Crowdsourcing and Bioinformatics

Paul Lu and Gane Wong
paullu@cs.ualberta.ca
University of Alberta
Canada
July 21, 2008
Acknowledgements

• Lois Banta and the other organizers
• Laura Frost
• Warren Gallin
• Janice Cooke
Overview

• What is crowdsourcing?
• What would crowdsourcing look like?
  – Illustrative example
• How would students and employers benefit?
Problem Statement(s)

1. How can some much-needed software be developed?
2. How can undergraduates (and others) be engaged and learn something about genomics, bioinformatics, and software?
3. How can students and employers benefit from #1?
Crowdsourcing

• Put simply: Motivating and harnessing (external) people to help solve a problem.

• For money:
  1. Netflix Contest: Movie recommender. $1 million prize.
  2. VMware Contest: Build VMs. First Prize $100,000. My team won Second Prize.
  3. X-Prizes in spaceflight, genomics, etc..

• For glory or the love of it:
  1. Open-source software development
  2. Wikipedia
Research vs. Non-Research

• Some problems require Ph.D.-level work, but not all
  – e.g., whole genome shotgun assembly is research

• But, many problems involve automating and customizing how data is gathered, manipulated, and analyzed and then presented.
  – e.g., automation of BLAST searches, or other tools
  – Many biologists and genome centres are bottlenecked on custom data processing (i.e., non-Ph.D. problems)
How does this apply to us?

• Think in terms of **course assignments**
  – Students can learn something and the work gets done

• Engage students and others to **write custom software**
  – Give credit for design, programming, testing, etc.
  – Focus on non-Ph.D. problems

• Engage students and others to **analyze biological data**
  – Similar, known analyses need to be performed on newly generated data
Current Problems

• My Personal Theory:
  – The “bar for entry” to most open-source projects is too high. Sink or swim.
  – The quality of work done on most projects is too low.

• Solution?
  – Problem definition, design, architecture, management, and quality control must be valued, along with programming.
  – Make it possible for biologists and computer scientists to work together

Design, Specification

Programming, Testing
Ultimate Goal

• Wiki-like Web site as repository for software.
• Architecture, design, and discussion integrated to Web site
• Automated testing (e.g., Fitnesse)
• Credit given for all aspects of the project.
• Contributions viewable for each participant, like a CV.
Ex: Remove Duplicates (1)

- Gather data from Sources A and B
- Find all sequences similar to another sequence in the database
- Remove these sequences, if appropriate
Ex: Design and Architecture

• Again, think in terms of course assignments…

• Input specification:
  – Get data from Source A using `wget`, already in FASTA
  – Get data from Source B using `wget`, then convert file format to FASTA

• Output specification:
  – Output in FASTA
  – Annotate each protein sequence with source information
  – Mock ups of data to clarify the specifications
Ex: Mock-Ups of Files

> Protein A
MGVTT
> Protein B
MTGVSM

Input

> Protein 1
MGVVT
> Protein 2
MTGSTTV

Input

> Protein A, Source=A, July 21, 2008
MGVTT
> Protein B, Source=A, July 21, 2008
MTGVSM
> Protein 2, Source=B, July 20, 2008
MTGSTTV

Output
Submission and Testing

• Via Web, code is submitted
• Code is automatically tested with other parts of the software pipeline
  – Programmer previously downloaded the entire pipeline inside a virtual machine
  – c.f. www.fitnesse.org
• Entire software pipeline is automatically tested as each new component is added
Why would a student join in?

• Contributions to design, programming, and testing are recorded (as with Wikipedia)
• An automatic CV can be generated for each contributor
• Many of the problems are as “easy” (aka well-formed) as course assignments
How would employers view it?

• Meritocracy: Students can document their “crowdsourcing credentials”, in great detail
  – Design
  – Programming
  – Testing
  – Management

• The work has more diversity and appeal than typical “summer jobs” with researchers

• If employers valued these credentials, then more people will participate…
What’s next?

• Get feedback from you, students, employers
• Some software infrastructure must be built
  – Nothing has been built yet
  – We could use crowdsourcing to do this initial work too
  – Some resources will be required
• Need to identify a good pilot project
• Need to deliver results