BCB420 - Computational System Biology

Boris Steipe

2019-01-04

Contents

Chapter 1

Wiki - Editing

(Wiki editing, namespaces; user page setup; copyright, a Course Journal stub page, and an insights! stub page)

1.1 Abstract:

This will likely be the first learning unit you work with, since your Course Journal will be kept on a Wiki, as well as all other deliverables. This unit includes an introduction to authoring Wikitext and the structure of Wikis, in particular how different pages live in separate "Namespaces". The unit also covers the standard markup conventions - "Wikitext markup" - the same conventions that are used on Wikipedia - as well as some extensions that are specific to our Course- and Student Wiki. We also discuss page categories that help keep a Wiki organized, licensing under a Creative Commons Attribution license, and how to add licenses and other page components through template codes.

1.2 Objectives:

- Provide an introduction to Wiki principles and Wikitext markup.
- Create first pages on your own on the Student Wiki.
- Learn about copyright, why we use Creative Commons licenses for the Student Wiki and how to place a license tag.

1.3 Outcomes:

- You are competent with basic Wiki markup and the extensions on this Wiki.
- You can create pages and add them to categories while taking care to create them into your own user space.
- You have created your own user page on the Student Wiki and added contents.
- You have created page stubs for a Course Journal and an insights! page.

1.4 Deliverables:

- Specified as "Tasks": There are no general deliverables for this unit; specific deliverables are described in the "Task" sections.

Collaboration is a common theme for modern lab work and a Wiki is a great way to share and seamlessly update information in groups - or just for yourself. Probably the most sophisticated Wiki software is MediaWiki, a set of PHP scripts that is under continuous development by the Wikimedia foundation; it is the same software that runs Wikipedia. This is open source, free software that is easy to install, is well documented and requires very little resources other than a machine that runs a MySQL database server and an Apache Webserver. Numerous extensions exist (and extensions are not hard to write); they enhance the already rich functionality. But let's start with small steps. You should by now have a user account on the Student Wiki, and I have configured that Wiki so that * only logged in users can view the pages; but … * all logged in users can create and edit (most) pages at will. * This means you could edit pages that don't "belong" to you. Respect the "House Rules" and don't edit other's things without permission, even if you can think of a particularly witty comment or hilarious prank. If you want to comment on a page: every page has an associated "Discussion" page that you can freely edit. Remember to "sign your name" to discussion entries.

1.4.1 Task:

1. Access the Student Wiki;
2. log in and navigate to your user page. (Your user page is linked to your name that appears at the top of every Wikipage once you are logged in.)
3. Create / edit the page, try out and experiment with the Wikitext syntax that this unit covers as you read about the different elements.
4. Work through the contents below.

For more extensive formatting help see: http://meta.wikimedia.org/wiki/Help: Editing For Math markup see: http://meta.wikimedia.org/wiki/Help:Formula

1.5 The Wiki concept

Wiki sites are collections of Web pages that allow you to view, edit and create pages from your browser, there is no need for special technology and basic editing is simple and intuitive with "Wikitext markup". The basic workflow of Wikis is straightforward: * Register an account and browse the site. * Whenever you find something that you can improve, edit it. * Whenever you find something that you would like to comment on, click on the "discussion" tab and share your views. * If you are interested in what becomes of your edits or the discussion, click on the "watch" tab, and the page will be added to a list of bookmarks to pages you are "watching". (You can even generate an RSS feed for recent changes or new pages). * No e-mail, no obligations. Do what you like, when you like, what you can. * Editing on the Course Wiki is only enabled for instructors. However you can freely edit all pages on the Student Wiki, once you have an account.

1.6 Editing basics

1.6.1 Start editing

To start editing a Wiki page, click on the "Edit" link at one of its edges. This will bring you to the edit page: a page with a text box containing the wikitext: the editable source code from which the server produces the webpage.

1.6.2 Preview before saving

When you have finished, press Show preview to see how your changes will look. Repeat the edit/preview process until you are satisfied, then click Save and your changes will be immediately applied to the article and accessible on the Web. They will also be stored in the main database for as long as the Wiki exists. Thus it is always possible to get back earlier versions of pages - back to the very first edit.

1.6.3 Basic text formatting

Here are some examples of the markup of Wikitext. It is not the same as HTML markup, however some HTML markup will work. In particular, the

Wiki applies styles through CSS technology (Cascading Style Sheets) and thus HTML tags can be used to apply consistent styles to individual page elements. Javascript won't work. What it looks like What you type You can emphasize text by putting two apostrophes on each side. Three apostrophes will emphasize it strongly. Five apostrophes is even stronger. You can ''emphasize text''by putting two apostrophes on each side. Three apostrophes will emphasize it''strongly''. Five apostrophes is''"'even stronger'"'.

A single newline has no effect on the layout. But an empty line starts a new paragraph. A single newline has no effect on the layout.

But an empty line starts a new paragraph.

You can break lines without starting a new paragraph. You can break lines without starting a new paragraph.

You can format text in a monospace font with a dashed box around it either by marking it with the HTML <pre> tag, or by putting a blank space at the beginning of a line. Example.

(This may not be very useful beyond the types of examples we show here, but it is a frequent source of confusion, when you find your text marked up this way by accident) You can format text in a monospace font with a dashed box around it either by marking it with the HTML <pre>" tag, or by putting a blank space at the beginning of a line.

Example.

(This may not be very useful beyond the types of examples we show here, but it is a frequent source of confusion, when you find your text marked up this way by accident)

Other special characters at the beginning of a line include: bulleted list numbered list term and definition Other special characters at the beginning of a line include:

- bulleted list

numbered list

; term and definition

You should "sign" your comments on discussion pages: Three tildes gives your user name - Boris (talk) Four tildes: user name plus date/time - Boris (talk) 22:18, 27 December 2012 (EST) Five tildes: date/time alone - 22:18, 27 December 2012 (EST) You should "sign" your comments on discusion pages: : Three tildes gives your user name - ~ : Four tildes: user name plus date/time - ~~ : Five tildes: date/time alone - ~~~

Use normal HTML character codes for special characters, or use Unicode. For example: > < & ° Å Ä ü → Use normal HTML character codes for special characters, or use Unicode. For example: > < & ° Å Ä ü →

You can use HTML tags, too, if you want. Some useful ways to use HTML: Put text in a typewriter font. The same font is generally used for computer code. Strike out or underline text, or write it in small caps. Superscripts and subscripts: x2, x2 Invisible comments that only appear while editing the page. You can use HTML tags, too, if you want. Some useful ways to use HTML:

Put text in a typewriter font. The same font is generally used for computer code.

Strike out or underline text, or write it IN SMALL CAPS.

Superscripts and subscripts: x2, x2

Invisible comments that only appear while editing the page.

For a list of HTML tags that are allowed, see HTML in wikitext. I tend to use Wiki-markup when I'm in a hurry, but use the HTML tag whenever I can't remember a Wiki-tag. It really doesn't make a difference. However: I never use Wiki-table markup. I find it less intuitive than HTML markup, more difficult to debug, and there's really no point in remembering both types of markup given that one really needs to be comfortable with HTML tables anyway.

1.6.4 Links

You will often want to make clickable links to other pages. What it looks like What you type Here's a link to a page named Sandbox. You can even say Sandboxes and the link will show up right. You can put formatting around a link. Example: Sandbox. Here's a link to a page named [[Sandbox]]. You can even say [[Sandbox]]es and the link will show up right.

You can put formatting around a link. Example: ''[[Sandbox]]".

You can link to an arbitrary piece of text with a piped link. Put the link target first, then the pipe character "|", then the link text - as in this example. You can link an arbitrary piece of text with a ''piped link". Put the link target first, then the pipe character "|", then the link text - as in [[Sandbox| this example]].

You can make an external link to a Web page just by typing an URL, e.g. http://igem.org Or you can link arbitrary text: iGEM. (Note: No "|" for external links, URL and text are separated by a blank, and only single square brackets!) Or you can generate a footnote-like link: [1]. You can make an external link to a Web page just by typing an URL, e.g. http://igem.org

Or you can link arbitrary text: [http://igem.org iGEM]. (Note: No "|" for ''"external"'' links, URL and text are separated by a blank, and only single square brackets!)

Or you can generate a footnote-like link: [http://igem.org].

Note: remember: internal links (using [[…]] tags to link to pages on this Wiki) are separated from linked text with a pipe character. External links (using […] tags to link to pages elsewhere on the Internet) are separated from linked text with a space character.

1.6.5 Special syntax

Two special syntax items need to be mentioned: “templates” and “magic words”: #### Templates Templates are pieces of Wikitext that are substituted where a code that links to them has been placed into a page. For example, if you enter {{Lorem}} on a page, the “Lorem ipsum dolor sit amet …” placeholder text is inserted in place of that code. Wikis make extensive use of templates. #### Magic words some reserved “magic”-words are replaced with dynamically created contents when the page is rendered. For example **TOC** forces placing a Table Of Contents at the position of this token rather than its default position, while **NOTOC** suppresses creation of a Table Of Contents on a page.

1.6.6 Creating a new page

To create a new page simply insert a link to a Wiki page, which has a page name that does not exist yet. The link will appear in red (except if you inadvertently used the name of a page that already exists), and the new page will be created when you click on the link. Page names can be long and contain blank spaces. Internally, all blank spaces are converted to underscore characters, but you can use the page name without underscores in links; the Wiki software translates this for you.

1.6.7 Namespaces

The Wiki maintains some pages in special collections, in so called “namespaces”. This is useful, because the behaviour of the software can be customized for different namespaces: for example you may be allowed to edit in the main- and the user- namespace, but not in the MediaWiki: namespace, where pages are held that affect the gears and wires of the Wiki. Page names without a prefix live in the main space. Some commonly used prefixes are: * User: - personal pages for user with an account on the Wiki; * Talk: - discussion pages for comments on pages, accessible via the “Discussion” tab; * Help: - this page for example; * Template: - pages with reusable text. * Special: - pages that implement special functionality (like login, user lists, or lists of recently changed pages); * Category: - an index of pages that have been given a common tag. This is a convenient way to access pages that are in some way related;

1.6.8 Categories

Once your page has been edited, you can associate it with one or more categories. Add the appropriate category tag by typing [[Category:BCH441_2013]] or [[Category:BCB410_2013]]. The page is then automatically linked from a page that collects all pages with that category tag. I would prefer that you do not create new categories; ask me if you feel a need for it.

1.6.9 Creating a new section or subsection on a page

To create a section or subsection, simply insert a section header into an existing section. Header levels are defined by the number of "=" characters before and after the header text. Click on an edit link of this page to see example code. Once a page has more than two headings, the Wiki automatically creates a table of contents. You can adjust the position of the table of contents by typing the "magic word" ___TOC___somewhere on your page (Note: double underscore), you can also suppress having a table of contents created with**NOTOC**.

1.6.10 Edit conflicts

If someone else makes an edit while you are making yours, the result is an edit conflict. Many conflicts can be automatically resolved by the Wiki. If it can't be resolved, however, you will need to resolve it yourself. The Wiki gives you two text boxes, where the top one is the other person's edit and the bottom one is your edit. Merge your edits into the top edit box, which is the only one that will be saved.

1.6.11 Reverting pages to a previous state

Sometimes a page needs to be reverted to a previous state. Access the page through a link to the Recent Changes special page: Special:Recentchanges. Find the page you need to revert, click on the hist link, click on the version you need and verify that it is the correct one. Then click on the edit tab at the top and Save page. A new version of the page is then created with the old text. Note that this does not actually overwrite anything - all edits are archived in the database.

1.6.12 Special markup on this Wiki

Here are some special templates and extensions installed on this Wiki:

- Vertical space - The template code {{Vspace}} will insert a two-line high space to help structuring text.

- References and footnotes - Enclosing text in <ref> … <\ref> tags will create a footnote reference and display the text wherever you place a tag on the page.
- Syntax highlight - The GeSHI syntax highlighter extension is installed on this Wiki. Type: for (i in 1:5) { print(i^2) } # 1 4 9 16 25 … to get: for (i in 1:5) { print(i^2) } # 1 4 9 16 25
- Pubmed Articles and abstracts - {{#pmid:15289071}} This inserts the article information in a

  , formatted by Template:Pubmed. Steipe (2004) Consensus-based engineering of protein stability: from intrabodies to thermostable enzymes. Meth Enzymol 388:176-86. (pmid: 15289071) [PubMed][DOI]

{{#pmid: 15289071 |Steipe2004}} * This formats the pubmed parser output for the Cite extension; A footnote mark will be inserted here[1] and the actual reference will appear beneath the section of the page.

- Math markup - H = - \sum_{i=0}^n p_i \log_{2} p_i</math>
  - see: http://meta.wikimedia.org/wiki/Help:Formula
- ToDo items - Type:
  ToDo:
- This …
- and that.

… to get: ToDo: This … and that.

- Notes Type:
  Note: take special care to …

… to get: Note: take special care to …

- Linking text to Wikipedia … using the {{WP|…}} template. If the linked text is the same as the Wikipedia page titke, simply type it: {{WP|Mutual information}} Mutual information

If the linked text is different, use the "|" pipe character to separate page-name and text: {{WP|Mutual information|'"WP article on Mutual Information"'}} WP article on Mutual Information

- Collapsible elements See: Manual: Collapsible elements
  Visible text …
- Collapsed text 1
- Collapsed text 2
- Collapsed text 3

Example: [Expand for poem] The Road Not Taken Robert Frost

Sample reference section ↑ Steipe (2004) Consensus-based engineering of protein stability: from intrabodies to thermostable enzymes. Meth Enzymol 388:176-86. (pmid: 15289071) [PubMed][DOI]

1.6.13 The "User space" and subpages

The User: namespace on the Student Wiki is especially important. Namespaces allow us to distinguish pages that share the same logical name. Every student will create a journal page, but of course there can be only one [[Journal]] page on the Wiki. Therefore each of these pages needs a distinct name. The obvious solution is to keep them in the User: namespace, and create them as subpages of everyone's User page. The page name of your user page is [[User:]]; subpages are created with a backslash, and therefore your Course Journal page should be [[User:/Journal]]. if you take more than one course, you can separate the journals like [[User:/BCH441-Journal]], [[User:/BCB410-Journal]], etc. Please do not create pages in the "Main space" of the Student Wiki! Do not omit the User:/ part of the page name.

1.6.14 Copyright

Over the last decades, in bioinformatics and many other fields of science, the paradigm under which we create value has profoundly changed. While we previously considered restrictions on the use of our insights important, tried to keep knowledge under control, and thought in terms of intellectual property, the modern paradigm is mindshare. We strive to make our work maximally useful to others, and to document how we are creating this utility. This does not mean that we are simply putting everything into the public domain: yes, people should use our ideas, but we must receive credit - as a currency for grant and scholarship applications and the like, to enable our future work. The right tool for this is copyright. Everything we write and create automatically falls under our copyright, there is no special copyright tag required. To have our material reused, we can either relinquish our copyright or grant a license to reuse. Material that is created in coursework will ideally be useful elsewhere, but it is only useful if its use is permitted and regulated. Wikis are tools for collaboration, and Wikipedia generally applies a site-wise license to all material. In our work we take a similar approach, but we apply licenses more specifically[1]. All material submitted for credit, including code, documentation, essays, manuals, images, lab journal entries, insights! pages etc. must be licensed with an appropriate open-source license. This is a strict requirement for the course. For code this is the MIT software license, for everything else this is the Creative Commons Attribution 4.0 International License. The MIT license for code guarantees that there are no restrictions on re-use other than fair and visible attribution of the authors' work. The CC license guarantees proper attribution of authorship but allows free use otherwise. Together, these licenses allow the material to be used, refactored, updated and republished and thus (hopefully) give it a fertile future life. In order to keep copyright and licenses consistent throughout the site, we use a template tag - simply insert it at the bottom of a page: Entering the template code … {{CC-BY}} creates the copyright message …

1.6.15 Task:

- Practice basic editing syntax by putting contents on your User Page:
  - enter your name,
  - your major(s), specialist program, year of study - or a link to your lab and your thesis theme if you are a graduate student;
  - enter your email address. I use this information a lot when I need to contact students, so make sure it is correct and current.
  - Add a category tag to your User page for the course you are taking. All pages with this tag are accessible via the link in the sidebar. What should the category tag say? Good question … go and find out.
  - Add a copyright template to the bottom of your user page by putting a {{CC-BY}} tag on its bottom.
  - Feel free to look at my User Page for code examples: clicking on the edit link will show you the source text. How do you find my User Page? Good question …
  - Create a subpage to your User Page; call it "Journal". Note: the link MUST be in your "User space". If you don't add the prefix User:yourname/… before your page name, the new page will end up in the main "namespace". I'll then have to delete it. That's not good. Make sure you know what you are doing, for example by looking at the code on my User Page, asking someone who knows, or asking on the mailing list.
  - Put some placeholder text on your journal page, you will fill it in when you work through the Journal unit.
  - Similarly, create an "insights!" page on a subpage to your User Page and add some placeholder text. That will be expanded when you work through the insights! unit. Play around some more. Feel free to ask how to go about achieving a particular effect that you may have seen elsewhere.

1.6.16 Self-evaluation

You should be familiar with the following: * How to Login to the Student Wiki and access your user page; * viewing a page's history; * basic text formatting and Wiki markup; * "signing" your name; * creating internal and external links; * creating section headers on a page on multiple levels; * reverting a changed page to an earlier version; * creating a new page (as a subpage of an existing page); * the concept of namespaces - especially the default ("main") and User: namespace; * the concept of categories and how to add a page to a category; * copyright on the Student Wiki, and how to insert a license note.

1.6.17 Notes

‘”Note”’ that additional rules for collaboration in the context of coursework derive from the rules for academic integrity and plagiarism. If some text is not copyrighted, this does not mean you can use it without reference and thus imply it is your own idea. That would be plagiarism. Further reading, links and resources

If in doubt, ask! If anything about this learning unit is not clear to you, do not proceed blindly but ask for clarification. Post your question on the course mailing list: others are likely to have similar problems. Or send an email to your instructor.

About …

Author: Boris Steipe boris.steipe@utoronto.ca Created: 2017-08-05 Modified: 2019-01-04 Version: 1.1 Version history: 1.1 Changed software license from GNU-GPL to MIT 1.0 Completed outcomes/objectives. Added copyright. First live version. 0.2 First contents imported from Help:editing. Added tasks. 0.1 First stub

Chapter 2

Your Course Journal

(How to keep a course- or lab journal)

2.1 Abstract:

Keeping a journal is an essential task in a laboratory. To practice keeping a technical journal, you will document your activities as you are working through the material of the course. A significant part of your term grade will be given for this Course Journal. This unit introduces components and best practice for lab- and course journals and includes a wiki-source template to begin your own journal on the Student Wiki.

2.2 Objectives:

- Introducing components and best practice of lab- and course journals
- Presenting sample wiki-text for Journal entries

2.3 Outcomes:

Upon concluding this unit you should be able to … * Begin a structured course journal on the Student Wiki using proper wiki text; * Write your own journal entries, including media images and code as required; * Cross-reference journal entries with links; * Link to external sources and deliverables on internal pages as appropriate; * Estimate the time you need for tasks, and develop a habit of improving your time-management skills.

2.4 Deliverables:

1. Your Journal: Your entire journal will be evaluated at the end of the course. Refer to the marking rubrics for details.
2. Insights: If you find something particularly noteworthy about this unit, make a note in your insights! page. **Caution:**
3. Your course journal is a deliverable of this course and it will be graded. Therefore all rules regarding plagiarism and other academic misconduct apply in full. In particular:
   - ‘”do not include any material from elsewhere without referencing it:”’ We are operating a “full disclosure” policy in this course. Anything that you did not write yourself, on the spot, must be referenced. In particular you need to reference if you are copying your own material from other courses.;
   - ‘”do not fabricate material that you are posting in your journal.”’ Fabrication could include things like: modifying results produced by your code, describing work that you have not actually done, or claiming a time for the journal entry that is not the time/date on which it was actually written. All of these are academic offences.;
   - ‘”Note:”’ Only journal entries that were written concurrently with the activity they describe will be evaluated for credit.
   - ‘”Note:”’ All journal pages on the Student Wiki—like all other submitted material—must contain a {{CC-BY}} template.

2.5 Prerequisites:

You need the following preparation before beginning this unit. If you are not familiar with this material from courses you took previously, you need to prepare yourself from other information sources: * ‘”Inquiry’‘: The scientific method; evidence based reasoning; how to design, execute and document an experiment; Conjecture, hypothesis and theory. *’‘Writing”’: Basic essay and report writing skills. How to format your submitted materials, how to quote, cite and avoid plagiarism. * This unit builds on material covered in the following prerequisite units: * 1

Work through this unit, then make your work with the “Plagiarism” Unit the first entry of your Journal!

Computational research embraces the same best-practice principles as any wet-lab experiment. We ensure our work is reproducible, we take great care that our conclusions are supported by data, and we keep notes to document our objectives, activities and how we arrived at our results. Those notes are more than just a handy collection of information: they need to become a robust, testable

record of activities. Paper notes are not very useful for bioinformatics work because they can't be cross-referenced easily with computer files. Ideally, bioinformatics journals will document results, and link to data files, code repositories, Webpages and other resources. Thus a technical solution needs to support incorporating or linking to results, data, code, workflow scripts, documentation, and much more. In this course, we use the open source Media Wiki software to support journal keeping[1]. Keeping a record of your activities is a habit, and habits need to be formed through practice. Is this going to be useful to you? I don't know, but neither do you unless this habit has been given a credible chance to form. Therefore we practice keeping journals in this course. As a welcome side effect, this creates a record of activities for future reference, and provide a basis for evaluation of your progress at the end of the course. Keeping a journal will help you work with other learning units or project components effectively, because this is all integrated over the entire course, and later units often make use of earlier results which you should have easily accessible. Remember: you are writing a lab notebook—not a formal lab report: a point-form record of your actual activities.[2] Write such documentation as notes to your (future) self. Record everything that's necessary, but be light and agile about your writing. Write your notes immediately, in parallel with your actual activities, don't draft them elsewhere and expect to enter and revise them later. Practice shows that delayed processing of journal notes creates an unmanageable burden. Therefore notes that are not written concurrently with the activity will not be considered for credit in this course. This too is about habit forming. But writing concurrently is so easy: since all of your computational work is done with a computer, begin every work-session by opening an editing window for its journal entry. Have the window open, and immediately record everything of importance. The Wiki is online, so you can even edit your journal from a library computer, and even (although it's awkward) from your phone. Obviously, the first step is to create a journal page in the User space of the Student Wiki - you have already done this in the Wiki editing unit.

2.6 Header

Write a header and give it a unique number. This is useful so you can refer to the header number in later text. Obviously, you should "hard-code" the number and not use the Wiki's automatic section numbering scheme, since the numbers should be stable over time, not change when you add or delete a section[3]. It is useful to add any new contents at the top of the page. Keeping the page in reverse chronological order, prevents you having to scroll to the bottom of the page every time you add new material. Note though, that the sections do not actually have to be in strict chronological order, like we would have them in a paper notebook. Typically you would number in a decimal system - like 1, 1.1, 1.2, 2, 3 etc. - so you can easily accomodate additions. It may be advantageous to give different subprojects their own numbering space - by adding a prefix to

the section number. This depends on how related the projects are. Everything you keep on the same page is easy to find with your browser's search function. But if search results come from different projects, that may be inconvenient. To decide what to put on the same page and what should go in different subpages, imagine what material you would search for and what search terms you might use[4]. Incidentally: the material in such a notebook is "permanent", since earlier versions of pages are always available via the history function. The Wiki never forgets. As well, they are automatically time-stamped. And that's actually a step beyond paper labnotes.

2.7 Objective

- State the objective.
- In one brief sentence, restate what your activity is supposed to achieve.

2.8 Estimate duration

The learning units in this course require you to estimate beforehand how long you will take, and to record how much time you actually took. Record your initial estimate (work-hours), how many hours you took, and how much time elapsed between start and end of your task. Make this a habit in your future coursework as well as in your future labwork. You will quickly note that you will become much better at time-management. The sample journal template that is included below contains wikitext to format a time estimate.

2.9 What to document

- Document the procedure - Note what you have done, as concisely as possible but with sufficient detail. "What is sufficient detail?" The answer is easy: detailed enough so that someone can reproduce what you have done. In practice that "someone" will often be you, yourself, in the future. I hope that you won't be constantly cursing your past-self because of omissions!
- Document your results.
  - You can distinguish different types of results -
  1. Static data does not change over time and it may be sufficient to note a reference to the result. For example, there is no need to copy a GenBank record into your documentation, it is sufficient to note the accession number, the refSeq ID, or the UniProt ID, or even better, to link to the relevant page on the external database server.

2. Variable data can change over time. For example the results of a BLAST search depend on the sequences in the database. A list of similar structures may change as new structures get solved and deposited in the PDB database. In principle you want to record such data, to be able to reproduce at a later time what your conclusions were based on. But be selective in what you record. For example you should not paste the entire set of results of a BLAST search into your document, but only those matches that were important for your conclusions. '"Indiscriminate pasting of irrelevant information will make your notes unusable."' Incidentally, the technology to expand and collapse paragraphs that we demonstrated in the Wiki editing unit can be put to excellent use to record data but keep it out of sight when not needed.
3. Analysis results - The results of sequence analyses, alignments etc. in general get recorded in your documentation. Again: be selective. Record what is important.

2.10 Conclusions

Note your conclusions. - An analysis is not complete unless you conclude something from the results. * Are two sequences likely homologues, or not? Just pasting the BLAST output is not enough. It's your call - '"record it"'. * Does your protein contain a signal-sequence or does it not? SignalP will give you a probability, but you must make the final call. * Is a binding site conserved, or not? The programs can only point out sections of similarity or dissimilarity. You are the one who interprets these numbers in their biological context. The analysis provides the data. In your conclusion you provide the interpretation of what the data means in the context of your objective. Were you expecting a signal-sequence but there isn't one? What could that mean? Sometimes your task will explicitly include to elaborate on an analysis and conclusion. But this does not mean that when analysis is not explicitly mentioned, you can skip the interpretation. In general you can never expect full marks if analysis and conclusions are missing.

2.11 Outlook for the next tasks

What's the next step? Note it here. Also include a link to the logically next entry - this way you can quickly hop through consecutive entries for a theme.

2.12 Cross references

Add cross-references. Cross-references to other information are supremely valuable as your documentation grows. It's easy to see how to format a link to a section of your Wiki-page: just look at the link under the Table of Contents at the top. But you can also place "anchors" for linking anywhere on an HTML page: just use the following syntax. <\span> for the anchor, and append #{some-label} to the page URL.

2.13 Media

2.13.1 Images

- Use discretion when uploading images
- Don't upload irrelevant images, don't upload copyrighted images, keep the size reasonable. Prepare your images well
- Don't upload uncompressed screen dumps. Save images in a compressed file format on your own computer. Then use the Image Upload link in the left-hand menu to upload images. The Wiki will only accept .jpeg, .png, .gif, or .svg images.
- Use the correct image types.
- In principle, images can be stored uncompressed as .tiff or .bmp, or compressed as .gif or .jpg or .png. .gif is useful for images with large, monochrome areas and sharp, high-contrast edges because the LZW compression algorithm it uses works especially well on such data; .jpg (or .jpeg) is preferred for images with shades and halftones such as the structure views you should prepare for several assignments, JPEG has excellent application support and is the most versatile general purpose image file format currently in use; .tiff (or .tif) is preferred to archive master copies of images in a lossless fashion, use LZW compression for TIFF files if your system/application supports it; The .png format is an open source alternative for lossless, compressed images. .bmp is not preferred for really anything, it is bloated in its (default) uncompressed form and primarily used only because it is simple to code and ubiquitous on Windows computers. Accordingly we don't support it here.

2.13.2 Image dimensions and resolution

Stereo images should have equivalent points displayed approximately 6cm apart. It depends on your monitor how many pixels this corresponds to. The dimensions of an image are stated in pixels (width x height). My notebook screen has a native display resolution of 1440 x 900 pixels/23.5 x 21 cm. Therefore a 6cm separation on my notebook corresponds to approximately 260 pixels. However

on my desktop monitor, 260 pixels is 6.7 cm across. And on a high-resolution iPad display, at 227 ppi (pixels per inch), 260 pixels are just 2.9 cm across. If your assigment or learning unit ask you to prepare stero images: adjust your images so they are approximately at the right separation and are approximately 500 to 600 pixels wide. Also, scale your molecules so they fill the available window and - if you have depth cueing enabled - move them close to the front clipping plane so the molecule is not just a dim blob, lost in murky shadows. Considerations for print (manuscripts etc.) are slightly different: for print output you can specify the output resolution in dpi (dots per inch). A typical print resolution is about 300 dpi: 6 cm separation at 300dpi is about 700 pixels. Print images should therefore be about three times as large in width and height as screen images.

2.13.3 Preparation of stereo views

- When assignments or leartning units ask you to create images of molecules, always create stereo views.
- Keep your images uncluttered and expressive
- Scale the molecular model to fill the available space of your image well. Orient views so they illustrate a point you are trying to make. Emphasize residues that you are writing about with a contrasting colouring scheme. Add labels, where residue identities are not otherwise obvious. Turn off side-chains for residues that are not important. The more you practice these small details, the more efficient you will become in the use of your tools.

2.13.4 Code

Always markup code using the GeSHi extension. This provides syntax highlighting, which is very useful to read the code. You simply place the code-block into opening- and closing “source” tags, and tell GeSHi which language it should assume. For R-code this looks like: Code …

[Expand] Expand for GeSHi rendered R code example … You can also use GeSHi to markup plain text - (although you can achieve a similar effect by simply beginning each line with a blank ” “). Lorem ipsum dolor sit amet …

Documents …

The section below contains Wiki-markup code that you can copy and paste for your course journal.

Wikitext Template

2.13.5 Self-evaluation

2.13.6 Notes

Here are some alternative applications – but (!) disclaimer, I myself don't use any of these (yet).. 1. Evernote - a web hosted, automatically syncing e-notebook. 1. Nevernote - the Open Source alternative to Evernote. 1. Google Keep - if you have a Gmail account, you can simply log in here. Grid-based. Seems a bit awkward for longer notes. But of course you can also use Google Docs. 1. Microsoft OneNote - this sounds interesting and if any one is using this, I'd like to hear from you. Syncing across platforms, being able to format contents and organize it sounds great. 1. RStudio projects - for development-focussed work – especially (but not exclusively) – in R, an RStudio project may be the right solution to keep your code, results, notes, manuscript drafts, literature and other assets all in one place. The great benefit is that it can all be under version control and it's super easy to share everything with colleagues on a team through GitHub Technically, GitHub documents are all publicly accessible if they are stored in repositories of free accounts - but you can commit binary files, so you can simply keep sensitive material in password-protected .zip files or otherwise encrypted.. The only downside that I can think of is that it's not possible to cross-reference and link to material. 1. I have come across "journal entries" that consist only of copy/pasted learning unit objectives… 1. If the Wiki automatically displays section numbers in its Table of Contents, you can turn that off in the preferences. 1. Media Wiki also has its own search functions that search for material everywhere on the Wiki, but this is likely not useful on the Student Wiki where many users may be writing about similar things.

2.13.7 Further reading, links and resources

If in doubt, ask! If anything about this learning unit is not clear to you, do not proceed blindly but ask for clarification. Post your question on the course mailing list: others are likely to have similar problems. Or send an email to your instructor.

About …

Author: Boris Steipe boris.steipe@utoronto.ca Created: 2017-08-05 Modified: 2019-01-05 Version: 1.3 Version history: 1.3 Emphasize habit forming and cuncurrent editing. Note on license. 1.2 Make time tags mandatory; warn against fabrication. 1.1 Add GeSHi example 1.0 First live version 0.1 First stub

Chapter 3

Prerequisites

This is a *sample* book written in **Markdown**. You can use anything that Pandoc's Markdown supports, e.g., a math equation $a^2 + b^2 = c^2$.

The **bookdown** package can be installed from CRAN or Github:

```
install.packages("bookdown")
# or the development version
# devtools::install_github("rstudio/bookdown")
```

Remember each Rmd file contains one and only one chapter, and a chapter is defined by the first-level heading `#`.

To compile this example to PDF, you need XeLaTeX. You are recommended to install TinyTeX (which includes XeLaTeX): https://yihui.name/tinytex/.

3.1 Attributions:

Original content for this book is from Boris Steipe BCB420 wiki resources licensed under  CC BY 4.0.

Icons are from the "Very Basic. Android L Lollipop" set by Ivan Boyko licensed under CC BY 3.0.