# My take on Getting Started with Research
# http://tandy.cs.Illinois.edu/warnow-reu.pdf

Tandy Warnow

# What is research, and why do it?

- Research is trying to understand something
- Research is tremendous fun – for me, it’s a bit like being a detective.
- Research is never the same, year to year –it is always changing
- Being a faculty member means you get to do your own research!

# This talk

- Being a good research assistant
- Developing as a strong researcher (most likely when you are a graduate student)
- Your future:
  - Developing a research agenda
  - Becoming a faculty member or joining a research lab
  - Having your own students
  - Making discoveries that excite you and others!

# Research assistants (RA) vs. Principal Investigator (PI)

- Undergraduate and beginning graduate RAs (e.g., you):
  - Implement methods designed by others
  - Test methods using existing benchmarks and make observations
  - Maintain artefacts (software, datasets, websites)
  - Assist with writing
  - Perhaps also: suggest experiments and algorithms, try new ideas
- Graduate student RAs:
  - All of the above
  - Innovate: Develop new methods, prove theorems, etc.
  - Interpret the results
  - Share responsibility for writing paper with PI
  - Help design study, and supervise other RAs
- PIs:
  - Develop the research agenda
  - Come up with a plan for making progress
  - Supervise the RAs
  - Interpret the results
  - Primary responsibility for writing the paper
  - Corresponding author (responsible for everything in a paper)

# Being a Good Research Assistant

- Be prepared to learn a lot of new skills (including "soft skills") –
  be patient with yourself
- Be reliable, responsible, and polite (even if others aren't)
  - Meet deadlines (give advance warning if you can't meet a deadline)
  - Show up on time for meetings
  - Don't read email or use your cellphone during meetings
  - Respond to email promptly
  - Respect your collaborators: be helpful with feedback, assist them in learning
- Learn good research skills:
  - Document everything you plan to do, before you do it
  - Save all your data and code: make it reproducible
  - Never plagiarize (so know what it means)

# Timeline (think towards the future):

- Now you are learning to be a good research assistant
- Over the next years, you will develop into a strong researcher
- Your future as a PI with your own research agenda
  - becoming a faculty member, having your own graduate students
  - joining a national lab (e.g., Argonne, Sandia National Labs)
  - Joining a research lab (e.g., Microsoft Research)
  - Making discoveries that excite you and others!

# Being a good PI (and also being a good graduate student)

- Research in most cases is meant for publication.
- Therefore, you need to think ahead, and realize you will be talking to the whole world:
  - Develop a research agenda: Why is the research topic interesting?
  - Are my findings interesting, and worth publishing?
  - Who will be interested?
  - Are my findings valid?
  - What questions will the readers have, and will I be able to answer them?
  - What does my research suggest for future endeavors?

# Coming up with a research agenda

Think of the bigger questions, and design your research to help move towards a larger goal.

- Your larger goal has to be important to some group of people (otherwise you can’t publish your work).
- Your larger goal shouldn’t be un-reachable, like “curing cancer”.
- To get good research ideas, you need to read the literature carefully, a lot. Over and over. And find and discuss weaknesses and strengths in the papers you read. What do they leave open? What did they not address?

# Coming up with a research agenda

My example: Algorithm design for large-scale phylogeny estimation

- Constructing phylogenies (i.e., evolutionary trees) is a basic part of biological research, but all the best approaches are computationally intensive (heuristics for NP-hard problems).
- New opportunities for research in algorithm development result, because of increased dataset sizes (hundreds of thousands of species).
- My longterm goal is: more accurate and faster estimation of ultra-large evolutionary trees, through better algorithm design and implementation.
- Biologists will benefit from my algorithms, if my algorithms are implemented and tested well, and my papers communicate the improvement (and people read the papers).

# Getting started: low-hanging fruit

Read papers to find ideas and opportunities:

# Getting started: low-hanging fruit

Read papers to find ideas and opportunities:

- if the paper designs a new algorithm but doesn’t implement it, then implement it and evaluate the method.

# Getting started: low-hanging fruit

Read papers to find ideas and opportunities:

- if the paper designs a new algorithm but doesn’t implement it, then implement it and evaluate the method.
- If there is a flaw in the evaluation (such as not comparing to the best competing method, or training and testing on the same data, or limiting the evaluation to very small datasets), then redo the evaluation.

# Getting started: low-hanging fruit

Read papers to find ideas and opportunities:

- if the paper designs a new algorithm but doesn't implement it, then implement it and evaluate the method.
- If there is a flaw in the evaluation (such as not comparing to the best competing method, or training and testing on the same data, or limiting the evaluation to very small datasets), then redo the evaluation.
- If the algorithm design is not great (in terms of running time/memory), see if you can modify the algorithm to improve its computational performance, without changing the accuracy.

# Getting started: low-hanging fruit

Read papers to find ideas and opportunities:

- if the paper designs a new algorithm but doesn't implement it, then implement it and evaluate the method.
- If there is a flaw in the evaluation (such as not comparing to the best competing method, or training and testing on the same data, or limiting the evaluation to very small datasets), then redo the evaluation.
- If the algorithm design is not great (in terms of running time/memory), see if you can modify the algorithm to improve its computational performance, without changing the accuracy.
- If the paper designs an algorithm for one problem but a slight modification would let it be used on another problem, do the modification

# Getting started: low-hanging fruit

Read papers to find ideas and opportunities:

- if the paper designs a new algorithm but doesn't implement it, then implement it and evaluate the method.
- If there is a flaw in the evaluation (such as not comparing to the best competing method, or training and testing on the same data, or limiting the evaluation to very small datasets), then redo the evaluation.
- If the algorithm design is not great (in terms of running time/memory), see if you can modify the algorithm to improve its computational performance, without changing the accuracy.
- If the paper designs an algorithm for one problem but a slight modification would let it be used on another problem, do the modification
- If the algorithm has not been implemented for parallel computing, then implement it

# Getting started: low-hanging fruit

Read papers to find ideas and opportunities:

- if the paper designs a new algorithm but doesn't implement it, then implement it and evaluate the method.
- If there is a flaw in the evaluation (such as not comparing to the best competing method, or training and testing on the same data, or limiting the evaluation to very small datasets), then redo the evaluation.
- If the algorithm design is not great (in terms of running time/memory), see if you can modify the algorithm to improve its computational performance, without changing the accuracy.
- If the algorithm has not been implemented for parallel computing, then implement it
- If some proposed computational problem is not known to be NP-hard or polynomial time, see if you can settle it (i.e., prove it NP-hard or solve it in polynomial time)
- If the paper designs an algorithm for one problem but a slight modification would let it be used on another problem, do the modification

# Reading the literature is important

- The more you read, the better your research
  - You’ll avoid doing something already published
  - You’ll get ideas you can use
  - You’ll find low-hanging fruit
- Don’t believe the authors’ claims of success
  - They may have used the wrong benchmarks
  - Claims of relevance to an application are often over-stated
  - Improvements are sometimes too small to be important
- So: read, read, read. And that means the literature, not only Wikipedia.
- Videos do not replace reading.

# Publishing your research

- Research in most cases is meant for publication.
- Therefore, you need to think ahead, and realize you will be talking to the whole world in your publication!
  - Why is the research topic interesting?
  - Are my findings interesting, and worth publishing?
  - Who will be interested?
  - Are my findings valid?
  - What questions will the readers have, and will I be able to answer them?
  - What does my research suggest for future endeavors?

# Why is the research topic interesting?

- Easy case: you are working on an established problem (e.g., Traveling Salesman) that interests many people
- Harder case: you are proposing a new problem (e.g., a new optimization problem no one has thought about before)
  - You will need to convince your readers this is worth doing!

# Are my findings interesting, and worth publishing?

- Results that are unlikely to get published:
  - Re-implementation of an existing method that doesn't improve anything
  - A new proof for an existing theorem that is even more complicated than the first proof
  - A demonstration of something that people already know
- Results that have merit but might be hard to publish:
  - A comparison of two methods on one dataset
  - A parameter exploration that doesn't change performance
  - A comparison between existing methods that shows a model condition where one method is generally better than another (depends on model condition)
  - A proof of a theorem or algorithm for a toy problem

# Are my findings interesting, and worth publishing?

- Results that could be published:
  - Re-implementation of an existing method that substantially improves running time or accuracy
  - A new proof for an existing theorem that is simpler than the first proof
  - A demonstration of something that people don’t know
  - A comparison of two methods on many datasets
  - A parameter exploration that changes performance
  - A proof of a theorem or algorithm for a well-motivated problem

# Doing valid research

- If you are doing theory, have you really proven your results? (Check all the details)
- If you are doing an implementation, is your code doing what it says it's doing? (Has it been checked?)
- If you are testing methods, have you made it reproducible?
  - Provided all the details (version numbers, commands)
  - Made all your data available
  - Made your source code available
- Have you tested for statistical significance?

# Are you falling into traps?

- Reporting only the results that favor your method?
- Only studying datasets where your method does well?
- Only using criteria where your method does well?
- Not comparing to the best competing methods?
- Testing on the same data on which you train your methods?
- Over-stating your findings?
- Mis-interpreting or mis-representing prior studies?

# Who is your audience?

- Who will be interested in your research?
  - Writing for theoretical computer scientists is different from writing for application-focused researchers.
  - Writing for people outside CS is also different.
- What questions will the readers have, and will I be able to answer them?
  - Thinking about your audience will improve your research impact, because it will drive you to do even better work
  - This is one reason why you need to make everything reproducible and valid
  - See “Retraction Watch”

# Typical experiences in writing papers

- The PI (professor) gives you a problem to work on, typically in a team.
- Several weeks or months of doing the research
- Paper writing is divided up between the team members
- Initial draft completed
- Revisions take a few weeks!
- Submission to conference.
- Rejection.
- Revision, resubmission, rejection, revision, resubmission, rejection,…
- Eventual acceptance, and it’s a much better paper than it started. Often in a better journal.

# Handling rejection

- My experiences:
  - Recent paper in Systematic Biology (top journal), initially rejected by PLoS ONE.
  - Recent paper accepted to Workshop on Algorithms in Bioinformatics, after 3 earlier rejections
  - And many others
- Easy to get discouraged.
- Realize that peer review is sometimes biased (and just wrong)
- Learn from the reviews, even if what you are learning is that you picked the wrong journal or conference.
- Most likely you need also to revise the study, the algorithm, or the writing.

# Avoiding embarrassing situations

- Keep very careful notes about everything you are doing.
- Never cherry pick data.
- Show all your results.
- Keep all your data – never throw out anything.
- Be completely open and honest with your co-authors about what you are seeing.
- Expect to make mistakes, but learn to find them so you can correct them.
- Expect to be rejected… it happens to everyone. Learn from reviews.
- Never plagiarize.
- Read Retraction Watch.

# Tips on writing: tremendously important

- Assume your audience doesn't know the research area, and write so that they will be motivated to read your paper and will understand what you are doing.
- Read the related literature (learn to use Google Scholar), and include a discussion in the paper.
- Find examples of well-written papers, and study them. (For conference submissions, read papers from the same conference and find the best ones.)
- Be meticulous about grammar, punctuation, etc.
- Avoid unnecessary notation, but make sure you are sufficiently precise.
- Know the definitions of the terms you use.
- Learn LaTeX.

# Research stimulates new research

Suppose you have a great new idea, and you write it up. Are you done?

- Much of the best research suggests new research questions. Think about the possibilities – what else could you do? What are the potential weaknesses in your approach, and how can you stress-test your techniques?
- What does my research suggest for future endeavors?
- What is your next study?

# Summary advice for RAs

- Realize you will need to learn many new skills (don’t be afraid to ask for help, and also realize this will take time)
- Always be honest and open with your collaborators and PI
- Be courteous and respectful of others
- Be ethical
- Read the literature carefully
- Document all your work, and make sure it’s reproducible
- Learn how to write well
- Think towards the future!

# Your future:

- Now you are learning to be a good research assistant
- Over the next years, you will develop into a strong researcher
- Your future as a PI with your own research agenda:
  - becoming a faculty member, having your own graduate students
  - joining a national lab (e.g., Argonne, Sandia National Labs)
  - Joining a research lab (e.g., Microsoft Research)
  - Making discoveries that excite you and others!