

# Introduction à l'éditeur emacs et à son mode ESS pour R

Christophe Pouzat

Laboratoire de Physiologie Cérébrale

CNRS UMR 8118 et Université Paris-Descartes

e-mail : [christophe.pouzat@parisdescartes.fr](mailto:christophe.pouzat@parisdescartes.fr)

web : [http://www.biomedicale.univ-paris5.fr/phycerv/C\\_Pouzat.html](http://www.biomedicale.univ-paris5.fr/phycerv/C_Pouzat.html)

4 janvier 2011

# Table des matières

- 1 Introduction** **3**
  
- 2 emacs** **4**
  - 2.1 Un peu d'histoire . . . . . 4
  - 2.2 Obtenir emacs . . . . . 5
  - 2.3 Par où commencer ? . . . . . 6
  - 2.4 Un peu de jargon « emacsien » . . . . . 8
  - 2.5 Fonctionnalités . . . . . 9
  - 2.6 Obtenir de l'aide . . . . . 12

# 1 Introduction

Ce texte reprend et élargit la présentation donnée pendant le séminaire. Ce qui est en plus correspond à des détails sur comment faire certaines des « actions » exposées. D'autres détails ont aussi été inclus pour essayer de répondre, à posteriori, à certaines des questions qui ont été posées pendant et après la présentation.

Bien que j'ai préparé ce document pour qu'il soit pratique à imprimer, j'ai commis une erreur de base en prenant l'essentiel de mes photos d'écrans dans ma configuration habituelle d'emacs, c'est-à-dire avec un fond noir et des caractères blancs. Le résultat est que ce que vous lisez maintenant n'est pas si pratique que ça à lire une fois imprimé et qu'en plus l'impression consomme beaucoup trop d'encre... Je m'en excuse et je corrigerai cette erreur avec une nouvelle série de figures dès que j'en aurai le temps.

**Pourquoi utiliser emacs ?** Les raisons pour utiliser emacs sont multiples et dépendent très probablement de l'utilisateur à qui cette question est posée. Comme emacs peut tout faire, ou presque, des gens très différents l'utilisent. Du gros des membres « réguliers » de la *R foundation* (c.-à-d. les membres de la core team) jusqu'à certains hellénistes, c.-à-d. des gens qui doivent lire et surtout écrire des choses du genre σοφιστής à longueur de journée. N'ayant pas la prétention de représenter l'ensemble de la communauté des utilisateurs et étant surtout trop paresseux pour me lancer de l'établissement d'une liste quasiment sans fin, je me limiterai à mon cas personnel. Les intérêts de l'auditeur, régulier ou non, du *semin-R* devraient recouper les miens sur un point au moins : R, que j'utilise pour ma part avec emacs (quelle surprise !). Mais j'utilise emacs aussi pour naviguer dans mes fichiers et répertoires ; pour lire et écrire mes courriels ; pour lire les fils RSS auxquels je suis abonné ; pour visiter certains sites Internet essentiellement textuels (comme le site de R) ; pour coder en C (ce qui m'arrive heureusement rarement maintenant) ; pour déboguer ces codes ; pour écrire des textes, comme le document que vous lisez en ce moment avec X<sub>Y</sub>TeX ; pour interagir directement avec le système par une « ligne de commande », ce que je fais lorsque je compile R ou emacs ; pour éditer mes fichiers html ; pour faire des calculs symboliques avec Maxima ou numériques avec Octave (un clone de Matlab sous licence GPL ; si j'avais Matlab sur ma machine, je pourrais aussi l'utiliser avec emacs). Il serait en fait plus rapide de lister ce que je ne fais pas avec emacs : visiter des sites Internet avec beaucoup d'images ; lire de gros fichiers pdf ; le multimédia (même si c'est faisable dans emacs avec EMMS, que je n'ai pas encore eu le temps d'essayer) ; modifier des fichiers Word (j'ai mal rien qu'à écrire ce mot) ou OpenOffice, voir même, comble du mauvais goût, Excel...

Comme R, emacs est *open source* et distribué sous licence GPL. Il est disponible sur tous les systèmes d'exploitation (GNU/Linux, FreeBSD, Mac OS X, MS Windows, MS DOS, Solaris, AIX, Sun OS, etc) ou presque. Il est activement développé. Enfin comme celle de R, la communauté emacs est une communauté de « haut niveau » ce qui a ses exigences et qui est parfois frustrant, mais ce qui, très (pour ne pas dire

trop) souvent représente une bouffée d'oxygène quand on est actif, comme moi, dans une communauté médiocre et suffisante, les neurosciences, dont les membres pensent qu'ils peuvent *tout comprendre* <sup>1</sup> et que toute bonne explication peut loger en quatre pages avec quatre figures... Cette remarque sur le haut niveau des communautés emacs et R (qui sont en partie les mêmes) ne devrait surtout pas décourager ou intimider le lecteur « débutant ». J'ai écrit ma thèse en 1998 avec Word et effectué les analyses contenues dans celle-ci avec Igor (un logiciel d'analyse de données qui ne vaut guère mieux qu'Excel); il ne faut donc pas désespérer, rien n'est irréversible! Il faut juste se dire qu'on apprend mieux et plus vite d'une communauté peuplée de gens plutôt intelligents... des fois on se sent forcément bête, comme quand on se fait « Ripleyed » <sup>2</sup> sur la *mailing list* de R; c'est le prix à payer. Avec un minimum d'humilité on s'en sort très bien.

## 2 emacs

### 2.1 Un peu d'histoire

Le nom emacs est un acronyme pour : *Editing Macros*; il a été développé à partir de 1976 par Richard Stallman et Guy L. Steele, Jr. et sa version GNU est aujourd'hui encore supervisée par Richard Stallman.

**Philosophie de emacs** L'introduction d'un *article* <sup>3</sup> de R. Stallman (1981) la définit très clairement :

EMACS is a real-time display editor which can be extended by the user while it is running.

Extensibility means that the user can add new editing commands or change old ones to fit his editing needs, while he is editing. EMACS is written in a modular fashion, composed of many separate and independent functions. The user extends EMACS by adding or replacing functions, writing their definitions in the same language that was used to write the original EMACS system. We will explain below why this is the only method of extension which is practical in use: others are theoretically equally good but discourage use, or discourage nontrivial use.

Extensibility makes EMACS more flexible than any other editor. Users are not limited by the decisions made by the EMACS implementors. What we decide is not worth while to add, the user can provide for himself.

---

1. Au sens où le fait que la réponse à une question puisse être trop subtile pour qu'ils la comprennent ne semble ne jamais leur avoir traversé l'esprit... Le contraste avec les statisticiens, par exemple, est flagrant.

2. Il faut entendre par là : « se faire remonter les bretelles par Brian Ripley, suite à une question dont la réponse se trouve, par exemple, sur la FAQ ».

3. <http://www.gnu.org/software/emacs/emacs-paper.htm>.

He can just as easily provide his own alternative to a feature if he does not like the way it works in the standard system.

A coherent set of new and redefined functions can be bound into a library so that the user can load them together conveniently. Libraries enable users to publish and share their extensions, which then become effectively part of the basic system. By this route, many people can contribute to the development of the system, for the most part without interfering with each other. This has led the EMACS system to become more powerful than any previous editor.

User customization helps in another, subtler way, by making the whole user community into a breeding and testing ground for new ideas. Users think of small changes, try them, and give them to other users—if an idea becomes popular, it can be incorporated into the core system. When we poll users on suggested changes, they can respond on the basis of actual experience rather than thought experiments.

To help the user make effective use of the copious supply of features, EMACS provides powerful and complete interactive self-documentation facilities with which the user can find out what is available.

A sign of the success of the EMACS design is that EMACS has been requested by over a hundred sites and imitated at least ten times.

Cette philosophie ne devrait pas paraître choquante aux utilisateurs de R que vous êtes...

## 2.2 Obtenir emacs

**Deux versions** Comme nous allons le voir très vite, l'élégante sobriété de l'interface graphique d'emacs peut paraître quelque peu déroutante à l'utilisateur « moderne » ; habitué qu'il est à voir son écran encombré de multiples bandeaux et « décorations » diverses, bardés de cibles qu'il ne cherche, d'ailleurs, à atteindre avec son « mulot » qu'une infime fraction du temps. Cet exposé est consacré à la version GNU de emacs dont le site est hébergé par le serveur de la *Free Software Foundation* à l'adresse : <http://www.gnu.org/software/emacs/>. À la fin du siècle siècle dernier, une version d'emacs plus « conviviale », avec plus d'icônes, a été développée et distribuée sous le nom de XEmacs <sup>4</sup>. Le développement de cette branche d'emacs semble aujourd'hui abandonné, car le gros des améliorations graphiques est désormais disponible « par défaut » dans emacs. Vous entendrez parfois des gens, souvent d'anciens utilisateurs de XEmacs, dire que le développement d'emacs est arrêté ; c'est inexact, une nouvelle distribution d'emacs apparaît à peu près une fois par an, ce qui n'est pas si rare que cela si on considère qu'un logiciel beaucoup plus jeune comme R en voit deux pendant la même période.

---

4. <http://www.xemacs.org/>

**Obtenir emacs** L'utilisateur de Linux trouvera emacs parmi les « paquets » de sa distribution favorite, mais s'il souhaite utiliser la dernière version, il aura intérêt à télécharger ses « sources »<sup>5</sup> et à compiler emacs lui même. Les instructions de compilation données dans le fichier INSTALL de la distribution sont suffisamment claires pour qu'un utilisateur ayant un minimum d'expérience avec cet exercice puisse le faire.

Les utilisateurs de Windows et de Mac OS n'auront qu'à télécharger les distributions de Vincent Goulet<sup>6</sup>. Ces distributions ont en plus le bon goût de contenir ESS et AUCTeX.

## 2.3 Par où commencer ?

**Quelques détails importants** Comme emacs a été développé avant la généralisation de l'emploi du « mulot »<sup>7</sup> et qu'une des fortes motivations à son développement été l'obtention d'une navigation rapide dans les fichiers, ses concepteurs ont introduits de nombreuses *fonctions* à cette fin. Certaines déplacent le curseur d'un mot ou d'un paragraphe, en avant ou en arrière, etc. Pour ne pas avoir à passer trop de temps à taper le nom de ces fonctions, un recours systématique aux raccourcis clavier s'est imposé. C'est ce qui rend l'apprentissage d'emacs un peu pénible, mais c'est une des caractéristiques qui le rendent terriblement efficace. Dans ce qui suit je vais souvent donner le raccourci clavier permettant d'effectuer un tâche spécifique. Les touches control et alt vont être abondamment sollicitées dans ce cadre, à tel point qu'il est d'usage de représenté la pression des touches control *et* x *simultanément* par le code : C-x ; et la pression des touches alt *et* g *simultanément* par le code : M-g. On a ainsi les commandes suivantes, où « l'accord de touches » C-x C-c signifie qu'on tape C-x *suivi de* C-c :

- C-x C-c : quitter définitivement emacs ;
- C-x C-f : lire un fichier ;
- C-x C-s : sauver un fichier ;
- C-g : abandonner une commande ;
- C-x u, C-\_ ou C-/ : annuler une modification ;
- C-s : rechercher en avant ;
- C-r : rechercher en arrière ;

emacs étant personnalisable, ces raccourcis peuvent évidemment être redéfinis par l'utilisateur.

---

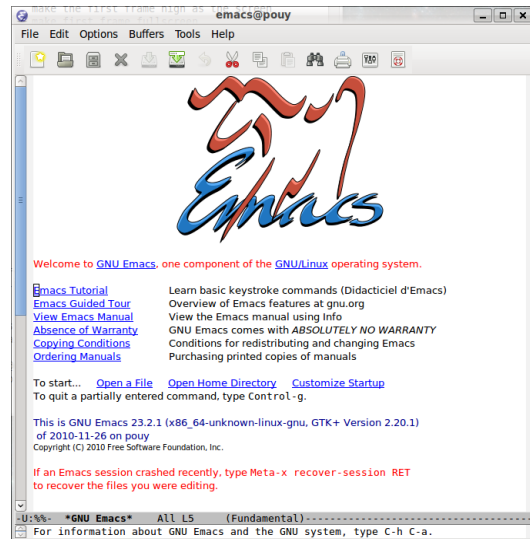
5. La liste des miroirs : <http://www.gnu.org/prep/ftp.html>.

6. Adresse : <http://vgoulet.act.ulaval.ca/ressources/emacs/>.

7. [http://fr.wikipedia.org/wiki/Souris\\_\(informatique\)](http://fr.wikipedia.org/wiki/Souris_(informatique))

Ceci étant clarifié, le débutant pourra commencer avec :

- la « visite guidée » (choisir Emacs Guided Tour sur l'écran de départ);
- le didacticiel (choisir Emacs Tutorial sur l'écran de départ);
- la page « HOWTO Débuter sous Emacs »<sup>8</sup>;
- le wiki<sup>9</sup> avec ses versions française et anglaise (plus fournie);
- emacs, comme la plupart de ces modes majeurs, vient avec une *reference card*, fichier pdf d'une ou deux pages décrivant les raccourcis clavier les plus fréquents; il est pratique d'imprimer ces documents.



Une fois les premières étapes franchies, l'utilisateur voulant aller plus loin pourra consulter :

- la documentation incluse dans emacs qui est complète et bien écrite (obtenue avec le raccourci : C-h i), mais comme celle de R il faut un peu de temps pour s'y habituer;
- le livre *Learning GNU Emacs* de O'Reilly en est à sa 3e édition (la 2e a été traduite en français);
- le livre *Teach Yourself Emacs in 24 Hours*, publié par Sams; c'est mon favorit (on peut en trouver une version HTML, *illégal*, sur la toile avec Google, on peut aussi l'acheter d'occasion, j'ai payé le mien 8 €, frais d'envoi compris);
- *The Woodnotes Guide to Emacs for Writers*<sup>10</sup> qui est très bien pour l'édition de textes;
- il y a aussi quelques vidéos plutôt bien, mais en anglais, sur youtube.

---

8. <http://guidespratiques.traduc.org/vf/Emacs-Beginner-HOWTO.html>  
9. <http://www.emacswiki.org/emacs-fr>  
10. <http://www.therandymon.com/papers/emacs-for-writers.pdf>.

## 2.4 Un peu de jargon « emacsien »

Avant de me lancer dans la descriptions des fonctionnalités « principales » d'emacs, j'introduis, suivant le document « [HOWTO Débuter sous Emacs](#) », un minimum de terminologie qu'il est nécessaire de connaître pour pouvoir lire la documentation d'emacs sans trop de peine.

**Tampons et Fichiers** Contrairement à de nombreux éditeurs, quand vous ouvrez un fichier dans emacs, celui-ci ne reste pas ouvert tout le temps que vous travaillez dessus. Au lieu de cela, emacs lit le fichier et le place dans un **tampon** (*buffer*) en mémoire. Tant que vous éditez le tampon et travaillerez avec les données rien ne sera changé sur le disque. C'est lorsque vous sauvegarderez explicitement le tampon que le fichier sur le disque sera modifié. Cette méthode possède des avantages et des inconvénients mais il est important que vous compreniez qu'emacs fonctionne de cette manière. Vous verrez ainsi le terme *buffer* utilisé dans la documentation d'emacs, dans les différents modes, dans les paquets, etc... Il faut juste réaliser que tampon (*buffer*) signifie « une copie du fichier qui se situe en mémoire ». Il faut aussi souligner que le tampon ne pointe pas toujours sur un fichier spécifique. Souvent emacs crée des tampons comme résultat d'une commande que vous avez exécuté, pour afficher une liste de choix à effectuer, et ainsi de suite.

**Points et Régions** En utilisant emacs, vous entendrez ou lirez souvent des références au sujet des **points**. Point signifie approximativement curseur ; la différence entre point et curseur n'est pas importante lorsque vous commencer avec emacs : le curseur est la représentation visuelle du point. Comme la plupart des éditeurs modernes, emacs permet d'effectuer des opérations sur une section du tampon courant (indentation, correction orthographique, coupure, collage, copie, etc). Vous pouvez **marquer** un bloc de texte en utilisant le clavier ou la souris et effectuer alors des opérations uniquement sur la zone sélectionnée. Dans emacs, ces blocs de textes sont nommés des **régions** (*regions*).

**Fenêtres et Cadres** L'emploi du terme **fenêtre** (*window*) porte maintenant probablement à confusion ; il a une origine historique qui remonte à l'époque où les ordinateurs n'avaient ni interface graphique ni gestionnaire de fenêtres. Une fenêtre dans emacs est une zone de l'écran dans laquelle un tampon est affiché. Quand emacs est lancé, vous avez seulement une fenêtre affichée. Quelques fonctions d'emacs, comme la documentation et l'aide, ouvrent souvent (temporairement) une fenêtre supplémentaire dans votre écran d'emacs. Les fenêtres d'emacs ne sont pas identiques à celle de votre gestionnaire de fenêtres qui sont elles appelées **cadres** (*frames*) dans le jargon emacs. Vous pouvez ainsi avoir plusieurs fenêtres dans un seul cadre. Nous allons en voir plusieurs exemples dans ce qui suit. Les fenêtres d'emacs sont en fait analogues aux onglets de votre navigateur Internet.



## 2.5 Fonctionnalités

**Un navigateur de fichiers** Le mode **dired** d'emacs permet de naviguer dans une arborescence de fichiers. Ce mode est activé par l'accord de touches : C-x d. On voit alors apparaître la liste alphabétique des fichiers (en noir), répertoires (en bleu) et des fichiers « de sauvegarde » (comme .bashrc~) en gris.

La **barre de menu** (*menu bar*) est en haut de l'écran. La barre (presque) en bas de l'écran est la **ligne de mode** (*mode line*); elle commence par `-U:%%-`, sur la figure. En dessous de celle-ci, où apparaît `Blink-Cursor mode disabled`, on trouve la **zone d'écho** (*echo area*) qui est la voix principale d'interaction avec emacs quand des fonctions sont explicitement utilisées. La ligne de mode comporte de nombreuses indications sur l'objet visualisé dans la **fenêtre** immédiatement au-dessus; je n'ai pas la place de les détailler toutes ici et vous invite à consulter la documentation (C-h i) de votre distribution emacs ou sa [version disponible sur la toile](#). La lettre **U** indique que les caractères et symboles sont codés en **UTF-8**; le symbole `~` indique que mon répertoire utilisateur (home directory) est montré; **Top** signifie que tout le répertoire n'est pas visible dans le fenêtre et que seulement la partie « du haut » est visible; `(12,57)` signifie que le curseur se trouve en ligne 12, colonne 57; `(Dired by name)` signifie que le mode dired est actif et que les fichiers sont ordonnés par noms. Il suffirait ici de taper `s` pour que les fichiers soient ordonnées par dates et `(Dired by date)` apparaîtrait à la place de `(Dired by name)`.

Si je cherche ici un répertoire particulier, comme mon répertoire `Talks_Lectures`, je peux m'épargner une navigation par les flèches ou avec la souris et la barre de défilement (voir plus loin) en utilisant les commandes de recherches. Pour cela, je tape d'abord C-s (rechercher en avant) ce qui provoque l'apparition, dans la zone d'écho de **I-search:** et l'indication de mode `(Dired by name)` est modifiée pour devenir

`(Dired by name Isearch)`, ce qui signifie que dans le « mode majeur » dired, le « mode mineur » Isearch a été activé. Je commence maintenant à taper le nom du répertoire ou du fichier qui m'intéresse `Ta` et au fur et à mesure que mes caractères sont entrés, ils apparaissent dans la zone d'écho et emacs déplace la partie du répertoire visualisé de sorte que les répertoires et fichiers dont le nom commence par `Ta` soient centrés dans la fenêtre. Lorsque le répertoire ou fichier recherché est sous le point (ou curseur), il suffit de taper la touche entrée (que nous re-

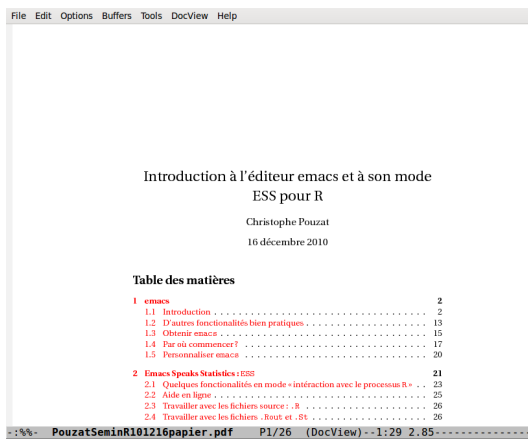
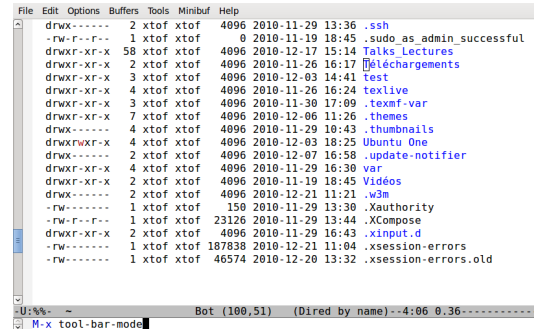
```
File Edit Options Buffers Tools Operate Mark Regexp Immediate Subdir Help
~/home/xtof:
total used in directory 816 available 833231672
drwxr-xr-x 71 xtof xtof 4096 2010-12-20 13:39 .
drwxr-xr-x 5 root root 4096 2010-12-17 14:57 ..
drwxr-xr-x 4 xtof xtof 4096 2010-11-26 16:18 administration
drwx----- 4 xtof xtof 4096 2010-12-01 18:57 .sedobe
drwx----- 1 xtof xtof 176 2010-11-29 13:44 .authinfo
drwxr-xr-x 3 xtof xtof 4096 2010-12-08 14:55 Base Biblio
drwx----- 7 xtof xtof 4096 2009-03-13 18:21 Base Figures
-rw----- 1 xtof xtof 3606 2010-12-20 11:36 .bash_history
-rw-r--r-- 1 xtof xtof 226 2010-11-19 18:35 .bash_logout
-rw-r--r-- 1 xtof xtof 3608 2010-11-29 13:44 .bashrc
-rw-r--r-- 1 xtof xtof 3103 2010-11-19 18:35 .bashrc~
-rw-r--r-- 1 xtof xtof 53176 2010-12-16 17:12 .bbdb
-rw-r--r-- 1 xtof xtof 53156 2010-12-15 16:48 .bbdb~
drwxr-xr-x 2 xtof xtof 4096 2010-12-14 09:35 bin
drwxr-xr-x 2 xtof xtof 4096 2010-11-19 18:45 Bureau
drwx----- 9 xtof xtof 4096 2010-12-20 13:32 .cache
-rw-r--r-- 1 xtof xtof 28607 2010-12-14 14:51 clavier.pdf
drwx----- 3 xtof xtof 4096 2010-11-30 11:08 .compiz
drwxr-xr-x 13 xtof xtof 4096 2010-12-07 17:00 .config
-rw----- 1 xtof xtof 60 2010-11-29 16:18 .cvspass
-U:%%- ~ Top (12,57) (Dired by name)--5:15 1.06-----
Blink-Cursor mode disabled
```

```
File Edit Options Buffers Tools Operate Mark Regexp Immediate Subdir Help
-rw-r--r-- 1 xtof xtof 675 2010-11-19 18:35 .profile
drwxr-xr-x 2 xtof xtof 4096 2010-11-19 18:45 Public
drwx----- 2 xtof xtof 4096 2010-12-20 11:23 .pulse
-rw----- 1 xtof xtof 256 2010-11-19 18:45 .pulse-cookie
drwxr-xr-x 48 xtof xtof 4096 2010-11-29 19:23 R
-rw----- 1 xtof xtof 34343 2010-12-21 11:13 .recently-used.xbel
-rw-r--r-- 1 xtof xtof 75 2010-12-14 14:48 .rprofile
drwxr-xr-x 9 xtof xtof 4096 2010-11-29 16:30 share
-rw-r--r-- 1 xtof xtof 509 2010-10-04 21:32 .signature
drwx----- 2 xtof xtof 4096 2010-11-29 13:36 .ssh
-rw-r--r-- 1 xtof xtof 0 2010-11-19 18:45 .sudo_as_admin_successful
drwxr-xr-x 58 xtof xtof 4096 2010-12-17 15:14 Talks Lectures
drwxr-xr-x 2 xtof xtof 4096 2010-11-26 16:17 Téléchargements
drwxr-xr-x 3 xtof xtof 4096 2010-12-03 14:41 test
drwxr-xr-x 4 xtof xtof 4096 2010-11-26 16:24 texlive
drwxr-xr-x 3 xtof xtof 4096 2010-11-30 17:09 .texmf-var
drwxr-xr-x 7 xtof xtof 4096 2010-12-06 11:26 .themes
drwx----- 4 xtof xtof 4096 2010-11-29 10:43 .thumbnails
drwxr-xr-x 4 xtof xtof 4096 2010-12-03 18:25 Ubuntu One
drwx----- 2 xtof xtof 4096 2010-12-07 16:58 .update-notifier
drwxr-xr-x 4 xtof xtof 4096 2010-11-29 16:30 var
drwxr-xr-x 2 xtof xtof 4096 2010-11-19 18:45 Vidéos
-U:%%- ~ 75% (99,53) (Dired by name Isearch)--11:23 0.61:
I-search: Ta
```

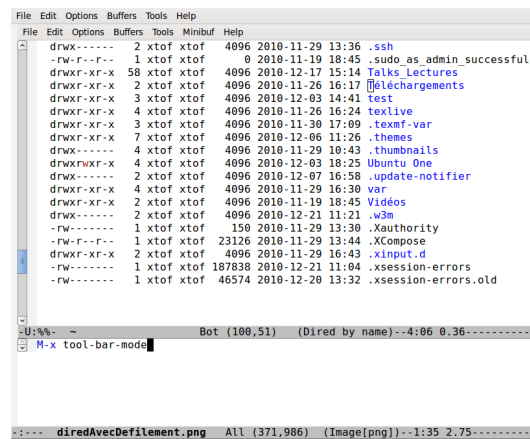
présenterons désormais par la convention d'emacs : RET). Remarquez aussi que les indications de la ligne de mode ont changé, **TOP** a été remplacé par **75%** ce qui indique que la position du point est à 75 % de la taille du fichier ; de même, les indications de position du point **(12,57)** ont changé pour **(99,53)**. Cela n'a l'air de rien, mais en pratique je gagne un temps considérable – un petit peu à chaque fois, mais je le fais très souvent – quand je navigue dans mes fichiers.

J'ai mentionné, dans le dernier paragraphe, la navigation avec la souris et la **barre de défilement** (*scroll bar*), mais aucune barre de défilement n'est présente sur mes deux dernières figures... Je peux facilement modifier cela en tapant **M-x**, ce qui fait apparaître **M-x** dans la zone d'écho et déplace le point dans celle-ci. Je tape ensuite **scroll-bar-mode** suivi de RET et une barre de défilement apparaît à gauche de la fenêtre. Je peux m'en débarrasser en répétant exactement le même séquence (**M-x scroll-bar-mode RET**). Quand on ne se souvient pas du nom exacte d'une fonction, on peut taper le nom partiel suivit de la touche TAB. Emacs coupe alors la fenêtre en deux et propose les compléments possibles dans la nouvelle fenêtre, il suffit de cliquer sur le nom de la fonction souhaitée, ou de taper les caractères maquant rendant la complétion unique suivis de TAB, pour que le nom complet apparaisse dans la zone d'écho et que la fenêtre présentant les complétions possibles disparaissent. On peut aussi activer ou désactiver la **barre d'outils** (*tool bar*) en tapant : **M-x tool-bar-mode RET**.

Les fichiers « simples », comme les fichiers PDF et les fichiers PNG (ou autres formats graphiques) peuvent être visualisés directement dans emacs. À partir de **direc**, il suffit de placer le point / curseur sur le fichier en question puis de taper RET. La figure suivante à gauche montre le présent document, un fichier PDF, visualisé dans emacs. La navigation se fait avec les touches **n** (*next* : page suivante), **p** (*previous* : page précédente), etc.



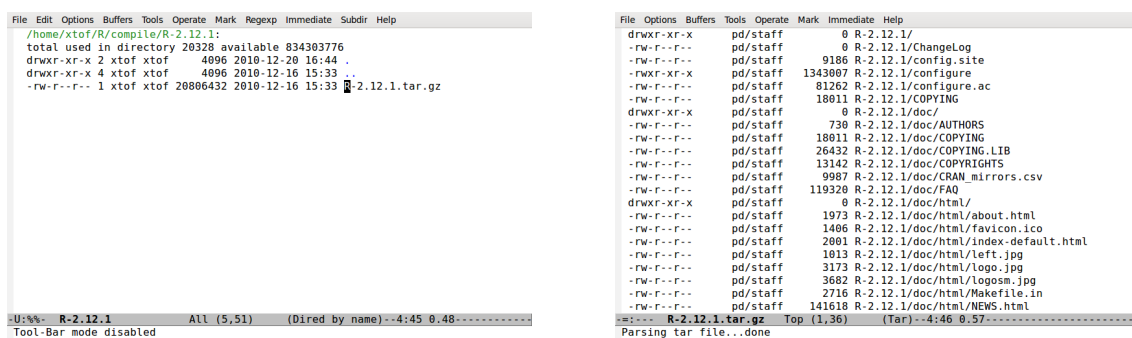
Un fichier PDF visualisé directement avec emacs grâce au mode DocView.



Un fichier PNG visualisé directement avec emacs grâce au mode Image.

Le mode **DocView**, automatiquement activé pour visualiser les fichiers PDF, ne permet pas de profiter des éventuels hyperliens présents dans ces fichiers. Pour visualiser les gros fichiers PDF ou pour profiter de la navigation par hyperliens, je recommande d'utiliser un logiciel dédié comme Acrobat Reader, xpdf ou evince. Ceux-ci peuvent être directement appelé depuis dired, sur le fichier choisi ; pour ce faire, il suffit, une fois que le point se trouve sur le fichier, de taper & suivi du nom de la commande invoquant votre logiciel de visualisation favori, comme, par exemple, acroread pour Linux. La partie droite de la figure précédente illustre la visualisation d'un fichier PNG directement avec emacs. Cette figure est un peu déroutante, je m'en excuse, puisque emacs est utilisé ici pour visualiser une photo d'écran, contenue dans le fichier diredAvecDefilement.png, de emacs... que voulez-vous, une fois qu'on est « tombé dans le GNU » on a du mal à sortir de l'auto-référence. Je pourrais ici utiliser, en mode dired, avec le point placé sur le fichier diredAvecDefilement.png, la commande : & gimp, pour ouvrir le fichier avec gimp, un logiciel de « manipulation d'images » (clone libre d'Adobe photoshop). La commande : & imagej m'ouvrirait le fichier avec ImageJ, un logiciel libre de traitement d'images.

Si vous avez l'habitude de compiler vous mêmes vos programmes, comme R ou emacs, vous aurez probablement, quelque part dans votre arborescence de répertoires, un ou des fichiers comprimés et archivés, c.-à-d. des fichiers se terminant par .tar.gz (ou .tgz) sous Linux ou par .zip sous Windows. De même si dans le cadre de collaborations, vous êtes amenés à partager et échanger de gros et multiples fichiers de données, vous vous êtes sans doute aperçus que ce type d'opérations est grandement facilité par l'usage de compression / archivage des fichiers. Bref, je vais supposer ici que vous avez un des ces fichiers, comme le dernier source de R, R-2.12.1.tar.gz lors de l'écriture de ce document, dans un de vos répertoires. Je suppose aussi que vous souhaitez voir ce que ce fichier contient sans utiliser explicitement votre logiciel de décompression favori. Heureusement, c'est très facile avec le mode dired d'emacs, comme l'illustre la figure suivante.



« Visite » du répertoire /home/xtof/R/compile/R-2.12.1 en mode dired, le point étant placé sur le fichier archivé et comprimé R-2.12.1.tar.gz, le fichier source de la dernière version de R lorsque ce document a été écrit.

Après avoir tapé RET il est possible d'examiner directement le contenu de l'archive montrée à gauche. Emacs passe alors automatiquement en mode Tar.

Si vous êtes satisfait du contenu de votre archive, comme le fichier R-2.12.1.tar.gz ci-dessus, vous voudrez sans doute décompresser celle-ci. Sur un système Unix / Linux ou sur un système Windows « civilisé » par l'installation de MinGW (Minimalist GNU for Windows), ceci se fait efficacement en ligne de commandes par la com-

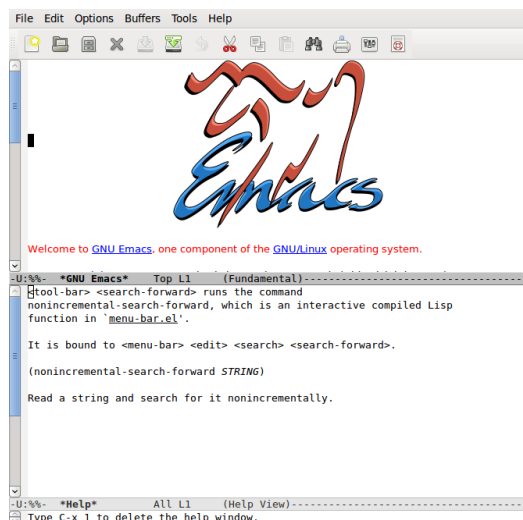
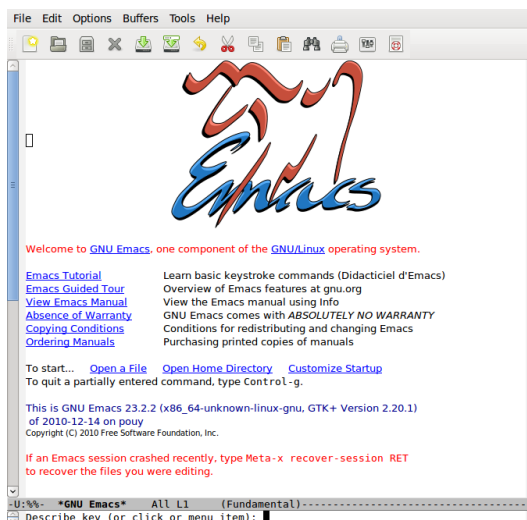
mande tar (si quelqu'un peut me renseigner sur comment faire cela sur MacOS je rajouterai l'info). La bonne nouvelle ici est que cette dernière peut être invoquée directement depuis le mode dired.

Lorsque le point est situé sur le fichier archivé, il suffit de taper « ! » ce qui fait apparaître dans la zone d'écho : `! on R-2.12.1.tar.gz :.` On tape alors : `tar -xzf RET`, et le tour est joué, un nouveau sous répertoire R-2.12.1 apparaît dans le répertoire courant.

```
File Edit Options Buffers Tools Minibuf Help
/home/xtof/R/compile/R-2.12.1:
total used in directory 28328 available 831152248
drwxr-xr-x 2 xtof xtof 4896 2011-01-03 17:18 ..
drwxr-xr-x 4 xtof xtof 4896 2010-12-16 15:33 .
-rw-r--r-- 1 xtof xtof 28886432 2010-12-16 15:33 R-2.12.1.tar.gz
-U:%%- R-2.12.1 All (5,51) (Dired by name)--5:19 2.72-----
! on R-2.12.1.tar.gz: tar -xzf
```

## 2.6 Obtenir de l'aide

Après ce rapide survole de quelques unes des fonctionnalités de emacs, le lecteur désirant « s'y mettre sérieusement » aura forcément besoin, plutôt tôt que tard, d'obtenir de l'aide. Tout comme R, emacs met à la disposition de ses, heureux, utilisateurs une aide abondante, qui, comme celle de R peut paraître un peu déroutante et « sèche » au premier abord. Elle est néanmoins extrêmement complète et apprendre à la maîtriser devrait être une priorité pour un débutant. Le plus simple est peut être de commencer par l'aide associée à une fonction accessible par un des menus déroulant ou par une des icônes de la barre d'outils. Cette dernière est activée par l'accord de touche C-h k. Dans la zone d'écho emacs indique alors : Describe key (or click or menu item) : comme illustré ci-dessous à gauche. Si je clique alors sur la paire de jumelles de la barre d'outils, emacs sépare le cadre en deux fenêtres, celle du haut étant identique à la précédente et celle du bas contient l'aide demandée (ci-dessous à droite).



Cette aide m'apprend que la touche que je viens de sélectionner, `<tool-bar><search-forward>` lance la *commande* ou fonction `nonincremental-search-forward` et que cette fonction est interactive – lorsqu'elle est appelée, emacs nous demande d'entrer une chaîne de caractères à chercher, comme

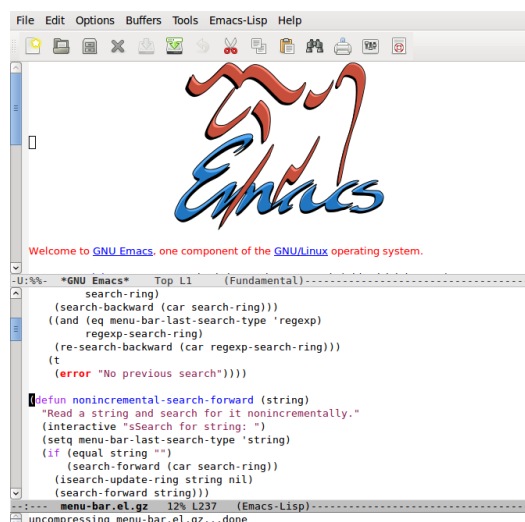
nous l'avons fait précédemment (page 9) – et est codée en Lisp. L'avant dernière ligne nous donne la syntaxe attendue par l'interpréteur lorsque la fonction est appelée :

`(nonincremental-search-forward STRING)`

Cette ligne correspond à ce qui se trouve dans la section Usage de l'aide de R. On voit clairement ici la différence avec la syntaxe employée par R, où on aurait par exemple :

`nonincremental.search.forward(STRING)`

De même que dans R, le code d'une fonction est toujours accessible en tapant le nom de la fonction *non suivi de parenthèses* puis RET en ligne de commande, le code d'une fonction Lisp de emacs peut être obtenu en sélectionnant ici, avec la souris, ou en déplaçant le point, sur `menu-bar.el` qui est le nom du fichier source (c.-à-d. équivalent à un fichier `.R` de R) contenant la définition de la fonction.



## Index

*buffer*, 8  
*echo area*, 9  
*frame*, 8  
*menu bar*, 9  
*mode line*, 9  
*reference card*, 7  
*region*, 8  
*scroll bar*, 10  
*window*, 8  
C-/, 6  
C-\_, 6  
C-g, 6  
C-h k, 12  
C-r, 6  
C-s, 6  
C-x, 6  
C-x C-c, 6  
C-x C-f, 6  
C-x C-s, 6  
C-x u, 6  
M-g, 6  
M-x, 10  
RET, 10  
*entrée*, 9  
*tool bar*, 10  
  
barre d'outils, 10  
barre de défilement, 10  
barre de menu, 9  
  
cadre, 8  
  
dired, 9  
DocView, 10, 11  
  
fenêtre, 8  
  
Image, 10  
  
ligne de mode, 9  
  
marquer, 8  
  
point, 8  
  
raccourcis clavier, 6  
région, 8  
  
tampon, 8  
Tar, 11  
  
zone d'écho, 9