Grouping Common Lisp Benchmarks

- or -

Bignums and Consing and Vectors, Oh My!

Christophe Rhodes

Centre for Computational Creativity

City University, London

Gabriel Benchmarks

Common Lisp, Scheme, ISLISP versions

Probably in use in other Language Communities

Richard Gabriel, 1995, MIT Press

- FFT
- FACTORIAL
- TAK
- ...

Lies, damned lies, and benchmarks...

cl-bench

Suite of benchmarks for Common Lisp

Maintained by Eric Marsden

Several new benchmark programs included

- CLOS.defclass
- COMPILER
- CORE-FILE-SIZE
- LOAD-FASL
- ...

Additional tests more task-oriented

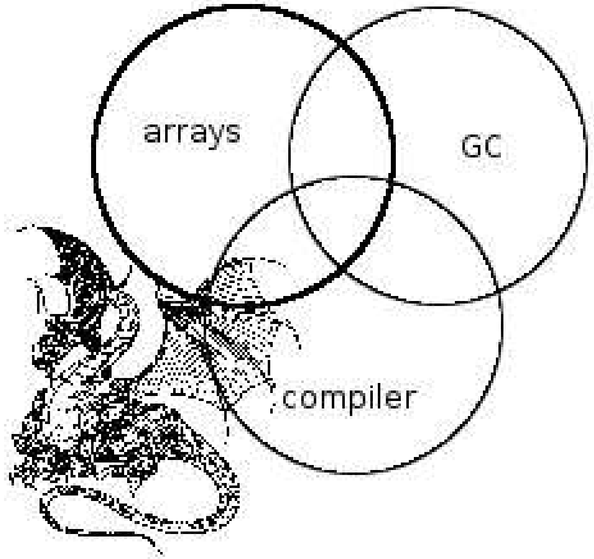
Report (to pdf) classifies benchmarks by task

Steel Bank Common Lisp

200,000 lines of Lisp (plus 20,000 C and asm)

- 15,000: each of six backends
- 50,000: compiler
- 70,000: library
- plus 10,000 in contributed modules

20+ year history, 3 years' personal involvment



Running the Benchmarks

Run the benchmarks on shiny new SBCL binary.

```
Benchmark                          Time / s
TAK                                    2.08
FFT                                   16.63
FACTORIAL                              2.30
CLOS/defclass                          1.75
...
```

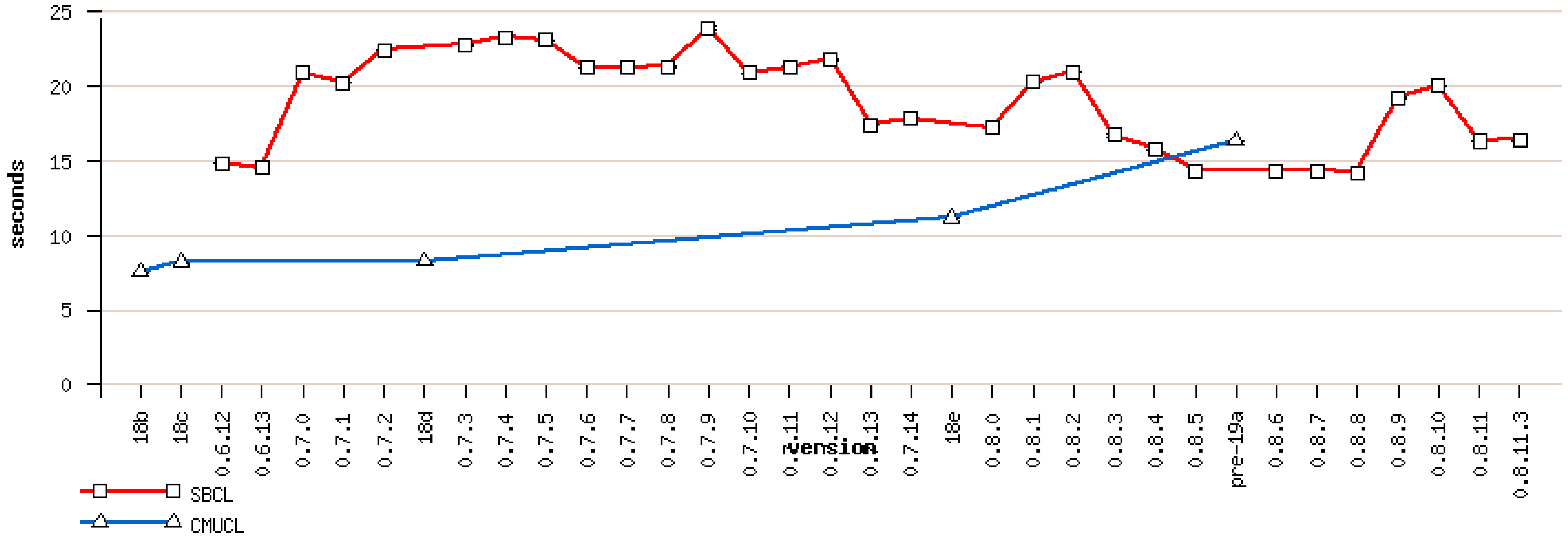
SBCL has

- reliable, scriptable build procedure
- disciplined revision control history

so can investigate effects of changesets

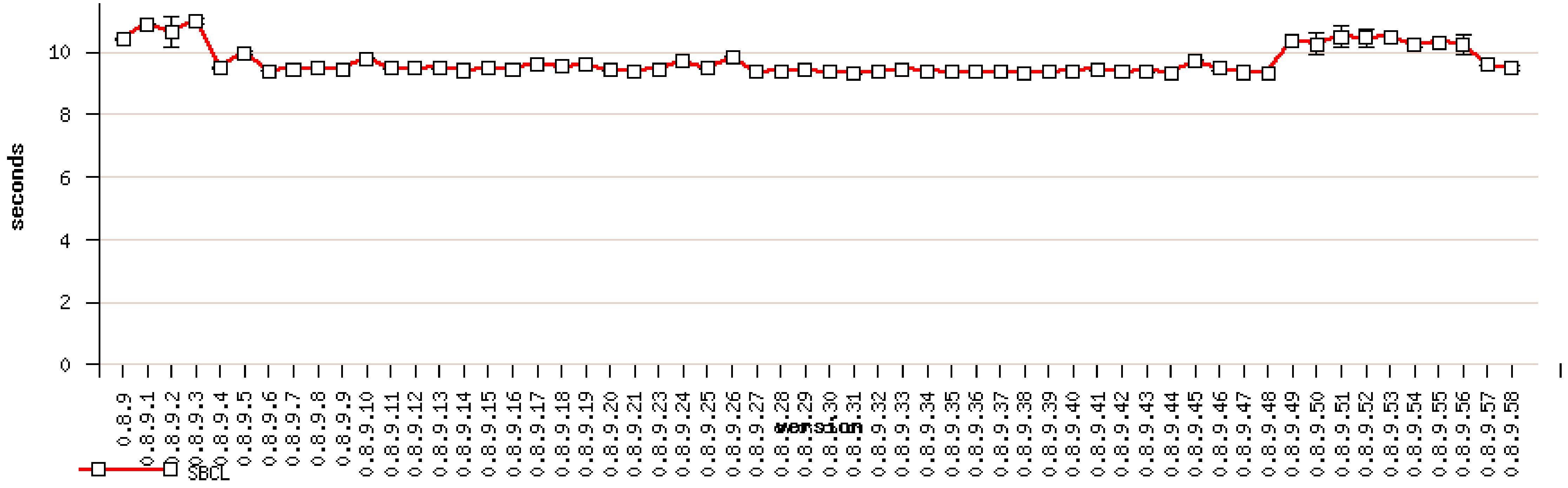
Raw results

CLOS.defmethod: released versions



Raw results (contd.)

COMPILER: SBCL 0.8.9 - 0.8.10 series



Analysis

Too much data (300 builds times 65 benchmarks)

Human eye is (too) good at finding patterns, so find a way to use it

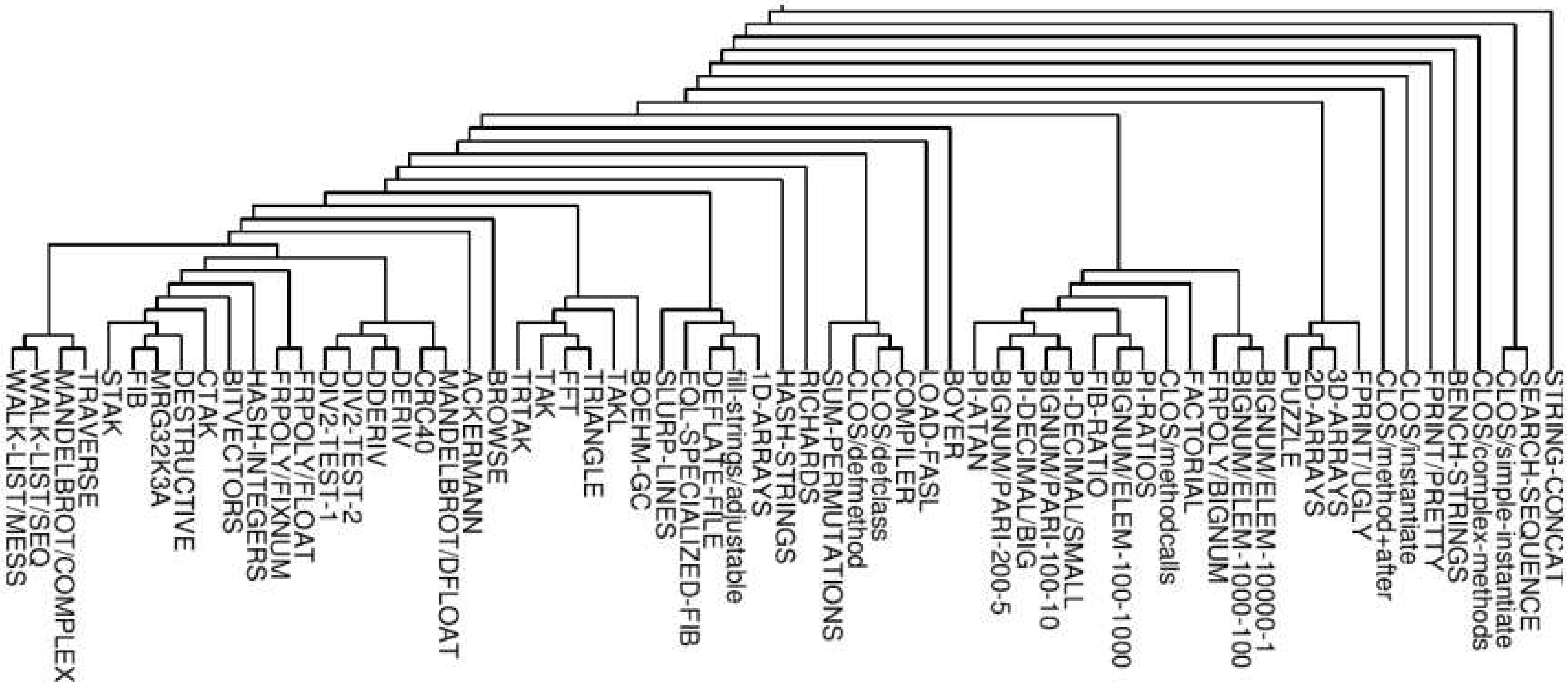
Apply clustering techniques (used in bioinformatics, musical data reduction)

Convert information into comparable form:

- difference between successive scores
- normalize by largest

Means of computing similarity (e.g. dot product)

Analysis (contd.)



Hierarchical agglomeration: clusters seen by eye

Other clustering methods exist: K-means, ...

Further Work

- Rerun with more data
- Compare with other implementations
- Correlate with test suite performance

Paul Dietz' ANSI-tests approaching 1.0: 16,000 tests

Each Common Lisp feature tested individually

Does fixing bugs cause performance changes?

Conclusions

Can perform experiments to find things out, even on computer programs

Managing complexity can be assisted by intelligent classification

Interpreting benchmark results can be tricky, but useful

Further information, downloads

Benchmarks

- http://www.dreamsongs.net/NewFiles/Timrep.pdf
- http://www.chez.com/emarsden/downloads/cl-bench.tar.gz

SBCL

- http://www.sbcl.org/
- http://sbcl.boinkor.net/benchmarks/