CHASM (Cancer-specific High-throughput Annotation of Somatic Mutations) Example workflow

Here is an example on how to run CHASM. For additional examples, download the software package from http://wiki.chasmsoftware.org and see the examples directory and readme file.

Files required to run CHASM

- The list of mutations from the tumor sequencing study formatted in genomic or transcript coordinates
- A passenger mutation rate table describing the mutation spectrum of the tumor type from which the mutations originated
- The list of features to use for classifier training

Mutation File formats:

Mutations can be specified as tab delimited lists of transcript and amino acid substitution with codon number ('transcript coordinates') or as genomic location and nucleotide substitution ('genomic coordinates'). In the later case, the –g flag should be used in CHASM command lines to indicate that mutations should first be mapped to transcript coordinates. The expected format for each coordinate system is described here:

Transcript coordinates:

- (optional) mutation identifier
- transcript
- amino acid substitution (e.g. G12D or H1047R)

Example of transcript coordinates:

```
# UID / Transcript / AA change
TR1   NM_001126116.1   D127Y
TR2   NM_001144919.1   R162Q
TR3   NM_000321.2      Q702K
TR4   NM_000245.2      A1108S
TR5   NM_004333.4      V600E
TR6   NM_001005862.1   G746V
```

Genomic Coordinates:

- (optional) mutation identifier
- chromosome (e.g. "chr1")
- 0-based start position of the nucleotide
- 1-based end position of the nucleotide
- strand of the nucleotides being reported in the following column*

- reference nucleotide
- alternative nucleotide

*If the reference and alternative nucleotides match the forward strand of the reference genome this should be “+”, if they match the reverse strand it should be “-”

Example of genomic coordinates:

# UID	Chr.	Start	Stop	Strand	Ref. base	Alt. base
TR1	chr17	7577505	7577506	-	G	T
TR2	chr10	123279679	123279680	-	G	A
TR3	chr13	49033966	49033967	+	C	A
TR4	chr7	116417504	116417505	+	G	T
TR5	chr7	140453135	140453136	-	T	A
TR6	chr17	37880997	37880998	+	G	T

Training a classifier:

CHASM should be run from the directory where it is installed.

The BuildClassifier script is used to construct a new CHASM classifier. BuildClassifier expects the following input arguments:

-m path to a passenger mutation rate table
-f path to a list of features
-i path to a list of mutations observed in a tumor study of the tissue of interest *
-o name to assign the trained classifier
-g **
-s random seed (optional)

* Passengers are generated in the set of transcripts observed to be mutated in a tumor sequencing study. This can be the same data set that the user wants to score with the classifier, or an independent data set of similar tumors.
** The -g flag should be used only if the list of mutations to be scored are in genomic coordinates.

To build a classifier when the input mutation data set is in transcript coordinates:

```
cd $CHASMDIR
./BuildClassifier -m ClassfierPack/contexts/TUMORTYPE.context -f
ClassifierPack/features/Features.list -o ClassifierName -i MutationFile
```

For genomic coordinates add the –g flag:

./BuildClassifier -m ClassfierPack/contexts/TUMORTYPE.context -f ClassifierPack/features/Features.list -o ClassifierName -i MutationFile -g

After running BuildClassifier, the following files will have been placed in the classifier directory:

- Mutations for classifier training (drivers.tmps and passengers.tmps) and null model mutations for estimating p-values (null.tmps)
- Feature information for the training set (Train.arff) and the null model mutations (Null.arff)
- The CHASM scores for the null model mutations (Null.classified)
- The list of features used to describe mutations (Features.list)
- The trained Random Forest (a file for each decision tree ending with the extension '.tree' and a file called train.forest)
- A human-readable dump of the Random Forest classifier (forest.out)
- A confusion matrix describing classifier performance if a CHASM score cutoff of 0.5 is used (train.conf)
- A file describing the contribution of each feature to classifier performance (train.imp)
- A file with out-of-bag scores for CHASM training set mutations (train.oob) used for estimating classifier generalization error. The use of out-of-bag scoring for Random Forest generalization error estimation is described in detail by Leo Breiman in his 2001 publication.

Scoring mutations with CHASM:

The RunChasm python script is used to score a set of missense mutations with a trained CHASM classifier. This script takes only two arguments: the classifier name and the mutation file. If the mutations are in genomic coordinates, the -g flag should be used.

Transcript coordinates:
./RunChasm ClassifierName MutationFile

or

Genomic coordinates:
./RunChasm ClassifierName MutationFile -g

Interpreting results:

The file ending with ".output" generated by RunChasm is designed for simple parsing such that CHASM scores, p-values and FDRs can easily be integrated into a larger spreadsheet with variant annotations. The input mutations can be matched to CHASM scores using the mutation identifier field. CHASM predictions can then be used to prioritize mutations for further analysis by sorting CHASM scores from smallest to

largest. This will rank the mutations by similarity to the driver mutation class of the CHASM training set versus simulated passenger mutations.

The output file generated by RunChasm can be sorted on CHASM score at the command line using:

sort -n -k 3 filename.output > filename.output.sorted

Alternatively, it may be convenient to rename the output file with a '.txt' extension and load it with a spreadsheet program, such as Microsoft Excel. After sorting, the top scoring driver candidates will be at the top of the file. In general, the top scoring driver candidates include well-known drivers as well as a subset of mutations that have not previously been implicated in the tumor type under study.