The Evolution of BEAGLE:
Confessions of a mongrel rule-breeder

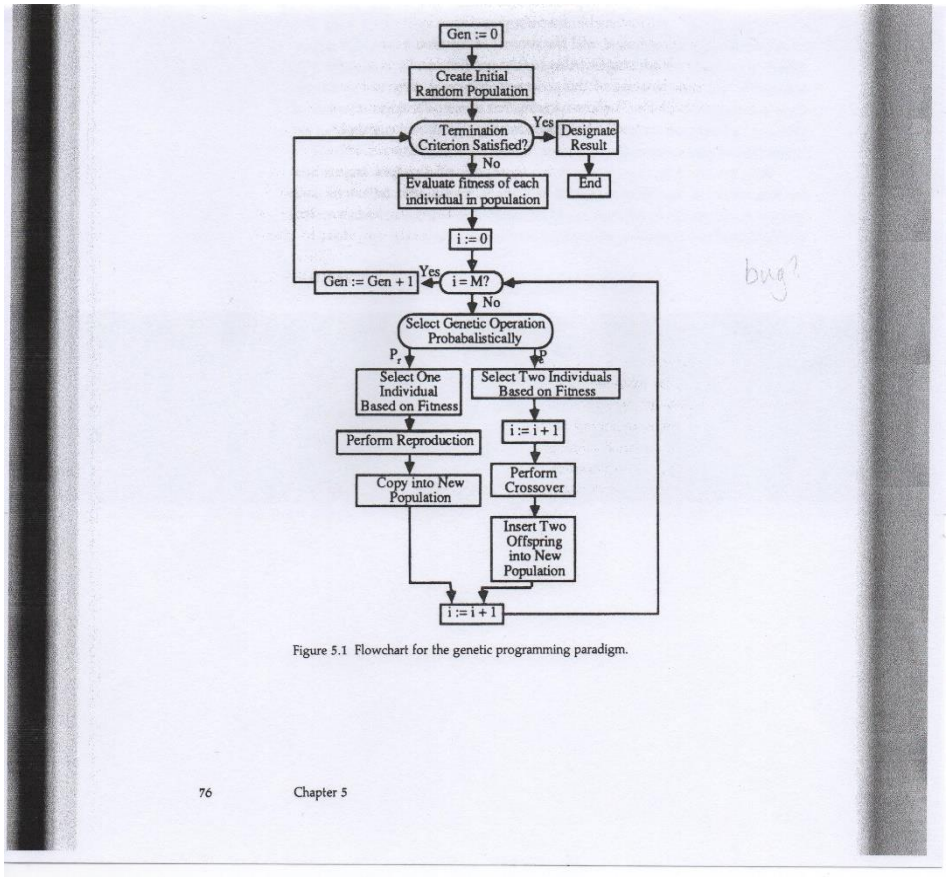
Richard Forsyth,
www.richardsandesforsyth.net

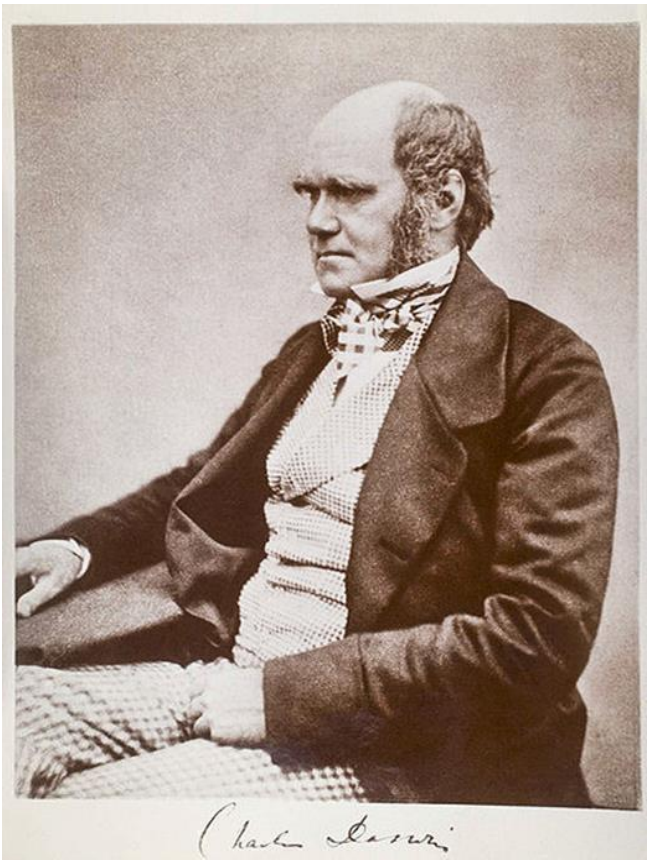
Outline

- (1) Biased retrospective on the genealogy of Genetic Programming
- (2) The BEAGLE revival: some lessons learned
- (3) Prospects of progress & prophecies of peril

But first, what is Genetic Programming?

- We wouldn't be here if we didn't know.
- Or would we?





Genotypes as Programs

- “The consequence of the ensuing encounter with the impinging multifactored environmental complex may be death, as the inevitable outcome of inadequate **genetic programming**; or it may be survival, with the genes then co-operatively spelling out the individual developmental tendencies.”
- Pulp and Paper Magazine of Canada, 59, 1958, page 193.

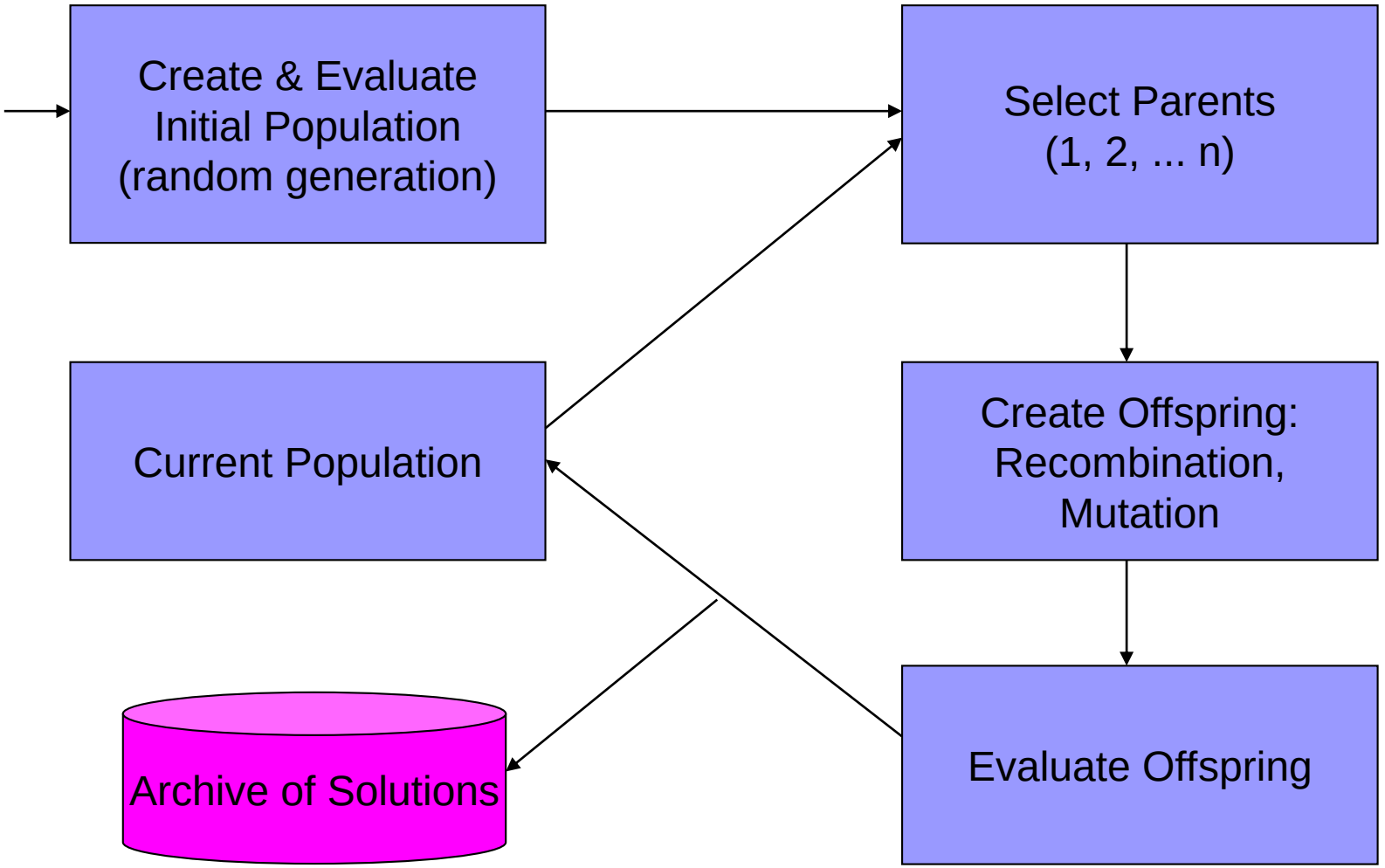
Genotypes from Programs

- “Hex settled down to flick through the pile of manuals. **Genetic programming** wasn't easy. He could see why Rose, even though he'd invented the concept and written the handbooks, had baulked at the details.” – R. Forsyth, Computer Weekly, 6 February, 1979.
- “This software is to be transferred from electronic equipment into living tissue via the **genetic programming technique** invented by Mike Rose -- a transformation that will turn the animal kingdom into a biological computing engine, rendering the Future System effectively immortal. It is the greatest amino acid trip since Genesis.” – R. Forsyth, Computer Weekly, 13 February 1979.

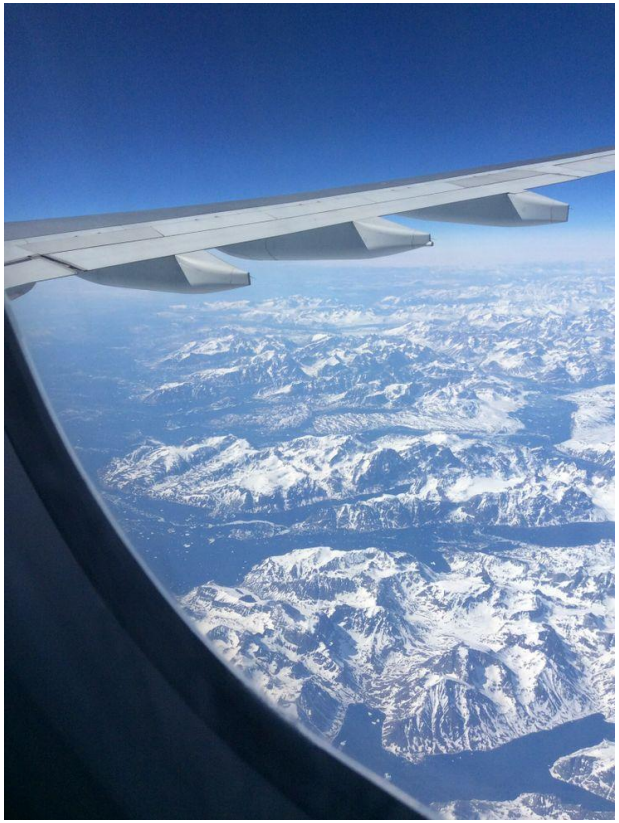
Programs as Genotypes

- "GP evolves **computer programs**, traditionally represented in memory as **tree structures**." – Wikipedia, 11 February 2016.
- "genetic programming evolves over populations of **program fragments** to assemble a final program that gives a solution." – Oxford Dictionary of Computing, Daintith & Wright (2008).
- "Genetic Programming is the extension of the genetic model of learning into the space of **programs**. That is, the objects that constitute the population are **not fixed-length** character srings … they are programs that, when executed, "are" the candidate solutions to the problem. These programs are expressed in genetic programming as **parse trees**, rather than as lines of code." – Hitchhiker's Guide to Evolutionary Computing, 2001. Accessed 11 February 2016: http://www.aip.de/~ast/EvolCompFAQ/Q1_5.htm

Basic Evolutionary Computing Cycle



The Genesis of Genetic Programming?

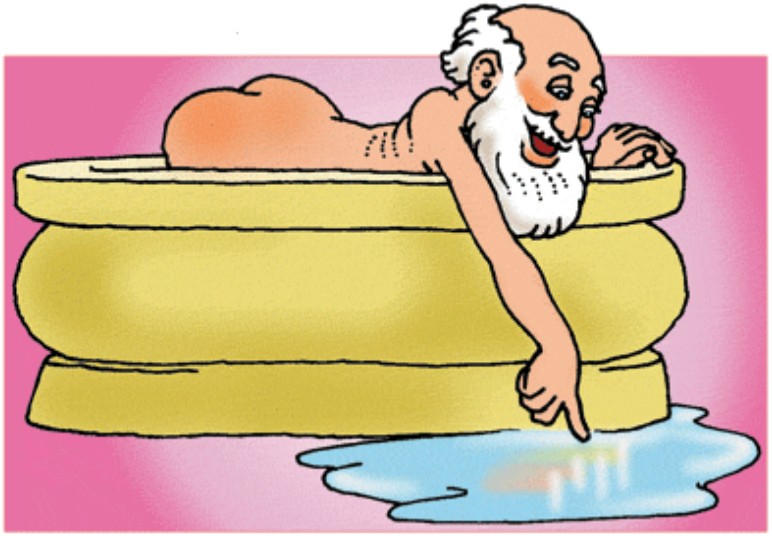


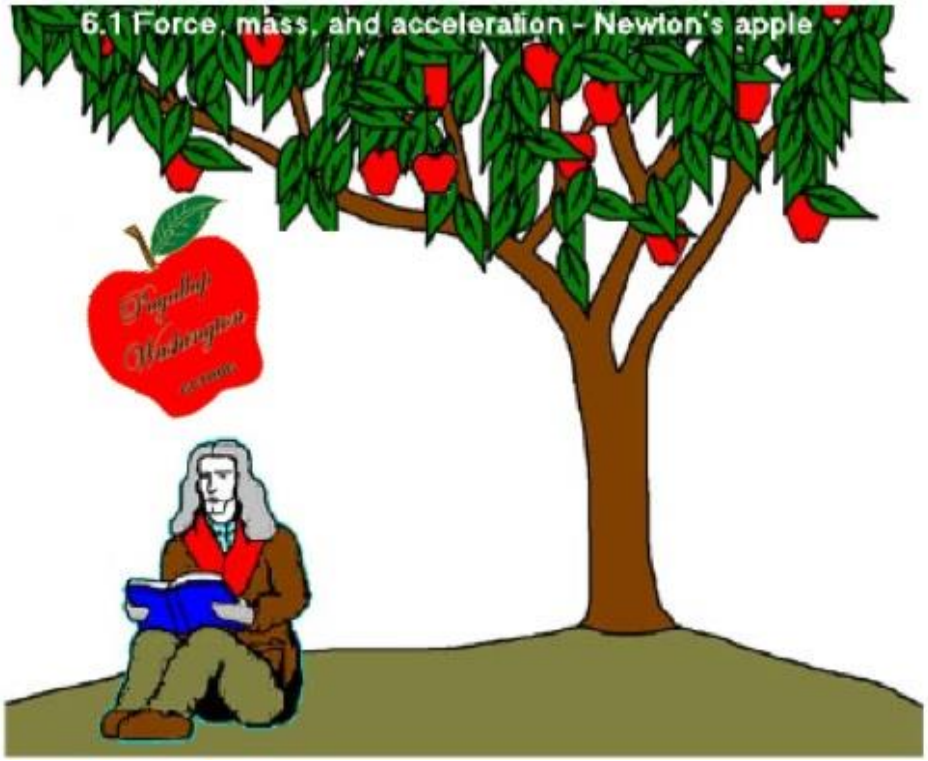
“Koza was 30,000 feet above Greenland when he asked himself why a genetic algorithm, so adept at refining pipelines, couldn't be used to evolve its own software.”

Popular Science, April 2006.

http://www.popsci.com/scitech/article/2006-04/john-koza-has-built-invention-machine

Eureka Moments in Science





Genetical Search: like so many neat computing ideas, Turing thought of it first

- Turing identified a third approach to machine intelligence in his 1948 paper entitled “Intelligent Machinery” (Turing 1948, page 12; Ince 1992, page 127; Meltzer and Michie 1969, page 23), saying:
- “There is the *genetical or evolutionary* search by which a combination of genes is looked for, the criterion being the survival value.”

Program evolution before 1987

- "The scheme sketched is really a natural selection on the processing demons. If they serve a useful function they survive, and perhaps are even the source for other subdemons who are themselves judged on their merits." -- Selfridge, O. (1959). p. 14.
- "From a more abstract viewpoint, then, an adaptive system is a schema for generating programs in accordance with the dictates of the environment." -- Holland, J. (1962) p. 217.
- "the generation procedure operates in parallel fashion, producing sets or populations of programs at each moment." -- Holland, J. (1962).
- "symbioorganisms will consist of numbers, and numbers in the machine can be interpreted as instructions according to any arbitrary code which can be established by writing an interpretive program." -- Barricelli, N. (1963). p. 2.
- "Thus, nonregressive evolution proceeds to find better and better programs for attacking the problem in hand." -- Fogel, L., Owens & Walsh (1966). p. 12.
- "this highlights the fact that the rules are really programs in a special-purpose language, which might lead to the conclusion that the system should ultimately generate LISP functions." -- Forsyth, R. (1981). p. 165.

My favourite definition of GP

- “**Genetic programming** applies **evolutionary search** to the space of **tree structures** which may be **interpreted as computer programs** in a language suitable to modification by **mutation and recombination**.” Bäck, T., Hammel, U. & Schwefel, H-P. (1997). IEEE Transactions on Evolutionary Computation, 1(1), 3-17.

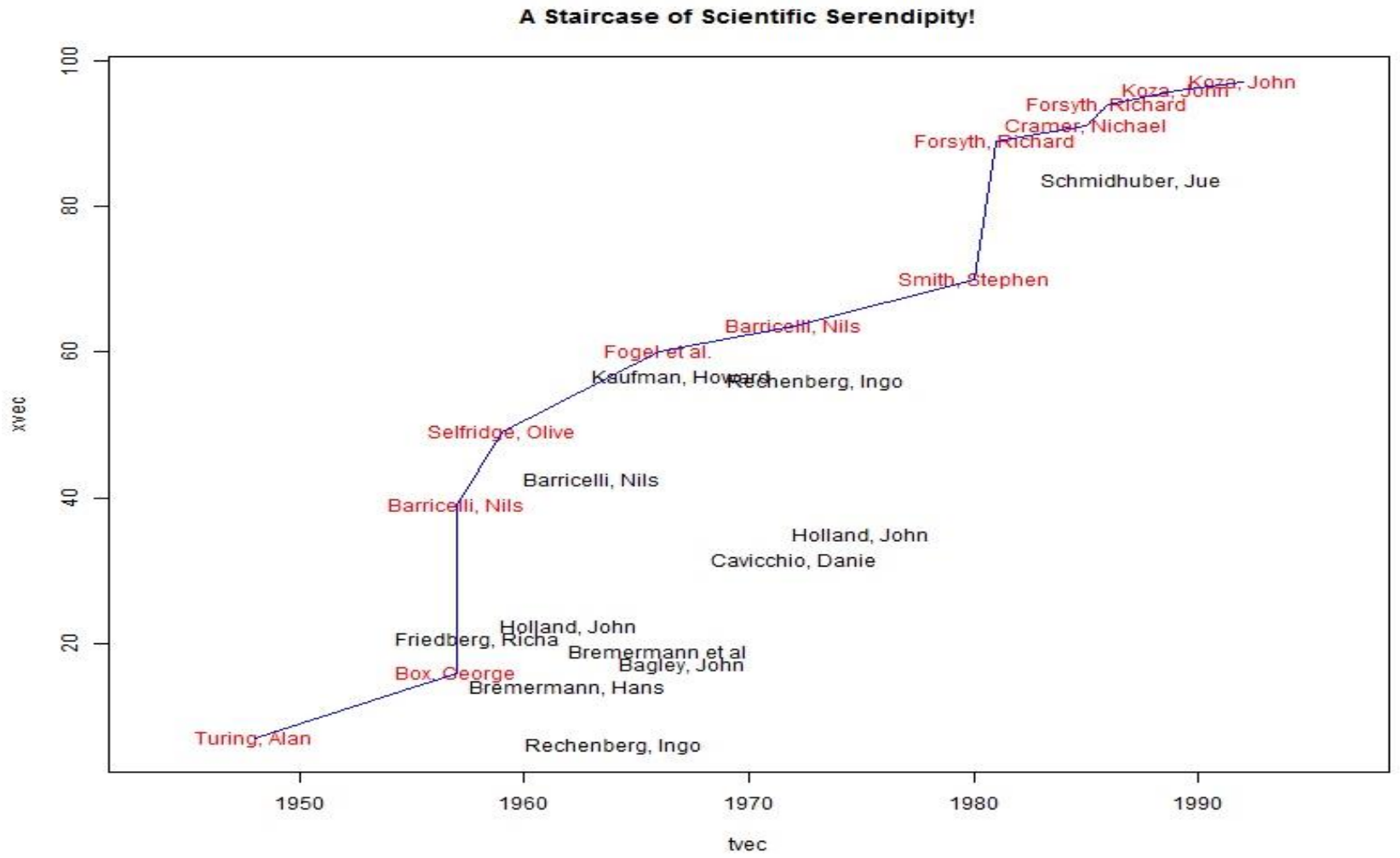
Beyond the bitstring barrier

- Tree-structured heritable material
- Variable-length heritable material
- Executable heritable material
- Syntax-aware crossover
- -- Kinnear (1994).
  - [Optimized by emulated evolution.]

One Giant Leap or many small steps?

“if Watson and Crick had not existed, the insights they provided in one single package would have come out much more gradually over a period of many months or years. Dr. B might have seen that DNA is a double-strand helix, and Dr. C might later have recognized the hydrogen bonding between the strands. Dr. D later yet might have proposed a complementary purine-pyrimidine bonding, ... Finally, we might have had to wait for Dr. G to propose the replication mechanism of DNA based on the complementary nature of the two strands. All the while Drs. H, I, J, K and L would have been confusing the issue by publishing incorrect structures and proposals.” Gunther S. Stent (1972). Scientific American, 227, p. 90.

The Ascent of GP



A spreadsheet of serendipity

weightings =		100		15	20	10	25	10	5	2	5	2	2	2	2
name	year	altitude	Clear Darwinian basis	Variable-length heritable material	Tree-structured heritable material	Syntax-aware crossover	Population members executable as programs	Population size exceeds 2	Uses crossover operation	Uses mutation operator	Export of executable software	Genotypes incorporate looping	Explicit submodule generation	System applied by others than originator	Selection at topmost level
Turing, Alan	1948	7	1						1		0			0	1
Box, George	1957	16	1	0	0	0	0	1	0	1	0	0	0	1	1
Barricelli, Nils	1957	39	1	1	0	0.5	0	1	1	1	0	0	0	0	1
Friedberg, Richard	1958	20.5	0.5	0	0	0.5	1	0	1	1	0	1	0	0	1
Selfridge, Oliver	1959	49	1	0	1	1	0	1	1	1	0	0	0.5	0.5	0
Bremermann, Hans	1962	14	1	0	0	0	0	0.5	1	1	0	0	0	0	1
Holland, John	1962	22.25	0.5	0.5	0		1	1		1	0			0	
Barricelli, Nils	1963	42.5	1	0.5	0	0	1	0	1	1	0	0.5	0	0	1
Rechenberg, Ingo	1964	6	1	0	0	0	0	0	0	1	0	0	0	1	1
Fogel et al.	1966	60	1	1	0	0.5	1	1	0	1	0	0.5	0	0	1
Bremermann et al.	1966	19	1	0	0	0	0	1	1	1	0	0	0	0	1
Bagley, John	1967	17	1	0	0	0	0	1	1	1	0	0	0	0	0
Kaufman, Howard	1967	56.5	1	1	0	1	0.5	1	1	1	0	0	0	0	1
Barricelli, Nils	1972	63.5	1	1	0	0.5	1	1	0.5	1	0	1	0	0	1
Cavicchio, Daniel	1972	31.5	1	0.5	0	0.5	0	1	1	1	0	0	0	0	1
Rechenberg, Ingo	1973	56	1	1	0.5	1	0	1	1	1	0	0	0	1	1
Holland, John	1975	35	1	0	0	0	0.5	1	1	1	0.5	0.5	0	1	0
Smith, Stephen	1980	70	1	1	0	1	1	1	1	1	0	0.5	0	0	1
Forsyth, Richard	1981	89	1	1	1	1	1	1	1	1	0	0	0	0	1
Cramer, Nichael	1985	91	1	1	1	1	1	1	1	0.5	0	1	0.5	0	1
Forsyth, Richard	1986	94	1	1	1	1	1	1	1	1	1	0	0	1	0
Schmidhuber, Juergen	1987	83.5	1	1	0.5	1	1	1	1	1	0.5	1	0	0	1
Koza, John	1989	96	1	1	1	1	1	1	1	0.5	1	1	0	0.5	1
Koza, John	1992	97	1	1	1	1	1	1	1	0	1	1	0.5	1	1

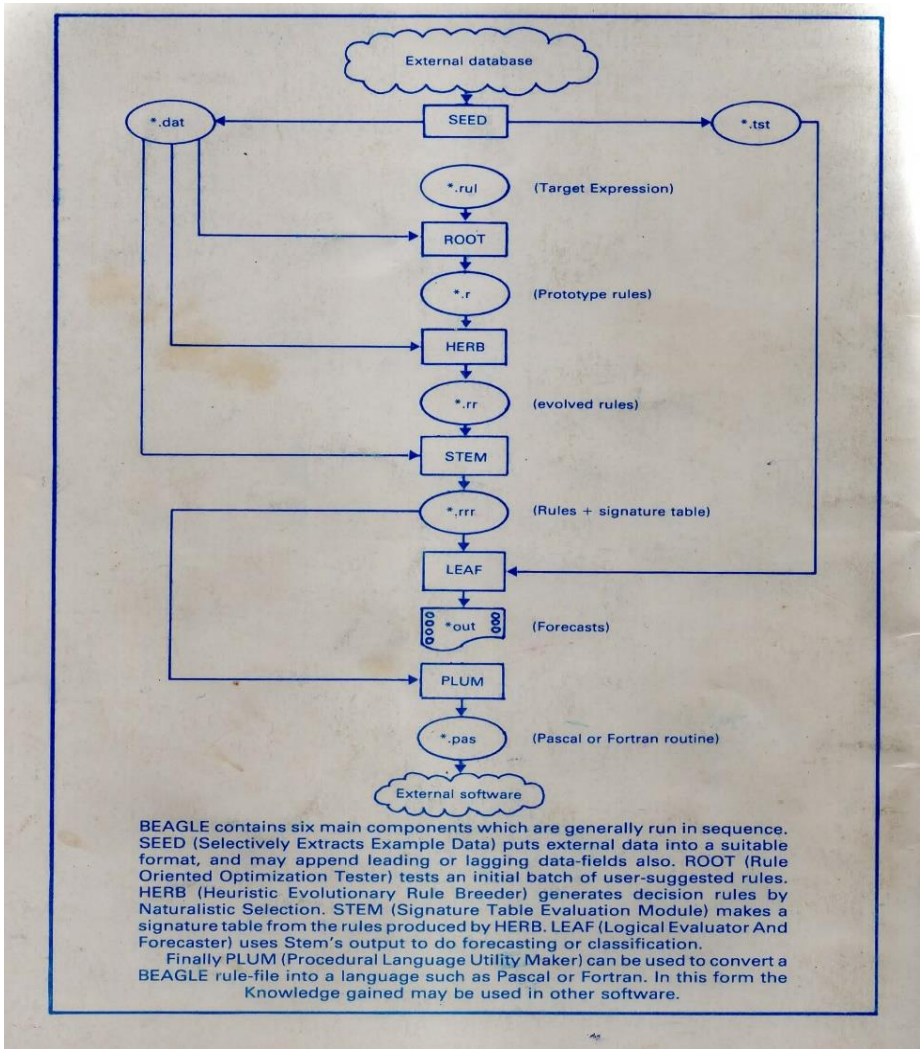
Tentative Conclusions

- Quantum leaps (i.e. little & often), not 1 huge jump
  - From 30,000 feet…
- Another illustration of the (Patrick?) "Matthew Effect"?
  - BEAGLE early but uninfluential

Generalizing GP

- “Although the preferred embodiment uses computer programs as entities, using **other types of entities** remain [sic] within the scope and spirit of the present invention.” J. Koza (1990) US Patent 4935877, p. 12.
- “Today, **different approaches** are considered as GP, from the evolution of **expression trees** to the evolution of electronic circuits or even architectural designs (structures, for short). The overarching principle is to subject all these kinds of **structures with variable complexity** to forces of evolution by applying mutation, crossover and fitness-based selection.” W. Banzhaf (2013) *Evolutionary Computation and Genetic Programming*.

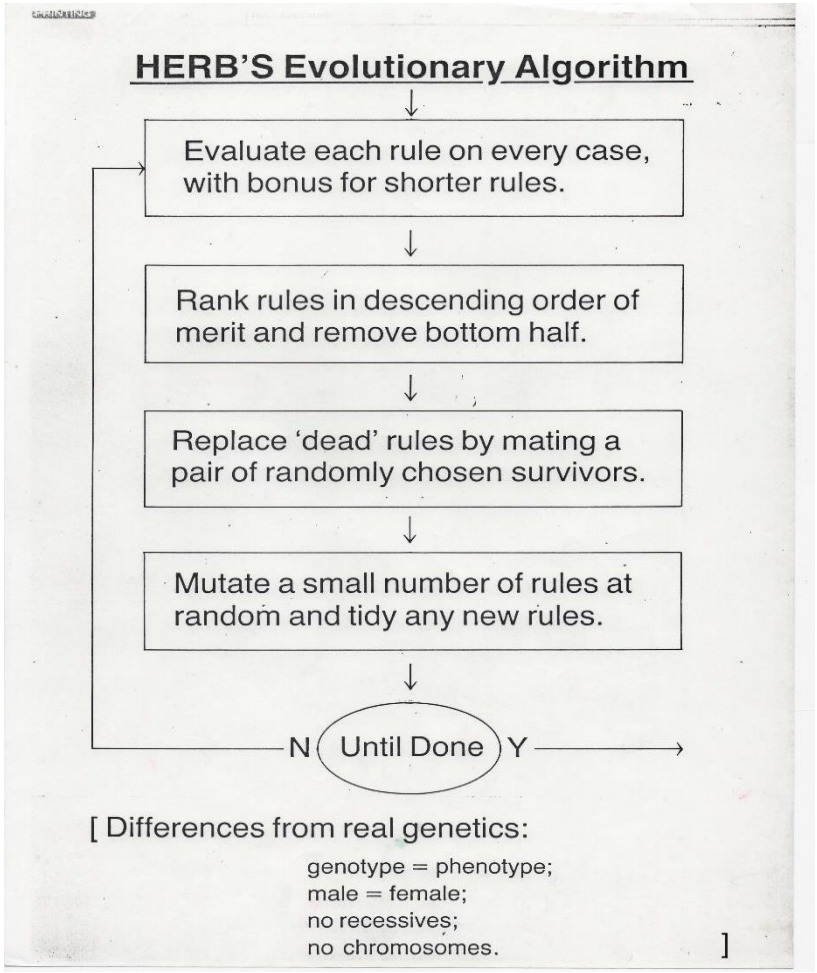
Refloating BEAGLE



BEAGLE contains six main components which are generally run in sequence. SEED (Selectively Extracts Example Data) puts external data into a suitable format, and may append leading or lagging data-fields also. ROOT (Rule Oriented Optimization Tester) tests an initial batch of user-suggested rules. HERB (Heuristic Evolutionary Rule Breeder) generates decision rules by Naturalistic Selection. STEM (Signature Table Evaluation Module) makes a signature table from the rules produced by HERB. LEAF (Logical Evaluator And Forecaster) uses Stem's output to do forecasting or classification.

Finally PLUM (Procedural Language Utility Maker) can be used to convert a BEAGLE rule-file into a language such as Pascal or Fortran. In this form the Knowledge gained may be used in other software.

Heuristic Evolutionary Rule Breeder



How many BEAGLEs?

- 3 ‘incarnations’ (‘generations’?)
  - Veteran/First-generation/grandmother: Kybernetes, 1981
  - Vintage/Second-generation/mother: PC/BEAGLE, 1985
  - Modern/Third-generation/daughter: Python3 BEAGLE, 2016

Good Points of 1980s BEAGLE

- It worked! (Fast enough on MS/DOS personal computers to solve practical problems.)
- Produced programs in C, Fortran or Pascal generated from example data. (Practical automatic programming.)
- Dealt with numeric targets (tabular regression) as well as logical target expressions (classification).
- Avoided 'bloat' with a prolixity penalty.
- Handled string fields as well as numeric variables.
- Bumbled beyond the bitstring.
- Didn't fall into the fitness-proportional selection trap. (Floating approximate median gave quasi-rank-based selection without expense of sorting.)

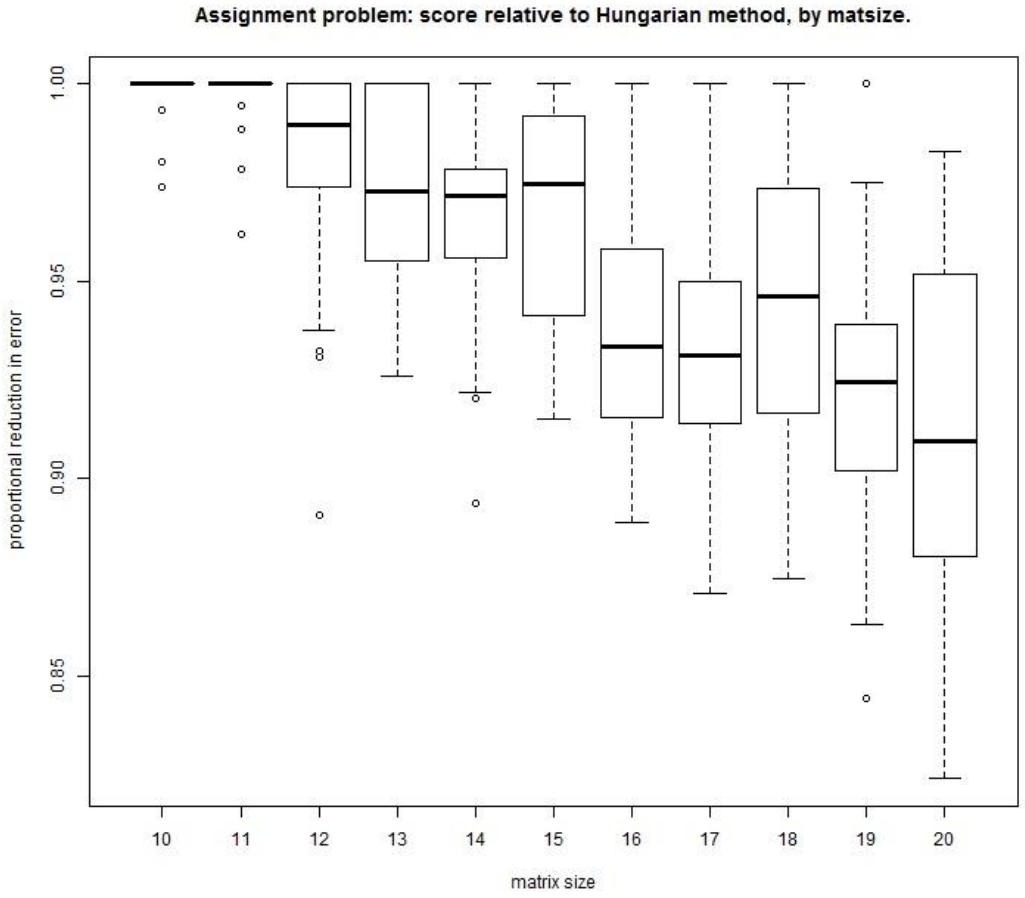
Points to Improve in 21st Century BEAGLE

- Move from generational to incremental search. (Mammals versus mayflies.)
- Handle multi-class classification problems.
- Devise a more intelligible string-handling technique.
- Have a slightly richer expression language
- Implement brevity-bias in a more principled manner. (Tidy **after** evolutionary process, not during!)
- Do Bayesian reasoning right! (Saundesonian??)
  - [“None of the above” mode ~ credibility of probability]
- Give it away free! (Nobody pays for software.)

“Sea Trials”

- 0. Informal trials just to make things work
- 1. Tests of the evolutionary engine
  - Selecting selection pressure (probings)
    - **Assignment Problem** (vs **Hungarian Algorithm**)
  - Exploring “continental collision” (partitioned populations)
    - **Travelling Salesrep Problem**
- 2. Tests of classification methods
  - Basic check against PC/BEAGLE (binary benchmarks)
  - Assessing effect of brevity bias in lexicographic fitness
  - Tabular versus Demonic decision-making

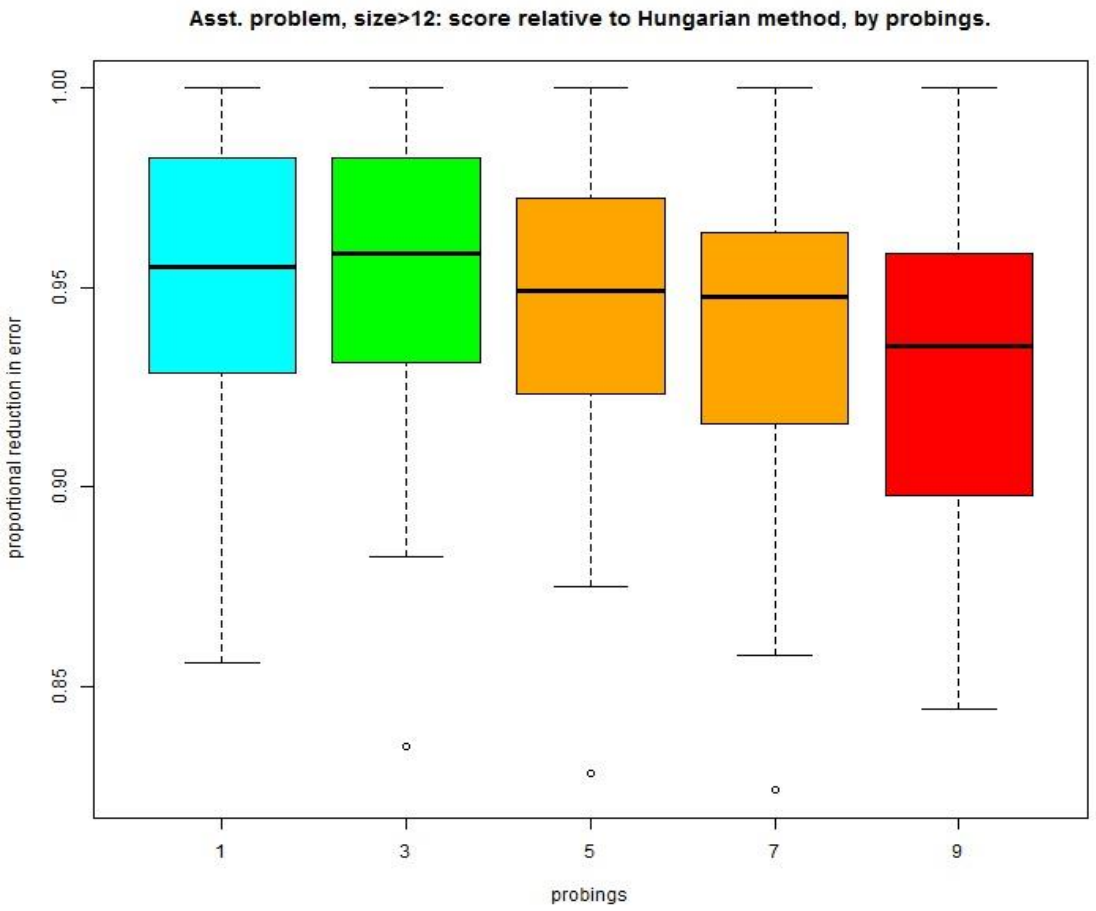
Assignment Problem Trial



Relscore = 1 – (Evscore-Hunscore) / (Randval-Hunscore)

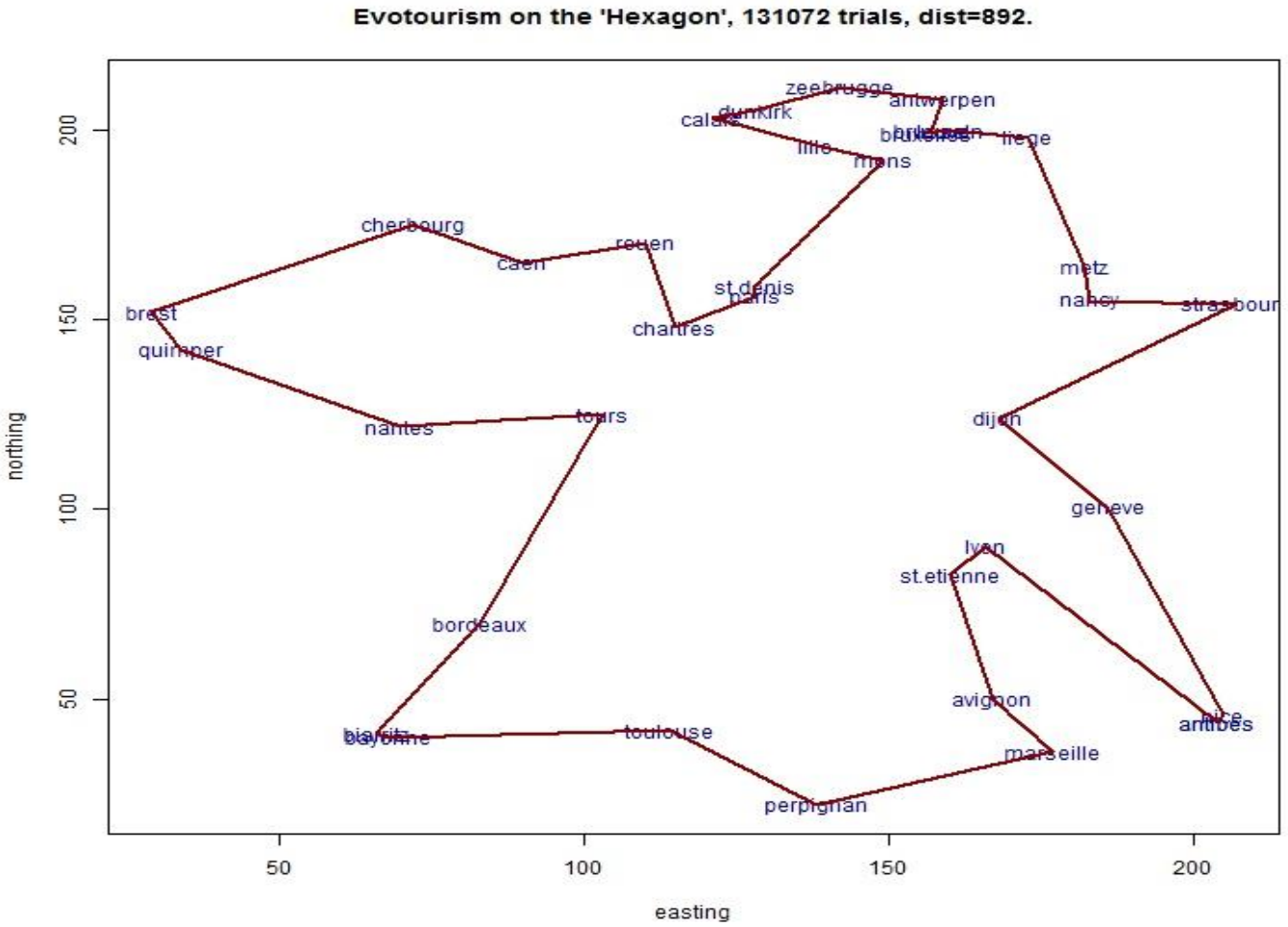
[~ proportional reduction in error measure.]

How many probes? Is a duel a tournament?

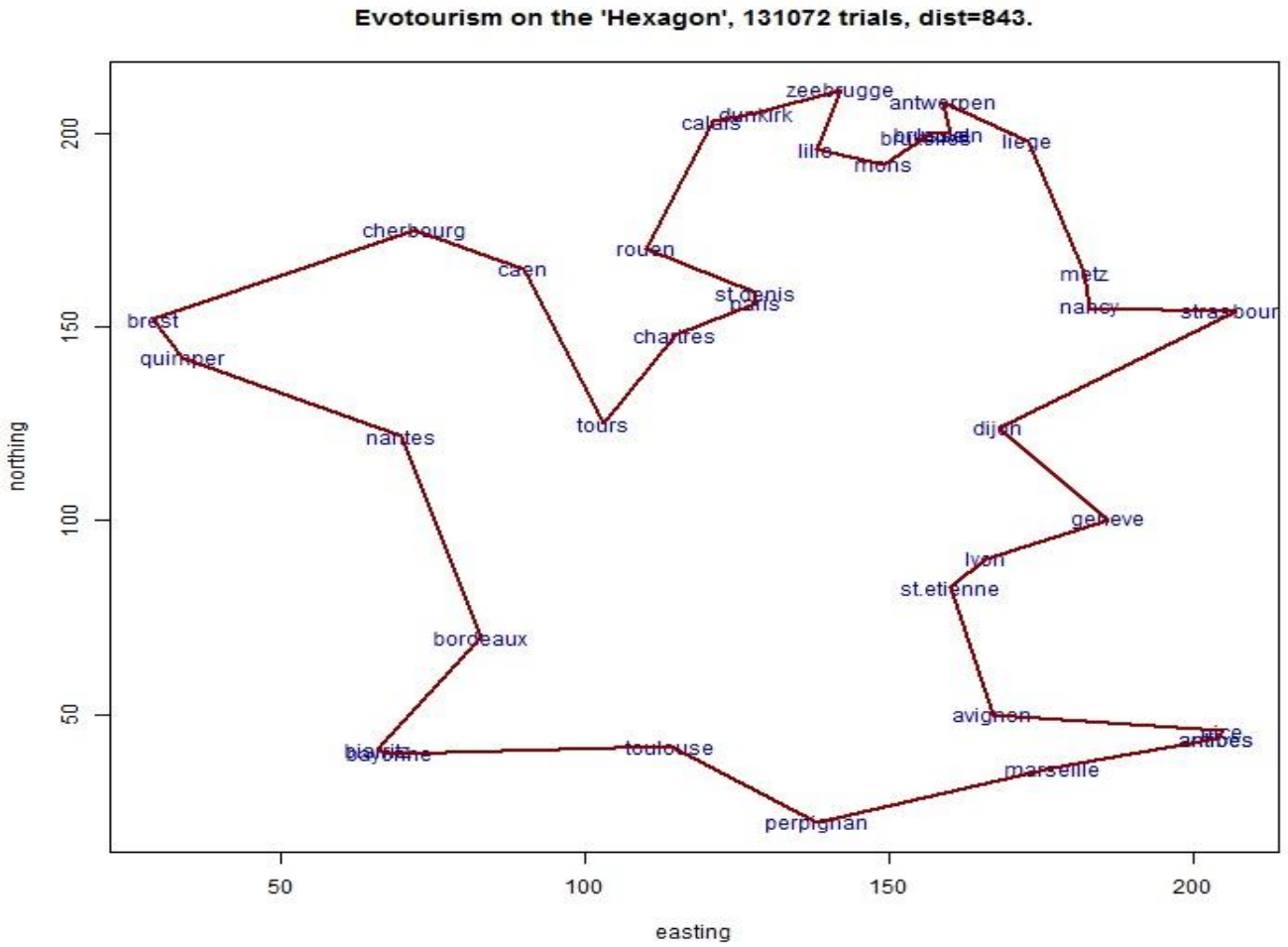


- You can have too much of a good thing
- ‘Duel’ selection is viable

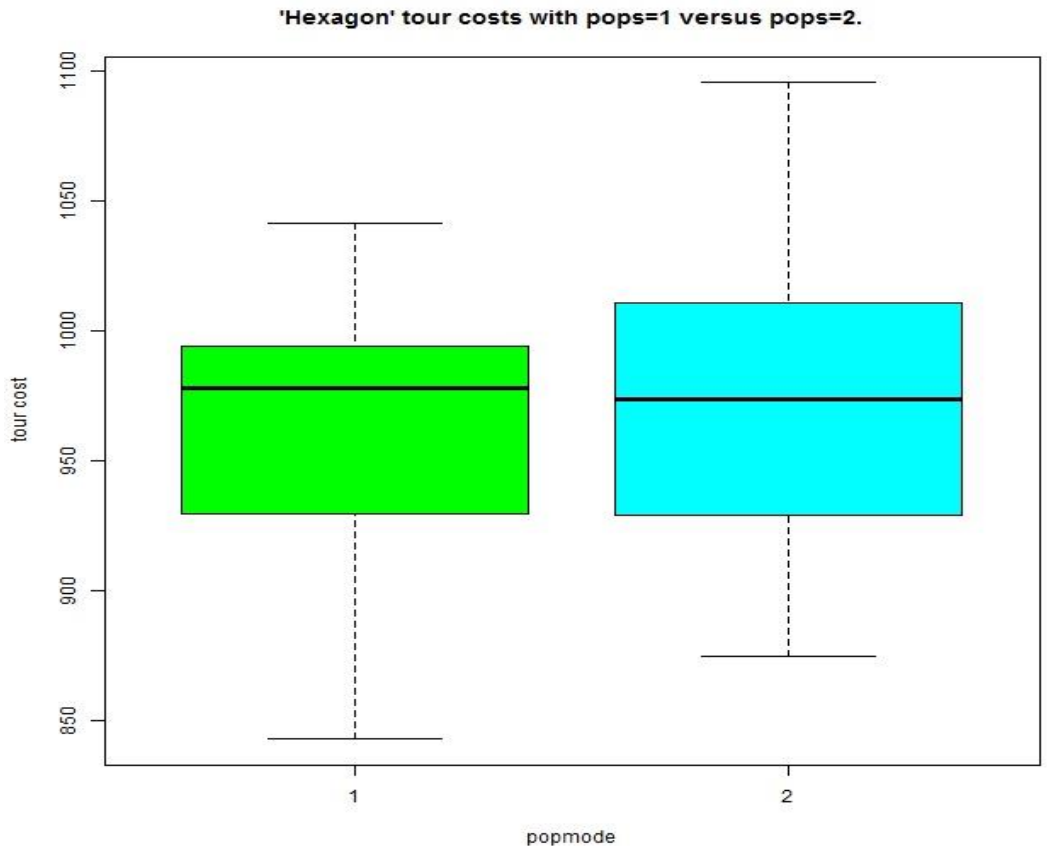
Touring the ‘Hexagon’



A shorter tour

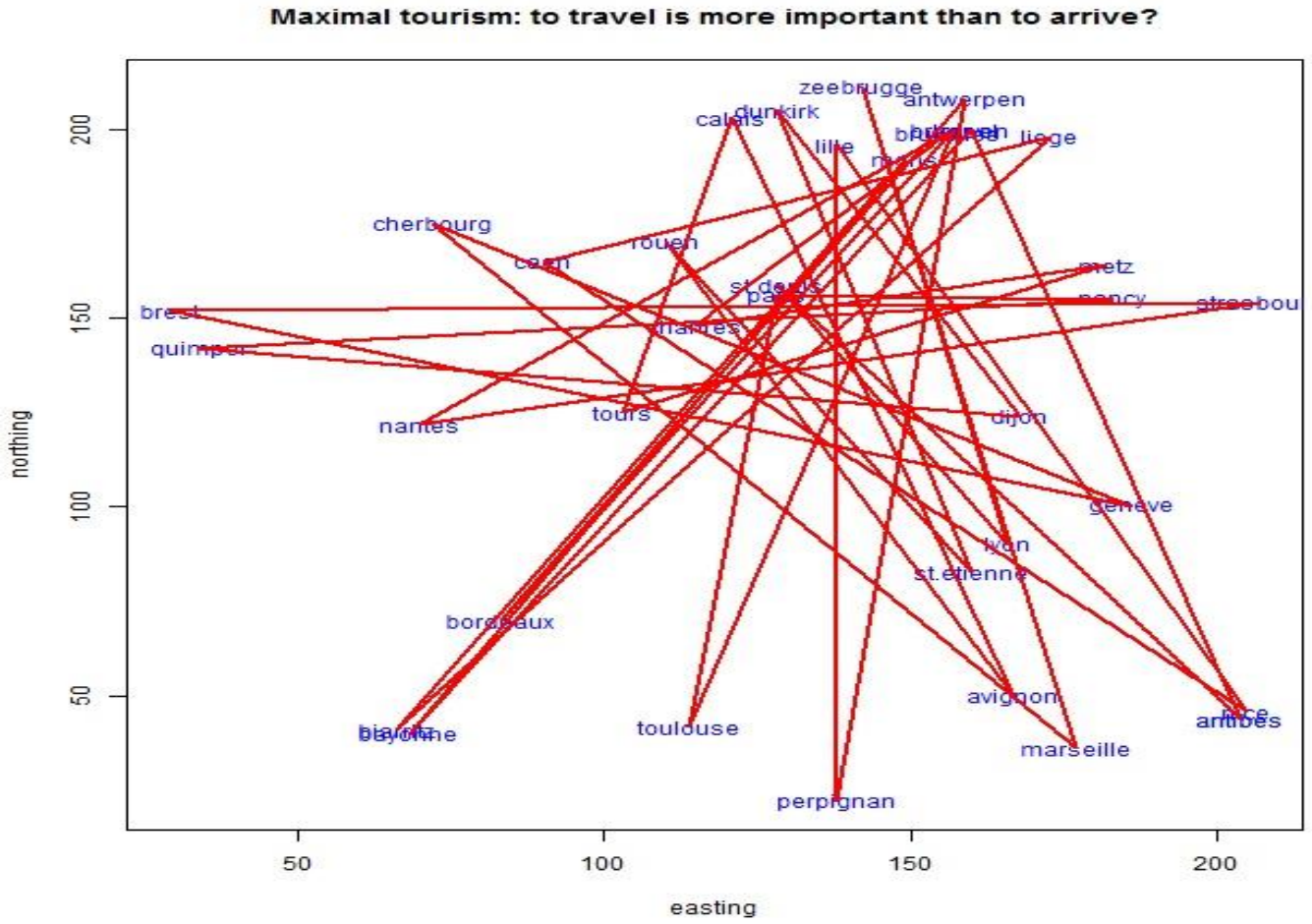


Partitioned Populations



Take-home message (16 reps, 2 modes) : population partitioning doesn't help enough to justify itself. (Not to be incorporated.)

Maximizing travel expenses ?



Getting back to BEAGLE

- Baseline binary benchmarks
- Brevity bias testing
  - Lexicographic fitness
    - [accuracy,size]
- Demonic versus Tabular linkage

What do BEAGLE rules look like?

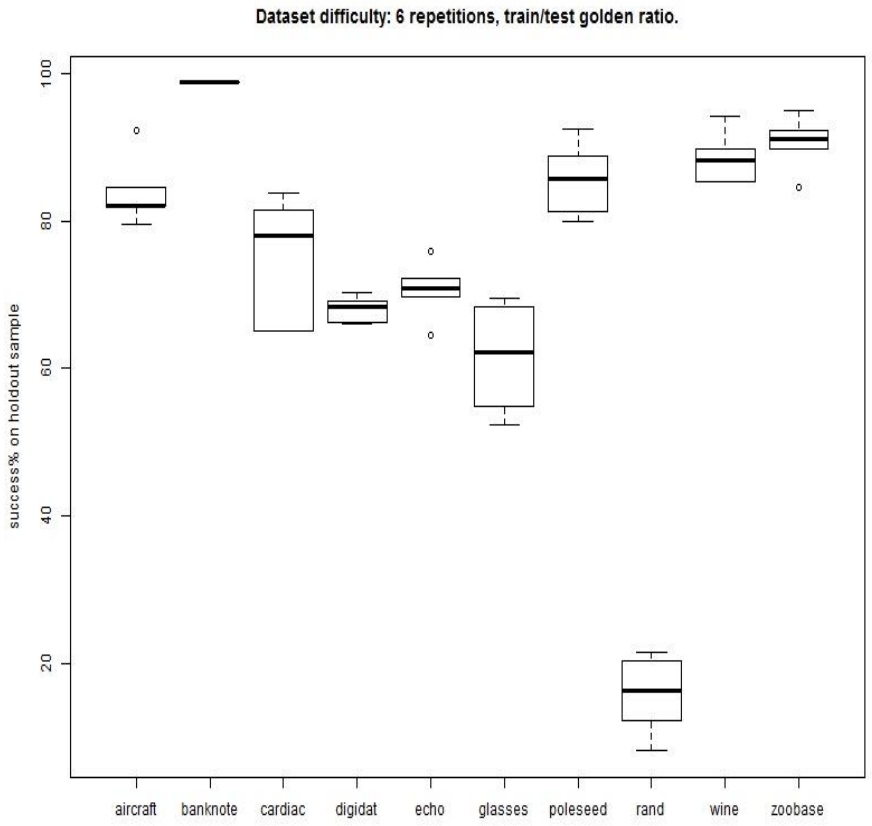
```
( BOMBING >    0.0000)$

(( FIGHTING - ( WINGSPAN >   16.4600)) <=
0.0000)
$     30      1      3     30
(( FUSELAGE >=    9.7652) & ( TOPSPEED <=
520.8080))
$     28      0      5     31
() $  -- from warplane.dat on 27/02/2016
at 16:14:27

        0.5156     64
000     3.0000     33     0.1034  9.0909
001     0.0000      0     0.5156
010     2.0000      3     0.6289 66.6667
011    28.0000     28     0.9833 100.0000
```

```
training data :
C:\beagwork\op\aircraft_dat1.dat
creation date : Tue Mar  8 10:11:56 2016
rule mode : tabular
64 16
fighting
[0, 1]
[0.4606961258558217, 0.5393038741441784]
$
( ( kgladen + loaddiff ) < ceiling )
( cannons > 3.5751266 )
$
00  [21, 2]
01  [0, 3]
10  [6, 24]
11  [0, 8]
[0.7037037037037037, -4.0]
$
```

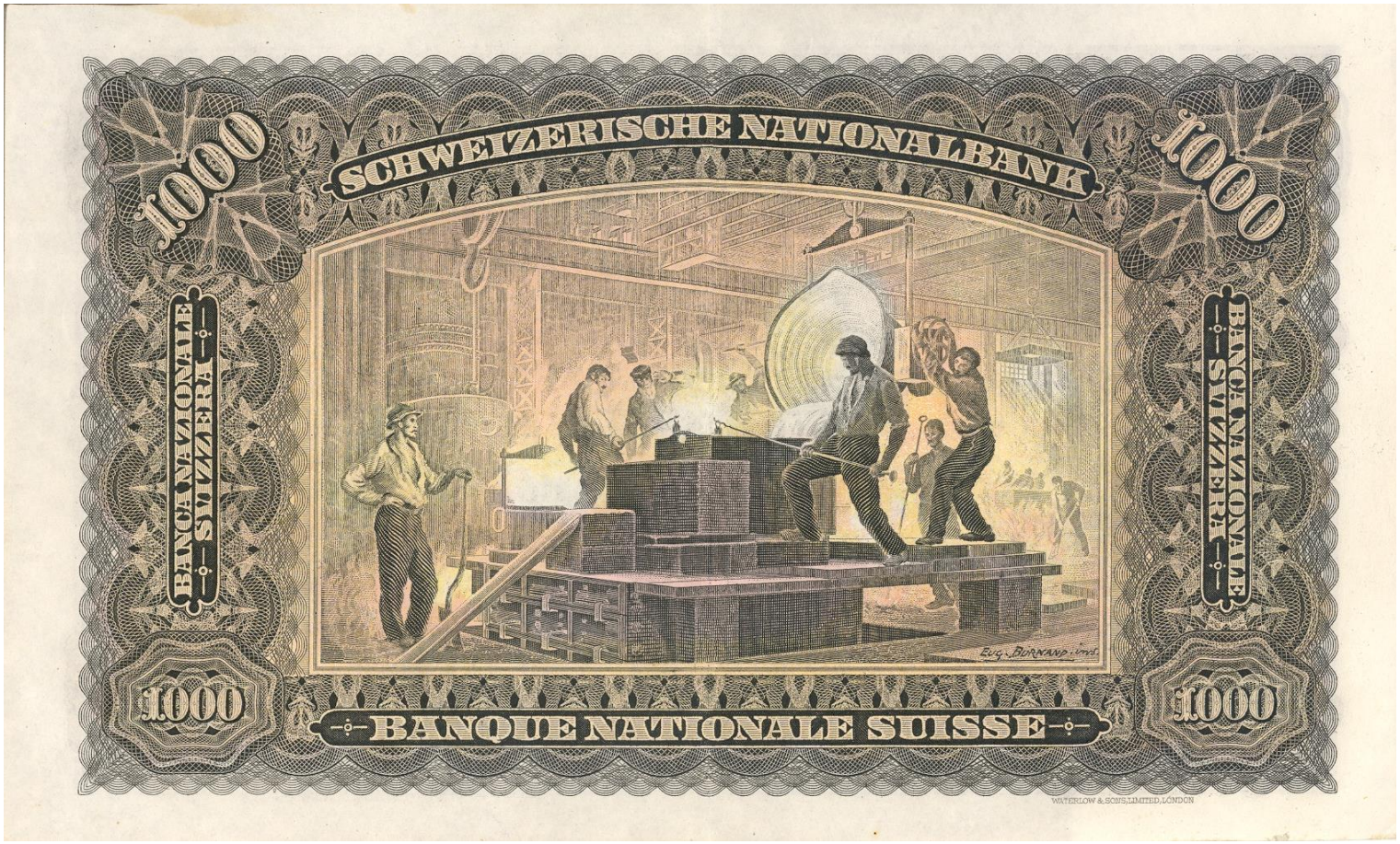
Aircraft dataset (103 WWII warplanes)

10 datasets for initial benchmarking

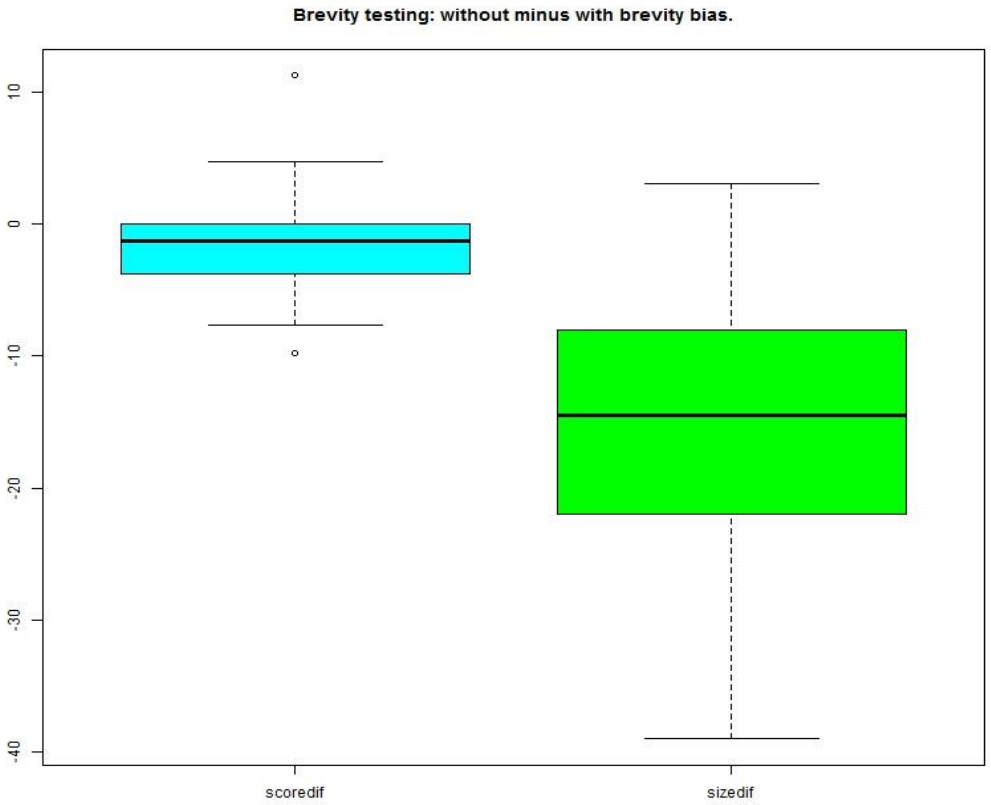
Dataset	rows	cols	target	cats
aircraft	103	16	fighting	2
banknote	206	8	genuine	2
cardiac	113	20	survived	2
digidat	1024	13	numeral	10
echo	208	61	category	2
glasses	214	10	catcode	6
poleseed	210	8	seedtype	3
rand	256	16	dice	6
wine	178	14	cultivar	3
zoobase	101	18	type	7



Swiss 1000 Franc notes (Flury & Riedwyl, 1988)

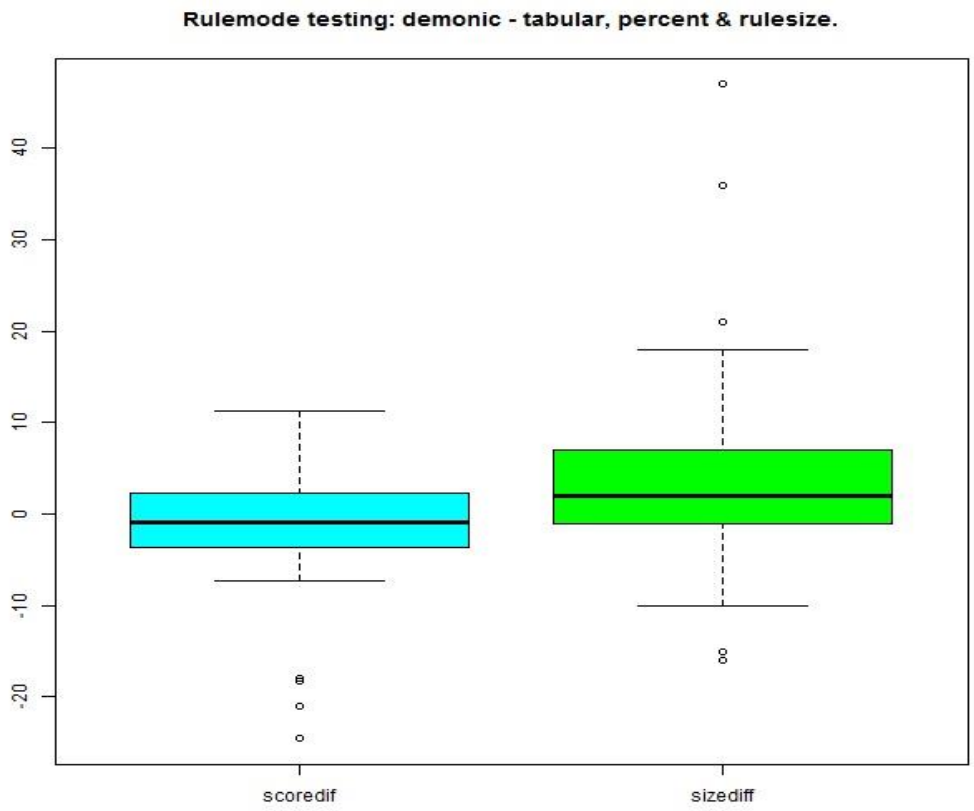


Brevity Bias: paying for parsimony?



- Conclusion:
- Size definitely reduced (mean 45%)
- Slight drop in accuracy (?) probably worth paying

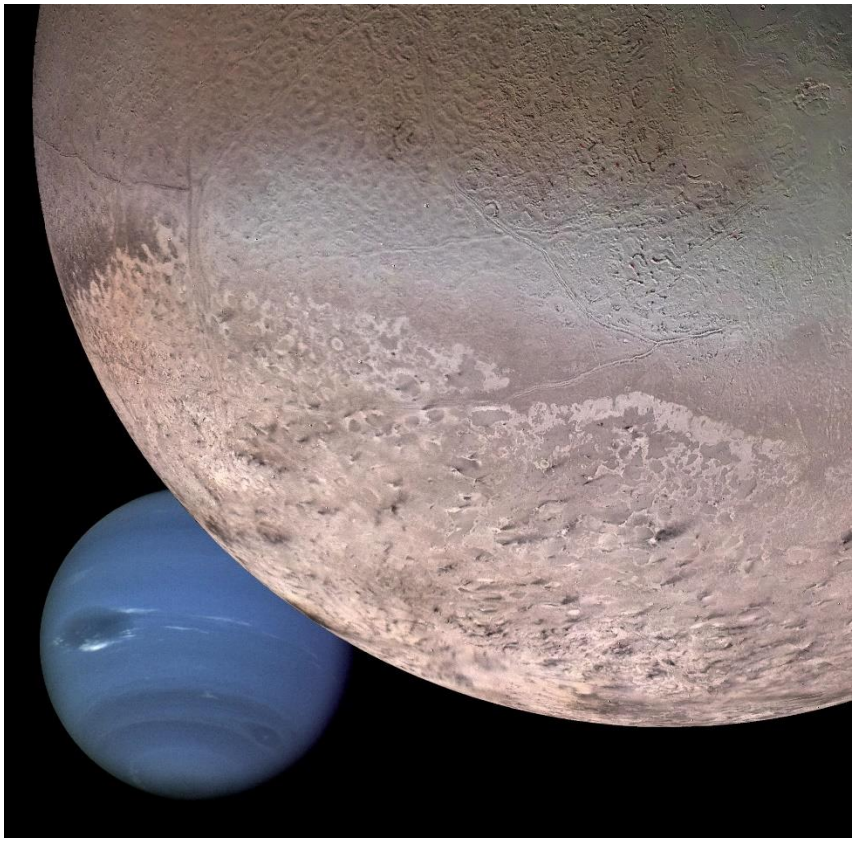
Demonic vs Tabular Rule Linkage



- Conclusion:
- Demonic linkage rules definitely larger, probably less accurate
- To be removed?

What about RUNSTER?

- Regression
- Using
- Naturalistic
- Selection
- To
- Evolve
- Rules



Moons & Planets

```
training data : c:\beagwork\ip\sats.dat
creation date : Fri Mar 11 14:38:31 2016
rule mode : reg1
33 6
reltime
[0.038244514, 0.058553387, 0.073129252, 0.093537415,
0.19047619, 0.247585724, 1, 2.335759272, 4.4880863, 61.25]
[33, 4.488086270393939, 12.949559136991365,
0.07360800099999999]
$
( $Root reldist * reldist )
$
0  [1, 0.219030957, 12.949559136991365]
[-0.0017889003012414605, -4]
$
```

Test on ‘planets’

```
dateline   Fri Mar 11 14:54:21 2016
progname   C:\beagwork\p3\tree.py
id         C:\beagwork\parapath\plansats.txt
testdat    c:\beagwork\ip\planets.dat
targvar    reltime

====holdout trial :
rank  safeness case  name                      pred:true      cellsize   abdsiff  diffsqrd
   1      0.45    7  Uranus                  84.4991 + 84.01          33      0.49      0.24
   2      0.17    8  Neptune                165.1079 + 164.8          33      0.31      0.09
   3      0.15    6  Saturn                  29.4725 + 29.4724        33      0.00      0.00
   4      0.13    5  Jupiter                 11.8679 + 11.87          33      0.00      0.00
   5      0.12    4  Ceres                    4.5954 + 4.6            33      0.00      0.00
   6      0.12    3  Mars                     1.8804 + 1.8805         33      0.00      0.00
   7      0.12    2  Terra Firma              1.0002 + 1              33      0.00      0.00
   8      0.11    1  Venus                     0.615 + 0.6152         33      0.00      0.00
   9      0.11    0  Mercury                  0.2406 + 0.2408         33      0.00      0.00
  10      0.08    9  Pluto                  249.1291 + 247.87         33      1.26      1.59
  11      0.03   10  Eris                   565.7393 - 559            33      6.74     45.42
++++++++++-

'success' percentage = 90.91
pearson correlation between predicted & true vals = 1.0
spearman rank-correlation between predicted & true vals = 1.0
mean abs.error = 0.8003
mean error ^ 2 = 4.3034
```

Or in R, if you prefer

```
runster_regrule = function (vals,stab) {
            ##  input vals should be a 1-row dataframe with appropriate colnames.
            ##  target variable is reltime.
            ##  rule mode is reg1.

            rule = c() ; bins = c('0','1')
            catlist = c(0.038244514, 0.058553387, 0.073129252, 0.093537415, 0.19047619, 0.247585724, 1,
2.335759272, 4.4880863, 61.25)
            priorvec = c(33, 4.488086270393939, 12.949559136991365, 0.07360800099999999)
            subrules = 1
            ##  compute rule values :
            rule[1] = (beag_root(vals$reldist) * vals$reldist)
            p = 0 ; b = c()
            while (p < subrules) {
                        p = p + 1  ##  early-r, late-py
                        v = rule[p] ; b = c(b,v)  ##  omit if tabular
                        }
            ##  reg1 mode :
            predval = v  ##  omit if tabular
            b = '0' ; cellvals = priorvec  ##  omit if tabular
            cellsize = priorvec[1] ; standev = priorvec[3]  ##  omit if tabular
            smalldif = priorvec[4]  ##  should work for both
            ##  tabular mode :

            return (list(cellcode=b,predval=predval,standev=standev,smalldif=smalldif,cellsize=cellsize))
            }
##  regression rule ends.
```

What next?

- For me
  - Plenty of software engineering
  - (plus documentation!)
  - (with help?)
  - “None of the above” categorization
- Whither GP?
  - Genetic bottleneck?
  - Mass extinction?!

4 billion years of field testing can't be bad. (Can it?)

Stuck at a local optimum?

- Why do I have to rewrite BEAGLE; why didn't it rewrite itself? It's had over 30 years!
- Why does AlphaGo use simulated neural nets and reinforcement learning?
- Why doesn't Google-Translate learn through GP?

3 Billion Years of Stromatolites

The Task: Dealing with Exponentially Hard Problems in Polynomial Time

- Fitness / Feedback:
  - Inconsistent
  - Sporadic
  - Multi-dimensional
- Robust Representations:
  - Andreas Wagner's genotype networks

Ecosystemic optimization?

- “Coral Reef Computing”?
- (before the door closes ;-)

Would success be worse than failure?

- Big Business + Big Data = Big Trouble

Thank you for your attention.

- ☺

Recommended reading

Banzhaf, W. et al. (1998). Genetic Programming: an Introduction. Morgan-Kaufmann.

Eiben, A.E. & Smith, J.D. (2003). Introduction to Evolutionary Computing. Springer-Verlag

Goldberg, D.E. (1989). Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley.

Holland, J.H. (1975). Adaptation in Natural and Artificial Systems. University of Michigan Press.

Koza, J.R. (1992). Genetic Programming. MIT Press.

Poli, R. et al. (2008). A Field Guide to Genetic Programming. www.gp-field-guide.org.uk

Wagner, A. (2011). The Origins of Evolutionary Innovation. Oxford: O.U.P.

Websites

- http://en.wikipedia.org/wiki/Evolutionary_computation
- http://www.cse.dmu.ac.uk/~rij/gafaq/top.htm
- http://www.genetic-programming.org/
- http://www.ra.cs.uni-tuebingen.de/software/JCell/tutorial/ch03s05.html
- http://bionik.tu-berlin.de/institut/
- http://cswww.essex.ac.uk/staff/poli/gp-field-guide/toc.html
- http://www.cems.uwe.ac.uk/~jsmith/ecbook/ecbook.html
- http://www.ieu.uzh.ch/wagner/research.html

Ancient versus Modern (binary benchmarks; golden ratio train/test)

Dataset	BEAGLE85 percent correct on test data	BEAGLE85 rule size	Py BEAGLE percent on test data (median of 5 runs)	Py BEAGLE median rule size	BEAGLE85 beats how many (of 5)?	RPART tree % on test data
Aircraft (bombing > 0)	87.18	14	76.92	4	5	71.79
Banknote (genuine > 0)	98.73	8	98.73	5	2.5	98.73
Cardiac (survived = 1)	67.44	32	69.77	9	1.5	62.79
Echo (Category > 15000) "rock" vs "mine"	70.89	28	72.15	18	2.5	69.62

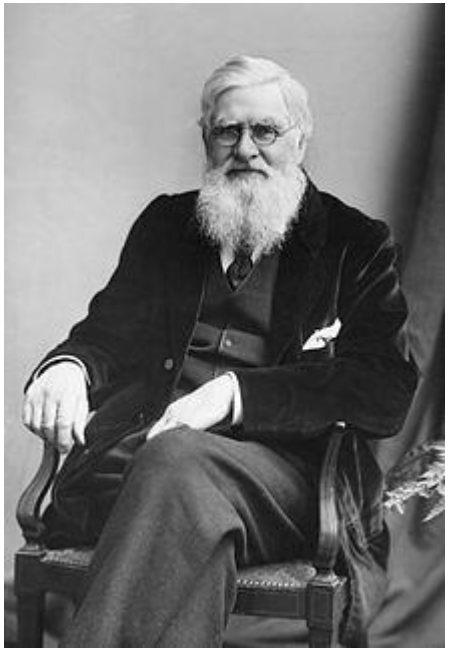
Dealing with unseen categories

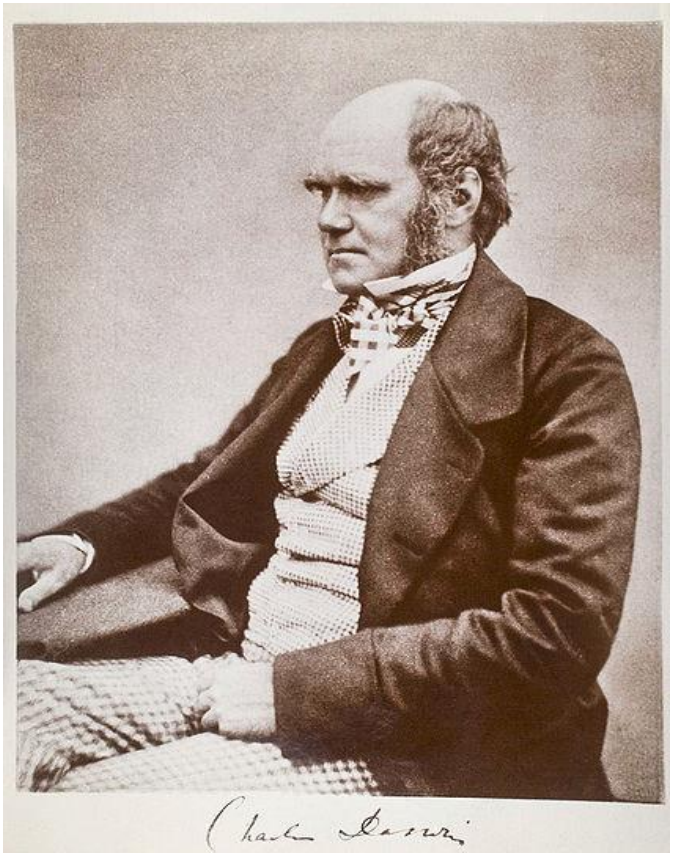
```
dateline   Fri Mar 18 15:05:57 2016
progname   C:\beagwork\p3\leaf.py
id         C:\beagwork\parapath\botlabs.txt
testdat    c:\beagwork\ip\bottest.dat
targvar    catname

====holdout trial :
```

rank	strength	case	textname	pred:true	cellsize	predvals	
1	0.97	2	saltaire_pride.txt	beer + beer	43	0.98	0.02
2	0.97	1	greene_king_ipa.txt	beer + beer	43	0.98	0.02
3	0.97	0	bud_strong.txt	beer + beer	43	0.98	0.02
4	0.96	21	paris_street.txt	wine + wine	44	0.02	0.98
5	0.96	20	la_paz_merlot.txt	wine + wine	44	0.02	0.98
6	0.96	19	ringtons_extra_fresh	wine ? soft	44	0.02	0.98
7	0.96	18	folkingtons_elderflo	wine ? soft	44	0.02	0.98
8	0.96	14	belvoir_elderflower_	wine ? soft	44	0.02	0.98
9	0.96	13	Theo_749f.txt	wine ? misc	44	0.02	0.98
10	0.96	12	sonn109.txt	wine ? misc	44	0.02	0.98
11	0.96	11	rosquijeau_breton_ci	wine ? misc	44	0.02	0.98
12	0.96	10	old_mout_cider.txt	wine ? misc	44	0.02	0.98
13	0.96	9	obertin_calvados_vso	wine ? misc	44	0.02	0.98
14	0.96	8	morrisons_lowalcohol	wine ? misc	44	0.02	0.98
15	0.96	7	montano_sidro_italia	wine ? misc	44	0.02	0.98
16	0.96	6	Lincoln1863Gettysbur	wine ? misc	44	0.02	0.98
17	0.96	5	kopparberg_cider.txt	wine ? misc	44	0.02	0.98
18	0.96	4	aspall_suffolk_cyder	wine ? misc	44	0.02	0.98
19	0.86	3	aspall_golden_malt_v	beer ? misc	9	0.93	0.07
20	0.68	17	fentimans_victorian_	beer ? soft	3	0.84	0.16
21	0.68	16	fentimans_ginger_bee	beer ? soft	3	0.84	0.16
22	0.68	15	fentimans_dandelion_	beer ? soft	3	0.84	0.16

```
+++++???????????????
```

Naturalists' Selection





The (Patrick?) Matthew Effect

- The “Matthew Effect”
  - [Stigler’s Law of Eponymy]
- “The differences of Mr Matthew’s view from mine are not of much importance: he seems to consider that the world was nearly depopulated at successive periods, and then re-stocked;” – C. Darwin (1861). *Origin*, third edition, historical sketch.
  - [Punctuated equilibrium?]

The Matthew Effect (chapter & verse)

- 26 His master replied, ‘You wicked, lazy servant! So you knew that I harvest where I have not sown and gather where I have not scattered seed? 27 Well then, **you should have put my money on deposit with the bankers**, so that when I returned I would have received it back with interest. 28 ‘So take the bag of gold from him and give it to the one who has ten bags. 29 For whoever has will be given more, and they will have an abundance. **Whoever does not have, even what they have will be taken from them**. 30 And throw that worthless servant outside, into the darkness, where **there will be weeping and gnashing of teeth**.’ [NIV]

"It has not escaped our notice that the specific pairing we have postulated immediately suggests a possible copying mechanism for the genetic material."

“I believe that if Watson and Crick had not existed, the insights they provided in one single package would have come out much more gradually over a period of many months or years. Dr. B might have seen that DNA is a double-strand helix, and Dr. C might later have recognized the hydrogen bonding between the strands. Dr. D later yet might have proposed a complementary purine-pyrimidine bonding, with Dr. E in a subsequent paper proposing the specific adenine-thymine and guanine-cytosine nucleotide pairs. Finally, we might have had to wait for Dr. G to propose the replication mechanism of DNA based on the complementary nature of the two strands. All the while Drs. H, I, J, K and L would have been confusing the issue by publishing incorrect structures and proposals.” Gunther S. Stent (1972). Scientific American, 227, p. 90.

Why am I here?

- “Truly new things one finds only in one’s youth. Later one becomes more experienced, more famous, and dumber.”
- Albert Einstein to Heinrich Zangger, 6 July 1917. (Tr. V. Tekavec; D. Overbye, 2000.)

An example rule, vintage BEAGLE

```
( SURVIVED =    1.0000)$

(((( CI <    216.0000) | ( SHOCKCAT <    3.0000)) = ( SHOCKCAT <=
4.5000)
) > ( HEIGHTCM <    153.9200))
$     39      8      1     22
(((( YO <   76.0000) >=  UO) <= (( DP >=   51.3976) <> ( UO >=   14.0200)
)) | ( AT <=   54.0000))
$     40     14      0     16
() $  -- from cardiac.dat on 27/02/2016 at 15:22:07
           0.5714      70
000        0.0000      10      0.0519      0.0000
001        1.0000      13      0.1122      7.6923
010        0.0000       6      0.0816      0.0000
011       39.0000      41      0.9422     95.1220
```

An example BEAGLE rule, current style

```
training data : C:\beagwork\op\cardiac_dat1.dat
creation date : Sun Mar  6 11:03:01 2016
rule mode : tabular
70 20
survived
[1, 2]
[0.5729490168751578, 0.42705098312484224]
$
( 63 > ap )
( 9 - uo )
$
00  [16, 6]
01  [19, 5]
10  [7, 0]
11  [3, 14]
[0.44, -3.0]
$
```

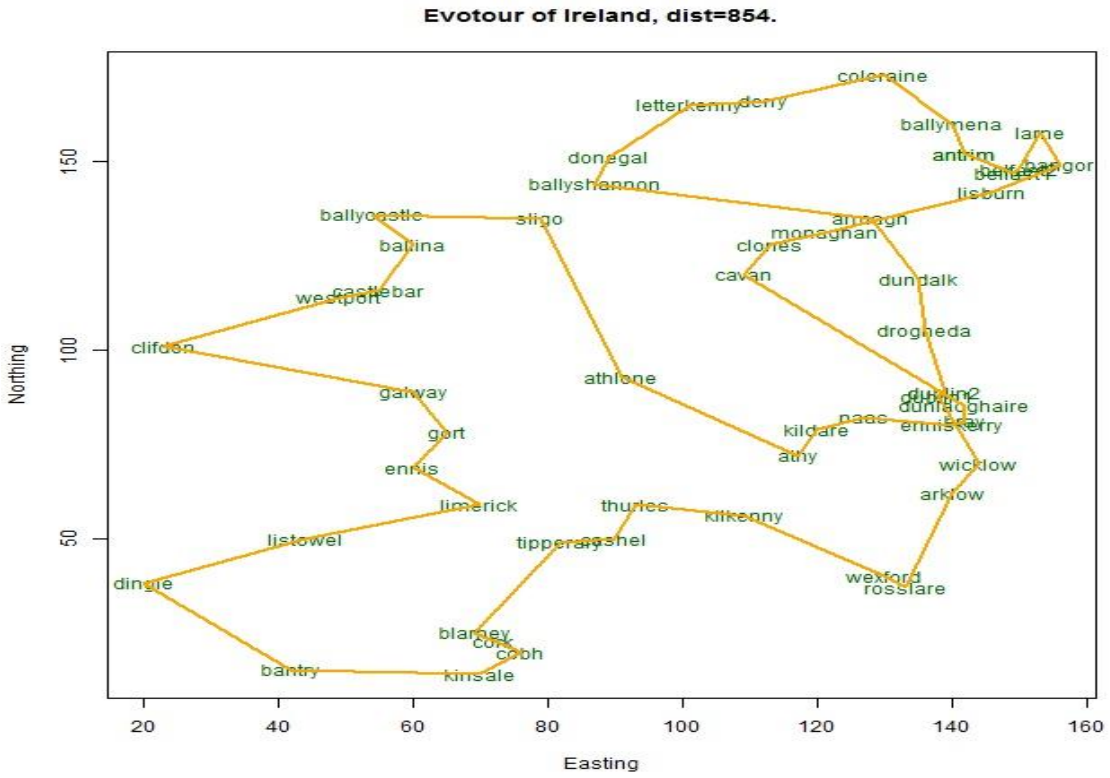
A rule with string tests

```
training data : c:\beagwork\ip\elements.dat
creation date : Mon Mar 14 17:40:26 2016
rule mode : tabular
118 9
Block
['d ', 'f ', 'p ', 's ']
[0.296125087484338 6, 0.24775602355205217, 0.28092892463013924, 0.17518996433347006]
$
( ( Atomic.no. > 3.3446655 ) \ ( Group > 2.13103 ) )
( Group < 12.9241236 )
( ( Description ? `Transition metal`) + ( Description ? `?`) )
$
000  [0, 0, 0, 1]
001  []
010  [0, 0, 0, 13]
011  []
100  [0, 0, 31, 0]
101  [0, 0, 5, 0]
110  [2, 28, 0, 0]
111  [38, 0, 0, 0]
[0.9743589743589743, -5.0]
$
```

Bottlabs rule

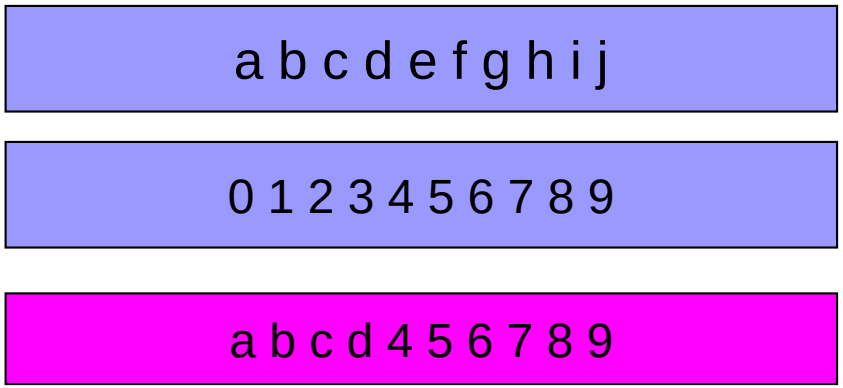
```
Resultant rule from all training cases :
(Rule size = 2 )
training data : c:\beagwork\ip\bottrain.dat
creation date : Sun Mar 13 14:59:24 2016
tabular
18 161
catname
['beer', 'wine']
[0.5278640450004206, 0.4721359549995794]
$
brewed
barley
$
00  [0, 44]
01  [9, 0]
10  [3, 0]
11  [43, 0]
[1.0, -1.0]
$
```

Round the emerald isle



Crossover operators

- Point crossover :



- Uniform crossover :