A LUST MANUAL

The entire process of using the LUST algorithm, beginning to end, is along one. This document is intended to help the user get started.

1. Getting data and pre-processing it

1.1. **Get TCGA data and transfer it to MATLAB.**

Go to Broad Institute GDAC Firehose portal.

Choose disease and go to data *browse*.

Download illuminahiseq normalized gene data and merged clinical data. (Can get microRNA and methylation also.)

Use Winzip to create .txt files, then put those into Excel.

Remove expression from surrounding tissues, identified by 11A in the TCGA barcode, keeping only 01A and 01B.

On a temporary copy, remove the text data in first column and first two rows.

Importdata from temp to a variable, say MM. You might have to save the Excel file as a text file to do this.

1.2. **Preprocess and determine conftol.**

Run *preproclm* on the temporary variable MM. The format is just

$$\text{MP} = \text{preproc(MM)}$$

where MM is the raw expression matrix just constructed. This log transforms and quantile-normalizes the data.

Save the result as *cancer_gene_exp.mat*.

1.3. **Make clinical files.**

The expression order is generally in alphabetical order by TCGA identifier. The clinical data is usually not; it will consist of several segments, in order within the segment. Moreover, there will be more clinical samples than expression samples, so the extras must be deleted. Make a copy of the clinical files and sort it until it agrees with the expression files, column by column. Save this as *cancer_clinical_sorted.xlsx.*

Now make a copy that contains only the TCGA identifier, survival time (either days till death or till last followup, depending on vital status), stage, censoring (0 for dead, 1 for alive). You will have to search for those rows. *Caveat:* The vital status, time to death and

Date: March 11, 2018.

time to last followup are in their own rows, and one or two places under *followups*. Use the most recent information. Save this as *cancer_clinical_sorted_short.xlsx*.

Finally, make the file *clinical_cancer.mat*, also called *surv_cancer.mat* or *cox_cancer.mat*. If there are N patients, then the *clinical* file is $N \times 4$ with columns survival, age, stage, censoring. (In the long run we may want more columns, but for now just this.)

2. Some data files

- *'TCGA_trimrids.mat'*, the gene names for 20,531 genes.
- *'stomach_gene_exp.mat'*, gene expression for 415 stomach cancer patients.
- *'gene_exp_12_stom.mat'* gene expression for 203 stage 1 and 2 stomach cancer patients.

3. Finding metagenes

This is step 1 of the algorithm, using the program *lust_find_metagenes.m*. Its form is

[kompg,overlaps] = lust_find_metagenes(rids_file, exp_file, dens, conftol)

The program takes expression data for a set of patients, and produces enough information to find the metagenes. This last part is not automated. The input is

- *rids_file*, the names of the genes. For our purposes this is usually *TCGA_trimrids.mat*, a list of the standard 20,531 genes used by TCGA.
- *exp_file*, the pre-processed mRNA expression file for the type of cancer (or subtype) you want to test.
- *density* is the desired density of $+1$ and -1's in the discretized expression matrix, usually 0.5.
- *conftol*, the confidence parameter that adjusts the sensitivity of the algorithm. This is normally in the range 0.7 to 0.8.

Thus a typical command line would be

```
[kompg,overlaps] = lust_find_metagenes
('TCGA_trimrids.mat','stomach_gene_exp.mat', .5, .75)
```

but on one line. The output is 32 candidate metagenes, listed on the screen but not as an output file (it could be, but you won't use most of them), plus:

- *kompg*, a table with the scores for each candidate metagene.

- *overlaps*, a table listing the number of genes in each candidate metagenes and their overlaps.

There is an art to extracting the metagenes from this mess, but this is the information you need.

Now make a *rids* file for each metgene you want to consider. Make a variable, say RR, with the names of the genes in the metagene; this can be done using copy and paste from the output (to the screen) of *lust_find_metagenes.m.* then use the *save* command:

save 'cancer_meta_X_rids.mat' RR

It is often useful at this point to extract the expression matrix for the genes in the metagene, using the program *extrakt.m*, and then another *save.* (But *extrakt.m* is embedded in the program *lust_meta_to_rids.m* for the next step, so it is not necessary to always do this.)

4. Finding signatures

This is step 2 of the algorithm, using the program *lust_meta_to_rids.m.* Its form is

oldknob = lust_meta_to_sigs (rids_file, exp_file, clinical_file,
metax, DE_rids, dens, confmin, confdelta, conftimes)

The program takes a metagene, expression data and clinical data to produce signatures (small sets of genes) predictive of survival. By design it produces more signatures than you want; only the top few are significant. The input is

- *rids_file*, the names of the genes in the metagene, which should be a subset of *TCGA_trimrids.mat.* Each metagene is selected from the output list of step 1. Using copy and paste, and the MATLAB 'save' command, you make a file containing the names of the genes in your metagene. See *'kirc_meta_C_rids.mat'* for an example.
- *exp_file*, the pre-processed mRNA expression file for the type of cancer (or subtype) you want to test; the program will extract the expression for the genes in the metagene only.
- *clinical_file*, an $N \times 4$ matrix where there are N patients. Column 1 is the survival time in days. Column 2 is the patients age at time of diagnosis, column 3 is the stage at diagnosis. Column 4 has a 1 if the patient is still alive (not censored), and a 0 if the patient is dead (censored). These are often names *cox_xxx.mat.*

- *'metax'* is a string expression for labeling the graphs with the cancer type and metagene, e.g., 'stomach_meta_A'.
- For free you get to see differential expression on a list of genes between long and short survivors as predicted by the signature. The current list is 'ICS_trimrids.mat', but this can be altered.
- *density* is the desired density of $+1$ and -1's in the discretized expression matrix, usually 0.5.
- *confmin, confdelta, conftimes*: the program runs with one or more values of *conftol*, starting at *confmin*, incrementing by *confdelta*, with *conftimes* the number of runs. Typical is 0.66, .04, 3 to run with *conftol* at .66, .70, .74.

Thus a typical command would be

```
oldknob = lust_meta_to_sigs
 ('kirc_meta_A_rids.mat','stomach_gene_exp.mat','cox_stom.mat,
'KIRC meta A on stomach','ICS_trimrids.mat', .5, .66, .04, 3)
```

but on one line. The output is 10 candidate signatures, plus a table *oldknob* of the top candidates considered. For each signature you get

- a list of genes in the signature,
- a graph with the Kaplan-Meier survival curves for the signature,
- the *p*-values for the log-rank test and Cox proportional hazards test, and the Fisher score $-\log p_1 - \log p_2$,
- a small table indicating the number of censored and uncensored patients for the top and bottom quartiles,
- a table of differential expression *p*-values for the chosen genes, e.g., immune checkpoint genes.

5. Testing signatures

You can also test a specific signature (or whole metagene), with or without the differential expression. These are the programs *signature_test.m* and *signature_test_with_DE.m.* The format of the latter is

```
[DE,fisher] = signature_test_with_DE( sig_rids_file, exp_file,
              clinical_file, string_label, DE_rids)
```

6. Whole cancer analysis

It can be useful to look at the survival properties and differential expression for the entire group, or a selected subset of patients. For this you use one of the two similar programs *whole_survival.m* or *whole_survival_with_DE.m.* The form of the former is

fisher = whole_survival(surv_data,metax,xtile,DE_rids,exp_file,binwidth)

where

- *surv_data* is the standard $N \times 4$ survival file,
- *'metax'* is a string expression for labeling the graphs with the cancer type,
- *xtile* is .25 for quartiles, .33 for thirds, etc.,
- *DE_rids* is the file with the names of the genes for which you want differential expression, typically *ICS_trimrids.mat.*
- *exp_file*, the pre-processed mRNA expression file for the type of cancer (or subtype) you want to test. the program will extract the expression for the genes in the metagene only.
- *binwidth* is the number of days per bin of the histogram for survival, typically 182.5, 250 or 365.

The latter program does not take a *DE_rids* file, but uses the whole TCGA list. But you only want the genes that show the most differential expression, so it has the input parameter *tol* for tolerance, and only lists those genes where the p-value for the differential expression is less than *tol.*

Caveat: When you test all the genes for differential expression, there is a serious multiple testing problem, and some of the results may be statistical flukes.

The output will be a histogram of survival times, survival graph and a list of differential expressions *for the set of patients who either died, or are still alive but have survived longer than the mean time for the whole set of patients.* This is one way with patients who are censored after a short time.