gkmSVM-R Tutorial notes

INSTALLATION for linux or mac:

Installation of Bioconductor packages:
$ R
> source('https://bioconductor.org/biocLite.R')
> biocLite('GenomicRanges')
> biocLite('rtracklayer')
> biocLite('BSgenome')
> biocLite('BSgenome.Hsapiens.UCSC.hg19.masked') (or other genomes)
> biocLite('BSgenome.Hsapiens.UCSC.hg18.masked')
> install.packages('ROCR')
> install.packages('kernlab')
> install.packages('seqinr')
> quit()

# Installation of gkmSVM package:
$ git clone https://github.com/mghandi/gkmSVM.git
$ R CMD INSTALL gkmSVM

Now, to run gkmSVM-R on the CTCF test set from Ghandi Lee, Mohammad-Noori, Beer, PLOS CompBio 2014:

Input files: ctcfpos.bed, nr10mers.fa, ref.fa, alt.fa from http://www.beerlab.org/gkmsvm

1. Generate GC, length, and repeat matched negative set and extract fasta sequence files for ctcfpos.fa and ctcfneg_1x.fa: (Larger negative sets can be generated by increasing xfold, and running time can be decreased by reducing nMaxTrials, at the cost of not matching difficult sequences. In general training on larger sequence sets will produce more accurate and robust models.)

$ R
> library(gkmSVM)
> genNullSeqs('ctcfpos.bed',nMaxTrials=10,xfold=1,genomeVersion='hg18',
outputPosFastaFN='ctcfpos.fa', outputBedFN='ctcfneg_1x.bed', outputNegFastaFN='ctcfneg_1x.fa')

2. Calculate kernel matrix:

> gkmsvm_kernel('ctcfpos.fa','ctcfneg_1x.fa', 'ctcf_1x_kernel.out')

3. Perform SVM training with cross-validation:

> gkmsvm_trainCV('ctcf_1x_kernel.out','ctcfpos.fa','ctcfneg_1x.fa',svmfnprfx='ctcf_1x',
outputCVpredfn='ctcf_1x_cvpred.out', outputROCfn='ctcf_1x_roc.out')

4. Generate 10-mer weights:

> gkmsvm_classify('nr10mers.fa',svmfnprfx='ctcf_1x', 'ctcf_1x_weights.out')

This should get AUROC=.955 and AUPRC=.954 with some small variation arising from the randomly sampled negative sets. You can then select the top weights with:

$ sort –grk 2 ctcf_1x_weights.out | head -12

which should give weights very similar to:

```
CACCTGGTGG      5.133463
CACCAGGTGG      5.090566
CACCAGGGGG      5.038873
CCACTAGGGG      4.833398
CCACCAGGGG      4.832404
CACCTAGTGG      4.782613
CACCAGAGGG      4.707206
CACTAGGGGG      4.663015
CACTAGAGGG      4.610800
CACTAGGTGG      4.580834
CCACTAGAGG      4.529869
CAGCAGAGGG      4.335304
```

5. To calculate the impact of a variant, in this case on CTCF binding, we use gkmsvm_classify to find the score difference between two alleles given in FASTA format in 'ref.fa' and 'alt.fa'. This is only different by a scale factor from deltaSVM calculated directly from SVM weights, as described in (Lee, Gorkin, Baker, Strober, Aasoni, McCallion, Beer, Nature Genetics 2015).

> gkmsvm_delta('ref.fa','alt.fa',svmfnprfx='ctcf_1x', 'dsvm_ctcf_1x.out')

If you find this tool useful, please cite:

Ghandi, Mohammad-Noori, Ghareghani, Lee, Garraway, and Beer, Bioinformatics (2016); and

Ghandi, Lee, Mohammad-Noori, and Beer, PLOS Computational Biology (2014).