The Cartwheel Project

(\textit{Python}) \textit{Tools for Regulatory Genomics}

C. Titus Brown
Caltech (now)
Michigan State (soon)
ged.cse.msu.edu
Outline

1. Introduction to regulatory genomics
2. A bioinformatics “in the medium” solution.
3. Digression: project maintenance.
Fruit fly: 20,000 genes

Worm: 20,000 genes

Sea urchin: 22,000 genes

Most vertebrates, including humans: 25,000-45,000 genes

Unknown, but many genes in common, esp with vertebrates
You can build a cathedral, ...
...and a train station, out of much the same material.
...because it’s the *architectural plans* that make the difference.

As with buildings, so with bodies: the developmental blue prints for animals are stored in our genomes, in both the genes and the **regulatory regions** that control gene expression.

Finding these regulatory regions is important.
Anatomy of a gene regulatory network node

Inputs
(signal)

Signalling cofactor
Repressor
Amplifier

Outputs
Gene / Transcript

Effect

Gene / Transcript

Transcription
Transitioning to a developmental gene network: connecting the genes
Endomesoderm Specification to 30 Hours

Maternal Inputs

Maternal G-cadherin
Mat SoxB1
Mat Otx

Mat cB
Mat Notch

EcnS

Unkn veget activ

Unkn mes activ

Unkn mes activ

S(U(H)

S(U(H):Nc

Zyg. N.

Gata (oral)

Germ (abo)

Endoderm

Endomesoderm

Mes

Skel

Veg1 Endo

Sm27
Sm50 Msp130 Msp-1

Ubq 1

Sm30 G-cadherin Ficolin Cyp

FvMo1,2,3

Decorin

Apopexc Gelsolin

OrCT Kapapo

OrCT Kapapo

Endo16

Mes

Ski

Not

VegF

Eve

Hox11/13b

FoxA

GataE

Krl

Brnl/2/4

FoxB

Bra

GataE

Hox11/13b

FoxA

Eve

Hox11/13b

GataE

Krl

Brnl/2/4

FoxB

Bra

GataE

Hox11/13b

FoxA

Eve

Hox11/13b

GataE

Krl

Brnl/2/4

FoxB

Bra

GataE

Hox11/13b

Copyright © 2001–2006 Hamid Bolouri and Eric Davidson

Ubq = ubiquitouk; Mat = maternal; activ = activator; rep = repressor;
unkn = unknown; Nucl = nuclearization; x = b-catenin source;
S(U(H):Nc = nuclearized b-catenin-Tcf1; ES = early signal;
ECNS = early cytoplasmic nuclearization system; Zyg. N. = zygotic Notch
cis-Regulatory regions connect upstream genes to their regulators.
For example, this regulatory region...
Eukaryotic transcriptional initiation involves many general factors, as well as specific enhancers.
...drives this tissue-specific gene expression in the sea urchin embryo.
What are we looking for?
What are we looking for?

- **Regulatory regions** consist of multiple binding sites that function combinatorially.
- **Binding sites** are 8-30 bp segments of DNA to which transcription factors bind specifically.
Why is this a hard problem?

- Genomes are *big*, regulatory regions are *small*.
  40-80 kb of DNA *per gene*; ~500 bp regulatory regions…

- No obvious statistical signature, unlike protein coding regions.

- No good way to *test* predictions without doing experiments (slow, expensive, difficult).
Comparative sequence analysis

- Look for regions conserved between two or more species.
- Basic logic: if a given gene is expressed in similar places in two organisms, molecular mechanisms (including *sequence*) driving expression may not have diverged.
for i in range(0, len(seq1) - w):
    for j in range(0, len(seq2) - w):
        m = count_identity(seq1[i:i+w],
                           seq2[j:j+w])
        if m >= threshold:
            record_match(i, j, m)

# forget reverse-complement for now
Family Relations / pair view
Cartwheel

- A system that lets biologists:
  - establish sequence analyses with custom params;
  - run on them on someone else’s compute server;
  - visualize and interact with the results via a client GUI

- Aimed at *bench biologists*.
- Intended to be extensible.
Web browser

Client GUI

HTTP

XML-RPC

Cartwheel server

database connection

Batching & queueing system

Third-party software
(BLAST, LAGAN, blastz, paircomp, etc.)
Cartwheel technology (all OSS)

- **Server:**
  - Linux
  - Python, Quixote (1.x), SCGI
  - PostgreSQL db, psycopg
  - Home-grown O/R adapter “cucumber”
  - Several subpackages (paircomp, motility) w/Python interfaces

- **Client:**
  - Started with Jython; switched to FLTK (C++). **Cross platform.**
  - Uses XML-RPC to communicate with server.
Comparative Sequence Analysis

This approach (not just Cartwheel...) has turned regulatory region search from a 2-yr technically difficult task into a 3-mo summer student project.
Cartwheel statistics

- Built progressively 2002-2004; several iterations.
- Server: 6k LoC (server); GUI: ~10k LoC
- ~600 total users, 50-100 labs, 30-40 institutions
- ~20 publications that used it
- 10-20 users/month

(Yes, scalability has never been an issue…)

---

**Cartwheel**

- **statistics**

  - Built progressively 2002-2004; several iterations.
  - Server: 6k LoC (server); GUI: ~10k LoC
  - ~600 total users, 50-100 labs, 30-40 institutions
  - ~20 publications that used it
  - 10-20 users/month

  (Yes, scalability has never been an issue…)

---
Digression: maintenance

- I’m the primary programmer (~95%).
- I do other things (experiments, PhD thesis, etc.)

Consequently,
- Web site became *very* fragile.
“Test-obsessed”

Need to test Web sites => built twill.
Need to run tests flexibly => use nose.
Need to target more tests => built figleaf.
Need to record Web tests => built scotch.

(yes, I got a little sidetracked…)
Automated testing rocks.

Probably ~30% of the Web app covered.
- unwarranted confidence in basic structure
- very easy to upgrade packages, transfer servers
  (~1 day to upgrade postgresql, switch to new version of Debian on a VMware server)
Sociological considerations

User population is computationally naïve.

Most tools in this area are either very simple or virtually impossible to use.

We provided a simple Web interface, an intuitive GUI interface, and a tutorial.

Biologists prefer making their own judgements.

…and they hate high false positives.
Do other programmers use it?

Provided simple interfaces (*XML-RPC*) for establishing, manipulating, downloading analyses.
Tried to make things easily extensible.
No documentation, of course.

Result: no one uses my code.

My small libraries are moderately popular (paircomp, motility) because they have a simple Python API.
Planning for the future

Technology update:
- I love Quixote for Web apps
- FLTK sucks (ugly!) => QT or KWWidgets (testing!)
- I hear Python is up to 3000 now??
- SQLAlchemy is a very nice O/R mapping system
- ExtJS javascript toolkit.
New functionality thoughts

Cartwheel originally built around targeted subsequencing of genomes.

Now we have a lot of big genomes sequenced!

⇒ Need to import information, interact with already-established alignments.

“for” loops don’t cut it -- move to pygr for genome manipulation/interaction?
New functionality thoughts

Motif searching is the new black.

Biologists want to search for binding sites (fuzzy 6-22 bp) and ask questions like:
- do these sites correlate well with conservation?
- do they cluster with themselves/others?
- are they near my favorite gene(s)?

Idea is to expand Cartwheel into a tool for regulatory genomics hypothesis generation and exploration.
Concluding thoughts

Write end-user software that works over software that is flexible.

Tutorials, ease of use are both really important (pick your target audience - devs? users?)

If you’re unlucky, your project will hang around for longer than you thought and you will have to think about maintenance.

Test.
Acknowledgements

Eric Davidson (old boss)
Marianne Bronner-Fraser (new boss)

Andy Cameron
Barbara Wold
Erich Schwarz
Tristan De Buysscher
Carolina Livi
Diane Trout