EMBL Bio-IT
The Portal for Computational Biology

Input-Output (IO) Redirections

Three IO "channels" are available by default:

- **Standard input (STDIN, Number: 0)**: The input for your program, normally your keyboard but can be an other program (when using pipes or IO redirection)
- **Standard output (STDOUT, Number: 1)**: Where your program writes its regular output to. Normally your terminal
- **Standard error (STDERR, Number: 2)**: Where your programs normally write their error message to. Normally your terminal

Input, output and error messages can be redirected from their default "targets" go others. If using the file descriptor numbers (0, 1, 2) in redirections, then there must be no whitespace between the numbers and the redirection operators.

Redirect to `/dev/null` to discard the output.

`cmd > afile`	Write the output of `cmd` into `afile`. This will overwrite `afile`.
`cmd >> afile`	Write the output of `cmd` into `afile`. This will add to `afile`
`cmd > /dev/null`	Discard the output of `cmd`
`cmd > afile 2>&1`	Write the output of `cmd` into `afile` (overwriting the file!) and write STDERR to the same place
`cmd >> afile 2>&1`	Add the output and error messages of `cmd` into `afile`
`cmd > afile 2> afile`	Same as above
`cmd >> afile 2>/dev/null`	Add the output of `cmd` to `afile` and discard error messages

`cmd > /dev/null 2>&1`
`cmd > /dev/null 2>/dev/null`
`cmd >& /dev/null`
	Three time the same: Discard output and error messages completely

`cmd1 < cmd2`	Use output of `cmd2` as standard input for `cmd1`

See also `http://www.catonmat.net/blog/bash-one-liners-explained-part-three/`, `http://www.catonmat.net/blog/bash-redirections-cheat-sheet/` and `http://wiki.bash-hackers.org/howto/redirection_tutorial`.

Frank Thommen, Structural and Computational Biology Unit, December 2012