Data wrangling

Paul M. Magwene

2/7/2018

Tidy data

To facilitate downstream analyses, data should be organized in a manner such that...

1. Each variable must have its own column.
2. Each observation must have its own row.
3. Each value must have its own cell.

Real-world data is often messy

Data files you generate or will be given may...

- Be poorly organized
- Have missing values
- Contain extraneous information
- Confounds variables and labels

Example: Starting messy data

	$cond_{1,t_1}$	$cond_{1,t_2}$	...	$cond_n$	$cond_{n,t_1}$	$cond_{n,t_2}$
$gene_1$	0.01	0.8	...		2.1	1.4
$gene_2$	1.1	NA	...		1.5	0.5
...	...	....	...		...	...
$gene_p$	3.14	1.4	...		NA	2.71

Problems

- Missing column headers
- Genes are cases rather than variables
- Confounds time and condition
- Blank columns – used for visual organization in spreadsheet, but interferes with analysis

Tidying data: Fixing headers, dropping extraneous columns

gene.name	$cond_{1,t_1}$	$cond_{1,t_2}$	...	$cond_{n,t_1}$	$cond_{n,t_2}$
$gene_1$	0.01	0.8	...	2.1	1.4
$gene_2$	1.1	NA	...	1.5	0.5
...	...	....	...	...	...
$gene_p$	3.14	1.4	...	NA	2.71

Tidying data: converting from "wide" to "long" format

gene.name	cond.and.time	expression
$gene_1$	$cond_{1,t_1}$	0.01
$gene_1$	$cond_{1,t_2}$	0.8
...	...	....
$gene_1$	$cond_{n,t_1}$	2.1
$gene_1$	$cond_{n,t_2}$	1.4
...	...	....
$gene_2$	$cond_{n,t_1}$	1.1
$gene_2$	$cond_{n,t_2}$	NA
...	...	....
$gene_p$	$cond_{n,t_1}$	NA
$gene_p$	$cond_{n,t_2}$	2.71

Tidying data: separating combined variables

gene.name	condition	time	expression
$gene_1$	$cond_1$	t_1	0.01
$gene_1$	$cond_1$	t_2	0.8
...	...	.....	...
$gene_1$	$cond_n$	t_1	2.1
$gene_1$	$cond_n$	t_2	1.4
...	...	.....	...
$gene_2$	$cond_n$	t_1	1.1
$gene_2$	$cond_n$	t_2	NA
...	...	...	....
$gene_p$	$cond_n$	t_1	NA
$gene_p$	$cond_n$	t_2	2.71

Tidy data facilitates visualization and analysis with minimum code

```
tidy.long %>%
  filter(gene %in% genes.of.interest) %>%
  ggplot(aes(x = time, y = expression, color = gene)) +
    geom_line() +
    facet_wrap(~ condition)
```

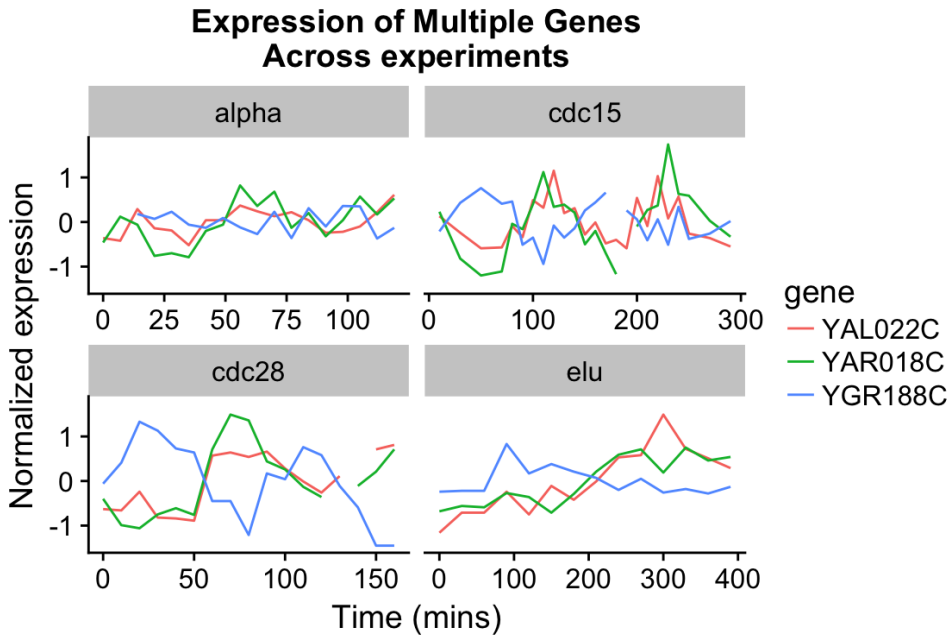


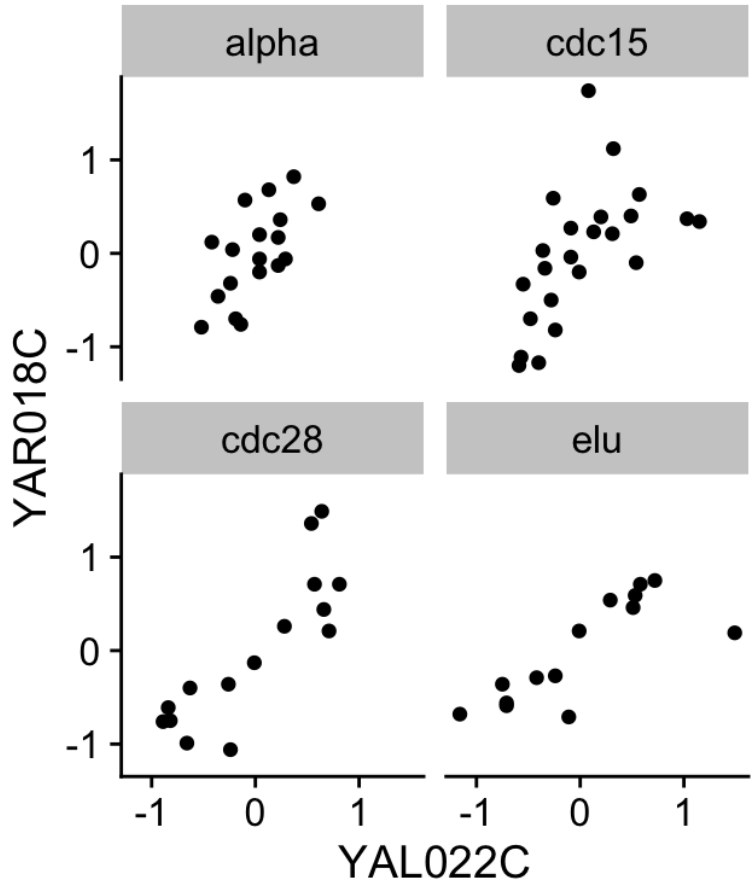
Figure 1: A visualization from tidy long data

Tidy, wide data is useful too if properly organized

condition	time	$gene_1$	$gene_2$	...	$gene_p$
$cond_1$	t_1	0.01	1.10	...	3.14
$cond_1$	t_2	0.80	NA	...	1.40
...	...	...	...	...	...
$cond_n$	t_2	1.40	0.50	...	2.71

A visualization from tidy, wide data

```
tidy.wide %>%
  filter(!is.na(YAL022C) & !is.na(YAR018C))%>%
  ggplot(aes(x = YAL022C, y = YAR018C)) +
    geom_point() +
    facet_wrap(~ condition)
```



Exploiting both long and wide tidy data allows us to create sophisticated visualizations and understand interesting patterns in our data

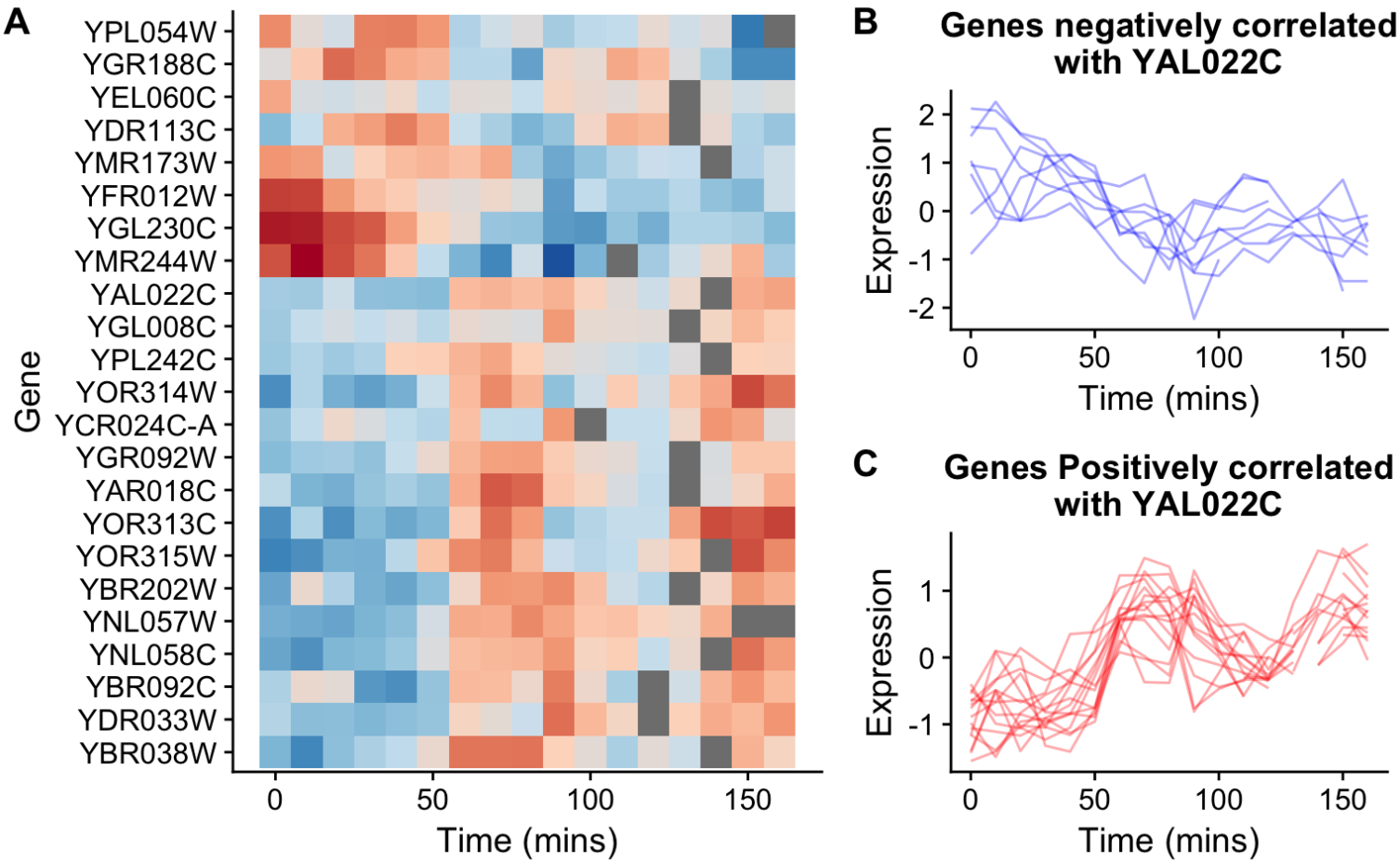


Figure 3: A visualization built by combining tidy long and wide data representations