Left/right motions	Text object motions	Search pattern special chars
Nh = Left N = Number of chars to move	Nw = Number of word/s forward	magic = "." non-magic = "\." = Matches any
NI = Right N = Number of chars to move	NW = Number of blank-separated word/s	single character
0 = First char in line	forward	^ = Matches start of line
^ = First non blank char in ine	Ne = Forward to the end of the Nth word/s	\< = Matches start of word
g0 = To first character in screen line	NE = Forward to the end of the Nth blank	\> = Matches end of word
g^ = To first non-blank character in screen	separated word/s	magic = "[a-z]" non-magic = " \[a-z]" =
line	Nb = N word/s backward	Matches a single char from the range
g\$ = To last character in screen line	NB = N blank-separated word/s backward	magic = "[a-z]" non-magic = " \[a-z]" = Matches
Nf{char} = To the Nth occurrence of {char} to the right	Nge = backward to the end of the Nth word/s	a single char not in the range
NF{char} = To the Nth occurrence of {char}	NgE = Backward to the end of the Nth	Marks
to the left	blank-separated word/s	$m{a-zA-Z} = Mark current position with mark$
Nt{char} = Till before the Nth occurrence of	N) = N sentences forward	{a-zA-Z}
{char} to the right	N(= N sentences backward	`{a-z} = Go to mark {a-z} within current file
NT{char} = Till before the Nth occurrence of	N} = N paragraphs forward	`{A-Z} = Go to mark {A-Z} in any file
{char} to the left	N{ = N paragraphs backward	`{0-9} = Go to the position where Vim was previously exited
Up/down motions	Dattorn convolves	" = Go to the position before the last jump
Nk = Up N lines	Pattern searches	" = Go to the position when last editing this
Nj = Down N lines	N/{pattern}/{offset} < Enter> = Search forward for the Nth occurrence of {pattern}	file
N- = Up N lines, on the first non-blank	from {offset}	`[= Go to the start of the previously
character	N?{pattern}?{offset} <enter> = Search</enter>	operated or put text
N+ = Down N lines, on the first non-blank	backward for the Nth occurrence of {pattern} from {offset}	`] = Go to the end of the previously operated or put text
character	N/ <enter> = Repeat last search, in the</enter>	`< = Go to the start of the (previous) Visual
G = Goto line N (default: last line), on the first non-blank character	forward direction to Nth occurrence	area
Ngg = Goto line N (default: first line), on the first non-blank character	N? <enter> = Repeat last search, in the backward direction to Nth occurrence</enter>	`> = Go to the end of the (previous) Visual area
Ngk = Up N screen lines	n = Repeat last search	`. = Go to the position of the last change in
Ngj = Down N screen lines	N = Repeat last search in opposite direction	this file
	N* = Search forward for the identifier under	:marks = Print the active marks
	the cursor to Nth occurrence	N <ctrl-o> = Go to Nth older position in</ctrl-o>
	N# = Search backward for the identifier	jump list
	under the cursor to Nth occurrence	N <ctrl-i> = Go to Nth newer position in</ctrl-i>
	Ng = Like ", but also find partial matches	jump list
	Ng# = Like "#", but also find partial matches	:ju = Print the jump list

Inserting text	Copying and moving text (cont)	Repeat
Na = Append text after the cursor (N times)	NP = Put a register before the cursor	N. = Repeat last change (with count
NA = Append text at the end of the line (N	position (N times)	replaced with N)
times)	N]p = Like p, but adjust indent to current line	q{a-z} = Record typed characters into
Ni = Insert text before the cursor (N times)	N[P = Like P, but adjust indent to current	register {a-z}
NI = Insert text before the first non-blank in	line	q{A-Z} = Record typed characters,
the line	Ngp = Like p, but leave cursor after the new	appended to register {a-z}
NgI = Insert text in column 1 (N times)	text	q = Stop recording
No = Open a new line below the current line, append text (N times)	NgP = Like P, but leave cursor after the new text	$N@{a-z} = Execute the contents of register {a-z} (N times)$
NO = Open a new line above the current line, append text (N times)	Changing text	Ngs = Goto Sleep for N seconds
	Nr{char} = Replace N characters with {char}	Undo/Redo
Deleting text	Ngr{char} = Replace N characters without	Nu = Undo last N changes
Nx = Delete N characters under and after	affecting layout	N <ctrl-r> = Redo last N undone changes</ctrl-r>
the cursor	NR = Enter Replace mode (repeat the	U = Restore last changed line
NX = Delete N characters before the cursor	entered text N times)	
Nd{motion} = Delete the text that is moved	NgR = Enter virtual Replace mode: Like	
over with {motion}	Replace mode but without affecting layout	
Ndd = Delete N lines	c{motion} = Change the text that is moved	
ND = Delete to the end of the line (and N-1	over with {motion} Ncc = Change N lines	
more lines)	NS = Change N lines	
NJ = Join N-1 lines (delete <eol>s)</eol>	NC = Change to the end of the line (and N-1	
NgJ = Like "J", but without inserting spaces	more lines)	
Copying and moving text	Ns = Change N characters	
"{char} = Use register {char} for the next	N~ = Switch case for N characters and	
delete, yank, or put	advance cursor	
Ny{motion} = yank the text moved over with {motion} into a register	g~{motion} = Switch case for the text that is moved over with	
Nyy = Yank N lines into a register	gu{motion} = Make the text that is moved over with {motion} lowercase	
NY = Yank N lines into a register		
Np = Put a register after the cursor position (N times)	gU{motion} = Make the text that is moved over with {motion} uppercase	
	N <ctrl-a> = Add N to the number at or after the cursor</ctrl-a>	
	N <ctrl-x> = Subtract N from the number at or after the cursor</ctrl-x>	