Multi-stage learning aids applied to hands-on software training

Kristian Rother, Magdalena Rother, Alexandra Pleus and Annette Upmeier zu Belzen

Submitted: 4th May 2010; Received (in revised form): 11th June 2010

Abstract
Delivering hands-on tutorials on bioinformatics software and web applications is a challenging didactic scenario. The main reason is that trainees have heterogeneous backgrounds, different previous knowledge and vary in learning speed. In this article, we demonstrate how multi-stage learning aids can be used to allow all trainees to progress at a similar speed. In this technique, the trainees can utilize cards with hints and answers to guide themselves self-dependently through a complex task. We have successfully conducted a tutorial for the molecular viewer PyMOL using two sets of learning aid cards. The trainees responded positively, were able to complete the task, and the trainer had spare time to respond to individual questions. This encourages us to conclude that multi-stage learning aids overcome many disadvantages of established forms of hands-on software training.

Keywords: education; training; didactic techniques; molecular viewer; molecular structure

INTRODUCTION
Practical exercises are an essential component of teaching and training. When preparing bioinformatics exercises, the heterogeneity of the trainees’ abilities is a frequent challenge, especially when practical skills using software, databases, and web servers are to be taught [1]. The reasons for this heterogeneity are manifold: the trainees may differ in background, level of experience, motivation and may be unfamiliar with the operating system or the technical terminology (also see the article by Schneider et al. in this issue). Other factors affecting the performance of individual students may be the room setup, availability of the teacher/trainer and in some cases language barriers. As a result, real participation and effective learning is often limited to a subgroup of trainees, with many others losing attention because the exercise is either too difficult or too easy for them.

A commonly applied solution of this problem is that the teacher reduces the complexity of the task: an exercise is divided into smaller independent problems or presented as precise step-by-step instructions. This approach has three disadvantages: first, trainees will not learn how to solve a complex task while following a ready-made recipe—they are deprived of the chance to learn by trial and error. Second, the material prepared highly depends on the ability of the trainer to predict the average skill level of his audience, which can be very difficult in summer school or roadshow-like settings. When the trainer chooses a wrong granularity of the task, the audience will be bored or frustrated. Because of that, the same material cannot be reused for an audience with a different skill level. Third, there is little room to reach out from the task at hand to related questions the trainees may be interested in.

Kristian Rother is teaching structural bioinformatics, programming and software development to bioinformaticians at the UAM Poznan. He is also contributor to the BioPython and PyCogent software libraries.

Magdalena Rother is teaching structural bioinformatics at the UAM Poznan. She is the main developer of the ModeRNA software for RNA 3D structure modeling.

Alexandra Pleus is a teacher of Biology and Physics is a PhD Student in the Department of Didactics of Biology at the Humboldt-University of Berlin. She researches about learners’ attitudes and the transition between primary and secondary school.

Annette Upmeier zu Belzen is teaching Didactics of Biology at the Humboldt-University of Berlin. Her main research topics are model competence, attitudes and interests of students as well as teacher-training and environmental education.
Flash cards have long been known as a learning tool that allows the trainee to actively interact with the subject [2]. They are sets of small cards on a given topic providing questions on one side, and answers on the other. A typical application is learning languages, in particular vocabulary, but flash cards on topics as diverse as chemical formulae and black jack can be found. They can be used to implement the Leitner system, a procedure to maximize the effect of rehearsal by increasing the intervals between repetitions [3]. Flash cards have been implemented in e-learning software such as Mnemosyne [4], in commercial systems to train sales strategies via mobile phones [5], and to teach pediatric facts to medical students [6]. Rehearsing with flash cards requires a rather large number of cards (50 at least), it is mainly found to exercise a body of facts, and they are mostly used outside the classroom. We wanted to explore how flash cards can be used to support a given task assignment in a computer-based tutorial setting.

In this article, we propose multi-stage learning aids [7]—a specialized form of flash cards—as an approach to train practical skills on bioinformatics programs. We exemplarily demonstrate their applicability on how to prepare and carry out a tutorial on the popular molecular viewer PyMOL [8], and the programming environment NetBeans [9].

MATERIAL AND METHODS
Multi-stage learning aids provide students with a set of flash cards that help them to solve a complex task [7]. Using these cards is optional—the trainees can decide themselves when and whether they need help. In practice, the learning aid cards can be printed (size A6 or smaller) in two ways:

- partial solutions: each card carries a detailed explanation of a sub-problem, providing incremental advice on the recommended path to the overall solution.
- hints & answers: the cards are printed duplex; the front side shows a hint or question that clarifies a particular aspect of the task to be solved. The back side presents the correct answer to the question or a precise instruction that brings the trainee one step ahead.

The set of learning aid cards have to follow a logical order; this can be the sequence in which corresponding sub-tasks need to be carried out. This approach often is advisable when a linear usage scenario on a software or web resource is to be taught. Alternatively, the solution for the most challenging part of the task can be given first (here most people will get stuck), and more detailed aspects thereafter.

For clarity, the sub-topics covered by the cards should be known or visible in advance. The number of cards should be around seven owing to the limited processing capacity of the brain [10]. To have enough time for the exercise, at least 5 min per card should be reserved.

Given a task, a trainer who wants to prepare learning aid cards has to decide what questions and hints to put there. Hänze [11] has formulated a number of guidelines that we would like to paraphrase: the cards should contain aids regarding both content and learning strategy. First, it needs to be made sure that the students have understood the task. A possible hint would be to reformulate the task in one’s own words, or identify properties of the desired solution. Second, focusing on the problem can be facilitated by highlighting important information, pointing out an analogous problem, or providing an ansatz to develop a solution. Third, the students can be requested to draw a sketch to visualize the approach to solve the problem. Fourth, necessary facts can be provided directly, including formulae, numbers and keywords. Finally, a way to verify a solution should be provided. The second to fourth point may vary in order, and some may occur more than once.

To illustrate this strategy for creating learning aid cards, we adapted an exercise from a Python programming course. The prepared learning aids include clarification of the problem at hand, assist in choosing the right data structure, visualization of a flow chart, point to key functions and checking the results. The exercise has been used in practice without the learning aids, but not yet with them. The material is available as Supplementary Data.

RESULTS
Case study 1: creating high-quality images in PyMOL
To demonstrate the applicability of multi-stage learning aids, we prepared a tutorial for the PyMOL molecular viewer. PyMOL is one of the most popular freely available viewers and has been used for many years by scientists to analyze 3D
structures of molecules and generate high-quality images [8]. It has a highly developed graphical and scripting interface. In the exercise, the trainees were to create an image showing the interaction between the two histidines and the iron atom in the heme group of hemoglobin (Figure 1). Three variations of this task were prepared, one for the PyMOL graphical interface, one for learning the command line interface, and one for writing a fully automated script that generates the picture. Two sets of learning aid cards were prepared (one for the graphical interface, the other for the scripting commands).

A trial group of seven bioinformatics students from first to fourth year carried out the exercise depending on their level of experience (two performing the basic task using the graphical interface and five the intermediate task on the command line). Four of the students were working in pairs, the other three alone. At the beginning of the exercise, a sample image and a sequence of seven steps necessary to create a high-quality image were displayed and explained during the first 5 min. The seven steps stayed visible during the entire exercise. A printed task description and a set of learning aid cards was made available to each trainee. Thus, it was possible to get immediate help on each stage if a trainee recognized that he got stuck there. The trainer was available during the entire exercise. The tutorial and the learning aid cards are available as Supplementary Data.

Six of the students completed the exercise successfully within 45 min, the seventh suffered a laptop battery failure. One of the resulting pictures is shown in Figure 1. During the exercise, the hint cards were used actively by all students. Two students ran short on time and used the solutions on the back sides of the learning aids to catch up within the last 5 min. A few students decided to start over during the exercise in order to repeat the individual steps. The trainer answered about seven short questions but was idle most of the time. The tutorial was closed by a slide show of the resulting images.

It is worth to note that we conducted the basic PyMOL tutorial with a different group of students earlier: 10 third year students, also working in pairs. The same set of learning aids was used, but in this study the sample result was not shown, and only a single stack of cards was placed in front of the classroom. We observed that the cards were not used by any of the students during the exercise. Also, the originally planned 1 h exercise took 2 h to finish, because the students had considerable problems to find their way around PyMOL despite active help by the tutor.

Case study 2: using the NetBeans development environment

A second application of learning aid cards was prepared to teach the most important functions of NetBeans, an integrated development environment (IDE). IDEs are tools offering a wide range of functions to programmers that help them organize, debug and refactor program code. NetBeans is a freely available IDE that supports many programming languages [9].

In the exercise, students were to familiarize with the program and to recognize its potential to save time when writing bigger programs. The eight most important operations were identified from an experienced developers’ point of view, and a set of eight one-sided cards was created, containing hints where to find particular commands in the program. The cards were formulated with an open result, rather inviting students to try out something than explaining in advance what will happen.

The exercise was carried out with seven fourth year bioinformatics students who had completed a 2-week Python programming course. The students were working with the program and the cards for 30 min (four in pairs, three alone). We observed that all could carry out the expected eight operations.

Figure 1: Oxygen binding in hemoglobin. The image was prepared by two 4th-year bioinformatics students using the PyMOL command line during a 60’ exercise with the program using multi-stage learning aids. The students had no previous experience with the PyMOL commands. The trainer was available for questions.
DISCUSSION

We have successfully applied multi-stage learning aids in a tutorial for the molecular viewer PyMOL. The technique of multi-stage learning aids has been examined in scientific subjects at secondary schools in Germany [7,11,12]. In a study involving 62 pupils from the 9th grade, 15 pairs of pupils were asked to solve a task from a physical subject using multi-stage learning aids, and a control group of another 15 pairs got a book text to as material for the same task [12]. Two pupils worked alone and were not considered in the evaluation. The quality of the solutions were evaluated by a score system. By video recordings, the communication of each pair was analyzed, and statements categorized into ‘scientific’ (relating to facts or conclusions), ‘strategic’ (planning of the solution) and ‘other’. It was found that the group using multi-stage learning aids communicated more in general, with twice as much statements in the ‘scientific’ category, and 1.5 times as much in the ‘strategic’ and ‘other’ categories. It was found that the number of points in the group using learning aids was significantly increased ($P < 0.05$).

Hänze et al. also reported that classes reacted very positively on this technique: more skilled pupils developed an ambition to solve the task at hand without the learning aids, giving the others time to catch up using the cards. Thus, the learning aids equalized the heterogeneity in a classroom and allowed all pupils to finish the exercise self-dependently and synchronously. The learning aids have been reported to work with individual, pair, and group exercises [11]. Our observations made during the PyMOL tutorial and the NetBeans example are fully in accord with these reports. One potential pitfall we would like to point out is that displaying a sample result and the structure of the task seemed essential in our example tutorial. Without either it was easy for trainees to get lost. As described in the ‘Results’ section, the availability of the learning aid cards is crucial: when we deposited one set of learning aid cards in front of the class in a previous PyMOL tutorial with the same task assignment, they were not used despite massive problems with the exercise. We think getting help had a submissive character in this scenario that inhibited the self-regulating effect of the learning aid. Although cooperation of students generally improves their performance [13], the students working in pairs did not help enough, probably because they lacked a reliable source of information. Therefore, we strongly recommend to make the cards available to the students directly.

One potential disadvantage of the method is that an unmotivated student may harvest the answers from all cards immediately, execute what is written on them, and finish the task—in instead of thinking of a solution first. With a written step-by-step instruction this could also happen, though. In other forms of tasks, a lazy student might try and get solutions from his classmates. Considering the latter alternative, a student that reads his instructions is still the next best thing after a student that develops ideas on his own. One way to prevent the ‘lazy student’ phenomenon would be to replace raw information in some cards by more strategic hints. The programming example in the ‘Materials and Methods’ section provides a few ideas how to achieve this.

With these caveats in mind, we see several advantages in using multi-stage learning aids: Once prepared, the same set of learning aids flexibly adapts to group size, differences in skill level, and time constraints. The trainees have a chance to work autonomously, and the trainer has plenty of room to react on individual problems—providing a backup mechanism. To quote one student: ‘If you get stuck, you don’t get stuck forever’ (translated from Polish). It is even possible to conduct an exercise at various levels of difficulty simultaneously, or as a step-by-step tutorial using the learning aids.

One purpose of the learning aids is that the trainer is less busy with troubleshooting. He should also be able to recognize whether the students are actively using the cards or not. It would be worthwhile to undertake a long-term study to see how grown-up students react on this or related training strategies. One possibility would include a web-based system to monitor card usage.

Taken together, multi-stage learning aids have several advantages over a step-by-step tutorial. To our best knowledge, this technique has not been applied to bioinformatics before. We are convinced as it has the potential to improve hands-on training of bioinformatics tools in summer school-like tutorials and to facilitate key stages of courses on particular software like the one on the R statistics package described in ref. [1].
SUPPLEMENTARY DATA

Supplementary data are available online at http://bib.oxfordjournals.org/.

Key Points

- Multi-stage learning aids are printed cards that guide trainees through a complex task and allow trainees to self-regulate their learning speed.
- They can be flexibly applied in a multitude of classroom situations, in particular tutorials of programs, databases, and web services.
- The practical applicability was shown on a PyMOL tutorial.
- We can recommend using multi-stage learning aids to bioinformatics training and encourage further communication on this subject.

Acknowledgements

We would like to thank the bioinformatics students of the Adam Mickiewicz University for participating in the tutorial, and Janusz M. Bujnicki for providing facilities for our didactic field trials.

FUNDING

This work was supported by the German Academic Exchange Service (DAAD; D/09/42768 to K.R.); and by the Faculty of Biology, Adam Mickiewicz University (PBWB-03/2009 to M.R.).

References