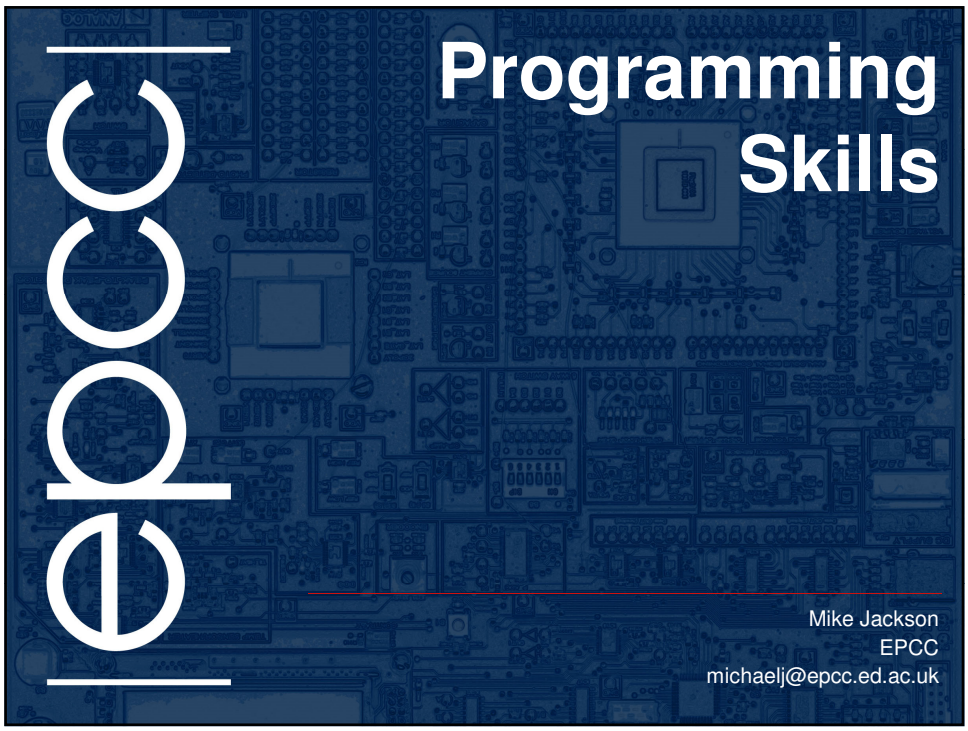


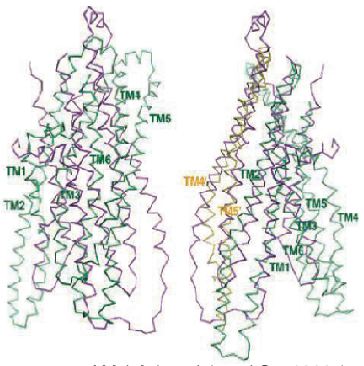
Scientific programming

epcc

- Model science and problems to be solved
  - Simulate scientific problem or environment
  - Input real data
  - Output simulated data
- Explore universe through simulation rather than experimentation
  - Test theories
  - Predict or validate experiments
  - Simulate “untestable” science
- Reproduce “real world” in computers
  - Simplified
  - Restricted dimensions and timescales

Correctness

epcc

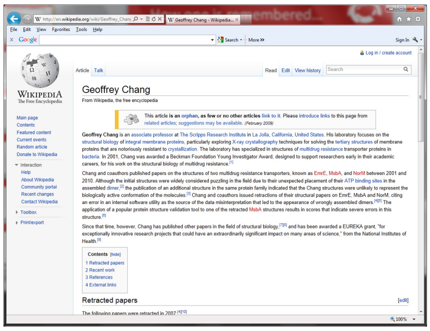


The structures of MsbA (purple) and Sav1866 (green) overlap little (*left*) until MsbA is inverted (*right*)

Dawson, R.J.P. and Locher, K.P. "Structure of a bacterial multidrug ABC transporter", Nature. 443(7108), pp180-5, September 2006.

"Chang was horrified to discover that a homemade data-analysis program had flipped two columns of data"

Miller, G. "A Scientist's Nightmare: Software Problem Leads to Five Retractions", Science 314(5807), pp1856-1857, December 2006. doi:10.1126/science.314.5807.1856.



https://en.wikipedia.org/wiki/Geoffrey_Chang

Comprehensibility

epcc

```
 through each SNP of interest
(my $x = 0; $x < scalar @pos; $x++)

and then each downstream SNP of interest
or(my $y = $x+1; $y < scalar @pos; $y++)

 #if SNPs within our chosen distance (500kb) and both present in the haplotypes file
 if((!($trait[$x] eq $trait[$y])) && (abs($pos[$x] - $pos[$y]) <= 500000) && (exists($legArrayPos{$pos[$x]})) && (exists($legArrayPo
 {
   my $snp1ArrayPos = "";
   my $snp2ArrayPos = "";
   my $snp1All = "";
   my $snp2All = "";

   #create output file for this SNP pair
   my $filename = "ConditionedResults2/$chr[$x].$pos[$x]-$pos[$y].EHH.GBR.2.txt";
   print "$filename\n";
   unless (-e $filename) {
     open(OUT, ">$filename");

     ###################CHANGE THESE IF NOT FOCUSING ON SECOND SNP#########################
     my $start = $pos[$y]-500000;
     if ($start < 1) {
       $start = 1;
     }
     my $end = $pos[$y]+500000;
     if ($end > $chrLengths{$chr[$x]}) {
       $end = $chrLengths{$chr[$x]};
     }
```

Courtesy of Carole Goble

Efficiency

- “More computing sins are committed in the name of efficiency (without necessarily achieving it) than for any other single reason – including blind stupidity”
  - Wulf, W. “A Case Against the GOTO”, Proceedings of the 25th National ACM Conference, pp791-97, August 1972.
- "Programmers waste enormous amounts of time thinking about, or worrying about, the speed of noncritical parts of their programs”
- “these attempts at efficiency actually have a strong negative impact when debugging and maintenance are considered”
- “premature optimization is the root of all evil.“
  - Knuth, D. “Structured Programming With Go To Statements”, Computing Surveys 6(4), pp261-301, December 1974.

Efficiency and optimization

- What is preferable?
  - An optimized code that produces incorrect results?
  - A non-optimized code that produces correct results?
  - Which is the most efficient use of resources?
- How do we prevent ourselves introducing bugs when we’re optimizing and parallelizing?
- Isn’t our time more valuable than a computer’s time?
- Hardware life < software life < our life
  - Computational Chemistry CASTEP code originates from 1990s.

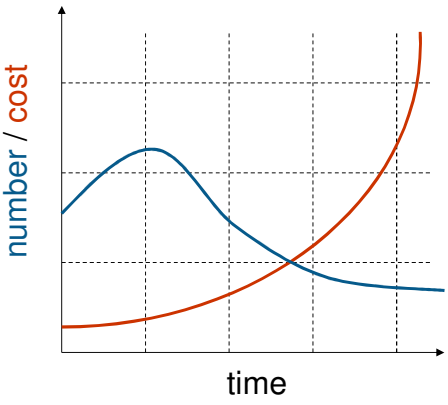
Technical debt

"large preponderance of design errors over coding errors on large-scale projects, not only with respect to numbers of errors, but also with respect to the relative time and effort required to detect them and correct them"

Boehm, B.W. McClean, R.K. and Urfrig, D.B. ""Some experience with automated aids to the design of large-scale reliable software", IEEE Transactions on Software Engineering 1(1), pp125-133, March 1975. doi:10.1109/TSE.1975.6312826

1-10-100 rule

"if it takes one unit of costs or effort to complete a job correctly,
it will take 10 times that effort to correct an error before it reaches the customer.
Once it has reached the customer, it will take 100 times the cost and effort to correct the situation, not to mention the loss of customer goodwill."



Technical debt

What is the most efficient way to detect bugs?

- Rigorous inspections can remove 60-90% of errors before the first test is run
  - Fagan, M.E. "Design and Code inspections to reduce errors in program development", IBM Systems Journal 15(3), pp182–211, NN 1975. doi:10.1147/sj.153.0182.
- The first review and first hour matter most
  - Cohen, J. "Best Kept Secrets of Peer Code Review", Smart Bear Inc, 2006, ISBN-10: 1599160676. ISBN-13: 978-1599160672.

Programming Skills

- Comprehensibility – write code that is readable, understandable, maintainable
  - Writing programs for people
  - Abstract data types and object-oriented programming
- Efficiency – optimizing yourself
  - Integrated development environments
  - Debugging using the command-line, Eclipse and DDT
  - Version control
  - Automated build
  - Continuous integration
- Correctness – write code that is correct, and can be refactored and optimised safely
  - Unit testing
- Efficiency – optimizing your code
  - Analysis, profiling, performance and timers

Coursework

- 1 – Code Review (20%)
  - Code review of a small program
  - What you submit: an individual report
  - **Deadline: 14:00 Thursday 15/10 (week 4)**
- 2 – Development (60%)
  - Group-based code design, development and testing in C, C++, Fortran, Java or Python
  - Groups are semi-randomly assigned on Tuesday 13/10 based on your preferred languages
  - What you submit: Source code and documentation plus non-assessed contribution questionnaire
  - **Deadline: 14:00 Friday 06/11 (week 7)**
- 3 – Performance (20%)
  - Performance experiments of the code you produced with your group in Coursework 2 above.
  - What you submit: Individual report
  - **Deadline: 14:00 Friday 27/11 (week 10)**

Coursework

Year	Lowest	Highest	Average	Standard Deviation
2014	57	81	71.55	7.56
2013	60	82	72.84	5.82
2012	56	83	69.29	6.79
2011	57	87	70.65	9.59

- Please feel free to ask questions to the class or the course organiser directly
  - Answers to general questions will be anonymised and sent to the class
- Provisional marks returned within 15 working days
  - Ideally 10!

Labs

- Computational Physics Lab 1028
  - ph-cplab.ph.ed.ac.uk
- MSc in High Performance Computing / High Performance Computing and Data Science / Theoretical Physics / Mathematical Physics
  - You should already have a lab account
- Everyone else
  - Your director of studies should arrange for you to get a lab account
  - Or, e-mail sopa-helpdesk@ed.ac.uk
  - Or, (last resort) visit Helpdesk Office, JCMB 4210 Mon-Fri, 09:00-12:30 or 13:30-17:00
  - Before Friday!