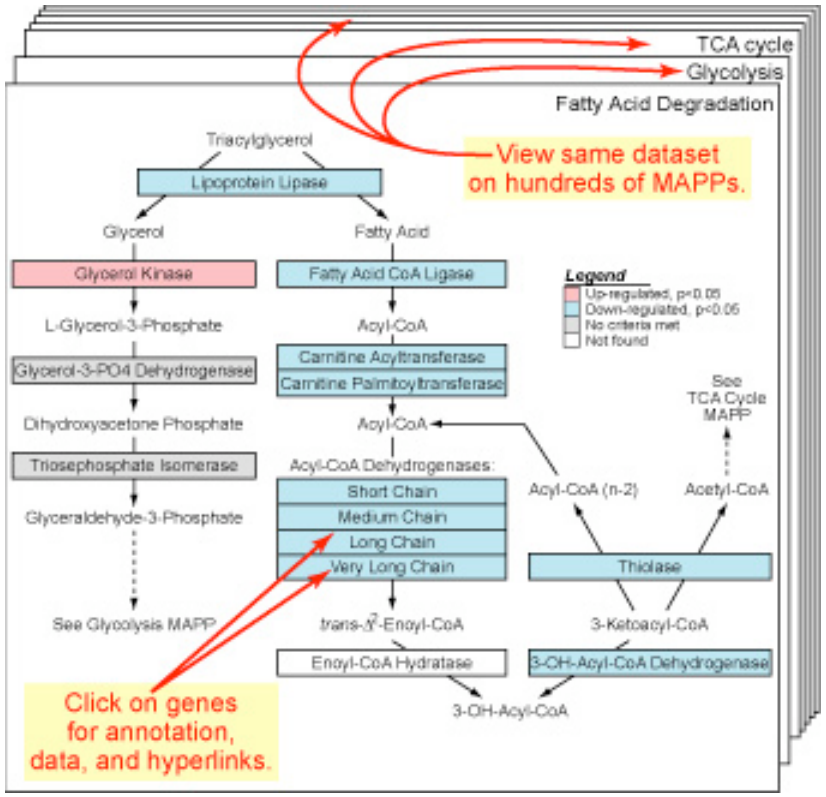
How GenMAPP Works

http://www.GenMAPP.org

- **Graphics tools make MAPPs that store gene IDs and vector coordinates for all graphical objects**

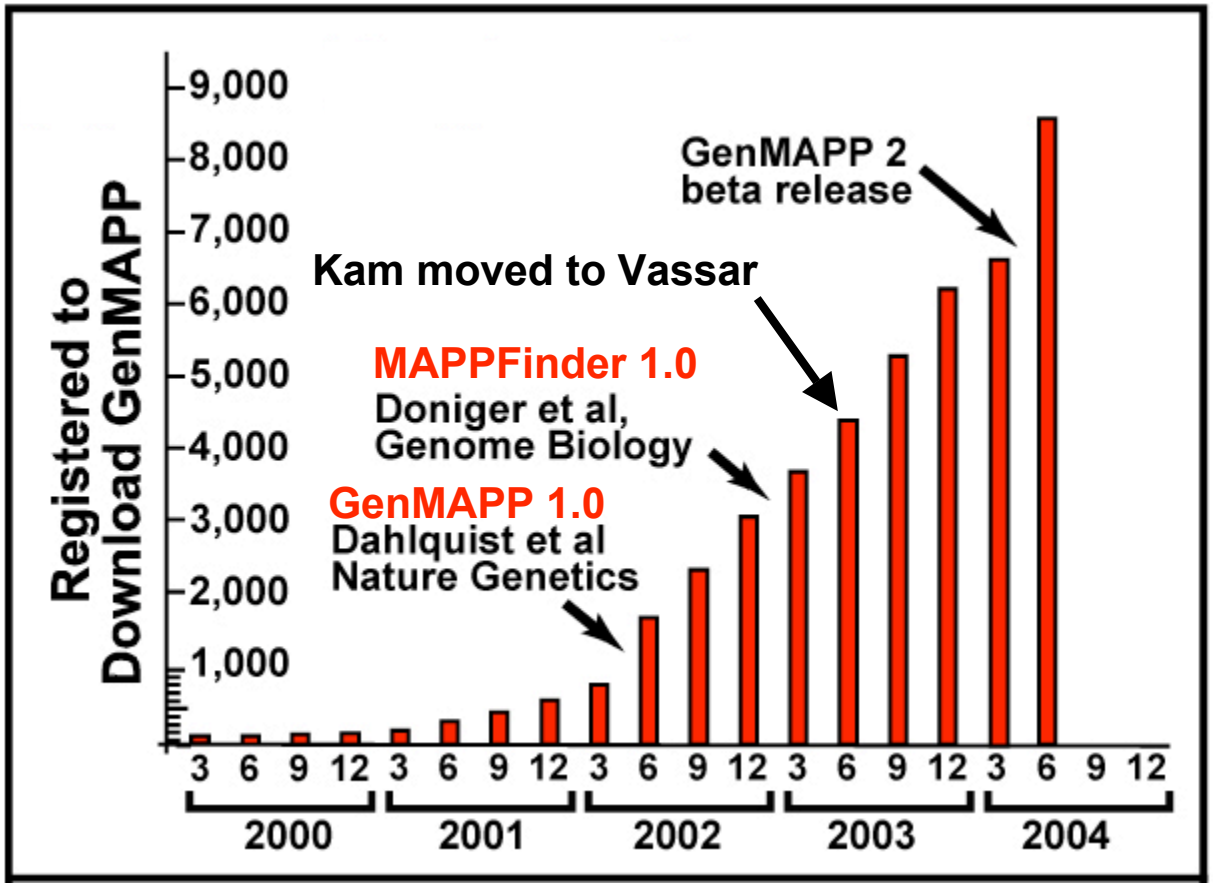
- **Separate Expression Dataset files store data and color-coding instructions**

- **Gene Databases store IDs, annotation, and hyperlinks to public gene and protein databases**



- **Stand-alone program implemented in Visual Basic, accessory files are Microsoft Access databases**

Number of Registered Users at www.GenMAPP.org



GenMAPP 2.0 release 12/2004

Kam moved to LMU 7/2005

Kam & Dondi Start collaborating 1/2006

Maintaining and Updating GenMAPP Gene Databases has been a Bottleneck for Development

- **Microarrays use different gene ID systems for annotation; users want as much information as possible.**
- **We need to capture and reliably relate gene data from different sources and keep the data updated.**
- **Gene Database design is data-driven; it tells GenMAPP what gene ID systems and relationships are present.**
- **Current GenMAPP Gene Databases are built from Ensembl as the main data source.**
  - **-- limited to (mostly) animal species**
  - **-- sensitive to changes in flat file formats**

Representative Sequenced Genomes

Species	Genome size (millions of b.p.)	Estimated # of genes
Haemophilus influenzae (bacteria)	1.83	1,743
Escherichia coli (bacteria)	4.6	4,288
Saccharomyces cerevisiae (yeast)	12	6,340
Drosophila melanogaster (fruit fly)	180	13,600
Caenorhabditis elegans (worm)	97	19,000
Arabidopsis thaliana (mustard plant)	130	*26,000*
Mus musculus (mouse)	3000	*30,000*
Homo sapiens (human)	3000	22,000

XMLPipeDB: A Reusable, Open Source Tool Chain for Building Relational Databases from XML Sources

Requirements:

- to create Gene Databases for other species (bacteria/plants) using UniProt as the main data source
- to be robust to changes in source file formats
- to use XML sources wherever possible
- to take advantage of existing open source tools
- to limit the manual manipulation of the data

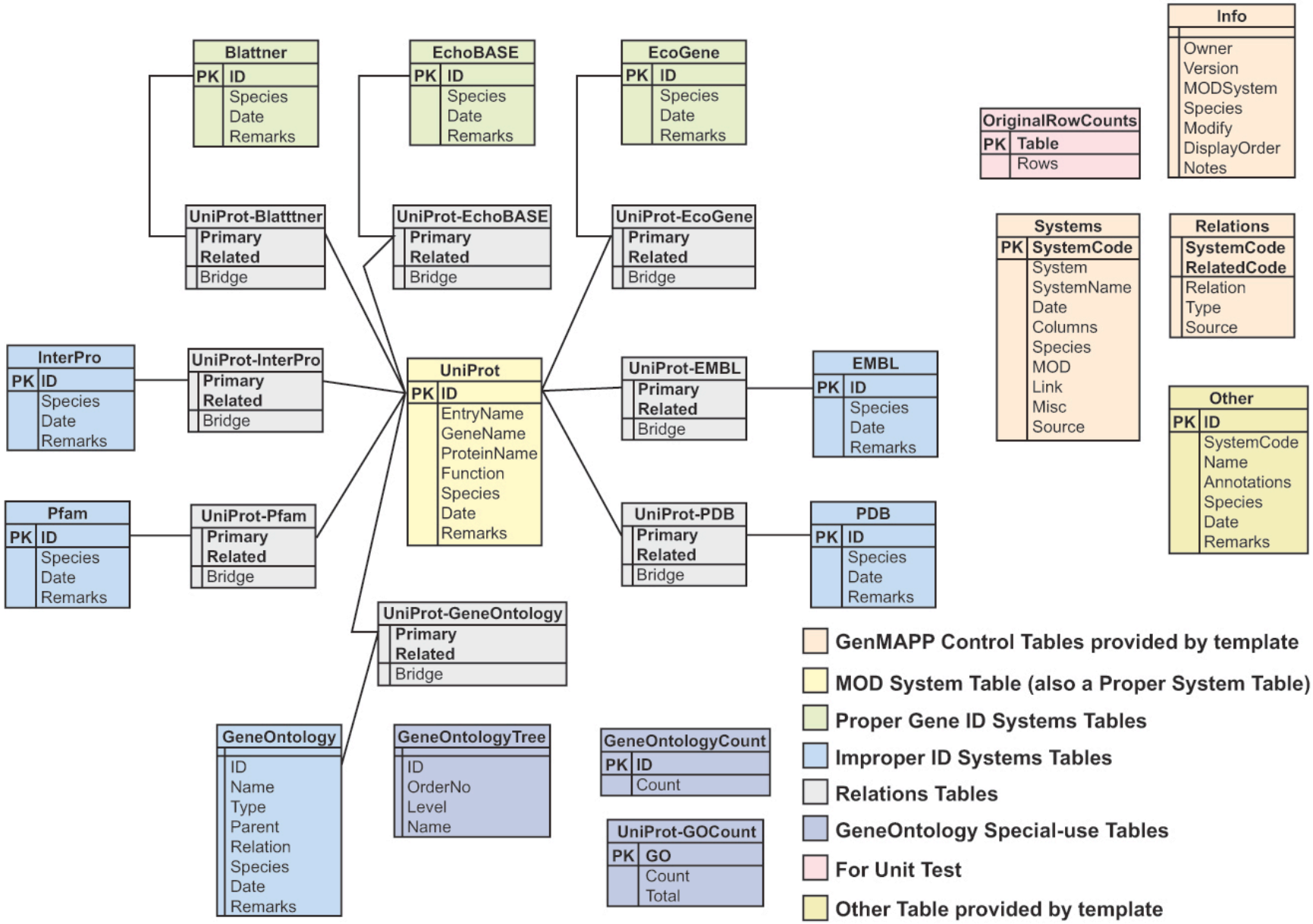
XMLPipeDB: A Reusable, Open Source Tool Chain for Building Relational Databases from XML Sources

Requirements:

- to create Gene Databases for other species (bacteria/plants) using UniProt as the main data source
- to be robust to changes in source file formats
- to use XML sources wherever possible
- to take advantage of existing open source tools
- to limit the manual manipulation of the data

First task was to build a GenMAPP Gene Database for *Escherichia coli* K12

GenMAPP Gene Database Schema for Escherichia coli K12



NOTE: Some Relations tables are not shown. All possible pairwise Relations tables exist between Proper ID systems and between Proper and Improper ID systems, but not between Improper ID systems (i.e., Proper-Proper, Proper-Improper, but NOT Improper-Improper).

Data Sources Required for a "Minimal" GenMAPP Gene Database

UniProt

- UniProt complete proteome sets for many species are made available as XML downloads by the Integr8 resource

Gene Ontology

- OBO XML format

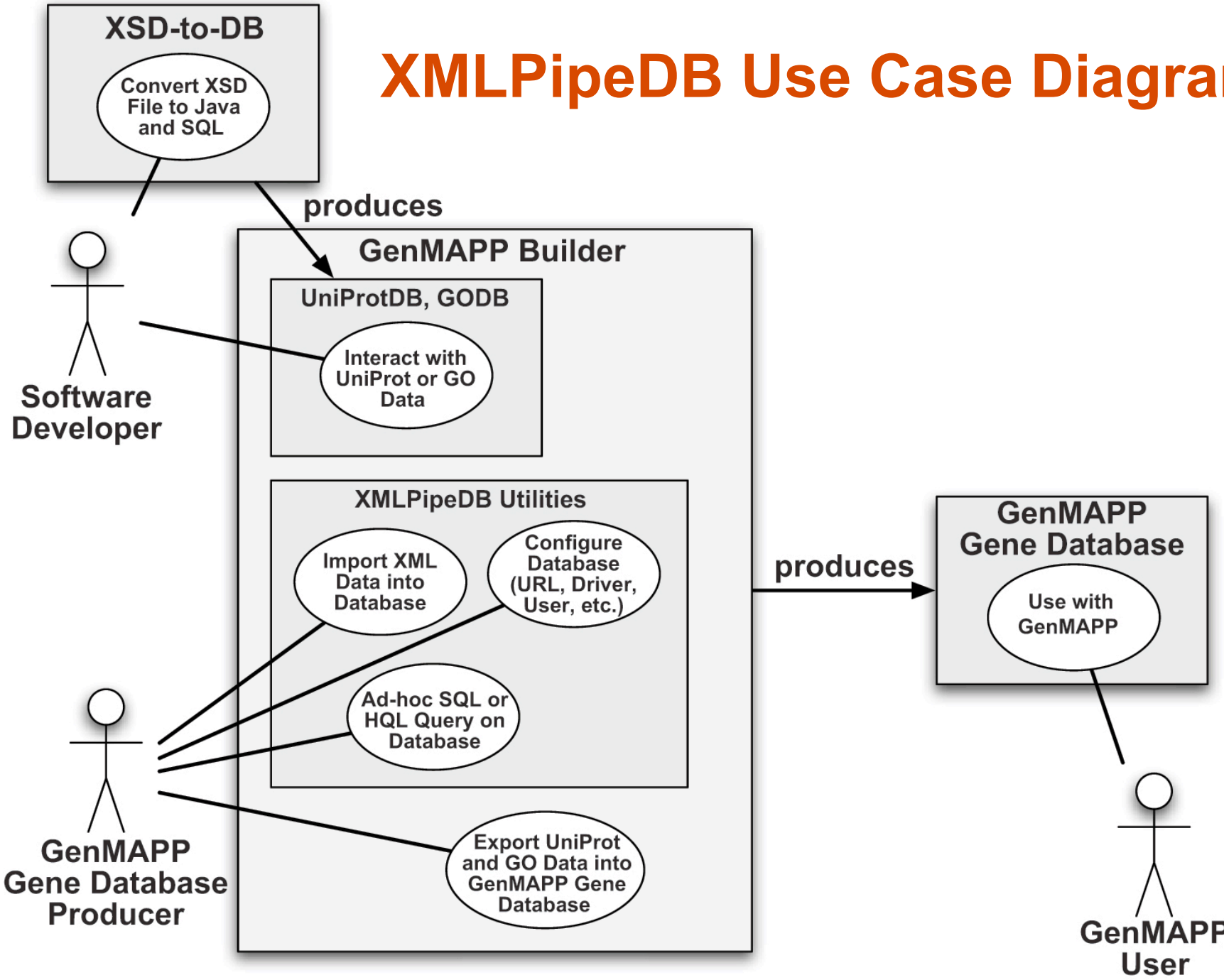
UniProt to GO associations

- GOA downloads also available at Integr8

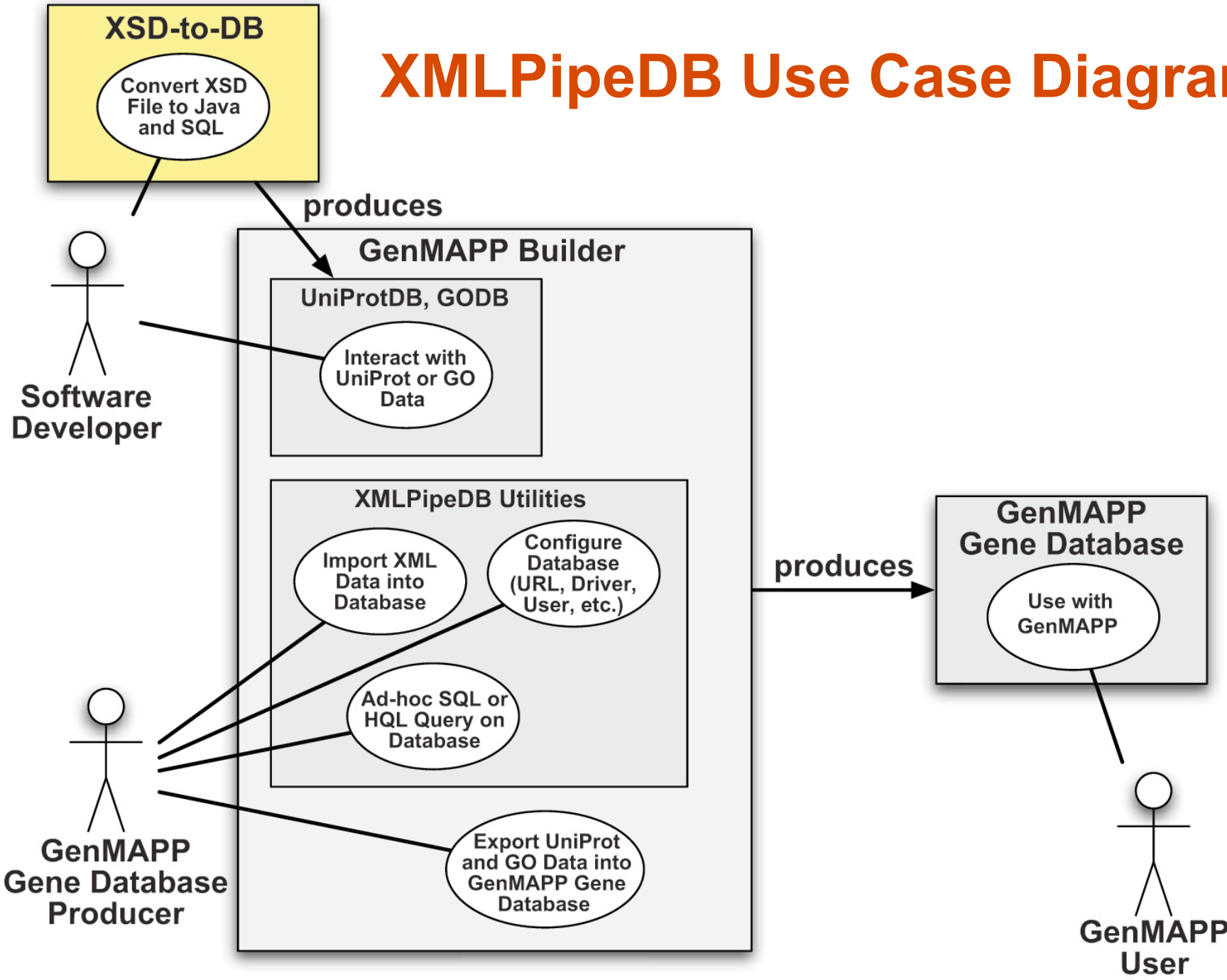
Outline

- Process
  - -- open source pedagogy
  - -- interdisciplinary collaboration
- Motivation
  - -- flood, deluge, tsunami (!) of genomic data
  - -- project requirements
- **XMLPipeDB: an open source tool chain for building relational databases from XML sources**
- Future Directions

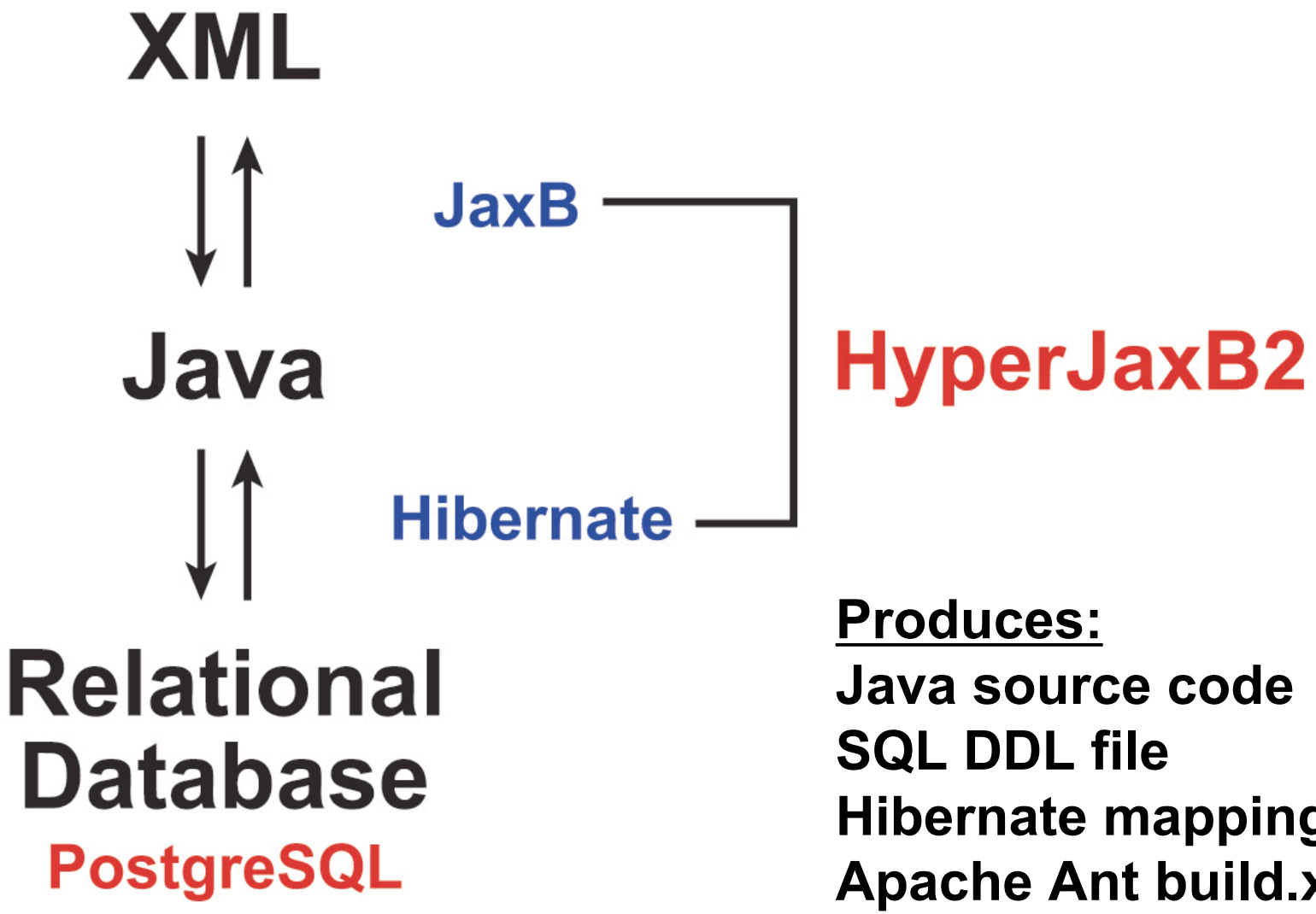
XMLPipeDB Use Case Diagram



XMLPipeDB Use Case Diagram



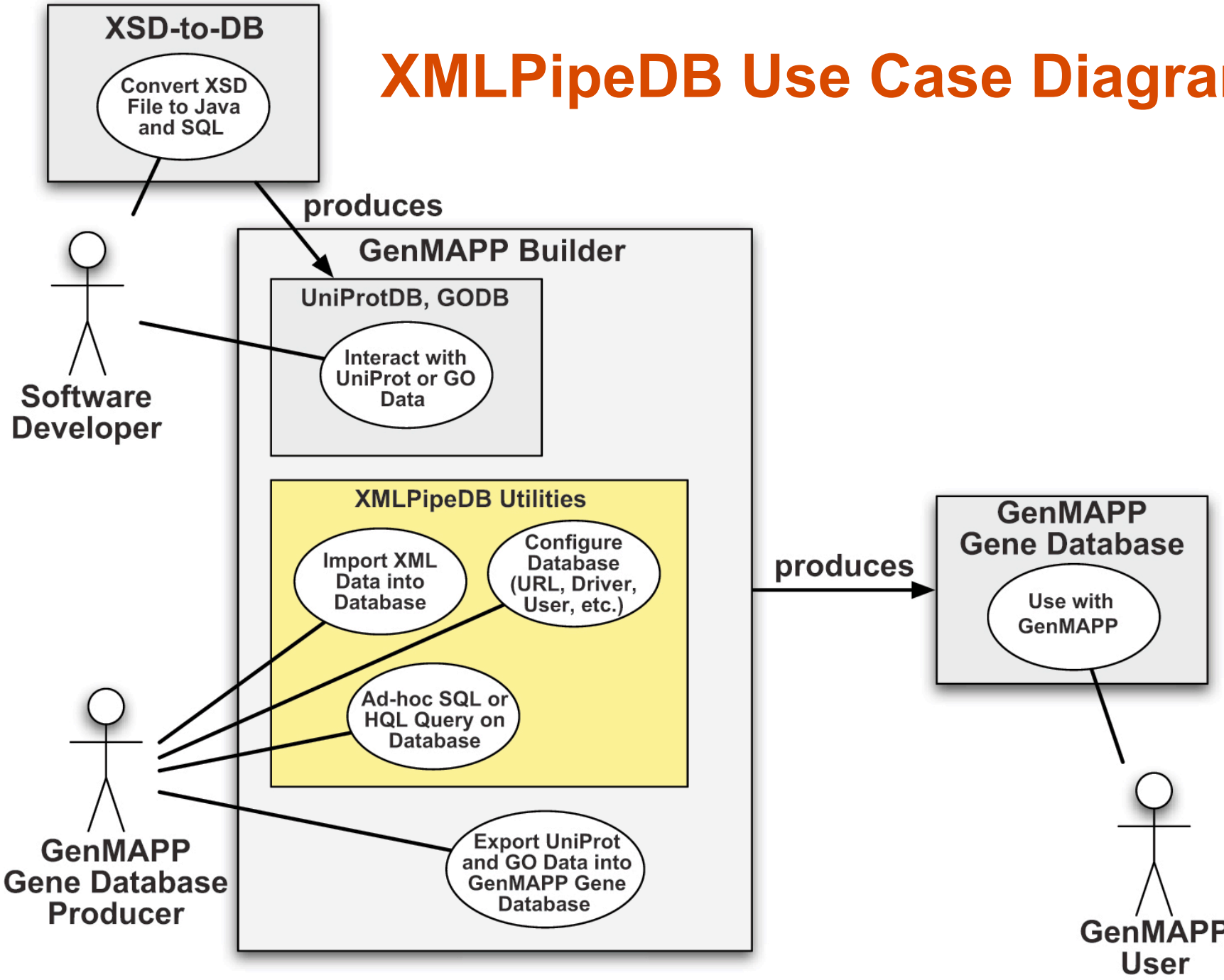
XSD-to-DB Stands on the Shoulders of other Open Source Tools



Produces:

Java source code
SQL DDL file
Hibernate mappings
Apache Ant build.xml

XMLPipeDB Use Case Diagram

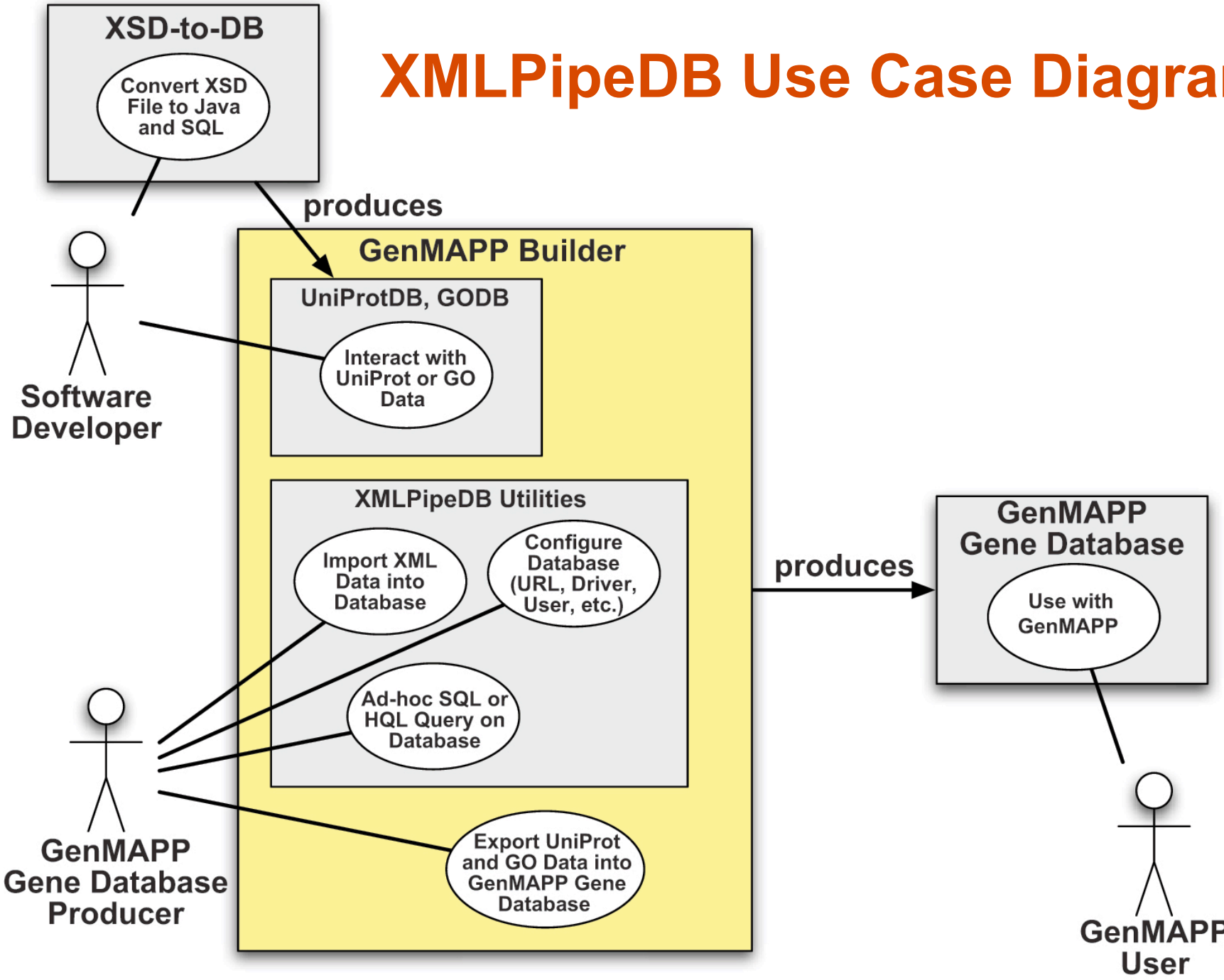


"Rule of Three"

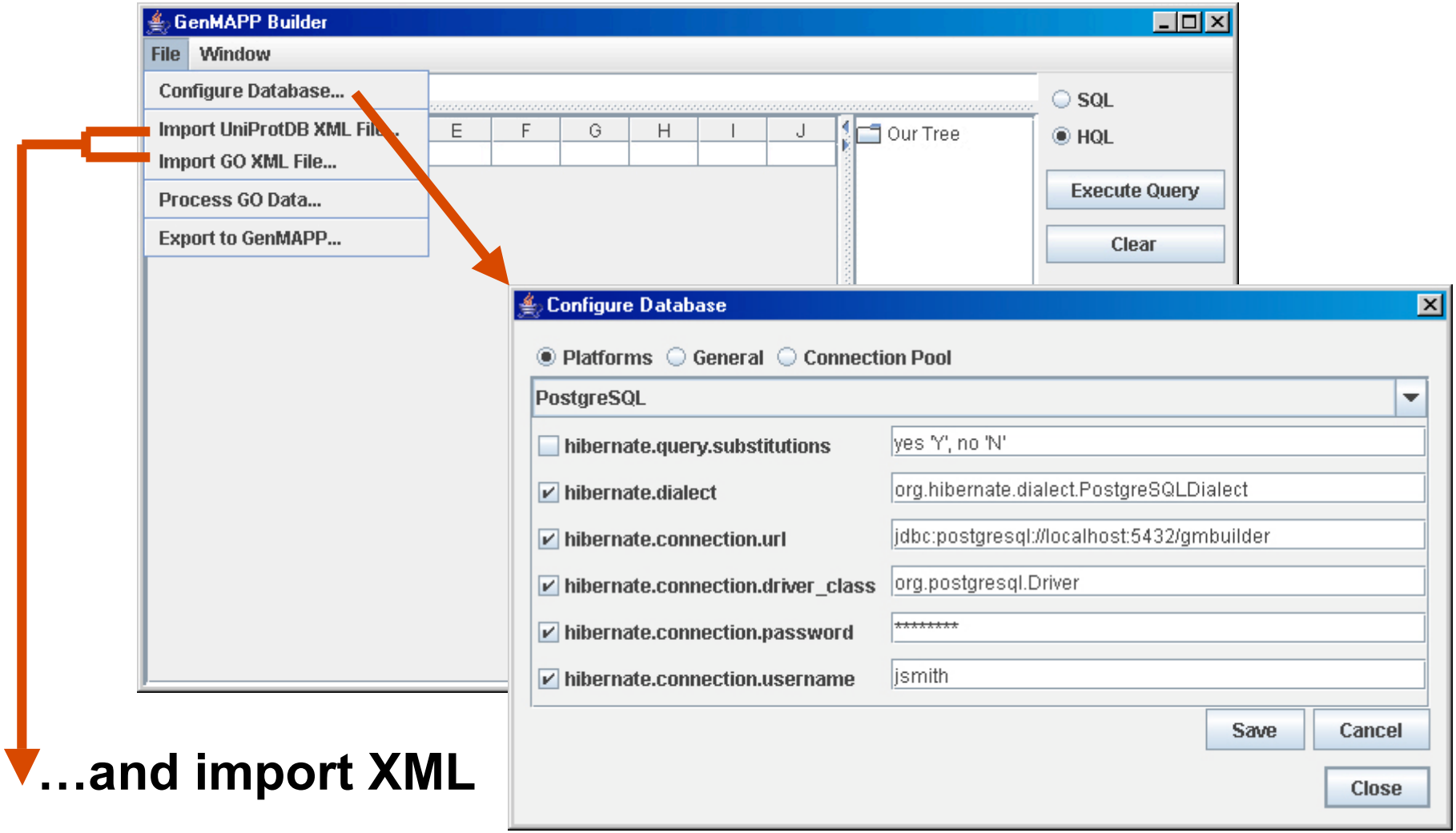
XMLPipeDB Utilities Library is a Suite of Java Classes that Provide Functions Common to Most XMLPipeDB Database Applications

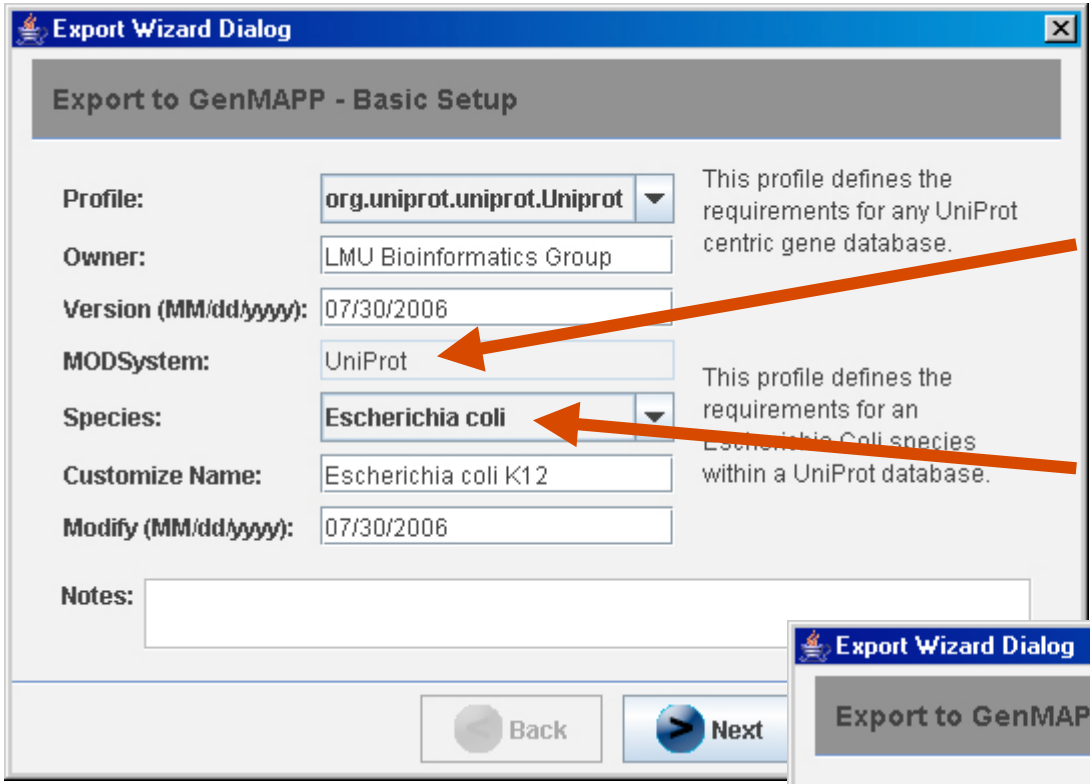
- **Loading of XML files into Java objects**
- **Saving XML-derived Java objects to a relational database**
- **Rudimentary query and retrieval of Java objects from the relational database**
  - **-- HQL (Hibernate Query Language), SQL query**
  - **-- object browser that shows results of query**
- **Configuring a client application to communicate with a relational database**

XMLPipeDB Use Case Diagram



GenMAPP Builder Uses the XMLPipeDB Utilities Library to Configure the PostgreSQL Database

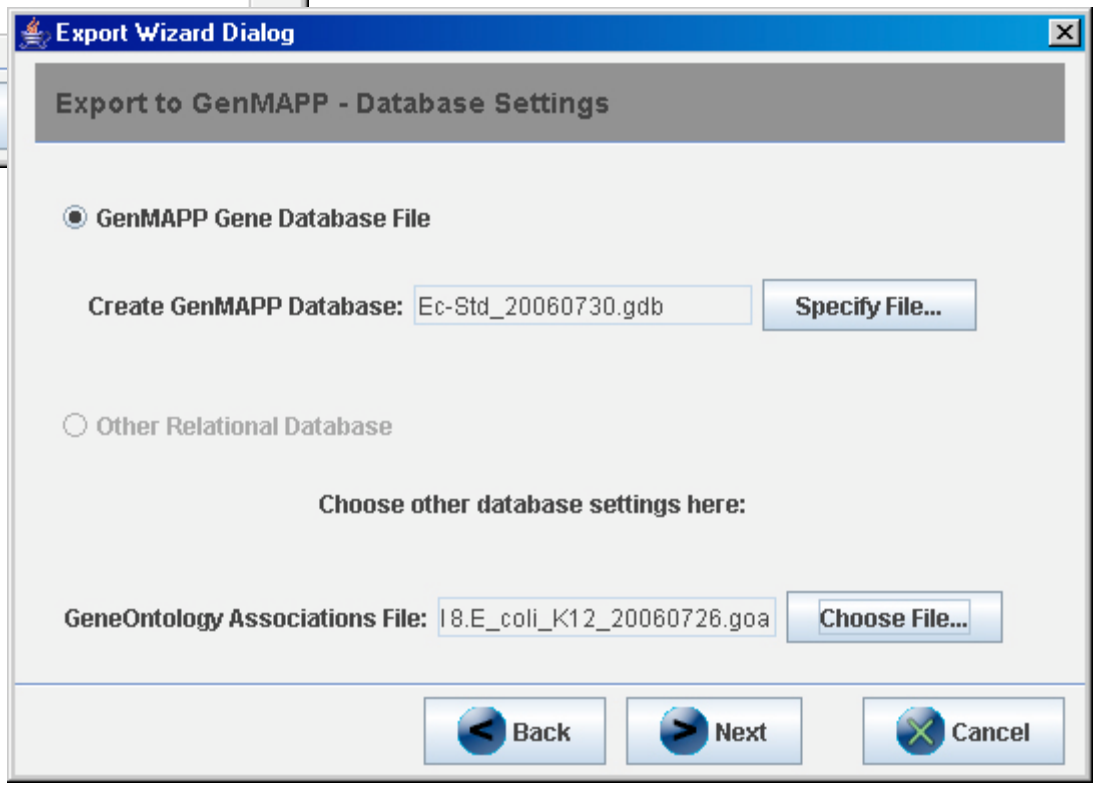


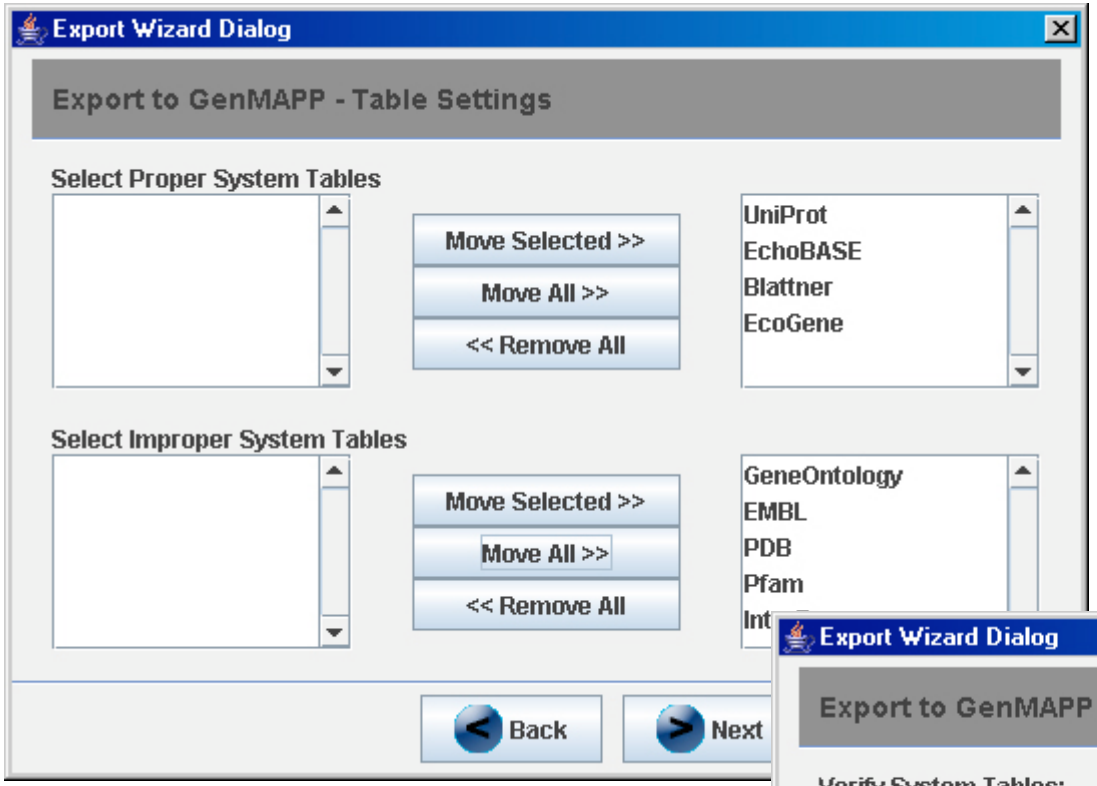


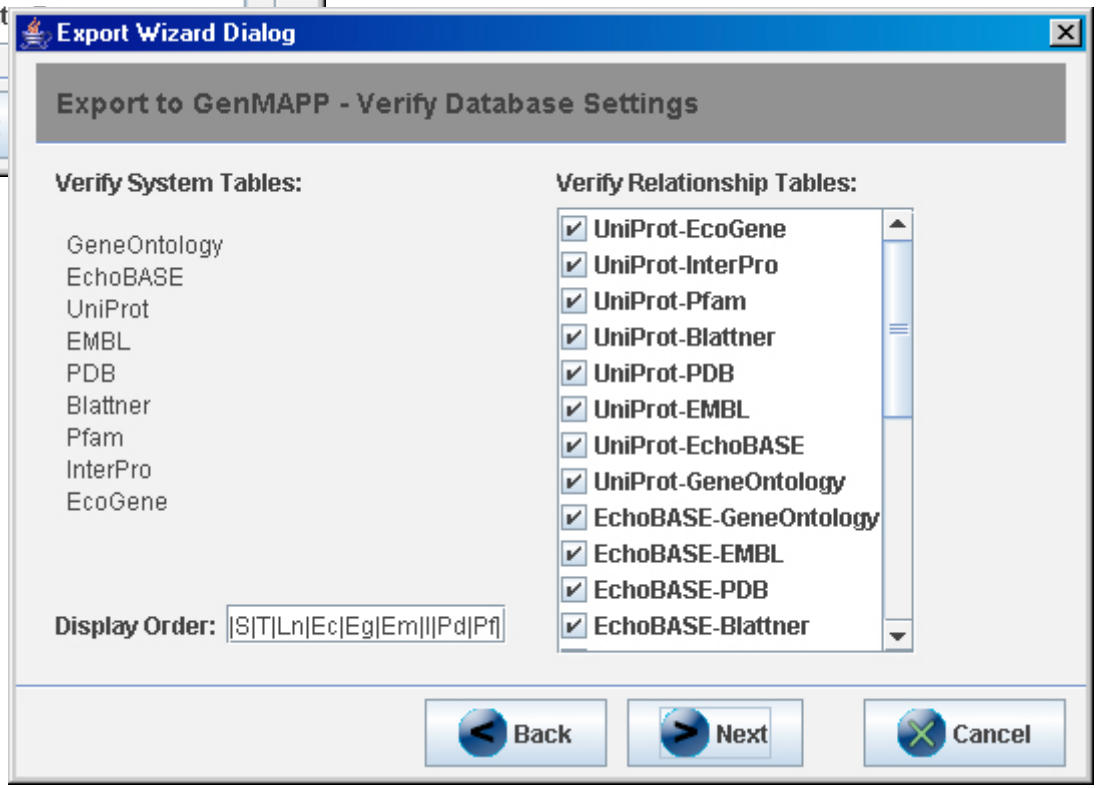
GenMAPP Builder Has Customized Profiles

-- for each Primary Data Source (e.g. UniProt)

-- for each species (e.g. *Escherichia coli*) based on TaxonID







The User Chooses Which Gene ID Systems and Relations to Export to the Gene Database

GenMAPP Gene Database for *Escherichia coli* K12 Was the First Milestone for XMLPipeDB

- **Loading the XML files into the PostgreSQL database took approximately 20 minutes**
  - **--UniProt XML (44 MB)**
  - **--GO XML (13 MB)**
- **Export of the Gene Database took approximately 2 hours**
- **Data integrity was checked by hand**
  - **--all 4329 records from UniProt were successfully exported to the Gene Database**
  - **--our Gene Database is missing 219 Blattner IDs**
  - **--the missing IDs were not present in the UniProt XML**

157 RNA genes
1 origin of replication
51 protein coding sequences
10 no feature designation

The Next Challenge is to Create a Gene Database for the Plant, *Arabidopsis thaliana*

- **The Arabidopsis UniProt proteome set has 34,304 proteins**
  - **--UniProt XML file is 228 MB**
  - **--order of magnitude larger than *Escherichia coli***
- **GenMAPP Builder failed to import the large XML file**
  - **--the file had to be broken into smaller individual files**
- **Export of the Gene Database took approximately 30 hours**
- **Need an automated solution for data integrity check of 34,304 proteins**
- **Need to automate the entire process of import/export/data integrity check**

Take-home Messages

- **Used an Open Source paradigm for Master's level course, resulting in useful bioinformatics software**
  - **-- software is NOT perfect, but acceptable for now**
  - **-- students will flow in and out of the project**

Take-home Messages

- **Used an Open Source paradigm for Master's level course, resulting in useful bioinformatics software**
  - **-- software is NOT perfect, but acceptable for now**
  - **-- students will flow in and out of the project**

- **GenMAPP Builder can make Gene Databases for any species represented in UniProt**
  - **-- produced a Gene Database for *Escherichia coli* K12**
  - **--Gene Database for *Arabidopsis thaliana* is in progress**

Take-home Messages

- **Used an Open Source paradigm for Master's level course, resulting in useful bioinformatics software**
  - **-- software is NOT perfect, but acceptable for now**
  - **-- students will flow in and out of the project**
- **GenMAPP Builder can make Gene Databases for any species represented in UniProt**
  - **-- produced a Gene Database for *Escherichia coli* K12**
  - **--Gene Database for *Arabidopsis thaliana* is in progress**
- **XMLPipeDB is a general set of tools that can be re-used for other bioinformatics and non-bioinformatics applications**
  - **-- LGPL license**
  - **-- *we have not experienced a change to an XSD yet***

Future Directions for XMLPipeDB

Near Term:

- Complete the Arabidopsis Gene Database
- Clean-up internal design for a generic species database
- Further automation
  - --data integrity checking (Tally Engine)
  - --building databases on a regular schedule (GMB Cruizer)
- Add data sources (TIGR CMR, NCBI Gene, Affymetrix)

Future Directions for XMLPipeDB

Near Term:

- Complete the Arabidopsis Gene Database
- Clean-up internal design for a generic species database
- Further automation
  - --data integrity checking (Tally Engine)
  - --building databases on a regular schedule (GMB Cruizer)
- Add data sources (TIGR CMR, NCBI Gene, Affymetrix)

Longer Term:

- Use XML sources to build MAPPs for GenMAPP
  e.g., KEGG-ML, BioPAX
- Applications that we haven't imagined yet

LMU Bioinformatics Group

http://xmlpipedb.cs.lmu.edu

LMU|LA
Loyola Marymount University

Kam D. Dahlquist

http://myweb.lmu.edu/kdahqui
kdahlquist@lmu.edu

John David N. Dionisio

http://myweb.lmu.edu/dondi
dondi@lmu.edu

XSD-to-DB
Adam Carasso
Jeffrey Nicholas
Scott Spicer

XMLPipeDBUtils
David Hoffman
Babak Naffas
Jeffrey Nicholas
Ryan Nakamoto

UniProtDB
Joe Boyle
Joey Barrett

GODB
Scott Spicer
Roberto Ruiz

GenMAPP Builder
Joey Barrett
Jeffrey Nicholas
Scott Spicer

Special Thanks
GenMAPP.org Development Group
Caskey L. Dickson, Wesley T. Citti
NSF CCLI Program (http://recourse.cs.lmu.edu)