Application Tutorial: OmicKriging

Keston Aquino-Michaels, Heather E. Wheeler, Vassily V. Trubetskoy and Hae Kyung Im

October 30, 2013

Method citation: Wheeler HE, et al. (2013) Poly-Omic Prediction of Complex Traits: OmicKriging. arXiv:1303.1788 `http://arxiv.org/abs/1303.1788`

To install from CRAN:

```
> install.packages("OmicKriging")
```

Start by loading OmicKriging functions into R:

```
> library(OmicKriging)
```

Define paths to the genotype (plink binary pedigree format), gene expression, and phenotype data files (paths may differ based on where the files are located). These files will later be passed to upcoming functions:

```
> "%&%" <- function(a, b) paste(a, b, sep="")
> gdsFile <- "gdsTemp.gds"
> ok.dir <- "~/OmicKriging-tutorial_data/"
> bFile <- ok.dir %&% "ig_genotypes"
> expFile <- ok.dir %&% "ig_gene_exon.txt"
> phenoFile <- ok.dir %&% "ig_pheno.txt"
```

Load the phenotype data into R:

```
> pheno <- read.table(phenoFile, header = T)
```

Load a pre-computed GCTA GRM into R (recommended):

```
> grmMat <- read_GRMBin(bFile)
```

Alternatively, to compute the GRM in R start by converting the genotype data from plink binary format into GDS format:

```
> convert_genotype_data(bFile = bFile, gdsFile = gdsFile)
```

Subsequently, compute a genetic relatedness matrix (GRM) from the GDS file:

```
> grmMat <- make_GRM(gdsFile = gdsFile)
```

By default, grmFilePrefix = NULL, however if specified, this function will save the computed GRM to disk in GCTA binary format. Additionally by default snpList = NULL and sampleList = NULL, however if specified, a list of individuals or snps will be retained in GRM calculation.

Load and calculate a gene expression relatedness matrix (GXM) with the following function:

```
> gxmMat <- make_GXM(expFile = expFile)
```

By default, gxmFilePrefix = NULL, however if specified, this function will save the computed GXM to disk in GCTA binary format.

Additional convenience functions are included to perform principal components analysis (PCA):

```
> pcMatXM <- make_PCs_irlba(gxmMat, n.top = 10)

> pcMatGM <- make_PCs_irlba(grmMat, n.top = 10)

> pcMat <- cbind(pcMatGM, pcMatXM[match(rownames(pcMatGM), rownames(pcMatXM)),])
```

The following convenience function allows the user to perform n-fold cross-validation. Specify the number of cores you wish to use (default = ”all”), the number of cross-validation folds desired (default = 10), covariates (by default covar.mat = NULL), the phenotype object, pheno.id (by default = 1 (the first phenotype in the file)), the h2 vector and a list of the correlation matrices to be included.

Note: The sum of the h2 vector must be between 0 and 1. In this example, we will give each matrix equal weight.

```
> result <- krigr_cross_validation(pheno.df = pheno,
        cor.list = list(grmMat, gxmMat),
        h2.vec = c(0.5, 0.5),
        covar.mat = pcMat,
        ncore = 2,
        nfold = "LOOCV")
```

This function will return a data.frame with column Ypred corresponding to the predicted values and column Ytest corresponding to the measured phenotypes.

Congratulations!
You have just completed the OmicKriging tutorial!

```
> result <- krigr_cross_validation(pheno.df = pheno,
+ cor.list = list(grmMat, gxmMat),
+ h2.vec = c(0.5, 0.5),
+ covar.mat = pcMat,
+ ncore = 2,
+ nfold = "LOOCV")
Detected 99 samples...
Set leave-one-out cross-validation...
With 2 logical core(s)...
Running OmicKriging...

Call:
lm(formula = Ytest ~ Ypred, data = res)
```

```
Residuals:
     Min       1Q   Median       3Q      Max
-2.08289 -0.65675  0.01653  0.68273  1.80417

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.01219    0.08644  -0.141    0.888
Ypred        0.62292    0.10924   5.702 1.28e-07 ***
---
Signif. codes:  0 "***" 0.001 "**" 0.01 "*" 0.05 "." 0.1 " " 1

Residual standard error: 0.86 on 97 degrees of freedom
Multiple R-squared: 0.2511,	Adjusted R-squared: 0.2433
F-statistic: 32.52 on 1 and 97 DF,  p-value: 1.276e-07

Finished OmicKriging in 0.556 seconds
```