

# Introduction to Shell Scripting Sed

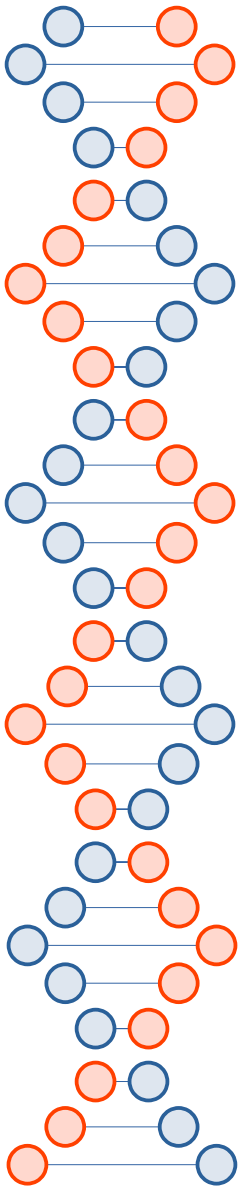
Skylar Chan - BSCI238G  
UMD  
February 24th, 2023

# Students will be able to

- Use tldr to see command examples
- Recall important grep flags
- Use sed to replace text



Swoobat

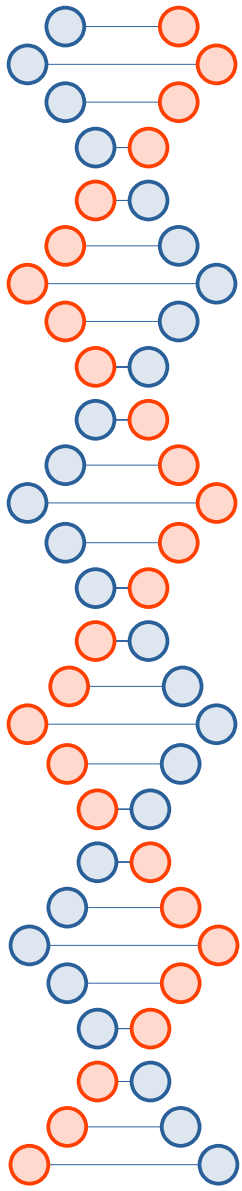


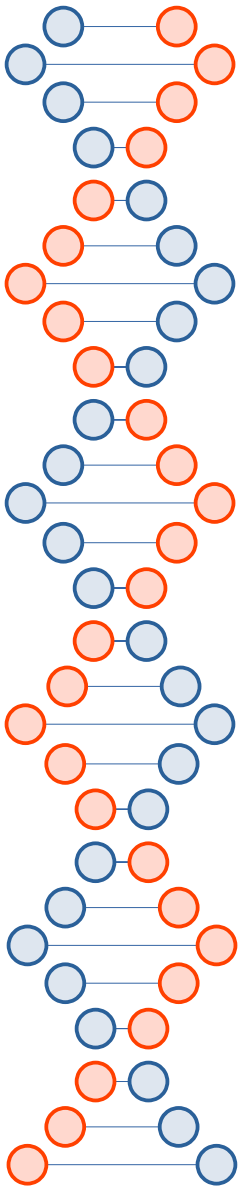
tldr

- Command examples online!
- Can be a useful resource
- Understanding the examples is the most important takeaway

# Important grep flags

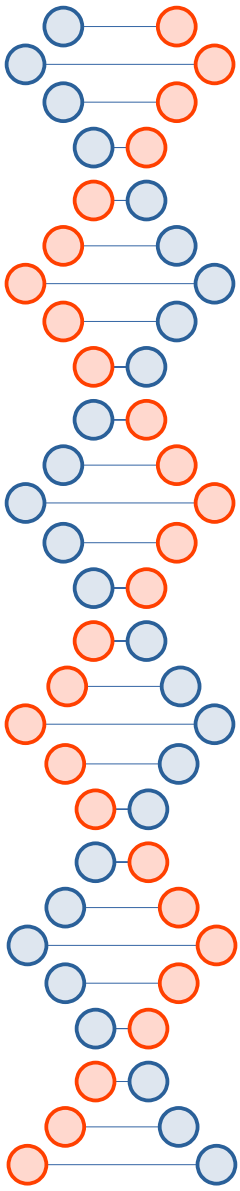
- -R is for recursive
  - Search a directory and its contents for matches
- -E is extended regular expression
  - No need to escape meta-characters ? + { | ( )
- A bunch of flags that can format the output differently





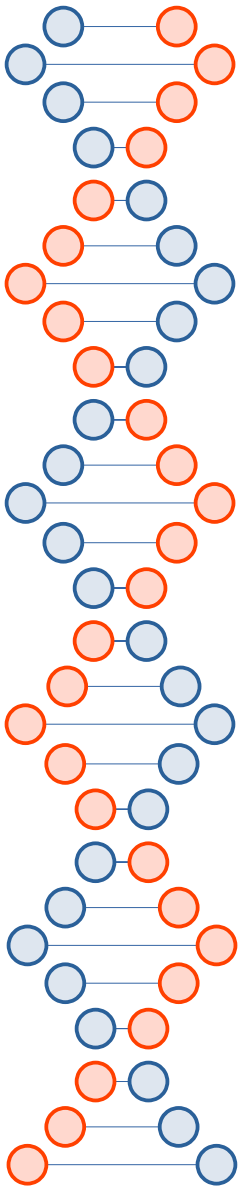
# Sed

- Stream editor
- Primarily used for text substitution
- Turing-complete programming language
  - We'll keep it simple



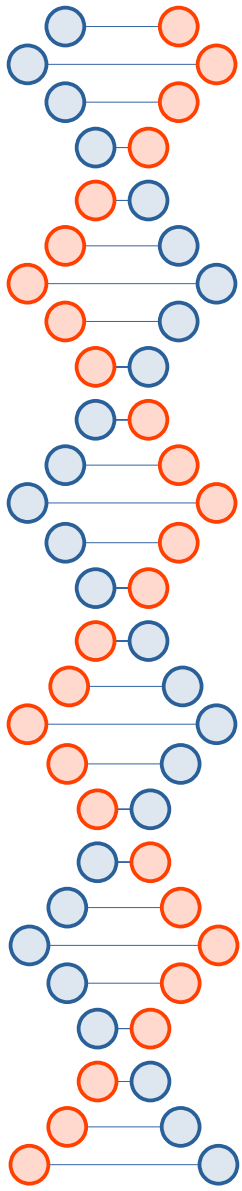
# Sed syntax

- `do_one_thing ; do_another_thing`
- Things we can do:
  - `s/find/replace/`
  - `s/find/replace/g`
- Can use backreferences
  - Group patterns with `( )` and reference them in substitutions
  - The pattern with the first `(` is `\1`, the second `(` is `\2`, and so on
  - `(a(b))` : `\1` is `ab`, `\2` is `b`
- Use `-E` for regex



# Sed one-liners

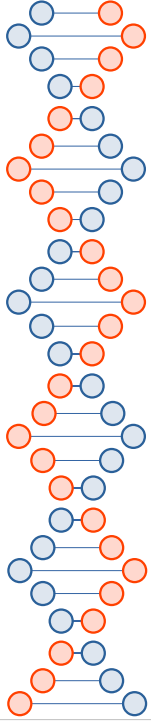
- **sed 's/a/b/'** - replace the first a of each line with a b
- **sed 's/a/b/g'** - replace all as with bs
- **echo -e 'ebbbb\nbbebb' | sed -e '/e/s/b/a/'**
- **echo -e 'ebbbb\nbbebb' | sed -e '/^e/s/b/a/'**
- Can use ; instead of -e, but -e is more readable
- Instead of / you can use another character like # or |



The End

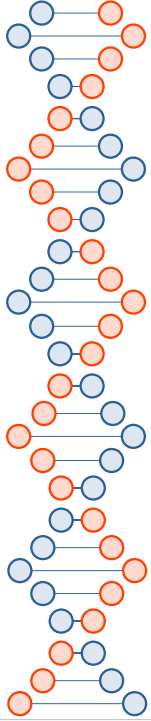
Today's Terminal Toy:  
fortune





# Introduction to Shell Scripting Sed

Skylar Chan - BSCI238G  
UMD  
February 24th, 2023

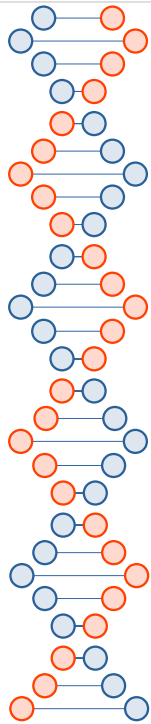


## Students will be able to

- Use tldr to see command examples
- Recall important grep flags
- Use sed to replace text

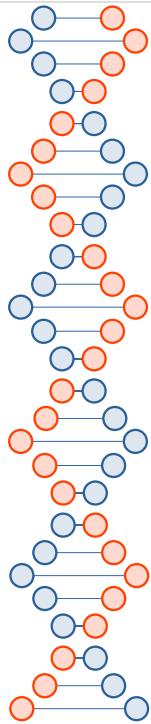


Swoobat



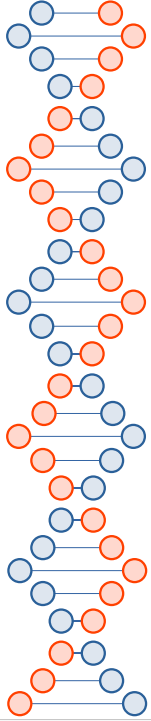
## tldr

- Command examples online!
- Can be a useful resource
- Understanding the examples is the most important takeaway



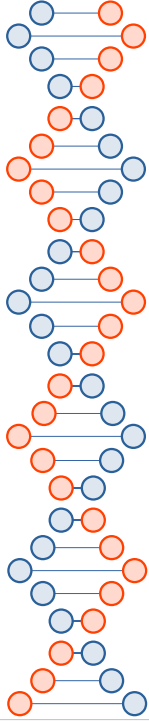
## Important grep flags

- -R is for recursive
  - Search a directory and its contents for matches
- -E is extended regular expression
  - No need to escape meta-characters ? + { | ( )
- A bunch of flags that can format the output differently



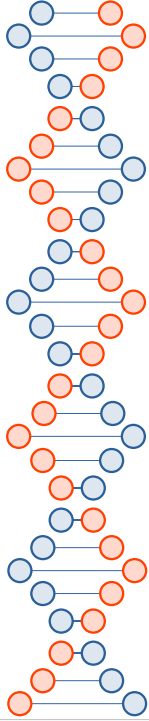
## Sed

- Stream editor
- Primarily used for text substitution
- Turing-complete programming language
  - We'll keep it simple



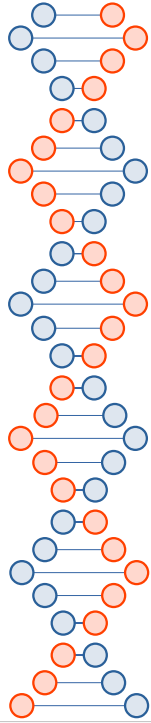
## Sed syntax

- `do_one_thing ; do_another_thing`
- Things we can do:
  - `s/find/replace/`
  - `s/find/replace/g`
- Can use backreferences
  - Group patterns with ( ) and reference them in substitutions
  - The pattern with the first ( is `\1`, the second ( is `\2`, and so on
  - `(a(b))` : `\1` is `ab`, `\2` is `b`
- Use `-E` for regex



## Sed one-liners

- **sed 's/a/b/'** - replace the first a of each line with a b
- **sed 's/a/b/g'** - replace all as with bs
- **echo -e 'ebbbb\nbbebb' | sed -e '/e/s/b/a/'**
- **echo -e 'ebbbb\nbbebb' | sed -e '/^e/s/b/a/'**
- Can use ; instead of -e, but -e is more readable
- Instead of / you can use another character like # or |



The End

Today's Terminal Toy:  
fortune