

# Una introduzione a GNU Emacs & AUCT<sub>E</sub>X

Mosè Giordano

Orlando Iovino

matteo.l

17 aprile 2012

## Sommario

In questo articolo vengono brevemente presentati il programma Emacs ed il pacchetto AUCT<sub>E</sub>X per le principali piattaforme: Windows, GNU/Linux e Mac OS. Dopo una introduzione ai programmi, verranno esposte le modalità per poter installare e configurare Emacs e AUCT<sub>E</sub>X per poter scrivere documenti L<sup>A</sup>T<sub>E</sub>X. L'articolo non ha la pretesa di essere un *manuale*, bensì un piccolo tutorial per far avvicinare gli utenti di L<sup>A</sup>T<sub>E</sub>X a questo programma, non così troppo complicato come spesso viene indicato.

<b>Indice</b>	<b>6</b>	<b>Ricerca diretta-inversa . . . . .</b>	<b>8</b>
<b>1 Introduzione . . . . .</b>	<b>1</b>	<b>7 Skeleton mode . . . . .</b>	<b>11</b>
<b>2 Installare Emacs . . . . .</b>	<b>2</b>	<b>8 Reference Card . . . . .</b>	<b>12</b>
<b>3 Installare AUCT<sub>E</sub>X . . . . .</b>	<b>3</b>	<b>9 Conclusioni . . . . .</b>	<b>12</b>
<b>4 Personalizzare Emacs &amp; AUCT<sub>E</sub>X . . . . .</b>	<b>5</b>	<b>Bibliografia . . . . .</b>	<b>14</b>
<b>5 Correttore ortografico . . . . .</b>	<b>7</b>	<b>Licenza d'uso &amp; Colophon . . . . .</b>	<b>14</b>

## 1 Introduzione

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Lo sviluppo di Emacs è iniziato negli anni 1970 ai laboratori di Intelligenza Artificiale del MIT, cioè molto prima che si diffondesse l'informatica che conosciamo oggi. Ciò comporta che Emacs utilizza una terminologia spesso diversa da quella utilizzata dalla stragrande maggioranza degli altri programmi che usiamo quotidianamente. Per questo motivo non bisogna sorprendersi se l'operazione di *tagliare* del testo (generalmente indicata in inglese con *cutting*) in Emacs viene chiamata *killing* e la successiva operazione di *incollare* il testo copiato (in inglese *pasting*) viene chiamata *yanking*.

Emacs è famoso per rendere possibile l'esecuzione di qualsiasi operazione usando solo la tastiera, cioè senza l'ausilio del mouse. A ogni comando può essere associata una combinazione di tasti. In questa guida seguiremo la notazione diffusa nel mondo di Emacs per indicare i tasti. Quindi il tasto `Control` verrà indicato con `C`, l'`Invio` viene indicato con `RET`, mentre `M` indica il tasto `Meta`.

Sulle odierne tastiere questo tasto è scomparso, si può ottenere lo stesso risultato con il tasto **Alt** oppure premendo e rilasciando il tasto **Esc**. Un trattino fra due tasti all'interno di una combinazione di tasti indica che i tasti vanno premuti contemporaneamente. Così, quando si dirà che per eseguire un comando bisogna usare la combinazione **M-x** si dovranno premere contemporaneamente il tasto **Alt** e la lettera **x** oppure premere il tasto **Esc**, rilasciarlo e premere la lettera **x**.


Il manuale di Emacs può essere consultato all'interno dello stesso editor di testo con le combinazioni di tasti **C-h r** (premere contemporaneamente il tasto **Control** e la lettera **h**, rilasciare questi tasti e premere la lettera **r**) oppure **C-h i d m Emacs RET** (premere contemporaneamente il tasto **Control** e la lettera **h**, rilasciare questi tasti e premere la lettera **i**, poi la lettera **d**, poi la lettera **m**, poi scrivere Emacs quindi premere il tasto **Invio**). Se si preferisce, il manuale può anche essere consultato dal menu

```
Help > Read the Emacs Manual
```

## 2 Installare Emacs

### Emacs per Windows

Per sistemi operativi Microsoft Windows non esiste una vera e propria procedura di installazione. Emacs si può utilizzare semplicemente procurandosi l'archivio che contiene gli eseguibili.

Dal sito ufficiale di Emacs per Windows ( **EMACS FOR WINDOWS**) si scarica l'archivio compresso `emacs-23.4-barebin-i386.zip` e lo si decompone in una posizione di comodo; una buona soluzione, per esempio, è decomprimere il suddetto archivio in:

```
C:/Programmi/emacs/
```

Se si vuole creare un collegamento di Emacs nel menù Start, è sufficiente un doppio clic sul file *addpm.exe* contenuto nella cartella:

```
C:/Programmi/emacs/bin
```

### Emacs per GNU/Linux

Emacs fa parte del progetto GNU quindi è presente in tutti i sistemi operativi della famiglia GNU/Linux. Se non è già installato, il metodo più semplice per ottenere Emacs in un sistema GNU/Linux è far riferimento al gestore pacchetti della propria distribuzione. Si possono naturalmente utilizzare i gestori di pacchetti a interfaccia grafica, riportiamo qui inoltre i comandi da terminale che possono essere eseguiti per installare Emacs in alcune delle principali distribuzioni: Debian e Ubuntu:

```
$ sudo apt-get install emacs
```

Fedora:

```
$ sudo yum install emacs
```

OpenSUSE:

```
$ sudo zypper install emacs
```

Il metodo più difficile, per i non avvezzi all'uso del terminale, consiste nel compilare Emacs a partire dal codice sorgente, tuttavia non è intenzione di questa guida spiegare come fare ciò.

Dopo averlo installato, Emacs potrà essere avviato facendo clic sul suo lanciatore oppure eseguendo da terminale il comando

```
$ emacs
```

Se si desidera utilizzare Emacs con interfaccia testuale bisogna aggiungere l'opzione `-nw` oppure `-no-window-system`:

```
$ emacs -nw
```

Possono essere aggiunti come argomenti da linea di comando il percorso del file (o dei file) che si vogliono modificare:

```
$ emacs file1.tex file2.tex
```

## Emacs per Mac OS

Sul mirror ufficiale (🐘 EMACS) non è presente una versione già compilata per Mac OS X, queste versioni sono rese disponibili da vari volontari come ad esempio il gestore del sito (🐘 EMACS FOR MAC OS X). La procedura per l'installazione è quella consueta: scaricata la versione più recente si apre il file dmg (se non è già stato aperto automaticamente al termine del download) e si trascina Emacs nella cartella Applicazioni.

In Mac OS le applicazioni avviate dall'interfaccia grafica non hanno accesso ai valori delle variabili d'ambiente, per rendere disponibile ad Emacs la variabile `PATH` è necessario dare da terminale il comando

```
$ defaults write ~/.MacOSX/environment PATH "$PATH"
```

Le nuove impostazioni sono effettive dopo aver eseguito un logout e un successivo login.

Il comando va ripetuto ogni volta che si installa qualche programma che modifica il valore di `PATH`, il caso che più probabilmente riguarda i lettori di questa guida è la distribuzione  $\text{T}_{\text{E}}\text{X}$ .

Per gli utenti che usano la tastiera italiana è utile impostare il tasto `cmd` come meta e il tasto `alt` per scrivere i caratteri speciali, in particolare le parentesi quadre, le graffe e la tilde, scrivendo queste righe nel file `.emacs`

```
(setq ns-command-modifier 'meta)
(setq ns-alternate-modifier nil)
```

## 3 Installare AUCT $\text{E}$ X

### AUCT $\text{E}$ X per Windows

Dal sito ufficiale di AUCT $\text{E}$ X per Windows (🐘 AUCT $\text{E}$ X FOR WINDOWS), si scarica l'archivio compresso `auctex-11.86-e23.2-msw.zip` e lo si decompone in

```
C:/Programmi/emacs/
```

Sia nell'archivio di AUCT $\text{E}$ X, che quello di Emacs, posizionato in precedenza nel percorso suddetto, vi è una cartella dal nome *site-lisp*: i file contenuti in queste cartelle vanno uniti in una sola cartella.

## AUCT<sub>E</sub>X per GNU/Linux

Anche il pacchetto AUCT<sub>E</sub>X è presente nei sistemi GNU/Linux. Come già detto per Emacs, per installare AUCT<sub>E</sub>X si consiglia di utilizzare il gestore pacchetti della propria distribuzione. Ecco i comandi da usare per installare da terminale il pacchetto nelle principali distribuzioni: Debian e Ubuntu:

```
$ sudo apt-get install auctex
```

Fedora:

```
$ sudo yum install emacs-auctex
```

OpenSUSE:

```
$ sudo zypper install emacs-auctex
```

## AUCT<sub>E</sub>X per Mac OS

Come per Emacs non è disponibile una versione di AUCT<sub>E</sub>X già compilata per Mac OS, in questo caso non esistono neppure versioni non ufficiali ed è quindi necessario utilizzare l'utility make. make viene installato insieme all'ambiente di sviluppo Xcode, chi non intende dedicare svariati GB di disco a un software utile soltanto a chi sviluppa software può sfruttare il fatto che make viene installato anche dal correttore ortografico cocoaspell la cui installazione è descritta nel paragrafo 5.

Dalla pagina relativa a Mac OS (🍎 AUCT<sub>E</sub>X FOR MAC OS X) bisogna scaricare l'archivio auctex-11.86.tar.gz.

Dopo aver estratto il contenuto dell'archivio in una cartella temporanea occorre aprire una sessione di terminale e portarsi nella cartella appena creata.

Configurare AUCT<sub>E</sub>X con il comando

```
$ ./configure --prefix=/Applications/Emacs.app/Contents/Resources/\
--with-emacs=/Applications/Emacs.app/Contents/MacOS/Emacs\
--with-lispdir=/Applications/Emacs.app/Contents/Resources/site-lisp/\
--without-texmf-dir
```

sostituendo se necessario a Applications<sup>1</sup> la cartella in cui si trova Emacs, in questo modo tutti i file saranno contenuti nel bundle Emacs.app.

Compilare e installare con il comando

```
$ make && make install
```

Dopo l'installazione bisogna impostare i visualizzatori per i vari file creati con L<sup>A</sup>T<sub>E</sub>X scrivendo nel file .emacs queste righe

```
(setq TeX-view-program-list
  '(("dvips and Skim" "%(o?)dvips %d -o &&\
/Applications/Skim.app/Contents/SharedSupport/displayline %n %f %b")
    ("Skim" "/Applications/Skim.app/Contents/SharedSupport/\
displayline %n %o %b")
    ("open" "open %o")))
(setq TeX-view-program-selection
  '(((output-dvi style-pstricks) "dvips and Skim")
    (output-dvi "Skim")
    (output-pdf "Skim")
    (output-html "open")))
```

<sup>1</sup>Utilizzando il terminale non bisogna utilizzare i nomi localizzati delle cartelle ma quelli reali, quindi Application è corretto anche se si utilizza la versione italiana di Mac OS

Come visualizzatore principale si utilizza Skim perché supporta la ricerca diretta e inversa dal pdf, i dettagli sulla configurazione di Emacs e Skim sono riportati nel paragrafo 6.

## 4 Personalizzare Emacs & AUCTeX

Una delle caratteristiche principali di Emacs è che è un editor di testo personalizzabile in ogni suo aspetto ed espandibile, in modo da renderlo adatto alle proprie esigenze. In questo paragrafo vedremo alcune delle personalizzazioni che possono risultare utili quando si lavora con documenti L<sup>A</sup>T<sub>E</sub>X. Il linguaggio utilizzato per espandere Emacs è l'Elisp, un dialetto del Lisp sviluppato appositamente per Emacs.

Per modificare le impostazioni relative alla gestione nativa di T<sub>E</sub>X da parte di Emacs è possibile usare la funzione `M-x customize-group RET tex RET`. In questo modo si aprirà un buffer nel quale sarà possibile modificare, tramite l'interfaccia, le opzioni desiderate. Per modificare in particolare AUCTeX è possibile usare la funzione `M-x customize-group RET AUCTeX RET`.

Di seguito saranno suggerite alcune porzioni di codice Elisp che permettono di personalizzare Emacs e che vanno inserite nel file di inizializzazione `~/.emacs`. Per rendere effettive le modifiche bisogna riavviare Emacs.

Se si usa AUCTeX, è possibile inserire macro nei propri documenti L<sup>A</sup>T<sub>E</sub>X con `C-c RET`. La comodità di questa funzione è che AUCTeX conosce le macro dei principali pacchetti L<sup>A</sup>T<sub>E</sub>X e permette quindi l'autocompletamento con il tasto TAB. Inoltre conosce gli argomenti e le opzioni che queste macro accettano, infatti usando `C-c RET frac RET` nel documento si otterrà `\frac{ }{ }` e il cursore si posizionerà all'interno del primo paio di parentesi graffe. Allo stesso modo, usando `C-c RET sqrt RET` verrà richiesto l'ordine *n* della radice da inserire (premere direttamente RET per non inserire nulla) e si otterrà `\sqrt[n]{ }`. Per fare in modo che AUCTeX conosca quali pacchetti L<sup>A</sup>T<sub>E</sub>X sono stati caricati bisogna permettergli di effettuare il parsing dei propri documenti e questo è possibile aggiungendo nel proprio file `.emacs` il seguente codice

```
(setq TeX-parse-self t) ; Attiva parsing al caricamento.  
(setq TeX-auto-save t) ; Attiva parsing al salvataggio.
```

In questo modo verrà creata una cartella chiamata `auto` nella stessa directory in cui si trova il documento nella quale verranno registrate le informazioni relative al proprio documento L<sup>A</sup>T<sub>E</sub>X, se non si desidera affollamento nelle proprie cartelle si potrebbe non voler attivare queste due opzioni.

Come detto, AUCTeX conosce le macro dei pacchetti principali, ma naturalmente non conosce tutte i pacchetti. Possiamo, però, fargli conoscere delle macro aggiuntive da utilizzare solo quando viene caricato nel documento L<sup>A</sup>T<sub>E</sub>X il pacchetto che fornisce le relative macro. Per esempio, AUCTeX non conosce le macro messe a disposizione dal pacchetto `siunitx`. Con il seguente codice da aggiungere nel file `.emacs` apprenderà che questo pacchetto fornisce i comandi `SI`, `si` e `num` di cui il primo accetta due argomenti, gli ultimi due solo uno:

```
(eval-after-load "tex"  
  '(TeX-add-style-hook  
    "siunitx"  
    (lambda ()  
      (TeX-add-symbols  
        ("SI" 2)  
        ("si" 1)  
        ("num" 1))))))
```

A volte in un documento L<sup>A</sup>T<sub>E</sub>X non si carica un certo pacchetto *A* poiché è automaticamente caricato da un altro pacchetto *B* che viene esplicitamente caricato. In questi casi, però, AUCTeX potrebbe non rendersi conto che il pacchetto *A* è stato effettivamente caricato e non fornirebbe le funzioni di

autocompletamento per le macro del pacchetto A, se è uno di quelli supportati. Per ovviare a questo problema è possibile inserire il seguente codice nel file di inizializzazione

```
(eval-after-load "tex"
  '(TeX-add-style-hook
    "B"
    (lambda ()
      (TeX-run-style-hooks "B")))))
```

Per esempio, il pacchetto `kpfonts` carica automaticamente il pacchetto `amsmath` e non è quindi necessario inserire nel proprio sorgente  $\text{\LaTeX}$  l'istruzione `\usepackage{amsmath}`. Per far sapere ad  $\text{\LaTeX}$  questa situazione si userà il codice

```
(eval-after-load "tex"
  '(TeX-add-style-hook
    "kpfonts"
    (lambda ()
      (TeX-run-style-hooks "amsmath")))))
```

Se si è soliti suddividere i propri documenti  $\text{\LaTeX}$  in più file da includere nel file principale con i comandi `\input` o `\include` può essere utile aggiungere al file `.emacs`

```
(setq-default TeX-master nil)
```

In questo modo, all'apertura dei documenti  $\text{\LaTeX}$  Emacs chiederà qual è il file principale associato e sul quale saranno eseguiti i comandi di compilazione.

Se si usa spesso  $\text{\LaTeX}$  per comporre documenti matematici potrebbe essere utile attivare la modalità `LaTeX-math-mode` con `C-c ~` oppure con il comando `M-x LaTeX-math-mode`. In questo modo, per esempio, per inserire il simbolo `\alpha` si potrà usare la scorciatoia `' a`. L'elenco delle altre scorciatoie per inserire i simboli può essere consultato dal menu `Math` nella barra dei menu. Per attivare automaticamente questa modalità ogni volta che si apre un documento  $\text{\LaTeX}$  bisogna aggiungere

```
(add-hook 'TeX-mode-hook 'LaTeX-math-mode)
```

al proprio `.emacs`.

Il codice

```
(setq TeX-electric-sub-and-superscript t)
```

aggiunto al proprio `.emacs`, permette di inserire automaticamente una coppia di parentesi graffe `{}` quando si scrivono i simboli `_` e `^` in modalità matematica.

Normalmente, quando si preme `RET` in un sorgente  $\text{\LaTeX}$  viene semplicemente aggiunta una nuova linea. Se si vuole che andando a capo la nuova linea sia automaticamente indentata bisogna premere `C-j` invece di `RET`. Si può però fare in modo che si ottenga questo risultato anche premendo normalmente `RET` aggiungendo il seguente codice al file di inizializzazione

```
(setq TeX-newline-function 'newline-and-indent)
```

Se si mette `reindent-then-newline-and-indent` al posto di `newline-and-indent` si avrà, in più, l'effetto che la riga attuale viene indentata.

## 5 Correttore ortografico

Per abilitare Emacs e AUCTEX a lavorare con la lingua italiana bisogna installare il correttore ortografico. Le procedure, poiché differenti, vengono di seguito esposte per tutti e tre i sistemi operativi. L'uso, invece, è invariante.

Per fare una revisione del documento che si sta scrivendo, bisogna premere la combinazione di tasti M-x e scrivere `ispell`: in questo modo il dizionario proporrà, in sequenza, la correzione dei termini che trova errati.

Un secondo modo, per scrivere correttamente, è attivare la modalità *flyspell* (correzione al volo). Se la si vuole attivare solo sul documento corrente basta andare in `Tools > Spell Checking > Automatic spell checking (Flyspell)`; se invece la si vuole tenere sempre abilitata si deve aggiungere la riga:

```
(add-hook 'LaTeX-mode-hook 'flyspell-mode)
```

al proprio file `.emacs`.

Quando viene rilevata una parola potenzialmente non corretta, questa verrà colorata (in genere) in rosso. Alcuni termini, però, vengono segnalati come errati anche quando sono corretti, cosa che accade perché quel termine non è presente nel dizionario: si può aggiungerlo, premendo il tasto centrale del mouse quando questo è posizionato sulla parola e scegliere `Save word`.

### Windows

Per usare il correttore ortografico in Emacs, su un sistema operativo Windows, si può installare il programma GNU Aspell (GNU ASPELL - WIN32 VERSION). L'ultima versione stabile per Windows, seppur datata Dicembre 2002, è la 0.50-3.

I file che bisogna procurarsi sono quelli che vengono di seguito riportati; si badi bene ad installarli nell'ordine in cui vengono dati:

1. il programma completo `Aspell-0-50-3-3-Setup.exe` (che rappresenta il programma vero e proprio);
2. il dizionario precompilato per l'italiano `Aspell-it-0.50-2-3.exe`.

Resta qualche altra operazione da fare. Come prima cosa si deve copiare il percorso della cartella dove sono contenuti gli eseguibili di Aspell nella variabile d'ambiente `PATH` di Windows. Secondo, aggiungere al proprio `.emacs` le seguenti righe di codice:

```
(setq-default ispell-program-name "aspell")  
(setq-default ispell-extra-args '("--reverse"))  
(setq ispell-dictionary "italiano")
```

dove, la prima e la terza sono quelle strettamente necessarie, mentre la seconda riga serve a risolvere dei problemi con le versioni precedenti di Emacs.

### GNU/Linux

Il correttore ortografico GNU Aspell può essere installato facilmente su GNU/Linux utilizzando, come al solito, il gestore pacchetti della propria distribuzione. Il dizionario italiano di GNU Aspell si chiama `aspell-it` quindi da terminale può essere installato in Debian con il comando

```
$ sudo apt-get install aspell-it
```

in Fedora con

```
$ sudo yum install aspell-it
```

mentre in openSUSE si può eseguire il comando:

```
$ sudo zypper install aspell-it
```

## Mac OS

Anche per Mac OS X è disponibile una versione di GNU Aspell (🍌 COCOASPELL)

Dopo aver installato cocoAspell.2.1 si possono installare i dizionari delle lingue che interessano scaricandoli direttamente dal sito di (GNU Aspell), ad esempio il dizionario italiano è contenuto nell'archivio `aspell6-it-2.2_20050523-0.tar.bz2`.

Dopo aver scaricato i dizionari bisogna estrarre il contenuto dell'archivio in una cartella temporanea, aprire una sessione di terminale e portarsi nella cartella appena creata. I dizionari si installano dando i seguenti comandi

```
$ ./configure
$ make
$ sudo make install
```

Come ultima cosa è necessario configurare Emacs aggiungendo al `.emacs` le seguenti righe:

```
(setq-default ispell-program-name "aspell")
(setq ispell-dictionary "italiano")
```

## 6 Ricerca diretta-inversa

Una delle principali funzioni che spesso l'utente  $\text{\LaTeX}$  cerca nel proprio editor, è quella della ricerca diretta-inversa, molto utile in fase di revisione finale del proprio documento, e specie quando quest'ultimo abbastanza corposo.

Alcuni editor di testo hanno un visualizzatore integrato preconfigurato per effettuare la ricerca dal sorgente `tex` al file `pdf` e viceversa, come per esempio  $\text{\TeX}$ works; altri, come Emacs, richiedono alcuni passaggi per arrivare allo scopo.

Le procedure per abilitare tale funzione in Emacs variano per i tre sistemi operativi, ma non solo, variano anche i visualizzatori che permettono tale funzione infatti in ambiente Windows, l'unico visualizzatore capace di fare questo è SumatraPDF, **in ambiente Linux si può usare [...]**, mentre in ambiente Mac OS si usa Skim.

### Ricerca diretta-inversa su Windows

Prima di entrare nel vivo dell'argomento, bisogna ovviare ad un bug che da tempo segue il programma Emacs su Windows, in particolare i sistemi Windows Vista e Seven (su Xp non dovrebbero esserci problemi), ovvero il problema che si ha quando si abilita il server in emacs.

La soluzione a tale problema è riportata in [4] alla voce *Waiting for Emacs server to start is ownership problem on ./emacs.d/server*.

Per i sistemi operativi Windows il solo visualizzatore capace di interagire con il file sorgente `.tex` creato con Emacs è SumatraPDF (🍌 SUMATRAPDF).

Per la verità, il visualizzatore sviluppato da Krzysztof Kowalczyk non supporta questa funzionalità, o meglio, non è ancora accessibile all'utente, pertanto, dopo averlo installato si può procedere nel seguente modo:



1. si scarica l'eseguibile Sumatra-TeX (🐞 SUMATRAPDF FOR T<sub>E</sub>X USERS) di William Blum che ha sviluppato appunto la funzionalità di ricerca diretta-inversa e fornito un modo per accedervi;
2. si rinomina l'eseguibile SumatraPDF-TeX in SumatraPDF;
3. si copia tale file nella cartella dove è contenuto quello installato precedentemente.

A queste operazioni segue la configurazione di SumatraPDF e di Emacs, mediante il proprio file *.emacs*.

Per la ricerca inversa si deve aprire SumatraPDF, andare in Impostazioni e quindi scegliere Opzioni... e digitare la riga

```
C:/Programmi/emacs/bin/emacsclientw.exe +%l "%f"
```

in *Set inverse search command-line*.

Per la ricerca diretta si deve scaricare lo script sumatra-forward (🐞 SUMATRA-FORWARD.EL) e copiarlo in una cartella accessibile a Emacs per esempio

```
C:/Programmi/emacs/bin/site-lisp
```

per poi richiamarlo all'interno del proprio file *.init* con la riga

```
(require 'sumatra-forward)
```

ed infine accertarsi di avere nella cartella degli eseguibili di Emacs, l'applicazione ddeclient, altrimenti scaricabile da (🐞 DDECLIENT).

Fatto questo resta da far capire a Emacs che si vuole la ricerca diretta inversa, e per farlo si possono scrivere nel proprio *.init* file, le seguenti righe:

```
(setq TeX-source-correlate-method (quote syntex))
(setq TeX-source-correlate-mode t)
(setq TeX-source-correlate-start-server t)
```

Per la ricerca inversa basta premere il tasto F8 e nel visualizzatore verrà evidenziata la riga corrispondente; per la ricerca diretta invece, basta fare doppio click nel visualizzatore e nel file *.tex* il cursore si sposterà all'inizio della riga che contiene quella parola.

Si vuole infine dare un suggerimento al lettore, qualora SumatraPDF non sia il proprio visualizzatore preferito. Infatti è possibile impostare SumatraPDF come visualizzatore predefinito di Emacs e non del sistema operativo (lasciando magari Adobe Reader), aggiungendo le seguenti righe a quelle sopra descritte:

```
(setq TeX-view-program-list (quote (("SumatraPDF" "SumatraPDF %o"))))
(setq TeX-view-program-selection (quote ( (output-pdf "SumatraPDF")
(output-dvi "Yap") (output-html "start"))))
```

## Ricerca diretta-inversa su GNU/Linux

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

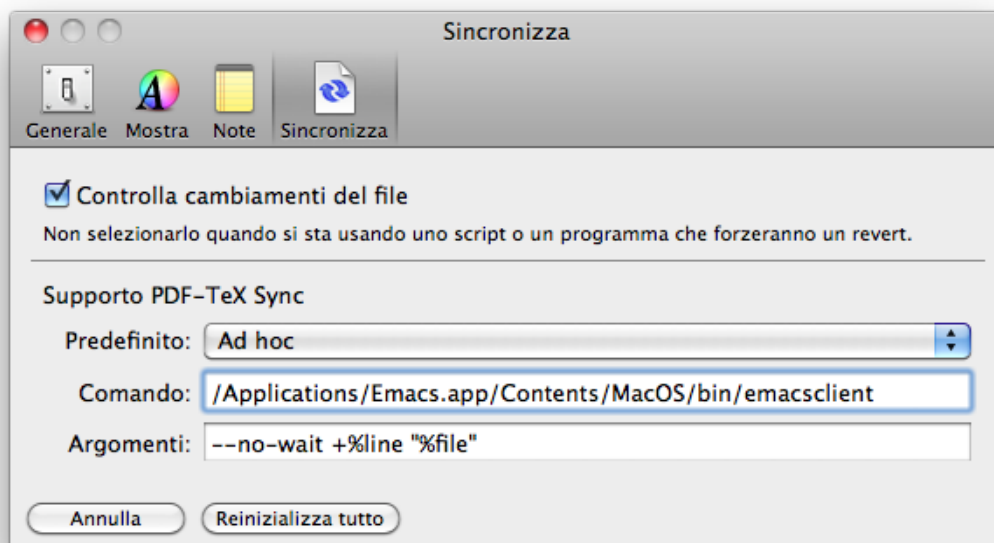


Figura 1: Preferenze di Skim per la ricerca diretta e inversa

## Ricerca diretta-inversa su Mac OS

L'unico visualizzatore di file PDF per Mac OS X che supporta la ricerca diretta e inversa associato a Emacs è Skim, già citato nel paragrafo 3, relativo all'installazione di AUCTEX.

Dopo averlo installato occorre configurarlo impostando quanto segue nella pagina *Sincronizza* delle preferenze:

- selezionare la checkbox *Controlla cambiamenti del file*
- scrivere `/Applications/Emacs.app/Contents/MacOS/bin/emacsclient` nel campo *Comando*
- scrivere `--no-wait +%line "%file"` nel campo *Argomenti*

Nella figura 1 sono riportate le impostazioni corrette.

La prima opzione fa sì che quando un file aperto in Skim viene modificato da un altro processo Skim chieda se deve essere ricaricato, scegliendo Automatico successivi aggiornamenti vengono ricaricati senza chiedere conferma.

La configurazione di Emacs è analoga a quella degli altri sistemi operativi, occorre scrivere queste righe in `.emacs`

```
(setq TeX-source-correlate-method 'synctex)
(setq TeX-source-correlate-mode t)
(setq TeX-source-correlate-start-server t)
```

Per passare dal file `.tex` al `.pdf` si utilizza la combinazione di tasti `C-c C-v`, per ritornare al file `.tex` bisogna fare click sulla parola che interessa tendendo premuto `shift` e `cmd`.

Con alcune versioni di Emacs precedenti alla 23.3 resta in primo piano la finestra del visualizzatore, in tal caso è necessario aggiungere queste righe al file `.emacs`

```
(defun ns-raise-emacs ()
  (ns-do-applescript "tell application \"Emacs\" to activate"))
(add-hook 'server-switch-hook 'ns-raise-emacs)
```

## 7 Skeleton mode

### Introduzione

La *skeleton mode* è uno speciale (ma potente) minilinguaggio, scritto in Emacs (un dialetto di Lisp) ed incluso di default in Emacs. Maggiori informazioni si possono ottenere digitando `C-h i d m` Autotype in Emacs.

La *skeleton mode* permette in maniera molto semplice di crearsi dei blocchi di codice (che magari si usano spesso) e di associare a questi una combinazione di tasti per richiamarli nel proprio documento.

Ad un livello successivo, si può anche usare questo linguaggio per creare dei veri e propri *templates*. Lo vedremo più avanti.

Per semplicità, per mostrarne l'uso, vengono di seguito riportati degli esempi presi da una discussione sul forum del GuIT

I codici si commentano quasi da soli:

```
(define-skeleton TeX-dollar-sign-skeleton "Make a pair of dollar
signs and leave point inside" nil "$_" "$" )

(global-set-key (kbd "C-x e") 'TeX-dollar-sign-skeleton)

(define-skeleton TeX-square-skeleton "Make a pair of square brackets
and leave point inside" nil "\\[" "\\]" )

(global-set-key (kbd "C-x s") 'TeX-square-skeleton)
```

Nella prima riga si dichiara lo *scheletro* con un nome, che verrà poi utilizzato per poterlo richiamare nel codice. Segue quindi una breve descrizione (facoltativa ma consigliata), e poi il codice vero e proprio. Infine con `global-set-key` viene creata una combinazione di tasti personali per richiamare lo *scheletro*.

Gli esempi appena mostrati sono veramente semplici e forse se ne potrebbe fare a meno, ma servono per far capire con quanta semplicità si possono *automatizzare* certe operazioni.

### Templates

Talvolta può capitare di dover scrivere documenti che hanno una forma molto simile tra loro, o che, all'opposto, utilizziamo così raramente che averne un modello *pronto uso* risulterebbe davvero comodo.

Ecco che allora ci viene in mente di creare un file, un *modello di documento* già configurato, da richiamare nel nostro editor preferito.

Per Emacs esistono diversi pacchetti che consentono di creare *template*, ognuno dei quali richiede una installazione. Per fare questo possiamo invece avvalerci anche della *skeleton mode* inclusa già in Emacs.

Supponiamo di voler creare un *modello di lettera* da usare velocemente all'occorrenza. Lo *scheletro* di codice che verrà incluso nel documento va appunto nel file `.emacs`. Un prototipo semplice potrebbe essere il seguente:

```
(define-skeleton LaTeX-skeleton-letter "Inserts a Latex letter
skeleton into current buffer. This only makes sense for empty
buffers." "Destinatario: "
"\documentclass[a4paper,10pt]{letter}\n"
"\usepackage[T1]{fontenc}\n"
"\usepackage[utf8]{inputenc}\n"
"\usepackage[italian]{babel}\n")
```

```
"\\name{Pinco Pallino}\\n"
"\\address{Via Marco Polo 32\\ \\ \\ Napoli 80100\\ \\ \\ Italy}\\n"
"\\begin{document}\\n"
"\\begin{letter}{\" str | \" *** Destinatario *** \" \"}\\n"
"\\opening{\" _ \"}\\n\\n"
"\\closing{Cordialmente}\\n"
"\\end{letter}\\n"
"\\end{document}\\n")
```

Come già detto dopo aver dichiarato il nome dello *scheletro* seguono una o due (o a piacere) righe di commento. La riga:

```
"Destinatario: "
```

è una riga di prompt che userà il *minibuffer* per interagire con l'utente quando verrà letto (o trovato) il simbolo `str` nel codice; il valore inserito dall'utente verrà proprio sostituito a `str`.

Il simbolo `'_'` indica la posizione dove verrà posto il cursore quando il codice *skeleton* è completamente inserito.

Il codice che si vuole inserire va sempre tra due doppi apici, i comandi vanno preceduti da un *backslash* e per ritornare a capo si usa `\n`.

Infine, come precedentemente detto, si può richiamare il *codice pronto uso* con una combinazione di tasti personale, ad esempio se si vuole usare la combinazione C-c l, bisogna scrivere nel file *.emacs*:

```
(global-set-key "\C-cl" 'LaTeX-skeleton-letter)
```

Ulteriori approfondimenti (oltre a quelli a cui si perviene con C-h i d m Autotype) si possono reperire on-line alla pagina di riferimento ( Skeleton Mode).

## 8 Reference Card

Oltre alla vasta documentazione che si può reperire in rete ed ai manuali ufficiali, di particolare utilità sono le *Reference Card*, disponibili sia per Emacs, sia per AUCT<sub>E</sub>X.

Si tratta di un elenco delle principali funzionalità e dei comandi a queste associate per poter, all'occorrenza, ricordare e trovare velocemente quello che serve.

Il tutto è riportato in maniera ordinata e concisa in una sorta di forma tabulare, eventualmente da stampare, e tenere a portata di mano sulla propria scrivania.

Le reference card sono disponibili sia in rete, sia sul proprio computer una volta installati Emacs ed AUCT<sub>E</sub>X. Sul proprio computer dovrebbero esserci anche i sorgenti pronti per essere compilati.

## 9 Conclusioni

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

<h2>GNU Emacs Reference Card</h2> <p>(for version 23)</p> <h3>Starting Emacs</h3> <p>To enter GNU Emacs 23, just type its name: <b>emacs</b></p> <h3>Leaving Emacs</h3> <p>suspend Emacs (or iconify it under X)    <b>C-z</b>  exit Emacs permanently    <b>C-x C-c</b></p> <h3>Files</h3> <p><b>read</b> a file into Emacs    <b>C-x C-f</b>  <b>save</b> a file back to disk    <b>C-x C-s</b>  save <b>all</b> files    <b>C-x s</b>  <b>insert</b> contents of another file into this  buffer    <b>C-x i</b>  replace this file with the file you really  want    <b>C-x C-v</b>  write buffer to a specified file    <b>C-x C-w</b>  toggle read-only status of buffer    <b>C-x C-q</b></p> <h3>Getting Help</h3> <p>The help system is simple. Type <b>C-h</b> (or <b>F1</b>) and follow the di-  rections. If you are a first-time user, type <b>C-h t</b> for a <b>tutorial</b>.</p>	<h2>AUCTeX Reference Card</h2> <p>(for version 11.86)</p> <h3>Conventions Used</h3> <p>Carriage Return or <b>C-m</b>    <b>RET</b>  Tabular or <b>C-i</b>    <b>TAB</b>  Linefeed or <b>C-j</b>    <b>LFD</b></p> <h3>Shell Interaction</h3> <p>Run a command on the master file    <b>C-c C-c</b>  Run a command on the buffer    <b>C-c C-b</b>  Run a command on the region    <b>C-c C-r</b>  Fix the region    <b>C-c C-t C-r</b>  Kill job    <b>C-c C-k</b>  Recenter output buffer    <b>C-c C-l</b>  Next error in <b>T<sub>E</sub>X/L<sup>A</sup>T<sub>E</sub>X</b> session    <b>C-c ‘</b>  Toggle debug of bad boxes    <b>C-c C-t C-b</b>  Toggle debug of warnings    <b>C-c C-t C-w</b>  View output file    <b>C-c C-v</b></p> <p>Commands you can run on the master file (with <b>C-c C-c</b>) or the  region (with <b>C-c C-r</b>) include the following (starred versions  are not available in all modes):</p> <p><b>T<sub>E</sub>X</b>    <b>*T<sub>E</sub>X</b>  <b>L<sup>A</sup>T<sub>E</sub>X</b>    <b>*L<sup>A</sup>T<sub>E</sub>X</b>  <b>ConT<sub>E</sub>Xt</b> (once)    <b>*ConT<sub>E</sub>Xt</b>  <b>ConT<sub>E</sub>Xt Full</b>    <b>*ConT<sub>E</sub>Xt Full</b></p>
---	---

## Riferimenti bibliografici

- [1] Onofrio de Bari. «GNU Emacs e AUCT<sub>E</sub>X per L<sub>A</sub>T<sub>E</sub>X». In: *ArsT<sub>E</sub>Xnica* 2 (ott. 2006), pp. 60–64. ISSN: 1828-2350. URL: <http://www.guit.sssup.it/arstexnica/>.
- [2] William Blum. *Configuring editors with SumatraPDF*. 2011. URL: <http://william.famille-blum.org/blog/index.php>.
- [3] William Blum. *SumatraPDF for TeX users*. 2011. URL: <http://william.famille-blum.org/software/sumatra/index.html>.
- [4] *EmacsW32*. 2011. URL: <http://www.emacswiki.org/emacs/EmacsW32>.
- [5] Richard Stallman. *GNU Emacs Manual*. 16<sup>a</sup> ed. Free Software Foundation, Inc. 2010. URL: [www.gnu.org/software/emacs/manual/emacs.pdf](http://www.gnu.org/software/emacs/manual/emacs.pdf).
- [6] Kresten Krab Thorup, Per Abrahamsen, David Kastrup et al. *AUCT<sub>E</sub>X, A sophisticated T<sub>E</sub>X environment for Emacs*. Free Software Foundation, Inc. 2010. URL: [www.gnu.org/software/auctex/manual/auctex.pdf](http://www.gnu.org/software/auctex/manual/auctex.pdf).

## Licenza d'uso & Colophon

Quest'opera è soggetta alla Creative Commons Public License versione 3.0 o posteriore: *Attribuzione, Non Commerciale, Condividi* allo stesso modo. L'enunciato integrale della licenza è reperibile sul sito ufficiale (<http://creativecommons.org/licenses/by-nc-sa/3.0/deed.it>).

La realizzazione di questo lavoro a sei mani si è resa possibile grazie al sistema di controllo di versione Git (<http://git-scm.com>). Per poter collaborare si può clonare il progetto tramite il sito Gitorius con `git clone git://gitorious.org/emacs-tutorial/emacs-tutorial.git`. È gradita la segnalazione di errori o refusi.