


The art of Programming

Evolutionary Algorithms

Juan-Julián Merelo.

@jjmerelo

U. Granada (Spain)



Get out and
smell the air

New is always better

Programming paradigms are changing on a **daily** basis

GPGPU

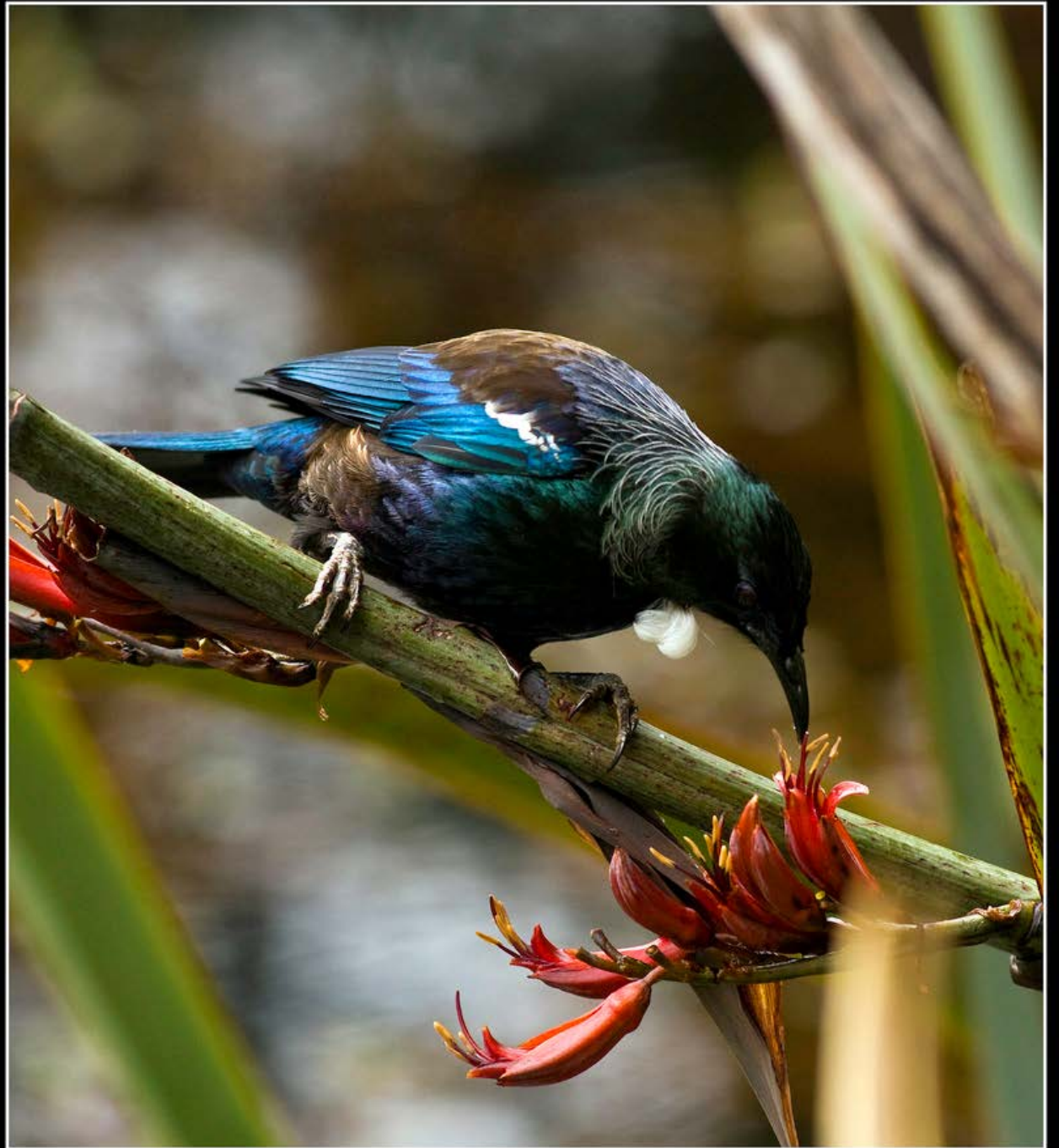
Cloud computing

NoSQL

Map/reduce

Internet of Things


Form
should fit
function



Tuí

And we
should
adapt to the
new





We should be able to write
publishable papers
artifacts painlessly
through efficient,
maintainable, scientific
programming

Mind your
environment



**Open source
your code and
data**



Aw, maaaaan!

- Open source first, then program
 - Scientific code should be born free.
- Science must be reproducible.
- Easier for others to compare with your approach
 - Increased H
 - Scientist heaven!
- Manifest *hidden* assumptions.

If you don't share you don't care!



Minimize bugs
via
test-driven
programming

Tests before code

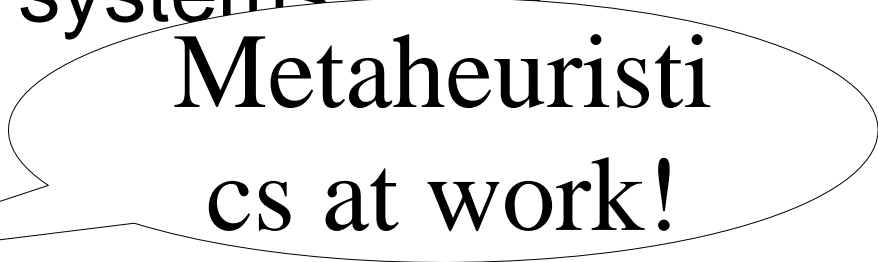
- What do you want your code to do?
 - Mutate a bit string, for instance.
- Write the test
 - Is the result from mutation different from the original?
 - Of course!
 - But will it be even if you change an upstream function?
Or the representation?
 - Does it change all bits in the same proportion (including first and last → corner cases)?



Control
the
source of
your
power

Source control systems save the day

- Source code management systems allow



Metaheuristics at work!

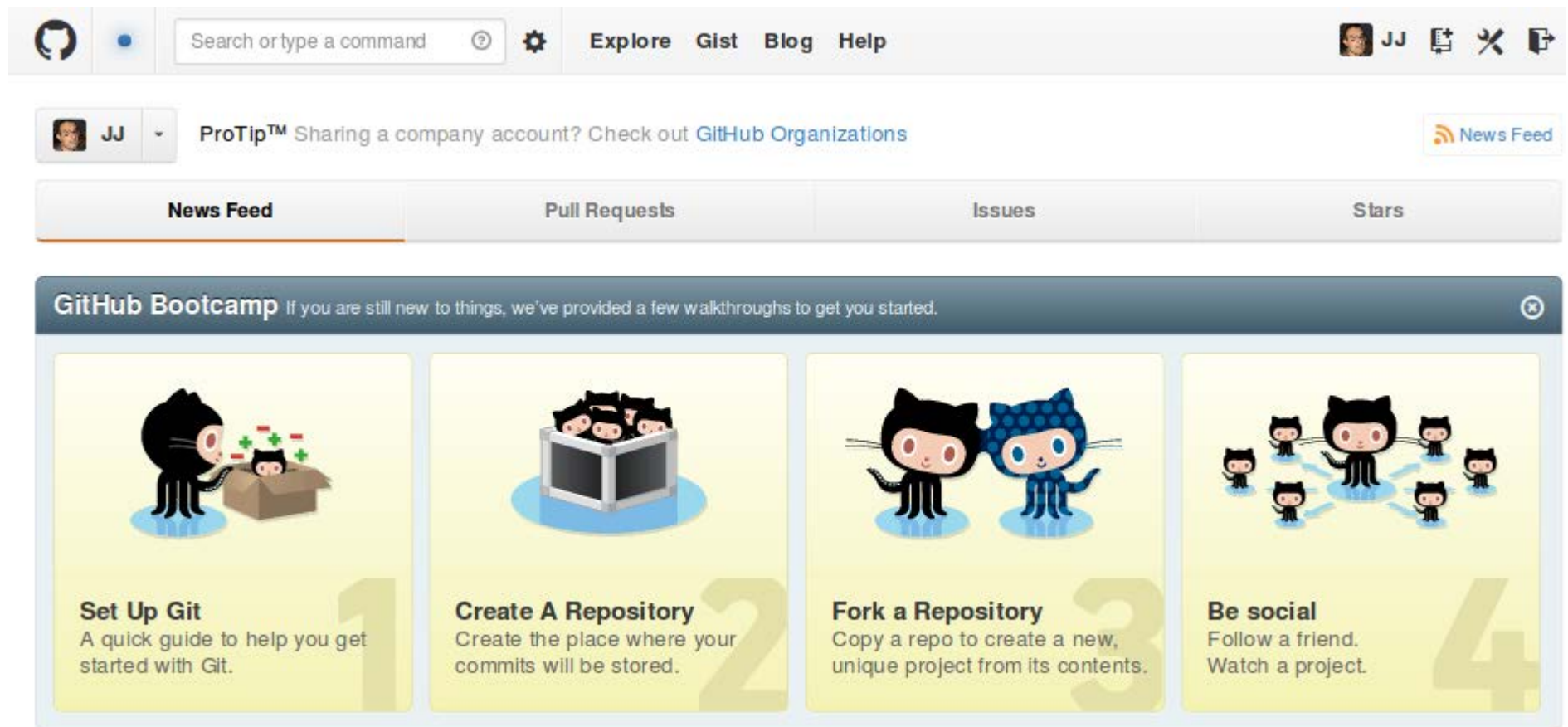
- Checkpoints
- *Stygmergic* interaction
- Individual responsibility over code changes
- Branches
- Distributed are *in*: git, mercurial, bazaar
- Centralized are *out*: subversion, cvs.
- Instant backup!

Code complete

- 1) Check out code/Update code
- 2) Make changes
- 3) Commit changes (and push to *central* repository)

Go with the Joneses

Use GitHub: <http://github.com>



The screenshot shows the GitHub homepage. At the top is a navigation bar with the GitHub logo, a search bar, and links for Explore, Gist, Blog, and Help. Below the navigation bar is a user profile section for 'JJ' with a 'ProTip™' message and a 'News Feed' link. The main content area features a 'News Feed' tab and a 'GitHub Bootcamp' section. The Bootcamp section contains four steps:

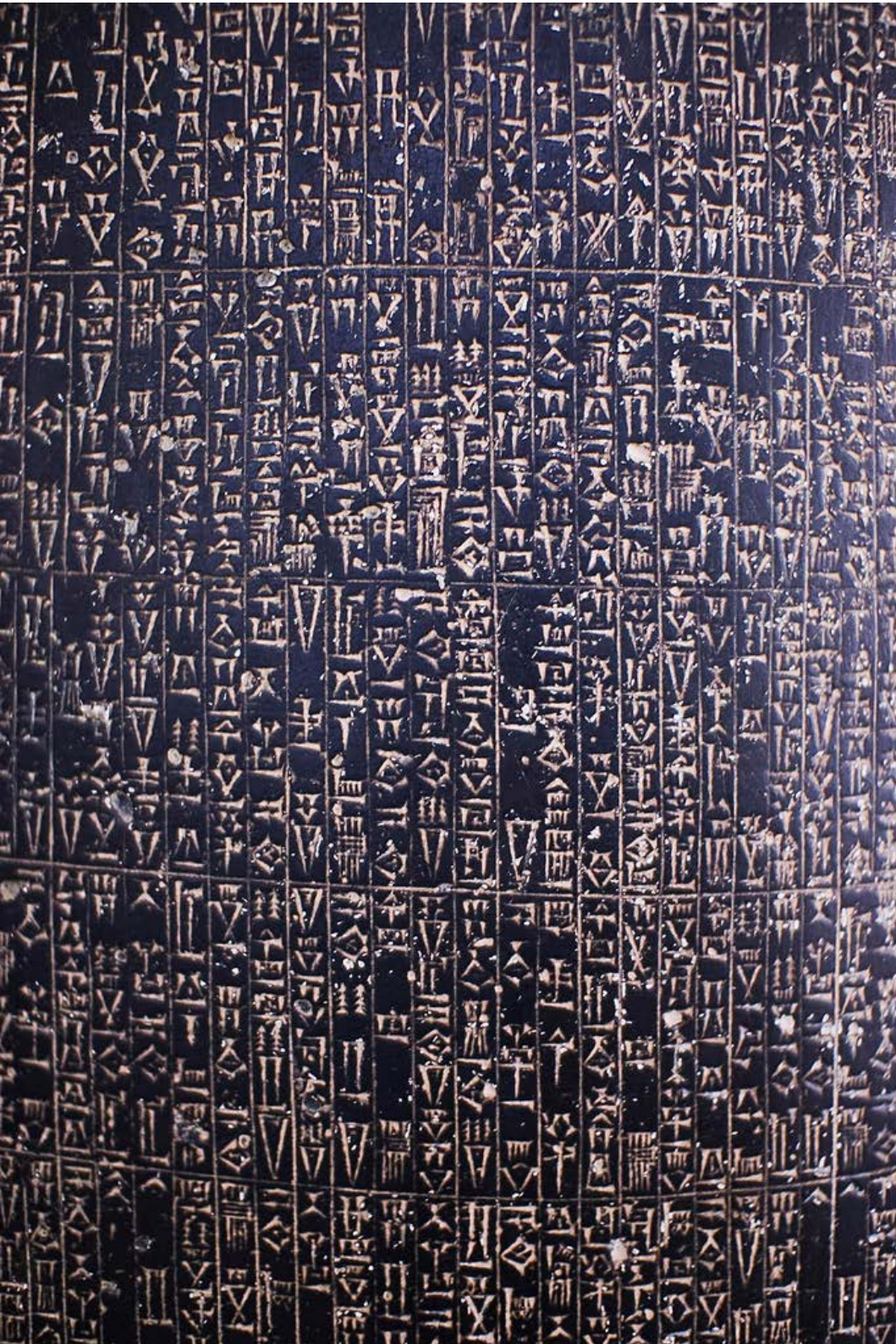
- Set Up Git**: A quick guide to help you get started with Git.
- Create A Repository**: Create the place where your commits will be stored.
- Fork a Repository**: Copy a repo to create a new, unique project from its contents.
- Be social**: Follow a friend. Watch a project.



Integrate

Pushing is not the end of the story

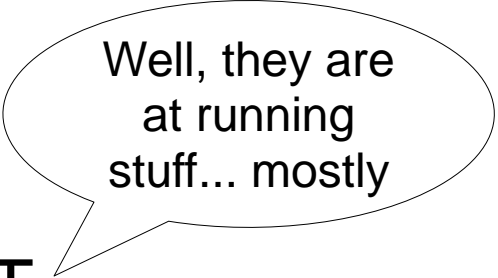
- Tests must be run, compilations made, checks and balances checked and balanced.
- Use Travis or Jenkins
 - If it's good enough for software developers, it's good enough for scientists!
- All this is free if you open source your code
 - Back to #2



Be language agnostic

Language shapes thought

- Don't believe the hype:
 - Compiled languages are faster... NOT
 - There is no free lunch.
- Avoid programming in C in every language you use
- Consider scripting languages: Python, Perl, Lua, Ruby, Clojure, Javascript... interpreted languages *are* faster.

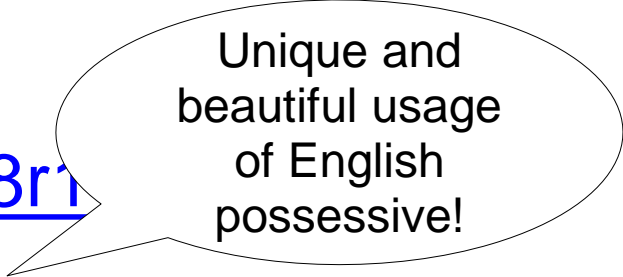


Well, they are
at running
stuff... mostly

Language agnoticism at its best

Evolving Regular Expressions for GeneChip Probe Performance Prediction

<http://www.springerlink.com/content/j3x8r1>



Unique and
beautiful usage
of English
possessive!

The regular expressions are coded in AWK scripts:

*Although this may seem complex, gawk (Unix' free
interpreted pattern scanning and processing*

Programming speed \gg
program speed



Scientists, not software engineers

- Our deadlines are for papers – not for software releases (but we have those, too).
- What should be optimized is speed-to-publish.
- Makes no sense to spend 90% time programming – 5% writing the paper.
- Scripting languages rock
 - and minimize time-to-publish.

Perl faster than Java?

Algorithm::Evolutionary, a flexible Perl module for evolutionary computation

<http://www.springerlink.com/content/8h025g83j0q68270/>

- Class-by-class, Perl library much more compact
 - Less code to write.
 - More time to write the paper, perform experiments....
- In pure EC code, Algorithm::Evolutionary was faster than ECJ.

Don't assume:



measure

Performance matters

- Basic measure: CPU time as measured by `time`

```
jmerelo@penny:~/proyectos/CPAN/Algorithm-  
Evolutionary/benchmarks$ time perl onemax.pl
```

```
0; time: 0.003274
```

```
1; time: 0.005438
```

```
[...]
```

```
498; time: 1.006539
```

```
499; time: 1.00884
```

Going a bit deeper:
profilers

Performance Profile Index

For run_experiment_instances.pl

Run on Wed Dec 8 09:43:05 2010
Reported on Wed Dec 8 09:55:31 2010

Profile of run_experiment_instances.pl for 166s (of 407s), executing 171728297 statements and 13126070 subroutine calls in 114 source files and 85 string evals.

Jump to file...

Top 15 Subroutines

Calls	P	F	Exclusive Time	Inclusive Time	Subroutine
2773830	4	2	35.7s	35.7s	Algorithm::MasterMind::check_combination
396600	1	1	26.9s	26.9s	Algorithm::Evolutionary::Wheel::first
198300	1	1	17.1s	43.9s	Algorithm::Evolutionary::Wheel::spin
405600	2	1	11.6s	12.2s	Algorithm::Evolutionary::Individual::Base::new
396600	1	1	10.8s	31.2s	Algorithm::Evolutionary::Op::String_Mutation::apply
396600	2	1	8.87s	9.38s	Algorithm::Evolutionary::Op::Base::check
699000	2	1	6.83s	40.7s	Algorithm::MasterMind::distance_taxicab
699000	1	1	6.27s	33.9s	Algorithm::MasterMind::matches
1593000	1	1	6.18s	27.6s	Algorithm::MasterMind::check_rule
396600	1	1	5.28s	17.2s	Algorithm::Evolutionary::Individual::String::clone
38	2	1	4.88s	165s	Algorithm::MasterMind::EvoRank::issue_next
661	1	1	4.47s	96.1s	Algorithm::Evolutionary::Op::Canonical_GA_NN::apply
584	3	1	4.06s	18.4s	Algorithm::MasterMind::partitions
198300	1	1	3.56s	13.6s	Algorithm::Evolutionary::Op::QuadXOver::apply
1718600	4	3	2.48s	2.48s	Algorithm::Evolutionary::Individual::Base::Fitness

See [all 2644 subroutines](#)

You can view a [treemap of subroutine exclusive time](#), grouped by package.

NYTProf also generates call-graph files in [Graphviz](#) format: [inter-package calls](#), [all inter-subroutine calls](#) (probably too complex to render easily).

You can hover over some table cells and headings to view extra information.
Some table column headings can be clicked on to sort the table by that column.

There's always
a better
algorithm/
data structure



And differences are huge

- Sort algorithms are an example
 - Plus, do you need to sort the population?
- Cache fitness evaluations
 - Cache them permanently in a database?
 - Measure how much fitness evaluation takes
- Thousand ways of computing fitness
 - How do you compute the MAXONES?
 - `$fitness_of{$chromosome} = ($copy_of =~ tr/1/0/);`
- Algorithms and data structures interact

Case Study: EAs as software programs

Time analysis of standard evolutionary algorithms as software programs

<http://dx.doi.org/10.1109/ISDA.2011.6121667>

Programs implementing EAs are analyzed; huge improvements can be achieved by changing random number generators or memory usage patterns

Implementation matters!

A photograph of three young people, two girls and one boy, wearing white chef hats and aprons, working with dough on a table. The girl on the left is wearing a pink and white striped cardigan over a red shirt. The girl in the middle is wearing a white sweatshirt with an American flag graphic. The boy on the right is wearing a white t-shirt with a blue patch. They are all focused on their task. The background is a white tent-like structure with some lighting equipment visible.

Learn the tricks of the trade

Two trades

- Evolutionary algorithms

- Become one with your algorithm.

- It does not work, but for a different reason than what you think it does



This is the Zen!

- Programming languages.

- What function is better implemented?

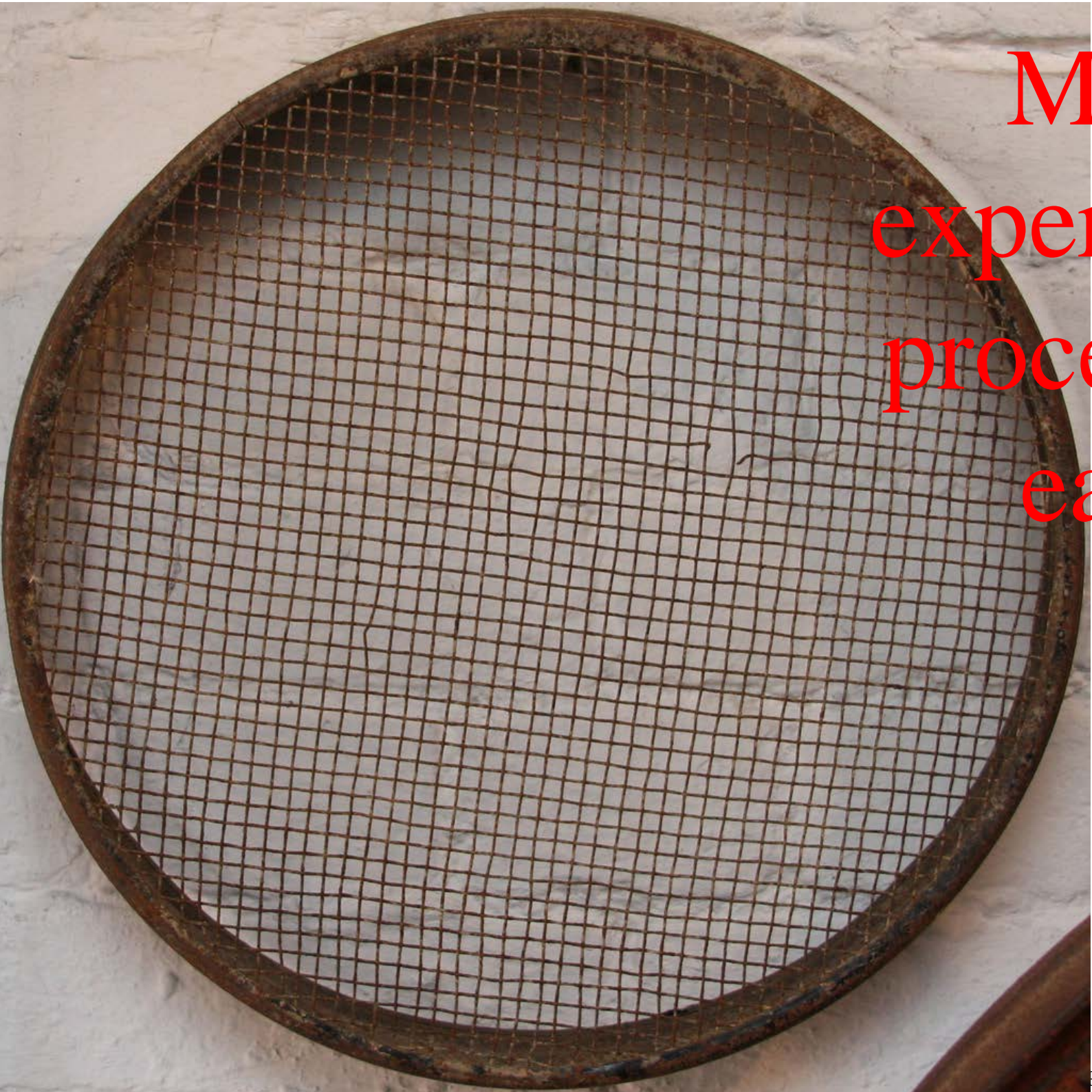
- Is there yet another library to do sorting?

- Where should you go if there's a problem?

- Even a third trade: programming itself.

Case study: sort

- Sorting is routinely used in evolutionary algorithms
 - Roulette wheel, rank-based algorithms
- Faster sorts (in Perl):
<http://raleigh.pm.org/sorting.html>
 - Sorting implies comparing
 - *Orcish* Manoeuvre, *Schwartzian* transform
 - Sort::Key, fastest ever
<http://search.cpan.org/dist/Sort-Key/>

A circular metal mesh screen, likely a sieve or filter, resting on a light-colored surface. The screen is made of a dark, possibly rusted, metal frame with a fine, square-patterned mesh. The mesh is slightly sagging in the center. The background is a light, textured surface, possibly concrete or stone.

Make
experiment
processing
easy

Avoid drowning in data

- Every experiment produces megabytes of data
 - Timestamps, vectors, arrays, hashes.
 - Difficult to understand after some time.
- Use serialization languages for storing data
 - YAML: Yet another markup language.
 - JSON: Javascript Object Notation.
 - XML: eXtensible Markup language.
 - [Name your own].

Case study: Mastermind

Entropy-Driven Evolutionary Approaches to the Mastermind Problem

Carlos Cotta et al., <http://www.springerlink.com/content/d8414476w2044g2m/>

- Output uses YAML.
- Includes:
 - Experiment parameters.
 - Per-run and per-generation data.
 - Final population and run time.

Open sourced (Follow #21)

When
everything
fails

visualize

backup
your data



Better safe than unpublished

- Get an old computer, and backup everything there.
 - If you do open science, you get that for free!
- In some cases, create virtual machines to reproduce one paper's environment
 - Do you think gcc 3.2.3 will compile your old code?
- Use rsync, bacula or simply cp.
- It's not if your hard disk will fail, it's when.



Keep stuff
together

Where did I left my keys?

- Paper: program + data + graphics + experiment logs + text + revisions + referee reports + presentations.
- Experiments have to be rerun, graphics replotted, papers rewritten.
- Use logs to know which parameters produced which data that produced which graph.
- And put them all in the same directory tree, or use sensible naming conventions.

Consider *literate* programming

- *Literate programming* means keeping program and document describing it and results in the same place.
- SWeave and Knitr integrate LaTeX and R in the same document.
 - Check availability for your favorite platform.
- Not the most popular way of writing papers.
- But check also <http://www.executablepapers.com/>

Keep a balance between fashions and efficiency

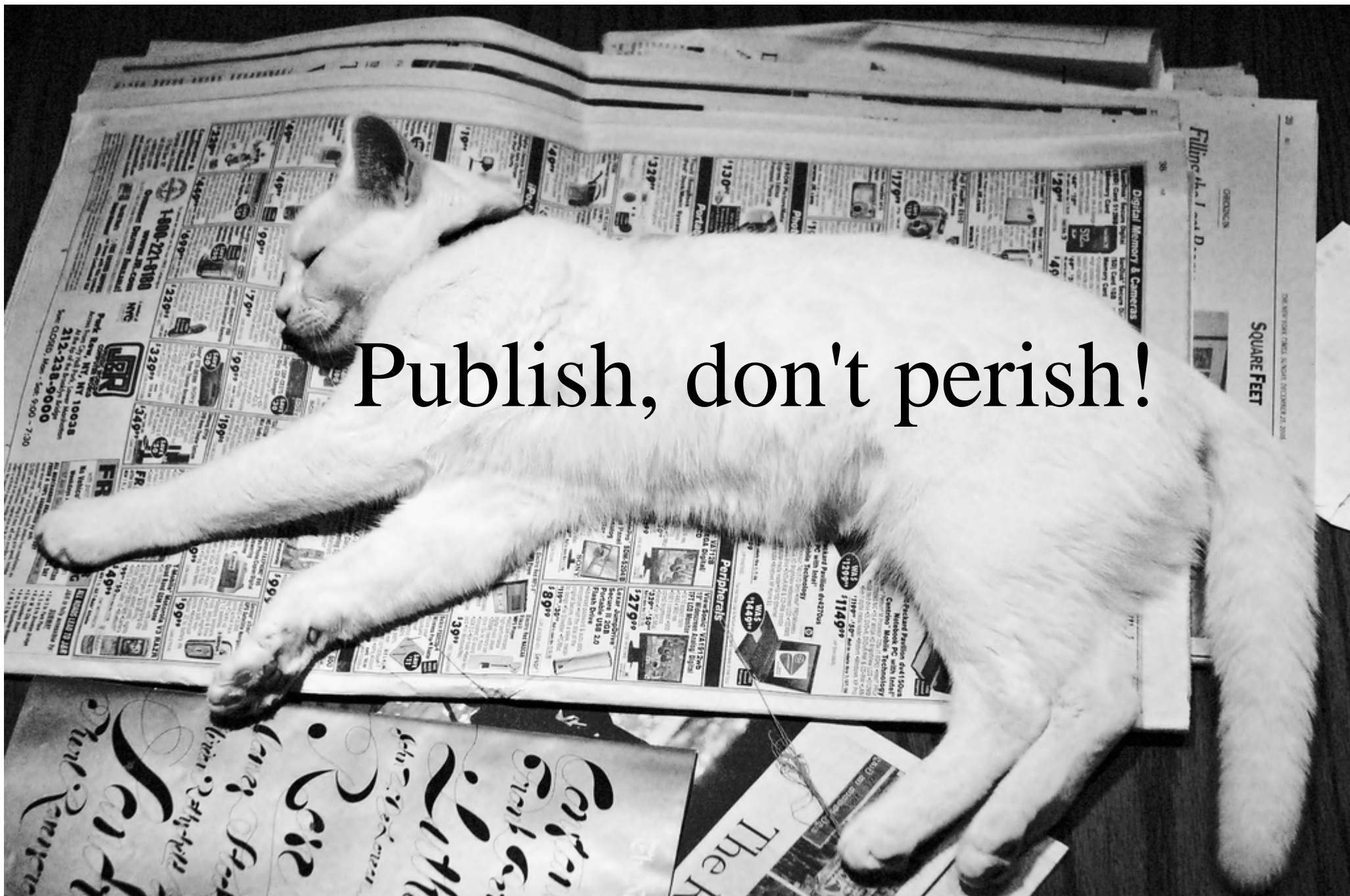


A photograph of a potted tree with green and reddish leaves in a blue oval planter. A yellow and black garden hose nozzle is spraying water onto the base of the tree. The scene is outdoors on a wooden deck with a grassy yard and a wooden fence in the background.

Nurture your
code

A moment of joy, a lifetime of grief

- Run tests periodically, or when there is a major upgrade of interpreter, upstream library or OS.
 - Can be automated.
 - See [#6](#).
- Maintain a roadmap of releases
 - Remember this is *free* software, **engage the community**.
- Your research is intended for the whole wide world.





Or camels!

(no cats were harmed preparing this
presentation)
Check me out at:

<http://twitter.com/jjmerelo>

<http://goo.gl/OFou1>

**Any (more)
questions?**

See you in Evostar 2014, Baeza:
<http://evostar.org>!