Sapphire: a research project in scientific data mining

- The focus of Sapphire is three-fold
  - research in robust, accurate, scalable algorithms
  - incorporating the research into parallel, portable, software modules within a flexible system architecture
  - application of the software to practical problems
- Analyze data from
  - simulations
  - experiments
  - observations
- Details, including publications, at
  - http://www.llnl.gov/casc/sapphire
The Sapphire view of the end-to-end data mining process

Our software is designed to address the diverse needs of our applications

- Not all problems require the entire process
- Not all algorithms are suitable for a problem
- Algorithms typically depend on several parameters
- Intermediate data must be handled appropriately
- Domain dependent and independent parts must be clearly identified
- Should be able to accommodate a growing data set

⇒ Our system design enables us to conduct research and develop software in the context of several problems.
The Sapphire approach: a flexible, portable, scalable system architecture

Components linked by Python

- De-noise data
- Extract features
- Sample data
- Multi-resolution-analysis
- Dimension-reduction
- Sample features
- RDB
- Data items
- Features
- Decision Trees
- Neural Networks
- k-NN
- k-means
- Genetic algo.
- ... Display Patterns
- User Input & Feedback
- Components linked by Python

Sapphire software: Version 1.0.0 released in September 2001 (C++, serial version)

- Toolbox
  - Scientific Data Processing
    - Several Wavelets in 1,2,3-D
    - Single/double precision
    - 14 linear, non-linear filters
    - Wavelet denoising options:
      6 shrinkage rules
      4 shrinkage functions
  - Decision Trees
    - 7 Split Criteria
    - Split Finders
    - Pruning
    - Creation of decision tree
    - Application of decision tree
- Simple Classifiers
  - Naive Bayes
  - Gibbs
- Domain Information
  - FITS Data
  - VIEW Data
  - Regular Data
  - Feature Vectors
- Evolutionary Algorithms
  - Many options for selection, crossover, mutation, initialization, and replacement

Sapphire & Domain Software

- Public Domain Software

- Sapphire Software
“Non-mathematical” lessons learned ....

- Understanding the data and formulating the problem takes time
- Must not under-estimate time to
  - obtain access to the data
  - read, write, and display the data
  - bring the data into a consistent format
- Obtaining consistently labeled data is non-trivial
- Handling a growing data set can be tricky