OmicKriging Tutorial

Heather E. Wheeler and Hae Kyung Im

March 8, 2013

To complete this tutorial, download the `OmicKriging-tutorial_data.zip` file from `http://www.scandb.org/newinterface/tools/OmicKriging.html` and unzip the directory. Also, if not already installed, download the GCTA software from `http://www.complextraitgenomics.com/software/gcta/download.html`.

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 the R package after downloading `OmicKriging_1.0.tar.gz` from `http://www.scandb.org/newinterface/tools/OmicKriging.html`:

```
> install.packages("~/Downloads/OmicKriging_1.0.tar.gz", repos = NULL, type="source")
```

To install from CRAN:

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

To start using the functions:

```
> library(OmicKriging)
```

Define paths to the genotype (plink format), gene expression, and subject ID data files, which will be used to call the GCTA software (paths may differ based on where the files are located):

```
> genotypeheader = "~/Downloads/OmicKriging-tutorial_data/hapmap3.regulome_QC"
> expressionheader = "~/Downloads/OmicKriging-tutorial_data/geneexon.txt"
> idfile = "~/Downloads/OmicKriging-tutorial_data/commonid.txt"
```

Define the path to the GCTA executable (path may differ based on where the file is located). NOTE: you may need to run `chmod a+x gcta64` from the command line to get the correct permission to execute the program.

```
> gcta = "~/bin/gcta64"
```

Specify output strings for the genotype output and the gene expression output:

```
> grmheader = "genotypes"
> gxmheader = "expression"
```

Compute a genetic relationship matrix (GRM) using the provided SNP genotype data and a subset of subjects from the `idfile`. If no `idfile` is specified, the computation will include all subjects in the `.fam` file. The following command will generate output files `genotypes.grm.gz` and `genotypes.grm.id` in the current working directory. The command is followed by the first few lines of output:

```
> computeGRM(genotypeheader,grmfullheader=grmheader,gctaname=gcta,idfile=idfile)
```

```
**********************************************************************
* Genome-wide Complex Trait Analysis (GCTA)
* version 1.11
* (C) 2010 Jian Yang, Hong Lee, Michael Goddard and Peter Visscher
* GNU General Public License, v2
* Queensland Institute of Medical Research
**********************************************************************
Analysis started: Wed Feb 27 12:04:59 2013

Options:
--bfile /Users/heather/Downloads/OmicKriging-tutorial_data/hapmap3.regulome_QC
--autosome
--make-grm
--keep /Users/heather/Downloads/OmicKriging-tutorial_data/commonid.txt
--out genotypes
.
.
.
```

Compute a gene expression correlation matrix (GXM) using the provided gene expression data and a subset of subjects from the `idfile`. The following command will generate output files `expression.grm.gz` and `expression.grm.id` in the current working directory. The command is followed by the first few lines of output:

```
> computeGX(expressionheader,gxmheader,idfile=idfile)

          [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]
  [1,] 1.0000000 0.9688564 0.9589840 0.9636058 0.9607907 0.9551637 0.9624089
  [2,] 0.9688564 1.0000000 0.9741036 0.9733889 0.9702171 0.9723358 0.9714017
  [3,] 0.9589840 0.9741036 1.0000000 0.9738285 0.9715436 0.9663458 0.9714970
  [4,] 0.9636058 0.9733889 0.9738285 1.0000000 0.9646634 0.9621896 0.9699591
  [5,] 0.9607907 0.9702171 0.9715436 0.9646634 1.0000000 0.9696719 0.9560634
  [6,] 0.9551637 0.9723358 0.9663458 0.9621896 0.9696719 1.0000000 0.9514174
  [7,] 0.9624089 0.9714017 0.9714970 0.9699591 0.9560634 0.9514174 1.0000000
  [8,] 0.9168412 0.9371357 0.9350952 0.9239409 0.9390961 0.9335457 0.9198963
  .
  .
  .
```

Generate a list of correlation matrices (`corlist`) to include in the `okriging` prediction:

```
> cortempo = data.frame(headers=c(grmheader,gxmheader),stringsAsFactors=F)
> corfilelist = c(cortempo[,1])
> corfilelist

[1] "genotypes"  "expression"

> corlist = readcorlist(corfilelist)
> corlist

[[1]]
            NA12750     NA11831     NA12146     NA11882     NA07056     NA12707
NA12750  1.06199900  0.09093205  0.06966235  0.08836764  0.08498611  0.06729684
NA11831  0.09093205  1.08269000  0.07351140  0.08632062  0.09684574  0.07509552
NA12146  0.06966235  0.07351140  1.06471500  0.07910301  0.08887303  0.09300449
NA11882  0.08836764  0.08632062  0.07910301  1.05780000  0.08782883  0.06807782
```

```
.
.
.
NA19207  0.99766930  0.07444655  0.07158796
NA19103  0.07444655  0.99196990  0.08180238
NA19099  0.07158796  0.08180238  1.01152600

[[2]]
          NA10846   NA10847   NA12144   NA12145   NA12146   NA12239   NA06994
NA10846 1.0000000 0.9688564 0.9589840 0.9636058 0.9607907 0.9551637 0.9624089
NA10847 0.9688564 1.0000000 0.9741036 0.9733889 0.9702171 0.9723358 0.9714017
NA12144 0.9589840 0.9741036 1.0000000 0.9738285 0.9715436 0.9663458 0.9714970
NA12145 0.9636058 0.9733889 0.9738285 1.0000000 0.9646634 0.9621896 0.9699591
.
.
.
NA19238 0.9677781 0.9452533 1.0000000 0.9733510 0.9794843
NA19239 0.9666349 0.9383375 0.9733510 1.0000000 0.9675642
NA19240 0.9678713 0.9603856 0.9794843 0.9675642 1.0000000
```

Compute the first 10 principal components from the GRM and the GXM to use as covariates in the `okriging` prediction. The commands below will generate the following files in the current working directory: `genotypes.eigenval`, `genotypes.eigenvec`, `expression.eigenval`, `expression.eigenvec`.

```
> computePC(grmheader, idfile=idfile, gctaname=gcta)
> computePC(gxmheader, idfile=idfile, gctaname=gcta)
```

Merge the principal components from the GRM and the GXM into one covariate file for use in the `okriging` prediction.

```
> tempoPC = read.table(paste(grmheader,".eigenvec",sep=""))
> names(tempoPC)[1:2] = c("FID","IID")
> tempoEG = read.table(paste(gxmheader,".eigenvec",sep=""))
> names(tempoEG)[1:2] = c("FID","IID")
> cova = merge(tempoPC,tempoEG,by.x=c("FID","IID"),by.y=c("FID","IID"))
> cova[1,]

  FID     IID      V3.x       V4.x     V5.x      V6.x      V7.x      V8.x
1 1334 NA10846 0.0828383 -0.0252644 -0.21174 -0.146945 -0.126587 -0.138558
        V9.x    V10.x     V11.x     V12.x      V3.y       V4.y      V5.y
1 -0.0107013 0.159496 -0.150518 -0.111634 0.0792288 -0.0740033 0.0515784
        V6.y      V7.y       V8.y    V9.y      V10.y    V11.y      V12.y
1 -0.0783922 0.0330657 -0.0625952 0.01623 -0.0932787 0.158431 -0.0779729

> covamat = as.matrix(cova[,!(names(cova) %in% c("FID","IID"))])
> rownames(covamat) = cova$IID
> covamat[1:2,]

             V3.x       V4.x       V5.x       V6.x       V7.x      V8.x
NA10846 0.0828383 -0.0252644 -0.2117400 -0.1469450 -0.1265870 -0.138558
NA10847 0.0790144  0.0028847  0.0721737  0.0503041 -0.0289619  0.258257
              V9.x      V10.x      V11.x      V12.x      V3.y       V4.y
NA10846 -0.0107013  0.1594960 -0.1505180 -0.1116340 0.0792288 -0.0740033
NA10847  0.1629440 -0.0711817 -0.0449313 -0.0766805 0.0803068 -0.0255578
             V5.y       V6.y        V7.y       V8.y      V9.y      V10.y
```

```
NA10846 0.0515784 -0.0783922  0.03306570 -0.0625952 0.0162300 -0.0932787
NA10847 0.0282218 -0.0426651 -0.00926951 -0.0542956 0.0234534  0.0220692
           V11.y      V12.y
NA10846 0.158431 -0.0779729
NA10847 0.106707 -0.0190641
```

Define the correlation matrix weights for the `okriging` prediction. In this example, equal weights are given to the GRM (`corlist[1]`) and GXM (`corlist[2]`).

```
> matwts = c(0.5,0.5)
```

Define the path to the phenotype file, read the file, and set the phenotype name:

```
> ptfile = "~/Downloads/OmicKriging-tutorial_data/growth.txt"
> pt = read.table(ptfile, as.is=T, header=T)
> rownames(pt) = pt$IID
> ptname = "igrowth"
> pt[1:3,]
```

```
          FID     IID   igrowth
NA06984  1328 NA06984 -1.199870
NA06985  1341 NA06985 -1.219671
NA06986 13291 NA06986  2.004806
```

Read the idfile:

```
> id = read.table(idfile,as.is=T,header=T)
> id[1:3,]
```

```
    FID     IID
1  1334 NA10846
2  1334 NA10847
3  1334 NA12144
```

Predict iGrowth by using one family at a time as the test set and the rest of the individuals as the training set.

```
> idfamlist = unique(id$FID)
> pred = data.frame()
> for(fam in idfamlist){
        idtest = id$IID[id$FID == fam]
        idtrain = id$IID[!(id$IID %in% idtest)]
        res = okriging(idtest,idtrain,corlist,matwts,pt,phenoname=ptname,Xcova=covamat)
        pred = rbind(pred,res)
        print(res)
  }
```

```
            IID       Ypred      Ytest
NA10846 NA10846 -0.98817953 -2.7583275
NA10847 NA10847  0.08282539  0.8529310
NA12144 NA12144  0.43317219 -0.3652869
NA12145 NA12145  0.05390511  0.9336448
NA12146 NA12146  0.47511462  0.8920761
NA12239 NA12239  0.40543085  1.0184136
            IID       Ypred      Ytest
NA06994 NA06994 -0.03865716  0.1800688
NA07000 NA07000 -1.47434915 -0.3121563
```

```
NA07022 NA07022  1.08175404 -0.8591054
NA07029 NA07029 -0.03939941  1.2154949
NA07056 NA07056  0.90701728  1.5139088
.
.
.
            IID     Ypred      Ytest
NA19238 NA19238 -1.576936 -0.6547554
NA19239 NA19239 -1.022883  0.6017974
NA19240 NA19240 -1.933233 -1.0170851
```

Test for association between the predicted iGrowth values (`Ypred`) and the true iGrowth values (`Ytest`) using Spearman's rank correlation:

```
> cor.test(pred$Ypred,pred$Ytest,method="spearman")

        Spearman rank correlation rho

data:  pred$Ypred and pred$Ytest
S = 274660, p-value < 2.2e-16
alternative hypothesis: true rho is not equal to 0
sample estimates:
      rho
0.5900107
```

Test for linear association between the predicted iGrowth values (`Ypred`) and the true iGrowth values (`Ytest`) using linear regression:

```
> summary(lm(Ypred~Ytest,data=pred))


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

Residuals:
     Min       1Q   Median       3Q      Max
-1.83828 -0.52606  0.01942  0.41035  2.13385

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.02209    0.05482   0.403    0.688
Ytest        0.45954    0.05283   8.699 4.26e-15 ***
---
Signif. codes:  0 "***" 0.001 "**" 0.01 "*" 0.05 "." 0.1 " " 1

Residual standard error: 0.6911 on 157 degrees of freedom
Multiple R-squared: 0.3252,        Adjusted R-squared: 0.3209
F-statistic: 75.68 on 1 and 157 DF,  p-value: 4.265e-15
```

Change the matrix weights to include only the GRM in the `okriging` prediction. Compare the linear regression results above to those below:

```
> matwts = c(1,0)
> pred = data.frame()
> for(fam in idfamlist){
        idtest = id$IID[id$FID == fam]
```

```
        idtrain = id$IID[!(id$IID %in% idtest)]
        res = okriging(idtest,idtrain,corlist,matwts,pt,phenoname=ptname,Xcova=covamat)
        pred = rbind(pred,res)
   }
> summary(lm(Ypred~Ytest,data=pred))

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

Residuals:
     Min       1Q   Median       3Q      Max
-1.90483 -0.50589 -0.00563  0.43485  2.27464

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.007775   0.058320  -0.133    0.894
Ytest        0.450343   0.056431   7.980 2.65e-13 ***
---
Signif. codes:  0 "***" 0.001 "**" 0.01 "*" 0.05 "." 0.1 " " 1

Residual standard error: 0.7422 on 160 degrees of freedom
Multiple R-squared: 0.2847,        Adjusted R-squared: 0.2802
F-statistic: 63.69 on 1 and 160 DF,  p-value: 2.654e-13
```

For larger datasets, it may be useful to speed up the computation time of the `okriging` prediction by using 10% of the sample at a time as the test set and the rest of the sample as the training set (10-fold cross-validation). The following commands show one way to do this. First, define the ten groups:

```
> idlength = length(id$IID)
> g = 1:10
> groupid = sample(g,idlength,replace=T)
> newiddata = data.frame(groupid,id$IID)
> colnames(newiddata) = c("FID","IID")
> newiddata
```

Note: the FID column will vary based on random sampling, but approximately 10% of the sample will be included in each group 1-10.

```
 FID     IID
1     7 NA10846
2     3 NA10847
3     7 NA12144
4     5 NA12145
5     6 NA12146
.
.
.
156   9 NA19194
157   5 NA19238
158   2 NA19239
159   9 NA19240
```

```
> idsubsetlist = unique(newiddata$FID)
> pred = data.frame()
> for(idsubset in idsubsetlist){
         idtest = newiddata$IID[newiddata$FID == idsubset]
         idtrain = newiddata$IID[!(newiddata$IID %in% idtest)]
         res = okriging(idtest,idtrain,corlist,matwts,pt,phenoname=ptname,Xcova=covamat)
         pred = rbind(pred,res)
         print(res)
   }

            IID        Ypred      Ytest
NA10846 NA10846 -0.016842637 -2.7583275
NA12144 NA12144  0.859234755 -0.3652869
NA06994 NA06994  0.614910728  0.1800688
NA11839 NA11839 -1.019719622 -0.7286911
NA11829 NA11829 -0.228355691 -2.5051105
NA10835 NA10835 -0.041154045 -0.3617874
NA12003 NA12003 -0.773329719 -1.0717198
NA12750 NA12750  0.578182967 -0.2614076
NA12864 NA12864 -0.008267252  0.2377773
NA12892 NA12892  0.639090743  0.9187230
NA18860 NA18860  0.927716734 -0.4235416
NA19159 NA19159  2.655615918  2.4220169
NA19143 NA19143 -0.267450895  0.1794527
NA19132 NA19132 -0.351116074 -0.2157961
            IID      Ypred       Ytest
NA10847 NA10847  0.8307533  0.85293100
NA07055 NA07055 -0.7097461 -0.21598447
.
.
.

> summary(lm(Ypred~Ytest,data=pred))

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

Residuals:
     Min       1Q   Median       3Q      Max
-2.16043 -0.53688 -0.00234  0.42142  2.19183

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.03197    0.06004   0.532    0.595
Ytest        0.44570    0.05786   7.703  1.4e-12 ***
---
Signif. codes:  0 "***" 0.001 "**" 0.01 "*" 0.05 "." 0.1 " " 1

Residual standard error: 0.7569 on 157 degrees of freedom
Multiple R-squared: 0.2743,        Adjusted R-squared: 0.2697
F-statistic: 59.34 on 1 and 157 DF,  p-value: 1.404e-12
```