File types in bioinformatics

BE_22 Bioinformatics SS 21

January Weiner

Most of the files in bioinformatic applications are either plain text files or compressed ("zipped") text files.

Plain text (ASCII or Unicode) text file can be opened with a text editor. I don't mean a word processor (like Microsoft Word), although you definitely can open them in Word. This is for example the "Notepad" program installed by default in Windows. However, there is a catch. If you are not using Windows 10, MacOS or Linux, read the following section.

Plain text file formats – line endings

Text files use special character codes to denote line endings (EOL, newline). This are often represented by a backslash and a letter, for example the "newline" character has the ASCII code 10 and is usually written as `\n`. Different systems use different characters to represent a line ending:

- Unix and Unix-like OS (Linux, MacOS) use a single character, `\n`
- Windows use two characters, `\r\n`
- Old MacOS (pre-MacOX) used `\r`

Opening a Unix text file on an older Windows OS (e.g. Windows 7) results in incorrect representation of newlines – all lines go into one single long line. Therefore, for such older OS you need to use another program (see the last section of this file).

File types and file extensions

Extensions are the little three or four letter abbreviations preceded by a dot at the end of the file, like `.txt` or `.docx`. They are used by many operating systems to choose with which program a particular file should be opened by default.

However, extensions are merely hints or suggestions. They do not dictate the file format. Neither they are in any way standardized. Nothing and noone prevents you from creating your own extensions or removing them at all: the file contents will not change when you do. You will merely have to indicate to the OS manually with which program you want the file to be opened.

Bioinformatics uses a myriad of extensions, usually created ad hoc. In most of the cases the extension just tells you the particular format of the text file, e.g. whether the columns in a table are separated by commas (`.csv`) or tabs (`.tsv`). But it is nonetheless a text file you can open in a text editor. For example, the original of this file has the extension `.md`, indicating that it is a markdown file, but that doesn't change the fact that I am viewing and writing it in a regular text editor.

There is one notable exception. Many files in bioinformatics are very large, and text files always take up a lot of space (because each character takes up a byte of space, but usually most of that byte are zeroes). Therefore, large text files are often compressed, frequently using the standard Unix tool, gzip. Such files often have two extensions, for example `.fastq.gz`, indicating that it is a FASTQ file (sequence with quality information, likely from next generation sequencing), which has been compressed.

When you download a file from editor (e.g. clustalw output), the file may or may not have a "correct" extension (`.aln`), but even if it has, that is not helpful. Windows or MacOS have no idea what this file is, so

you need to use "Open With..." or a similar context menu entry to select an editor in order to view it. If you have problems with that, try chaning the extension to `.txt`.

What editors to use

There are the default editors, which are suitable for opening text files and viewing them, but without many helpful features, then there are programmers editors and finally programming environments (IDE, integrated development environment). The division between the latter two is fuzzy.

1. Default editors

- MacOS: TextEdit
- Windows: Notepad (note: on Windows 7 and earlier, this will not work correctly with Unix-like files)
- Linux: depends on your distribution, but usually gedit

2. Programmers editors

- vi(m), Emacs: old school. Has been around since the dawn of mankind. Dinosaurs like me use them. Extremely powerful, extremely steep learning curve, they don't work like any other editor you have seen. Learning basics of vi(m) might be useful, since you are guaranteed to have vi installed on virtually every Unix server.
- Windows: Notepad++ – versatile and powerful
- Linux: gedit already has loads of features like syntax highlighting.
- Atom: powerful editor from Microsoft/github (available for all systems). Open source. Can be used as a git/github user interface as well.
- Visual Studio Code: another one from Microsoft. Also Open Source, also embedded git support.

3. IDE (programming environments)

- Eclipse
- Bluefish something between an editor and an IDE.
- Visual Studio IDE from Microsoft