

Procesadores de texto Markup: más allá de MS Word

H. Saiz^{1*}

(1) Departamento de Biología y Geología, Física y Química Inorgánica, Escuela Superior de Ciencias Experimentales y Tecnología, Universidad Rey Juan Carlos. C/ Tulipán s/n, 28933 Móstoles, España.

* Autor de correspondencia: H. Saiz [saizhugo@gmail.com]

> Recibido el 10 de noviembre de 2017 - Aceptado el 29 de noviembre de 2017

Saiz, H. 2017. Procesadores de texto Markup: más allá de MS Word. *Ecosistemas* 26(3): 110-111. Doi.: 10.7818/ECOS.2017.26-3.14

Con la aparición del lenguaje de programación R los ecólogos hemos empezado a perder el miedo a los programas informáticos que sustituyen los clics de ratón en menús y pestañas por el uso del teclado para escribir líneas de comando. El principal motivo de ello es que R proporciona unas ventajas que otros programas de análisis estadístico no tienen (p.ej. es gratuito y de código abierto, permite realizar casi cualquier análisis y reproducirlo en otro ordenador, se actualiza constantemente, etc.), por lo que el esfuerzo dedicado a aprender R es una inversión que en el futuro nos permitirá resolver nuestros problemas de manera más eficiente. En esta nota voy a presentaros alternativas a los procesadores de texto tradicionales (p.ej. MS Word) y comentaré qué beneficios pueden suponer para realizar trabajos científicos.

Los procesadores de texto por comandos se basan en una idea sencilla. Cuando escribimos un código informático, cada línea de código que escribimos le indica al programa la acción a realizar. Por ejemplo, cuando en R escribimos:

```
x <- mean(c(1,2,3))
```

estamos indicando al programa que cree un objeto "x" que contenga la información "promedio de los números 1, 2 y 3". Al ejecutar esta línea obtenemos un objeto x con valor dos.

Del mismo modo, en procesadores de texto basados en comandos podemos indicar mediante código qué formato queremos que tengan las palabras (p.ej. cursiva, negrita), los párrafos (p.ej. sangrados, separación entre párrafos) e incluso el documento entero (p.ej. numeración de páginas, interlineados). Por ejemplo, en Markdown (un lenguaje para escribir mediante comandos) cuando escribimos:

```
**Negrita** y *cursiva*
```

el resultado es "**Negrita** y *cursiva*".

Existen diferentes procesadores de texto basados en comandos, como [LaTeX](#) o los programas basados en lenguaje Markdown. Además, incluso R tiene un paquete específico ([Rmarkdown](#)) que permite crear textos dentro de la propia consola de R y combinarlos fácilmente con otras salidas del programa. Así, es posible añadir las figuras y tablas en el texto directamente conforme las vamos

creando en R. El poder combinar diferentes funciones como analizar datos y escribir textos dentro de un mismo programa supone varias ventajas, como por ejemplo permitir el intercambio de información entre usuarios sin tener que recurrir al envío de varios ficheros. Es por ello que en los últimos años varios usuarios han propuesto el uso generalizado en ciencia de estos programas sobre los procesadores de texto tradicionales.

En general, los procesadores de texto basados en comandos presentan diferentes ventajas deseables en investigación:

Permiten incluir ecuaciones de manera más sencilla. Existen disciplinas científicas como matemáticas y física donde el uso de ecuaciones en artículos científicos es común. Por ello, es necesario contar con un procesador de texto que permita escribir ecuaciones de forma rápida y simple. Si bien es cierto que MS Word ha mejorado mucho incluyendo complementos como [MathType](#), muchos matemáticos y físicos prefieren utilizar procesadores de texto por comandos para escribir sus trabajos científicos ([Brischoux y Legagneux 2009](#)).

Simplifican el cambio de formato. Cuando se envía un trabajo científico a una revista, es habitual tener que enviar el texto con un formato particular (*sensu* bibliografía, numeración de páginas, etc.) que puede cambiar en función de la revista. Esto implica que cada vez que se envía un trabajo hay que invertir tiempo en comprobar que el trabajo se acomoda a dicho formato. Sin embargo, en los procesadores de texto por comandos el formato se define en líneas de código y, aunque aún no es común en ecología, muchas revistas proporcionan plantillas preformateadas. Además, existen repositorios de internet como [citationstyles](#) o paquetes de R como [rticles](#) que facilitan enormemente el formateo del manuscrito, incluyendo la bibliografía. De esta forma, simplemente habrá que cambiar estas líneas de código para tener el artículo con el nuevo formato que nos interese.

No tienen problemas de compatibilidad y ocupan poco espacio. Los textos generados por estos procesadores suelen estar en formatos de texto universales (p.ej. ".txt") que pueden ser abiertos por cualquier ordenador. De hecho, existen revistas científicas que permiten enviar los trabajos en estos formatos de texto. Además, los archivos de texto ocupan menos espacio que los archivos de los procesadores tradicionales como los ".doc", por lo que trabajar sobre el documento es mucho más eficiente.

Facilitan el intercambio de archivos y la reproducibilidad en la ciencia. Al poder combinar dentro de un único archivo texto con otros componentes, como análisis estadísticos, cualquiera puede reproducir exactamente un trabajo científico al completo. De esta forma, al recibir un trabajo no sólo tendríamos acceso a la información escrita en el mismo, sino que podríamos hacer los análisis, comprobar los datos, etc. En resumen, tendríamos toda la información necesaria para replicar el trabajo. Además, si compartimos los archivos en repositorios como [GitHub](#), el sistema permite llevar un registro de todos los cambios que se hagan sobre el documento, pudiendo identificar en todo momento cuándo y quién ha realizado qué modificación, y haciendo muy sencillo recuperar versiones anteriores del trabajo.

Es software libre y gratuito. Programas como MS Word requieren comprar licencias de usuario, que además no garantizan el acceso a nuevas versiones del producto. Sin embargo, los procesadores de texto por comando son gratuitos y al ser de código abierto, cualquiera puede modificarlos para el beneficio de (normalmente) todos los usuarios.

Por supuesto, el uso de este tipo de procesadores no está exento de problemas. El principal de ellos es que requieren un proceso de aprendizaje mayor que los procesadores de texto tradicionales, lo cual puede resultar problemático para usuarios no acostumbrados al uso de comandos. Por ejemplo, un estudio de la universidad de Giessen ([Knauff y Nejasmic 2014](#)) encontró que los nuevos usuarios de MS Word escribían de forma más eficiente que los nuevos usuarios de LaTeX. Aunque el estudio mostró que LaTeX también presentaba algunas ventajas (p.ej. más eficiente para escribir ecuaciones, produce menos cansancio), la respuesta general fue que trabajando en MS Word se escribía más rápido y con menos errores. Por ello, muchos usuarios de procesadores de texto tradicionales consideran que el esfuerzo necesario para aprender procesadores alternativos no compensa las posibles ventajas que éstos puedan tener.

Cómo escribir un trabajo científico es una decisión personal que depende de cada autor. Mi objetivo no es convencer a nadie para que cambien MS Word por algún procesador de texto basado en comandos. Simplemente, es dar a conocer esta herramienta de trabajo tan común en otras disciplinas científicas. Al final, cada uno debe trabajar con el método en que se sienta más a gusto. Personalmente, al trabajar en equipo lo más eficiente es que todos los colaboradores trabajen usando las mismas herramientas. Sin embargo, nunca está de más aprender nuevas herramientas que puedan

resultar necesarias para trabajos futuros. Por ejemplo, las cada vez más comunes colaboraciones entre físicos y ecólogos ([Vermeulen et al. 2013](#)) pueden hacer necesario el conocimiento de este tipo de herramientas. De este modo, cuando nos llegue el momento de trabajar juntos podamos todos comunicarnos en el mismo idioma.

En esta nota he tratado de presentaros una alternativa para escribir textos científicos. Ahora ya depende de cada uno sumergirse en el mundo de los procesadores de texto por comandos. Existen muchos manuales abiertos y gratuitos ([Oetiker et al. 1995](#); [Baumer et al. 2014](#)) para comenzar a manejar este tipo de programas, y decidir qué método de trabajo nos parece el más adecuado. Sin embargo, no debemos olvidar que el objetivo de la ciencia es transmitir los descubrimientos para que todo el mundo pueda beneficiarse de ellos. Más importante que el "cómo", es el "qué" escribimos. Y este "qué" siempre debe ser lo más riguroso posible, ya escribamos en MS Word o en cualquier otro programa.

Agradecimientos

Gracias al grupo de ecoinformática de la AEET, en especial a I. Bartomeus, I. Mendoza, A.J. Pérez-Luque y F. Rodríguez-Sánchez, por su revisión de esta nota. Hugo Saiz cuenta con una beca Juan de la Cierva-Formación del MINECO. Esta nota ecoinformática está escrita en RMarkdown. El archivo donde se incluye todo el código para reproducir el texto está incluido como material suplementario (S1).

Referencias

- Baumer, B., Cetinkaya-Rundel, M., Bray, A., Loi, L., Horton, N.J. 2014. R Markdown: Integrating a reproducible analysis tool into introductory statistics. *arXiv preprint arXiv:1402.1894*.
- Brischoux, F., Legagneux, P. 2009. Don't format manuscripts. *The Scientist* 23: 24.
- Knauff, M., Nejasmic, J. 2014. An Efficiency Comparison of Document Preparation Systems Used in Academic Research and Development. *PloS one* 9: e115069.
- Oetiker, T., Partl, H., Hyna, I., Schlegl, E. 1995. La introducción no-tan-corta a LATEX2 ϵ . Disponible en: http://portal.uned.es/pls/portal/docs/PAGE/UNED_MAIN/LAUNIVERSIDAD/UBICACIONES/01/OFERTAESTUDIOS/GRADOS_CIENCIAS/SOFTWARE%20DESCARGAS%20LINUX/MANUALES/LATEX_INTRO_153PAGS.PDF
- Vermeulen, N., Parker, J.N., Penders, B. 2013. Understanding life together: A brief history of collaboration in biology. *Endeavour* 37: 162-171.