Using org-mode in research

senthil

October 23, 2016

1 Introduction

- Emacs is a text editor (short for Editor MAcroS)
- Text editors are much more useful than to just display/edit text
- Plain (UTF-8) text is the best form to preserve data (not a Excel file not a JPEG image of a plot!)
- Emacs includes a ‘mode’ called org-mode that is very useful to organize data and experiments, carry out analysis using scripts and other external programs and store results and all the procedures applied to the data in a single location as a simple text file.
- Documenting in org-mode is useful in reproducible research.

2 Org-mode in note taking

2.1 Sections and subsections

- Section titles have one ‘*’, two ‘**’ indicate subsection titles
- Easy to increase or decrease levels of different sectioning
-
- an item in a list (see below)
-

2.2 Lists

2.2.1 Bullet points

- We already saw how itemized lists are shown in the 1

2.2.2 Numbered lists

- Numbered lists appear like this:
    1. First item
    2. Second item
    3. ...etc.
    4.
    5.
    6.
    7.
    8.

2.2.3 Check list [3/3]

- ☒ Finish first item
- ☒ Then second item
- ☒ Finally do the third item in the check list

2.3 Todo lists [100%]

2.3.1 DONE Download 16S rRNA sequences from NCBI

2.3.2 DONE Align to SILVA database using mothur

2.3.3 DONE Manually curate using BioEdit, SeaView, etc.

2.3.4 DONE Apply hard and soft filter

2.3.5 DONE Calculate distance matrix, construct NJ tree

2.3.6 DONE Construct ML and MP tree

2.3.7 DONE Edit trees in iTOL

2.3.8 DONE Beautify in Inkscape or Illustrator

2.4 Tables

- Tables can be created by hand
- Spreadsheet like capablities

Subject	Score
Phy	90
Chem	89
Biol	70
Math	79
Total	249
Avg	83

- Tables can also be created from tab-limited or csv-limited text

- Use command: `M-x org-table-convert-region`

bin	N50	size	ctgs	genes	markers	%cov	dup_mark
b1_m4	6467	7.29	1708	8588	136	108	294
b2_m4	5206	5.96	1609	6954	38	30	23
hi_m4	21756	1.78	123	1979	58	46	5
meta_m4	4738	61.25	20730	76130	139	111	1407

2.5 Figures

3 Org-mode in reproducible research

3.1 Many published results not reproducible

- A figure or a plot is useful to describe results
- 53 deliberately chosen cancer research papers (novel approaches)
- Only 6 were reproducible (11 % cases)
- 73 % authors: NO RESPONSE to data request (Psychology)

3.2 Solution: Include data with your figures

- The following illustrate a trivial example
- But there are real world examples (see email)

3.3 Toy example

- 454 reads assembled using genome assembly program (Newbler)
- Two varying parameters:
  - Minimum overlap length in bp(ml): 5, 10, 20, 30, 40, 50
  - Minimum percentage identity (mi): 75, 80, 85, 90, 95
- The following snippets of code runs newbler gets the stats

3.3.1 Newbler assembly

```
# /mnt/hit2g/senthil_files/PYROPHAGE/bin/try_runassembly.sh
#!/bin/bash
# Senthil / UNLV / October 14, 2014
# Try different -ml and -mi values for assembling pyrophage data
for mi in 75 80 85 90 95;
do
    for ml in 5 10 20 30 40 50;
    do
        # with urt
        runAssembly -o ../results/URT_NEW_PYRO_${mi}_${ml} -force \
            -ml ${ml} -mi ${mi} -nobig -cpu 6 -urt \
            ../data/Hot_Springs_metagenome_G7162.fasta;
    done
done
```

3.3.2 Get assembly stats

- Use R to check assembly statistics

- Extract stats to output file

```
for i in $(find ../results/ -name "454AllContigs.fna");
do
    j=$(echo $i | cut -d "/" -f 3);
    k1=$(echo ${j} | cut -d"_" -f 4);
    k2=$(echo ${j} | cut -d "_" -f 5);
    k3=$(echo $j | cut -d "_" -f 1);
    echo -ne "${j}\t${k1}\t${k2}\t\"${k3}\"\t";
    read_fasta -i ${i} | analyze_assembly -x \
        | cut -d ":" -f 2 | tr ’\n’ ’\t’ | sed -e ’s/---//g’;
    echo;
done > urt.out;
```

3.3.3 Experimental output table

name	mi	ml	n50	lc	asize	ctgs
URT_NEW_PYRO_95_20	95	20	1686	17476	8327669	7880
URT_NEW_PYRO_95_30	95	30	1628	17514	8246873	8012
URT_NEW_PYRO_80_30	80	30	1571	15253	8427227	8594
URT_NEW_PYRO_95_5	95	5	1733	17514	8294156	7524
URT_NEW_PYRO_95_50	95	50	1511	17486	8058979	8446
URT_NEW_PYRO_75_50	75	50	1471	16189	8256061	8971
URT_NEW_PYRO_75_5	75	5	1677	17515	8490416	8100
URT_NEW_PYRO_95_10	95	10	1733	17476	8290778	7563
URT_NEW_PYRO_75_40	75	40	1526	15253	8340378	8746
URT_NEW_PYRO_80_10	80	10	1679	17515	8499064	8069
URT_NEW_PYRO_85_40	85	40	1528	15253	8314699	8702
URT_NEW_PYRO_80_40	80	40	1528	11866	8333476	8752
URT_NEW_PYRO_85_20	85	20	1624	15253	8535743	8465
URT_NEW_PYRO_80_50	80	50	1472	15253	8246656	8947
URT_NEW_PYRO_90_10	90	10	1681	17515	8494919	8079
URT_NEW_PYRO_85_50	85	50	1472	15253	8245814	8930
URT_NEW_PYRO_75_20	75	20	1623	15253	8533799	8483
URT_NEW_PYRO_85_30	85	30	1569	15253	8432496	8618
URT_NEW_PYRO_85_10	85	10	1677	17553	8499299	8089
URT_NEW_PYRO_90_20	90	20	1619	15253	8528845	8487
URT_NEW_PYRO_85_5	85	5	1677	17515	8492951	8077
URT_NEW_PYRO_90_30	90	30	1577	15253	8423223	8583
URT_NEW_PYRO_95_40	95	40	1575	17486	8125789	8085
URT_NEW_PYRO_75_30	75	30	1578	14477	8420677	8547
URT_NEW_PYRO_90_50	90	50	1477	15253	8231116	8889
URT_NEW_PYRO_80_5	80	5	1677	13349	8496575	8092
URT_NEW_PYRO_75_10	75	10	1677	17553	8498313	8074
URT_NEW_PYRO_80_20	80	20	1617	16530	8547131	8506
URT_NEW_PYRO_90_5	90	5	1680	17515	8490953	8065
URT_NEW_PYRO_90_40	90	40	1528	15253	8330050	8703

3.3.4 Plot

```
library(ggplot2)
mydat <- read.table('urt.out', head=T, row.names=1)
pdf('n50_ovlgth.pdf', useDingbats=FALSE)
p <- ggplot(mydat, aes(ml, n50)) +
    geom_point(aes(colour=factor(mi))) +
    xlab('Seq overlap (bp)') +
    ylab('N50 (bp)')
p <- p + labs(colour='% identity')
p + ggtitle('Sequence identity and overlap vs N50')
dev.off()
```

3.4 Final result

- The length of the sequence overlap between reads and the N50 are inversely related (Fig 1), higher identity (95%) resulted in slightly better N50.

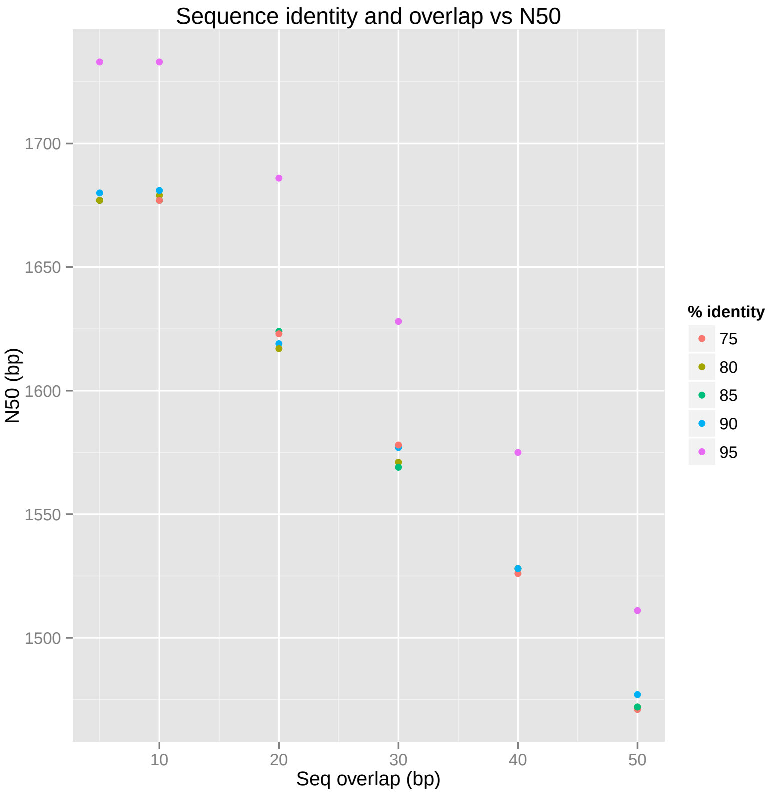

Figure 1: N50 is inversely related to sequence identity and overlap

4 Exporting to other formats

- So far, we saw how everything is text (scripts, results, documentation, etc)
- Org-mode allows exporting the text to PDF, HTML
- Use ~C-e C-l C-p' to get PDF
- Use ~C-e C-h C-h' to get HTML