Making Toast: Using analogies to explore concepts in bioinformatics

Kate L. Hertweck

1Department of Biology, The University of Texas at Tyler, Tyler, TX

Abstract

Contemporary biology is moving towards heavy reliance on computational methods to manage, find patterns, and derive meaning from large-scale data, such as genomic sequences. Biology teachers are increasingly compelled to prepare students with skills to meet these challenges. However, introducing biology students to more abstract concepts associated with computational thinking remains a major challenge. Analogies have long been used in science classrooms to help students comprehend complex concepts by relating them to familiar processes. Here I present a multi-step procedure for introducing students to large-scale data analysis (bioinformatics workflows) by asking them to describe a common daily task: making toast. First, students describe the main steps associated with this procedure. Next, students are presented with alternative scenarios for materials and equipment and are asked to extend the analogy to accommodate them. Finally, students are led through examples of how the analogy breaks down, or fails to accurately represent, a bioinformatics analysis. This structured approach to student exploration of analogies related to computational biology capitalizes on diverse student experiences to both clarify concepts and ameliorate possible misconceptions. Similar methods can be used to introduce many abstract concepts in both biology and computer science.

INTRODUCTION

The emergence of large-scale data collection in both molecular biology and biodiversity sciences is increasing the demand for scientists proficient in computational data analysis (1-3). The response to this demand are lessons or entire classes that teach bioinformatics to students with varying familiarity with programming, statistics, and even basic computer skills. Instructors in bioinformatics classes face the daunting task of making high-level, abstract computational concepts accessible to students who have no frame of reference for basic concepts in computer science, and who may even be acutely intimidated by them.

An obvious solution to encourage student learning of complicated concepts is to relate these foreign terms and processes to things with which students are familiar. Such metaphors and analogies have been widely used throughout the history of biology to elucidate concepts, and their use in science classrooms has been explored and evaluated extensively (e.g., 4). Although “metaphor” and “analogy” are often used interchangeably in science education literature, analogies are generally more structured and require explicit connections between concepts (5).

Analogy: Making Toast

Analogies are often incorporated into both textbooks and classroom teaching to explain myriad biological topics, including cell/molecular biology, genetics, and evolution (e.g., 6-8). Moreover, analogies can act as an introduction to scientific modeling (a formal method of scientific inquiry that replicates the natural world in a conceptual framework), as modeling requires students to draw on reasoning by analogy (9). Analogies are also frequently used in popular media to describe complex topics related to bioinformatics (10). Despite this prevalence, analogies have a contentious history in biology education (4). For example, use of analogies may sometimes confound student understanding of a topic (11), both by reinforcing previous misconceptions and failing to build an appropriate framework for abstract concepts.

Despite possible drawbacks, I contend that analogies are an invaluable tool to explore and improve students’ understanding of foundational concepts in bioinformatics. An appropriately designed analogy should bridge the gap between a familiar, tangible concept and a new idea that may be unfamiliar and/or intangible. The following example describes the application of an analogy to introduce students to an essential aspect of computational thinking.
SCIENTIFIC TEACHING THEMES

Active learning
By asking students to relate complex or abstract concepts in biology and computer science to common, familiar events and processes, analogies promote active engagement in learning. Describing and expanding an analogy require students (individually or in pairs) to evaluate both their own capacity to understand the analogy, as well as their classmates’ attempts to describe bioinformatics processes. Finally, breaking down the analogy is an essential step for students to personally identify potential pitfalls or misconceptions about the process being described.

Assessment
Analogies can be used in various ways in science classrooms. The in-class “making toast” example is useful as an explorative exercise, and instructors can quickly and easily gather informal feedback about the extent to which students can conceptualize procedures and steps in algorithms. However, the concept of analogies can also be incorporated into formal assessment. Following the in-class introduction to computational thinking, I asked students to describe their own common, daily task that can represent a bioinformatics workflow (see instructions and associated rubric in S1). At the end of the semester, I included a question in the summative (cumulative) assessment for the class that asked them if they could identify any additional ways in which their selected analogy breaks down. Because the initial “making toast” exercise occurs at the beginning of the semester, students are much better equipped in this summative assessment to describe specific relationships, extensions, and drawbacks between their analogy and bioinformatics workflows.

Inclusive teaching
The use of analogies encourages students to share knowledge about diverse cultures and experiences in the science classroom. In the “making toast” example, students are reminded about plurality in food choices, both in terms of diet (what foods are eaten with toast, including dietary restrictions) as well as preparation (what equipment is available for making the toast). Assessing student learning through formation of their own analogies encourages students to share experiences important to them. For example, students assigned to analogize a daily task to a bioinformatics workflow chose processes ranging from quilting to baling hay. Finally, some genomics analogies have been extended to physical models, allowing visually impaired and blind students to engage in lessons about gene sequencing (12), and other analogies could be similarly developed to encourage inclusivity.

MAKING TOAST: ALGORITHMS FOR NOVICE PROGRAMMERS

A challenging aspect of learning bioinformatics is developing requisite skills in computational thinking, a type of analytical thinking requiring abstraction at multiple levels of complexity (13). Given that analogies require abstraction of the familiar to relate the unfamiliar, this is an example well-suited to the strength of analogy use in bioinformatics education. This exercise introduces students to computational thinking through the process of making toast. This process of food preparation is commonly applied in computer science to help students learn about algorithms, or the step-by-step process of solving a problem with a computer (14-15). In this example, the process of making toast is generalized to represent steps in the process of a bioinformatics/data analysis workflow. The steps described below focus on the second phase of the Focus, Action and Reflection (FAR) model of teaching and learning science with analogies (16). This teaching process is designed for biology students at the high school, undergraduate, or graduate level, and is structured to highlight particular challenges to student understanding of bioinformatic workflows.

Introducing the analogy
The introduction of the analogy should help students actively practice skills in computational thinking, as well as develop an understanding for how parts of a bioinformatics process work together. Show students a picture of a loaf of bagged bread and a toaster (Figure 1) and ask them to think about how they would describe the process of making toast. Have students work with a partner to write down the steps required. Some students may have questions, such as “Is the toaster plugged in?” and “How much are we assuming?” I generally gesture toward the picture and tell them to base their steps on what they see there. You may choose to give them additional parameters, ranging from very specific (“Imagine a child in your house needs instructions”) to very general (“Imagine you are telling a Martian/extraterrestrial how to make toast”).

MAKING TOAST

What assumptions did you make about the start and end points for the process? Most students will start with a step to untwist the fastener on the bag of bread, but some will include steps for walking to the place on the counter where the bread is located. Similarly, some students will leave the bread in the toaster, while others will include steps to eat the toast. Bioinformatic analysis requires specific input and output, identified from the start of the experiment.
• Did anyone include a way to test whether the bread is toasted to the appropriate level? Some groups may have specified the length of time for the bread to stay in the toaster. A few students may use color of the toast to identify whether it is completely done. Validation is important in bioinformatics, so assessing the success of the process is essential.

• Did anyone include conditional statements to make toast? Some students may have used if/then steps to determine the process for making toast. These are directly relatable to common coding methods (for loops, if/then, if/else) in algorithms and bioinformatics analysis, and are essential to offering flexibility and transferability to code.

Expanding the analogy
The expansion of the analogy will allow you to more explicitly relate parts of making toast to a bioinformatics process by offering alternative conditions for the steps students have described. Ask pairs of students to trade their list of steps with other student pairs. Ask students to compare their methods with the other group to gain better understanding of the variation in how steps are described. Do the other group’s steps accomplish the process effectively? You can then select from a series of different options to modify the process of making toast and deepen students’ understanding:

• Type of bread (Figure 2A). Ask how many more steps are required to complete the process of making toast if the initial materials differ. Some steps, like cutting the loaf of bread into slices, may seem trivial, while others, like baking bread, may require another whole series of steps. Relate this to steps that may be required to pre-process (filter, parse, or quality control) data prior to analysis.

• Toasting apparatus (Figure 2B). Ask students what needs to change in their steps to allow for different methods of toasting the bread. A toy toaster oven, for example, may not have enough power to bake bread to completion, while an open-fire oven may need to be managed more carefully. Relate this to equipment available for bioinformatics analysis, and contrast a desktop/laptop computer with a high-performance computing resource.

• Resulting food (Figure 2C). Ask students which steps need to be changed, or if additional steps need to be added, if the desired food is modified from basic toast. The case of toaster strudel is akin to starting with a different type of data, while adding an egg to the toast is similar to combining data types.

• What happens when something goes wrong? Ask students what steps are required if there is an interruption or deviation in the process. For example, what if the electricity goes out? Can the process be restarted? These questions can help students prepare themselves for troubleshooting their own bioinformatics workflows later in the semester.

Breakdown of the analogy
Understanding how the analogy inaccurately represents the biological or computational process is essential to correcting possible misconceptions in student learning. Remind students that all analogies fail in at least one way. Ask them to describe how making toast fails to accurately represent a bioinformatics process. There are many possible student responses, depending on their familiarity with data analysis and computer science, and many of them will likely be superficial (i.e., “You can’t eat the results of a bioinformatic analysis”). The following are example prompts that can help students think more critically about the suitability of the analogy.

Did anyone include a way to test whether the bread is toasted to the appropriate level? Some groups may have specified the length of time for the bread to stay in the toaster. A few students may use color of the toast to identify whether it is completely done. Validation is important in bioinformatics, so assessing the success of the process is essential.

Did anyone include conditional statements to make toast? Some students may have used if/then steps to determine the process for making toast. These are directly relatable to common coding methods (for loops, if/then, if/else) in algorithms and bioinformatics analysis, and are essential to offering flexibility and transferability to code.

Expanding the analogy
The expansion of the analogy will allow you to more explicitly relate parts of making toast to a bioinformatics process by offering alternative conditions for the steps students have described. Ask pairs of students to trade their list of steps with other student pairs. Ask students to compare their methods with the other group to gain better understanding of the variation in how steps are described. Do the other group’s steps accomplish the process effectively? You can then select from a series of different options to modify the process of making toast and deepen students’ understanding:

• Type of bread (Figure 2A). Ask how many more steps are required to complete the process of making toast if the initial materials differ. Some steps, like cutting the loaf of bread into slices, may seem trivial, while others, like baking bread, may require another whole series of steps. Relate this to steps that may be required to pre-process (filter, parse, or quality control) data prior to analysis.

• Toasting apparatus (Figure 2B). Ask students what needs to change in their steps to allow for different methods of toasting the bread. A toy toaster oven, for example, may not have enough power to bake bread to completion, while an open-fire oven may need to be managed more carefully. Relate this to equipment available for bioinformatics analysis, and contrast a desktop/laptop computer with a high-performance computing resource.

• Resulting food (Figure 2C). Ask students which steps need to be changed, or if additional steps need to be added, if the desired food is modified from basic toast. The case of toaster strudel is akin to starting with a different type of data, while adding an egg to the toast is similar to combining data types.

• What happens when something goes wrong? Ask students what steps are required if there is an interruption or deviation in the process. For example, what if the electricity goes out? Can the process be restarted? These questions can help students prepare themselves for troubleshooting their own bioinformatics workflows later in the semester.

Applications for Analogies in Bioinformatics Education
There are opportunities to incorporate analogies into bioinformatics curriculum from both biological and computer science perspectives (Table 1). The dominance of genomics as a commonplace approach in biological science is a prime example for the application of analogies in teaching, as biology teachers are increasingly compelled to expose students to topics in the field. However, genomics includes processes that are both complex and emergent (patterns at the organismal level arise through interactions among smaller units, like

CourseSource | www.coursesource.org
2016 | Volume 03
Making Toast: Using analogies to explore concepts in bioinformatics

<table>
<thead>
<tr>
<th>Type of Content</th>
<th>Bioinformatics Concept</th>
<th>Analogy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Biological</td>
<td>Genome function</td>
<td>Instruction manual/blueprint (1)</td>
</tr>
<tr>
<td>Biological / Computational</td>
<td>Genome assembly</td>
<td>Jigsaw puzzle (2)</td>
</tr>
<tr>
<td>Computational</td>
<td>Algorithms and workflows</td>
<td>Making toast (this article)</td>
</tr>
<tr>
<td>Computational</td>
<td>Parallel computing</td>
<td>Jigsaw puzzle, desert island, long distance phone call (3)</td>
</tr>
</tbody>
</table>


... (rest of the content)