SOFTWARE MATTERS:

A **Snapshot** of Software Development Practice in Operations Research

Huseyin Ergin & Mesut Yavuz

Summary

- We have analyzed 2017 INFORMS Journal on Computing papers.

Why?

- **Serious research requires a solid software like a strong building requires a solid foundation.**

We have been there and done that!

- “I copied my whole project and tweaked a bit for another research!”

We have been there and done that!

- “I modified one part of my project but some other part failed for this!”

We have been there and done that!

- “I couldn’t understand a piece of code I have implemented 6 months ago!”

We have been there and done that!

- "I modified my code outside of my norm to satisfy a reviewer's request [in a quick and dirty way]!"

We have been there and done that!

- “I have used external libraries with cryptic (or non existent) documentation!”

We have been there and done that!

- “I tracked down a code of an algorithm in a paper and ended up in a desert!”

Software is important!

- But why?

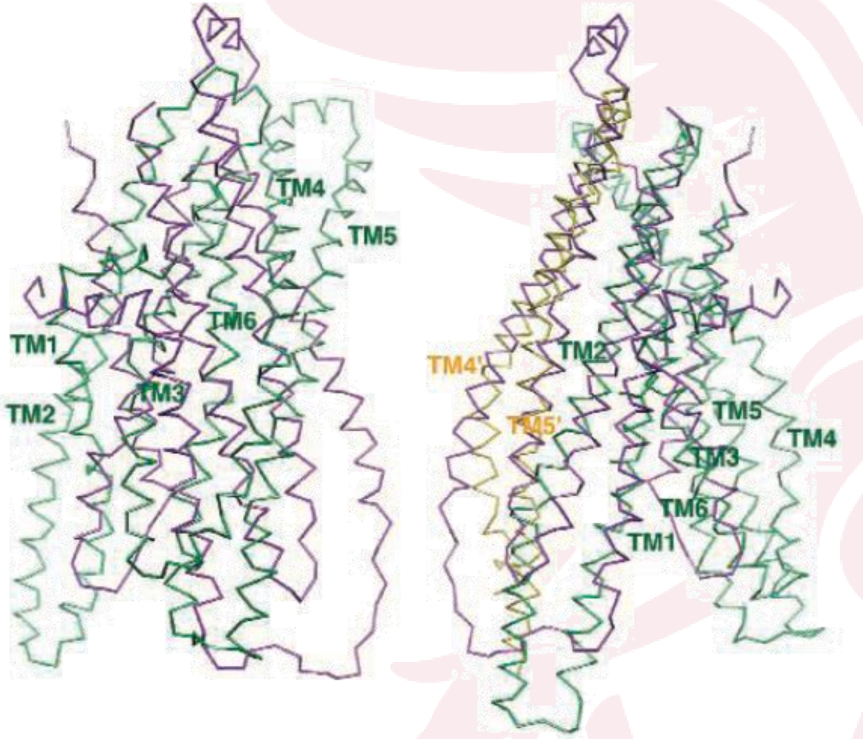
SCIENTIFIC PUBLISHING

A Scientist's Nightmare: Software Problem Leads to Five Retractions

Until recently, Geoffrey Chang's career was on a trajectory most young scientists only dream about. In 1999, at the age of 28, the protein crystallographer landed a faculty position at the prestigious Scripps Research Institute in San Diego, California. The next year, in a ceremony at the White House, Chang received a Presidential Early Career Award for Scientists and Engineers, the country's highest honor for young researchers. His lab generated a stream of high-profile papers detailing the molecular structures of important proteins embedded in cell membranes.

Then the dream turned into a nightmare. In September, Swiss researchers published a paper in *Nature* that cast serious doubt on a protein structure Chang's group had described in a 2001 *Science* paper. When he investigated, Chang was horrified to discover that a homemade data-analysis program had flipped two columns of data, inverting the electron-density map from which his team had derived the final protein structure. Unfortunately, his group had used the program to analyze data for

2001 *Science* paper, which described the structure of a protein called MsbA, isolated from the bacterium *Escherichia coli*. MsbA belongs to a huge and ancient family of molecules that use energy from adenosine triphosphate to transport molecules across cell membranes. These so-called ABC transporters perform many



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

Sciences and a 2005 *Science* paper, described EmrE, a different type of transporter protein.

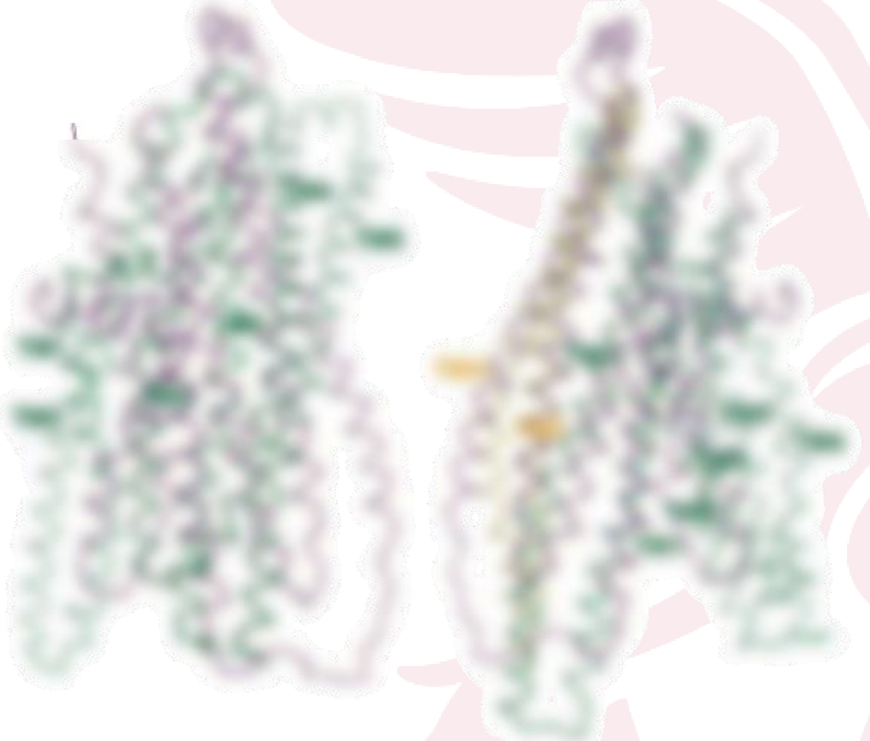
Crystallizing and obtaining structures of five membrane proteins in just over 5 years was an incredible feat, says Chang's former postdoc adviser Douglas Rees of the California Institute of Technology in Pasadena. Such proteins are a challenge for crystallographers because they are large, unwieldy, and notoriously difficult to coax into the crystals needed for x-ray crystallography. Rees says determination was at the root of Chang's success: "He has an incredible drive and work ethic. He really pushed the field in the sense of getting things to crystallize that no one else had been able to do." Chang's data are good, Rees says, but the faulty software threw everything off.

Ironically, another former postdoc in Rees's lab, Kaspar Locher, exposed the mistake. In the 14 September issue of *Nature*, Locher, now at the Swiss Federal Institute of Technology in Zurich, described the structure of an ABC transporter called Sav1866 from *Staphylococcus aureus*. The structure was dramatically—and unexpectedly—different from that of MsbA. After pulling up Sav1866 and Chang's MsbA from *S. typhimurium* on a computer screen, Locher says he realized in minutes that the MsbA structure was inverted. Interpreting the "hand" of a molecule is always a challenge for crystallographers,

SCIENTIFIC PUBLISHING

A Scientist's Nightmare: Software Problem Leads to Five Retractions

[illegible] San Diego, California. The next year, in a ceremony at the White House, Chang received a Presidential Early Career Award for Scientists and Engineers, the country's highest honor for young researchers. His lab generated a [illegible]

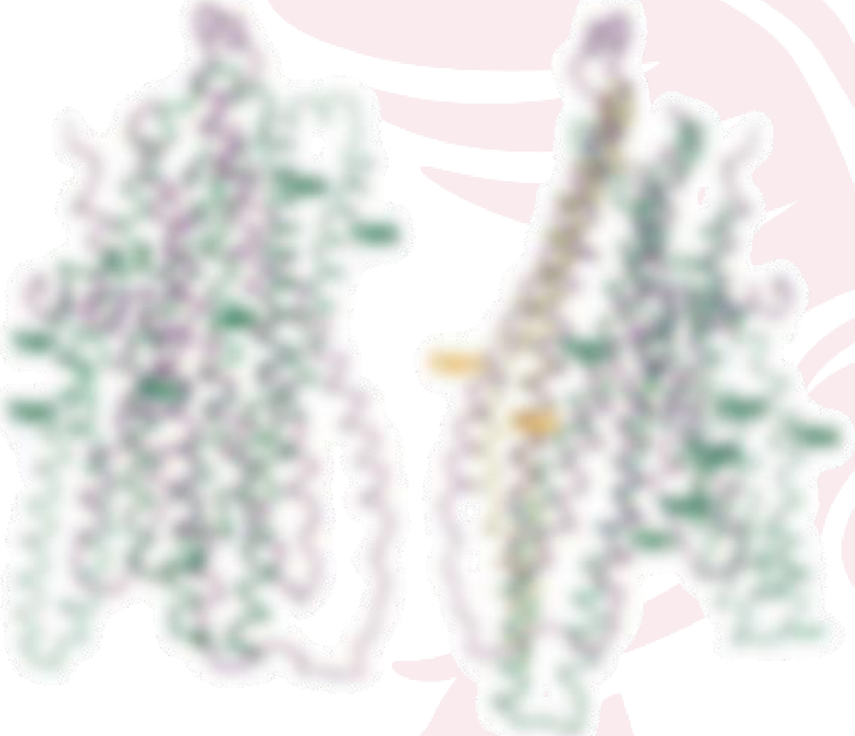


[illegible]

SCIENTIFIC PUBLISHING

A Scientist's Nightmare: Software Problem Leads to Five Retractions

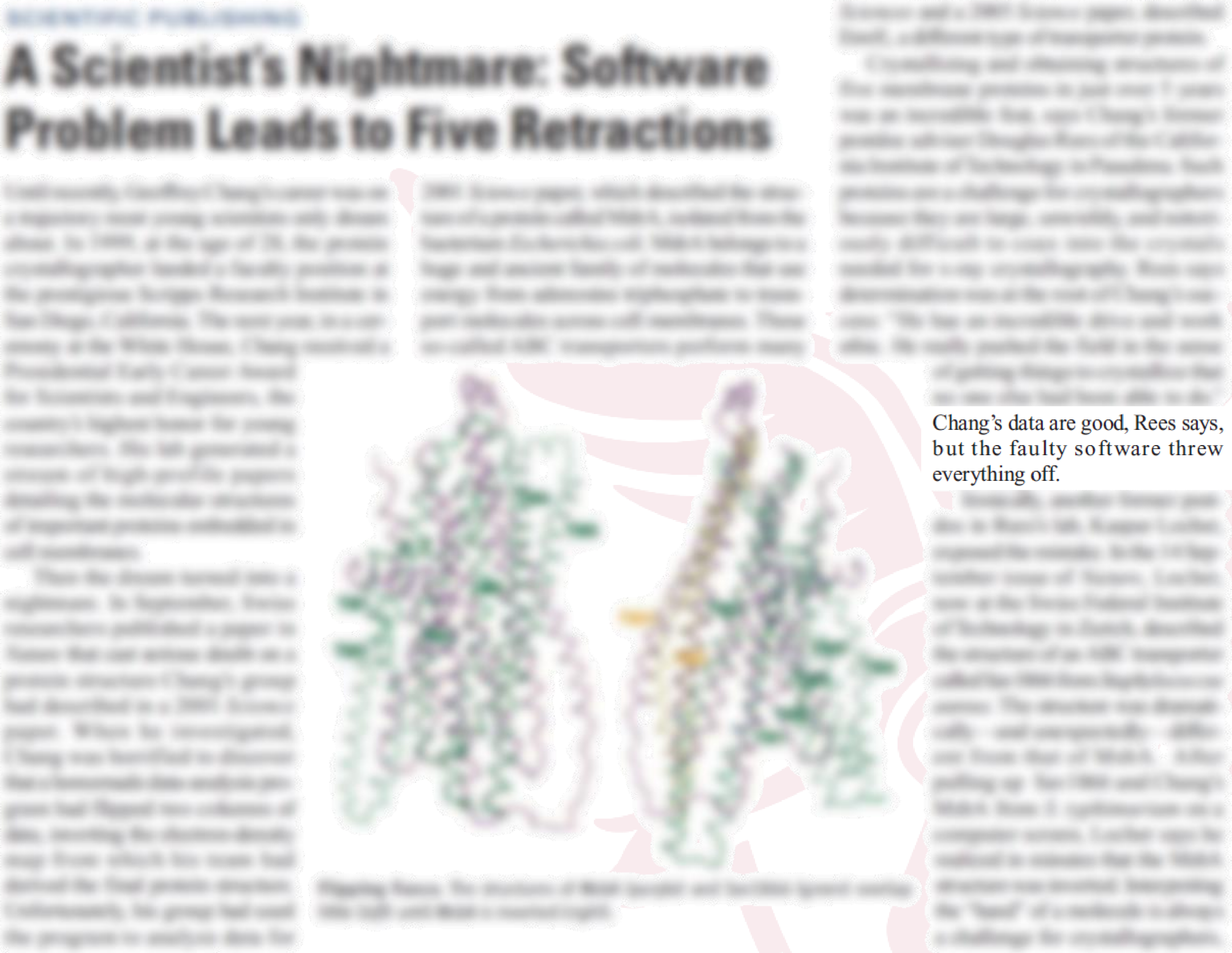
[illegible]



[illegible]

paper. When he investigated, Chang was horrified to discover that a homemade data-analysis program had flipped two columns of data, inverting the electron-density map from which his team had derived the final protein structure.

[illegible]



SCIENTIFIC PUBLISHING

A Scientist's Nightmare: Software Problem Leads to Five Retractions

[illegible]

Chang's data are good, Rees says, but the faulty software threw everything off.

[illegible]

How good can we write software?

- “Scientists typically spend 30% or more of their time developing software.”
- “90% or more of them are primarily self-taught.”
- “Most scientists learn most of what they know about developing software on their own or informally from their peers, rather than through formal training.”

Hannay JE, Langtangen HP, MacLeod C, Pfahl D, Singer J, et al.. (2009) How do scientists develop and use scientific software? In: Proceedings Second International Workshop on Software Engineering for Computational Science and Engineering. pp. 1–8. doi:10.1109/SECSE.2009.5069155.

Replication crisis

- Chang and Li tried to replicate 67 economics papers from 13 well-regarded economics journal and ended up replicating only 49% of the studies with the extended help of the authors.

Andrew C. Chang and Phillip Li. Is Economics Research Replicable? Sixty Published Papers from Thirteen Journals Say 'Usually Not'. SSRN Electronic Journal, 2015(83):1–26, oct 2015.

Replication crisis

- Ioannadis et al. evaluated the replicability of analyses from 18 genetics articles and reproduced only 2 of them entirely and 6 of them with discrepancies.

John P A Ioannidis, David B Allison, Catherine A Ball, Issa Coulibaly, Xiangqin Cui, Aedin C Culhane, Mario Falchi, Cesare Furlanello, Laurence Game, Giuseppe Jurman, Jon Mangion, Tapan Mehta, Michael Nitzberg, Grier P Page, Enrico Petretto, and Vera van Noort. Repeatability of published microarray gene expression analyses. Nature Genetics, 41(2):149–155, feb 2009

It is your turn:

- “I consider myself a good software developer!”

It is your turn:

- Would you use a car that is not designed by engineers?

It is your turn:

- Would you trust a nurse who self-trained on things he/she does?

Why?

- Why do we continue acting like we ALL can do it?
- Why do we produce software without [adequate, formal] training and build our precious research on top of it?

Consider

- How many of you consider yourself a good developer?
- What more could you achieve if there is a bunch of perfect developers on your team?

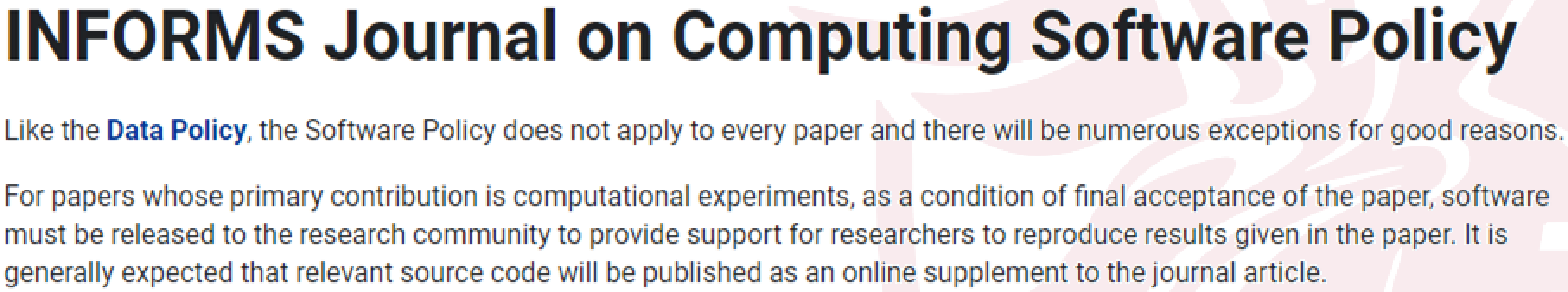
Code policies

- Stodden et al. analyzed 170 journals regarding code sharing and supplemental material policies and found out that during the consecutive years they did the study, the code adoption policy increased 30% by the journals.

Victoria Stodden, Peixuan Guo, and Zhaokun Ma. Toward reproducible computational research: An empirical analysis of data and code policy adoption by journals. PLOS ONE, 8(6):1–8, 06 2013

INFORMS JoC Software Policy

- How many of you have known there was a policy?
  - https://pubsonline.informs.org/page/ijoc/softwarepolicy



INFORMS Journal on Computing Software Policy

Like the **Data Policy**, the Software Policy does not apply to every paper and there will be numerous exceptions for good reasons.

For papers whose primary contribution is computational experiments, as a condition of final acceptance of the paper, software must be released to the research community to provide support for researchers to reproduce results given in the paper. It is generally expected that relevant source code will be published as an online supplement to the journal article.

Statistics

- Now, let’s look at some statistics!

Meta: year

2017

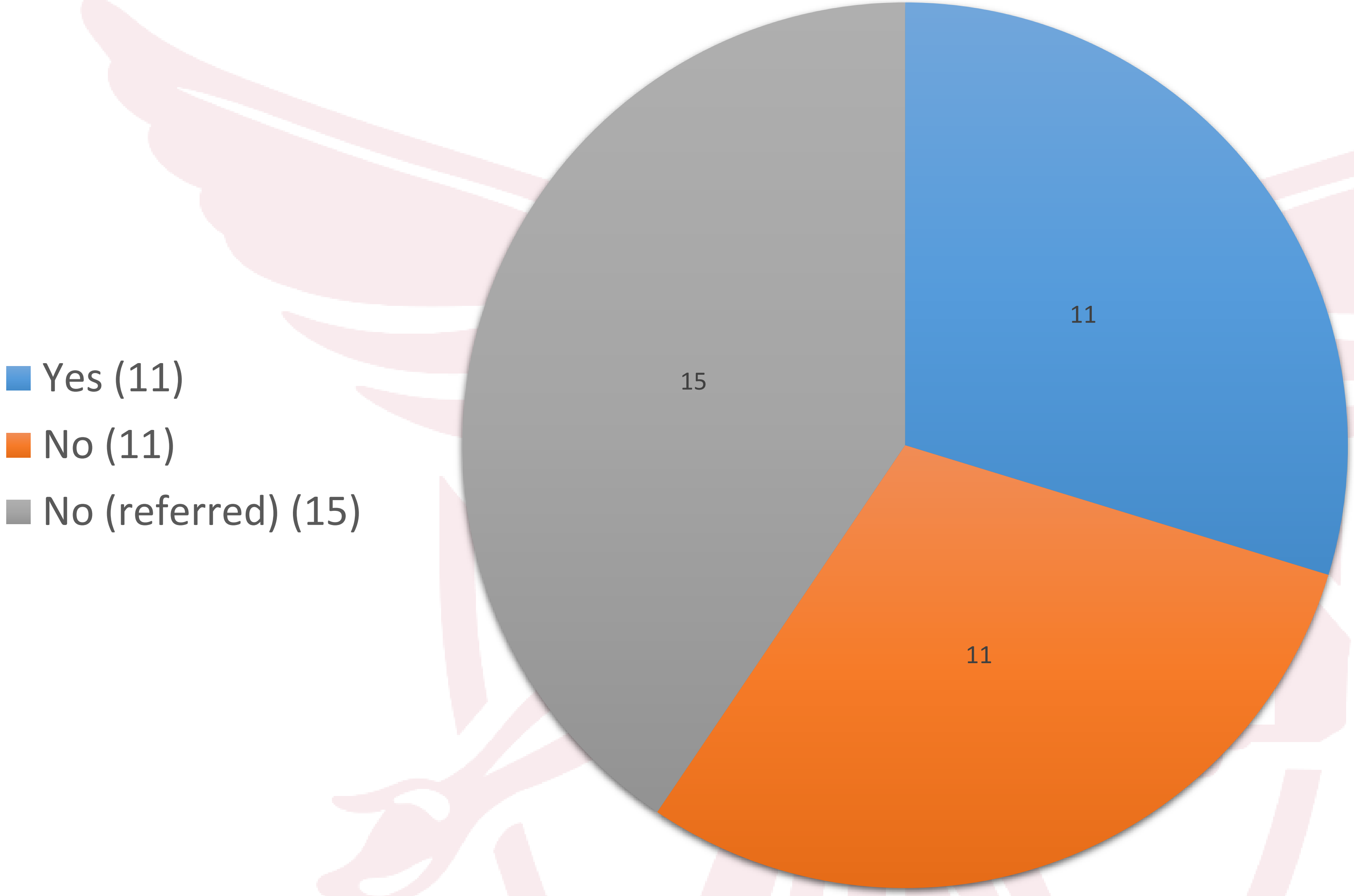
Meta: number of papers

37

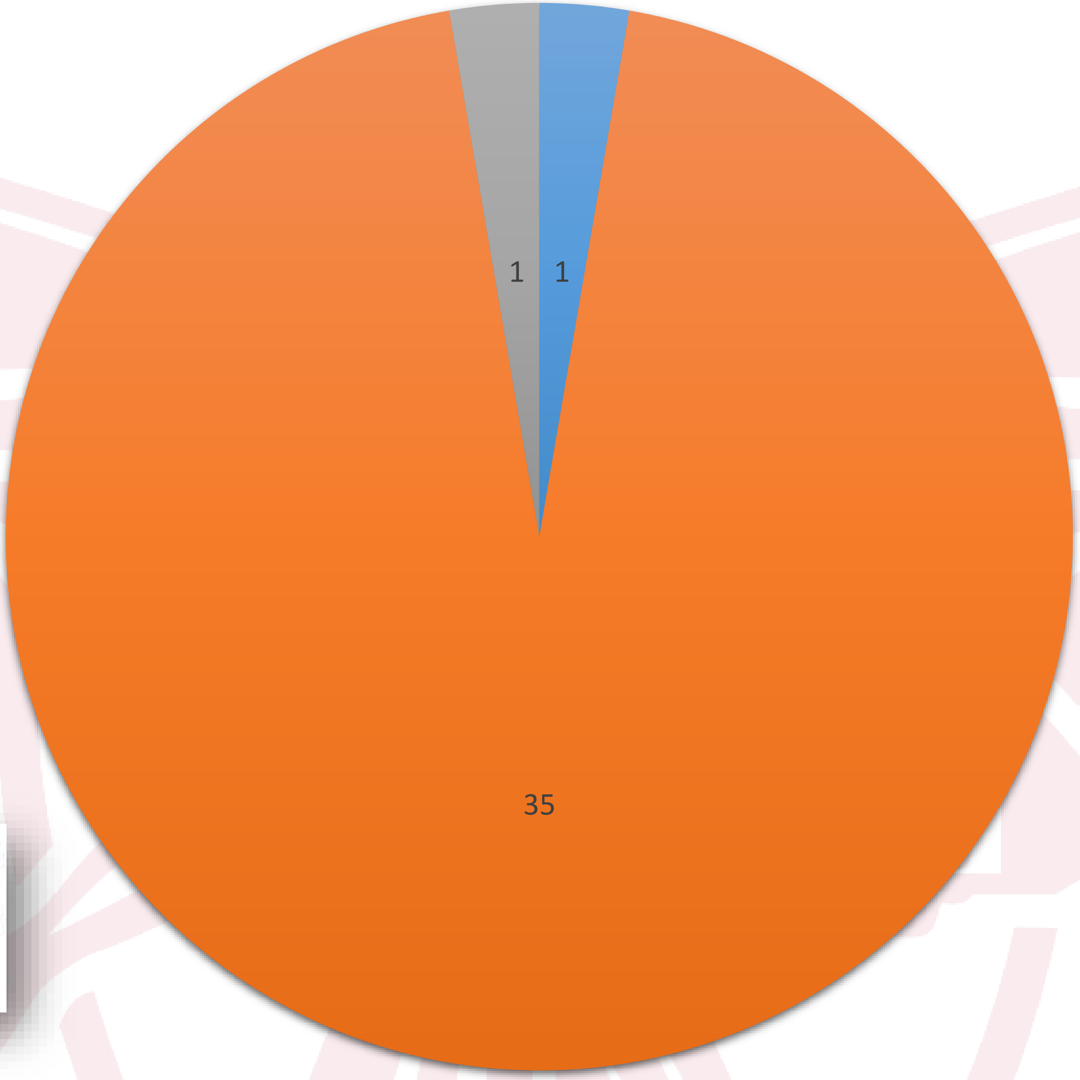
Meta: collected data

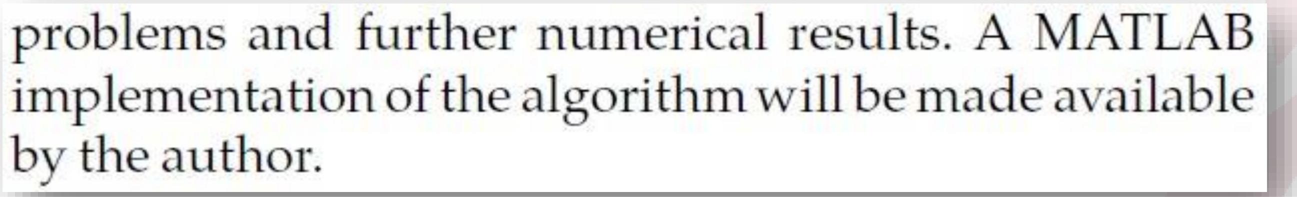
Data availability
Code availability
Programming language
Pseudo-code
Comparison
Software practices
Technical information

Stat: data availability

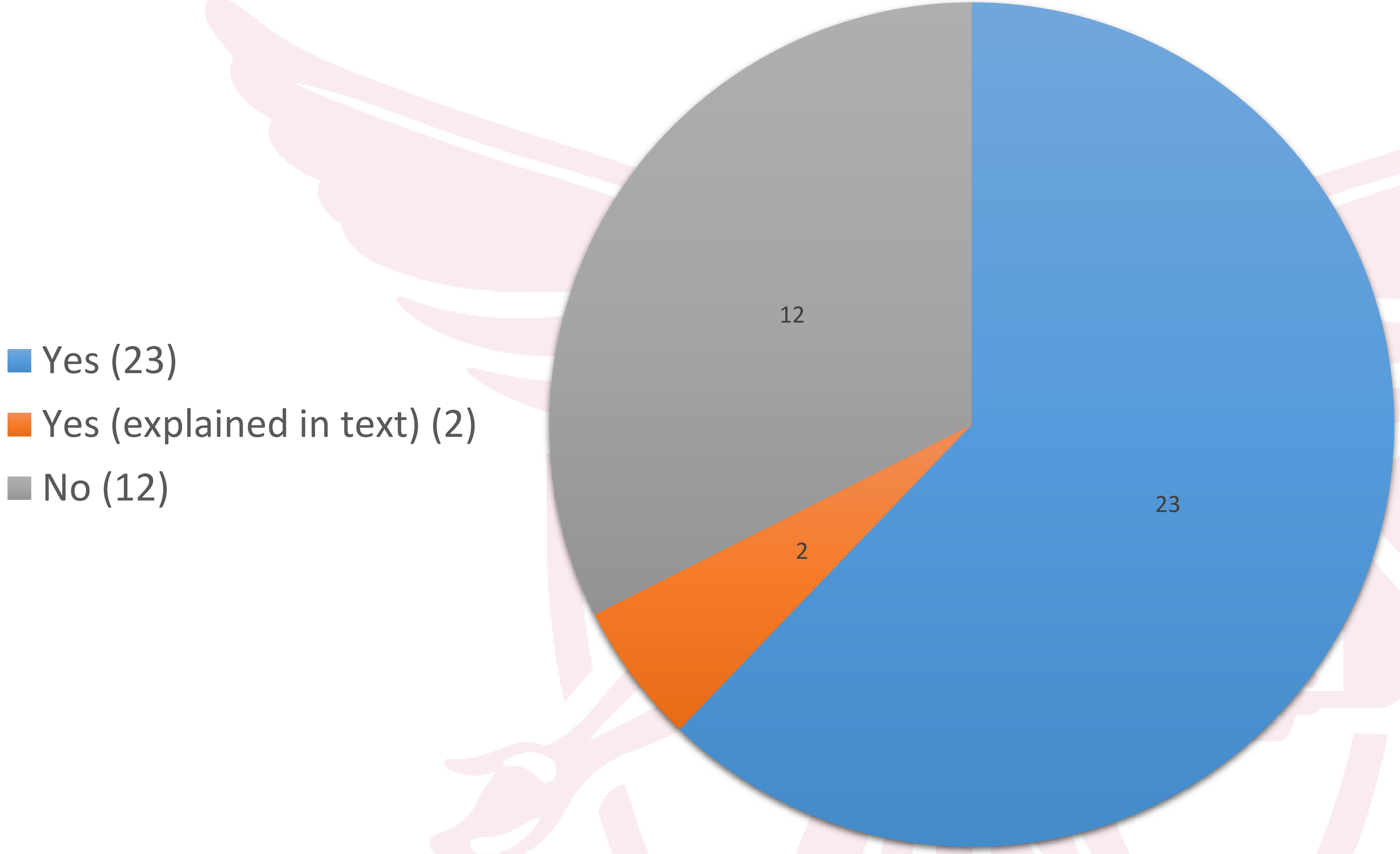


Stat: code availability

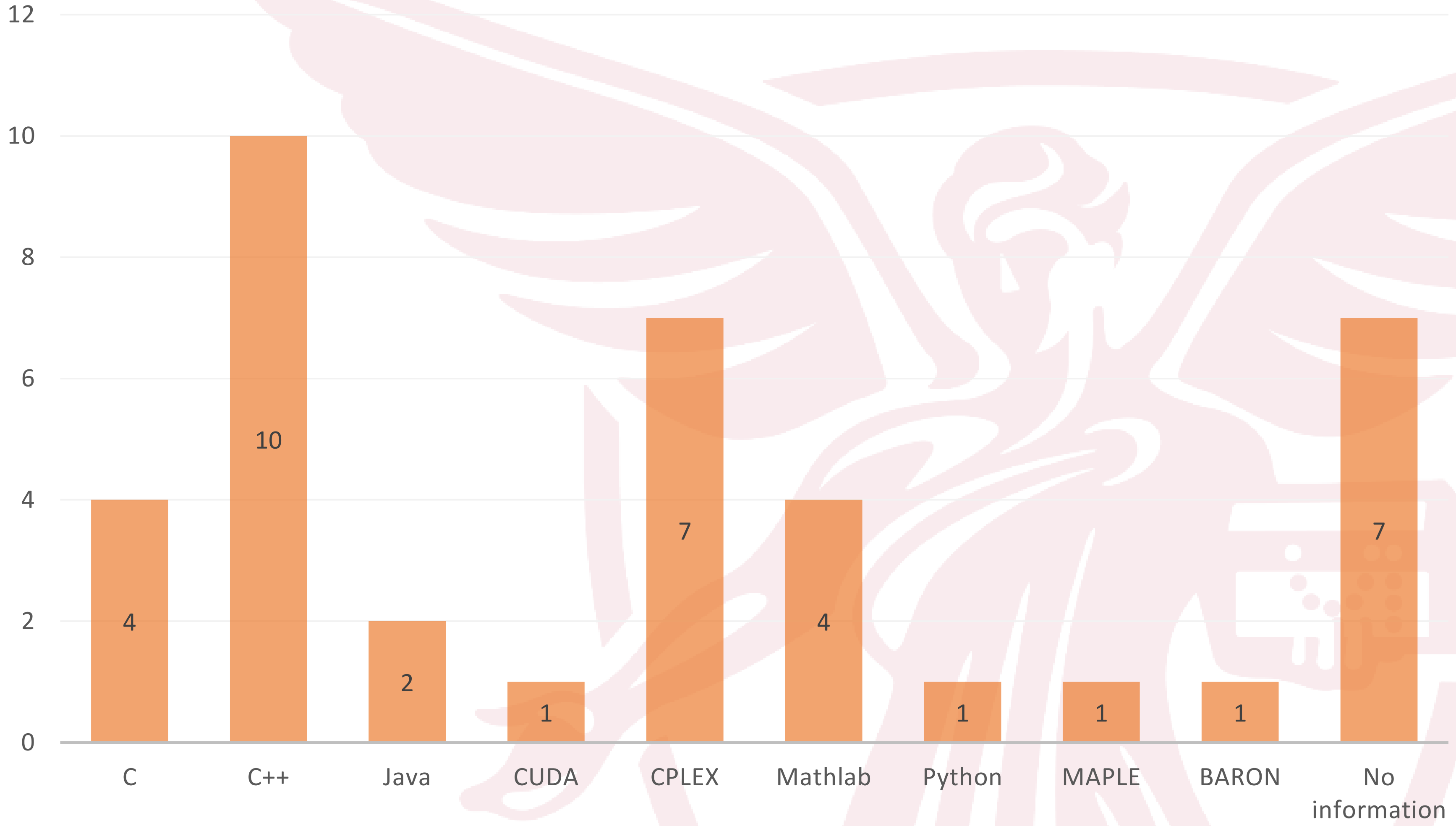



problems and further numerical results. A MATLAB implementation of the algorithm will be made available by the author.

Stat: pseudo-code



Stat: programming languages



Stat: papers mentioning software practices

0/37

Stat: computers

Intel Xeon E3 1,240 V2 processor at 3.40GHz and 24 GB Ram (win7)

Intel Xeon X5650 2.67 GHz and 24 GB Ram (openSUSE)

Intel Xeon E3-1220 3.1 GHz and 16 GB Ram

Intel Core i5-2430M CPU, 16GB Ram, (win7 64bit)

3.1 GHz iMAC with 16 GB ram

Dell Precision T1650 workstation with 3.3. GHz intel core i3-212- CPU, 3.7 GB Ram

Intel Core2duo @1.65 GHz with 8GB DDR2 RAM

64-bit Xeon X5650 2.66 GHz

Intel core 2.8 GHZ and 16 GB Ram

Intel Xeon ES-2637 3.5 GHz with 128 GB Ram, Linux oracle server

Stat: papers with not computer info

7/37

You are not alone!

- These problems exist in a lot of other fields!

Some suggestions

- More collaboration with domain experts

Some suggestions

- Modularity and unit testing

Some suggestions

- Version control

Some suggestions

- Documentation while coding

Some suggestions

- Publish your most frequent codes as libraries and fully test them

Some suggestions

- Include examples

Some suggestions

- Put an immutable copy of your code as a supplement

Some suggestions

- Code reviews

What’s in it for you?

- Shorter development times

What’s in it for you?

- Easily switch between algorithms on the same data

What’s in it for you?

- Focus on your primary area of expertise [rather than code development]

What’s next?

- We are planning to extend this study with active participation of authors.
- A survey in the upcoming weeks that wants to learn some insider information about YOU!
- We will distribute it through various channels.
  - Email us directly:
    - Huseyin Ergin: hergin@bsu.edu Mesut Yavuz: myavuz@culverhouse.ua.edu