Annex I. Common metric calculation for Estonian sites

Richard Meitern

14-01-2020

Introduction

The aim of this document is to reproduce the new Fish Index calculation done in the "Intercalibration Technical Report: Cross-GIG Rivers – Fish Fauna." (hereafter: CrossGIG). It also aims to highlight the minimal input data needed to perform the calcualtions. In order to work this script needs the models to compute the predicted metrics and the input data tables. These can be found in the data forlder in the github repository of this project:

The orignal CrossGIG database as well as sample scripts how to calculate the Index and the needed extra tables was kindly provided by Olivier Delaigue (personal communication). We have currenlty not asked for premission to publish this data. Hence only the minimal set of data needed to do the calcualtions for Estonian sites is included in this report.

```
#change this where you hold the data
datadir<-"data/"
```

The structure of the ".rda" files in the data directory is matching the respective exported tables from the original CrossGIG Database.

Read required input data

Loading required tables. These tables have the same structure as in the crossGIG MS Access DB. However the data provides is just the Estonian subset of the original CrossGIG data.

```
#Fishing_Occasion
occas <- readRDS(paste0(datadir,"tables/","Fishing_Occasion.rda"))

#Catch
catch <- readRDS(paste0(datadir,"tables/","Catch.rda"))
```

```
# loading species that are considered to be intolerant of oxygen depletion
intorerant_fish_list <- c(
  "Alburnoides_bipunctatus",
  "Cobitis_calderoni",
  "Coregonus_lavaretus",
  "Cottus_gobio",
  "Cottus_poecilopus",
  "Eudontomyzon_mariae",
  "Hucho_hucho",
  "Lampetra_planeri",
  "Salmo_salar",
  "Salmo_trutta",
  "Salmo_trutta_fario",
  "Salmo_trutta_lacustris",
  "Salmo_trutta_macrostigma",
  "Salmo_trutta_marmoratus",
  "Salmo_trutta_trutta",
  "Salvelinus_fontinalis",
  "Salvelinus_namaycush",
  "Salvelinus_umbla",
  "Thymallus_thymallus"
)
```

Creating the table for calculating predicted values

The new table is called "calcTbl" and is created from Fishing_Occasion and Catch tables. Some modifications to original data are made to match the expected output table. The expected output table in needed to generate the exact same metric values as in the CrossGIG report.

```
intol_fish_caught <- intersect(catch$Species,intorerant_fish_list)
```

```
#these species have been excluded from aggregate values
#TODO find out why? likely something to do with biology
exclude_species <- c(
  "Ameirus_nebulosus",
  "Barbatula_quignardi",
  "Barbus_balcanicus",
  "Chondrostoma_vardarense",
  "Cottidae_sp",
  "Cyprinidae_sp",
  "Gasterosteidae_sp",
  "Gobio_obtusirostris",
  "Lampetra_sp",
  "Oxynoemacheilus_pindus",
  "Petromyzontidae_sp",
  "Phoxinus_bigerri",
  "Rutilus_rutilus_x_Abramis_brama",
  "Rutilus_rutilus_x_Scardinius_erythrophthalmus",
  "Salmo_salar_x_Salmo_trutta",
  "Squalius_laietanus"
  )
```

```
occas$captures <- NA
occas$captures <- sapply(occas$Site_code, function (site) {
  tbl <-as.data.frame(catch[catch$Site_code == site &
            !(catch$Species %in% exclude_species)  ,"Run1_number_all"])
  if(nrow(tbl) > 0){
    sum(tbl)
  }
  else{NA}
  })
```

```
occas$lDR <- log(occas$E_catchsize)
```

```
occas$totSpecies <- sapply(occas$Site_code, function (site) {
  tbl<-catch[catch$Site_code == site &
            !(catch$Species %in% exclude_species) &
              catch$Run1_number_all > 0,]

  #Sabanejewia_romanica and Sabanejewia_vallachica
  #are counted as one species in EFI+

  res<-nrow(tbl)
  sabsp<-c("Sabanejewia_romanica", "Sabanejewia_vallachica")
  if(sabsp[1] %in% tbl$Species & sabsp[2] %in% tbl$Species){
    res <- res - 1
  }
  return(res)

  })
```

Calculate observed metrics

First we need a list of species requiring a rheophilic reproduction habitat, i.e. preference to spawn in running waters This list originates from EFI+.

```
rheophilReproHabitatFish <-c(
 "Abramis_sapa",
 "Achondrostoma_arcasii",
 "Achondrostoma_occidentale",
 "Achondrostoma_oligolepis",
 "Acipenser_baeri",
 "Acipenser_gueldenstaedtii",
 "Acipenser_naccarii",
 "Acipenser_nudiventris",
 "Acipenser_oxyrinchus",
 "Acipenser_ruthenus",
 "Acipenser_stellatus",
 "Acipenser_sturio",
 "Alburnoides_bipunctatus",
 "Alosa_alosa",
 "Alosa_fallax",
 "Alosa_immaculata",
 "Anaecypris_hispanica",
 "Aristichthys_nobilis",
 "Aspius_aspius",
 "Barbus_barbus",
 "Barbus_bocagei",
 "Barbus_caninus",
 "Barbus_comizo",
 "Barbus_cyclolepis",
 "Barbus_euboicus",
 "Barbus_graellsii",
 "Barbus_guiraonis",
 "Barbus_haasi",
 "Barbus_meridionalis",
 "Barbus_microcephalus",
 "Barbus_peloponnesius",
 "Barbus_petenyi",
 "Barbus_plebejus",
 "Barbus_prespensis",
 "Barbus_sclateri",
 "Barbus_tyberinus",
 "Chondrostoma_arrigonis",
 "Chondrostoma_genei",
 "Chondrostoma_miegii",
 "Chondrostoma_nasus",
 "Chondrostoma_soetta",
 "Chondrostoma_toxostoma",
 "Chondrostoma_turiense",
 "Cobitis_calderoni",
 "Coregonus_maraena",
 "Coregonus_oxyrinchus",
 "Cottus_gobio",
 "Cottus_petiti",
 "Cottus_poecilopus",
 "Ctenopharyngodon_idella",
 "Dicentrarchus_labrax",
 "Eudontomyzon_mariae",
 "Eudontomyzon_vladykovi",
 "Gobio_albipinnatus",
 "Gobio_gobio",
 "Gobio_kesslerii",
 "Gobio_uranoscopus",
 "Gymnocephalus_schraetser",
 "Hucho_hucho",
 "Huso_huso",
 "Hypophthalmichthys_molitrix",
 "Iberochondrostoma_almacai",
 "Iberochondrostoma_lemmingii",
 "Iberochondrostoma_lusitanicum",
 "Knipowitschia_punctatissima",
 "Lampetra_fluviatilis",
 "Lampetra_planeri",
 "Lethenteron_zanandreai",
 "Leuciscus_cephalus",
 "Leuciscus_keadicus",
 "Leuciscus_leuciscus",
 "Leuciscus_lucumonis",
 "Leuciscus_muticellus",
 "Leuciscus_pleurobipunctatus",
 "Leuciscus_souffia",
 "Leuciscus_svallize",
 "Luciobarbus_sclateri",
 "Oncorhynchus_gorbuscha",
 "Oncorhynchus_kisutch",
 "Oncorhynchus_mykiss",
 "Oncorhynchus_tschawytscha",
 "Padogobius_martensii",
 "Padogobius_nigricans",
 "Petromyzon_marinus",
 "Polyodon_spathula",
 "Pseudochondrostoma_duriense",
 "Pseudochondrostoma_polylepis",
 "Pseudochondrostoma_willkommii",
 "Pseudophoxinus_beoticus",
 "Romanogobio_belingi",
 "Romanogobio_vladykovi",
 "Rutilus_frisii",
 "Rutilus_pigus",
 "Rutilus_rubilio",
 "Sabanejewia_balcanica",
 "Salmo_salar",
 "Salmo_trutta",
 "Salmo_trutta_fario",
 "Salmo_trutta_lacustris",
 "Salmo_trutta_macrostigma",
 "Salmo_trutta_marmoratus",
 "Salmo_trutta_trutta",
 "Salvelinus_alpinus",
 "Salvelinus_fontinalis",
 "Salvelinus_umbla",
 "Zingel_asper",
 "Zingel_streber",
 "Zingel_zingel",
 "Thymallus_thymallus",
 "Tropidophoxinellus_spartiaticus",
 "Vimba_vimba"
)
```

```
#extending the list with some missing species names
rheophilReproHabitatFish <- c(rheophilReproHabitatFish,
                              "Parachondrostoma_miegii",
                              "Parachondrostoma_turiense",
                              "Romanogobio_uranoscopus",
                              "Luciobarbus_graellsii",
                              "Luciobarbus_guiraonis"
                              )
```

```
occas$Obs_RHPAR <- sapply(occas$Site_code, function (site) {
  tbl <-catch[catch$Site_code == site &
              catch$Species %in% rheophilReproHabitatFish,]
  if(nrow(tbl)>0){
    tbl<-tbl[tbl$Run1_number_all > 0,]
    nrow(tbl)
  }
  else{0}
  })
```

Then we need a list of species intolerant to oxygen depletion, always more than 6 mg/l O2 in water This list is somewhat the same as "intolerant_fish_list" exept that the list has been extended to incorporate more species.

```
o2IntorerantFish <- c(
    "Alburnoides_bipunctatus",
    "Cobitis_calderoni",
    "Coregonus_lavaretus",
    "Cottus_gobio",
    "Cottus_poecilopus",
    "Eudontomyzon_mariae",
    "Hucho_hucho",
    "Lampetra_planeri",
    "Phoxinus_phoxinus",
    "Salmo_salar",
    "Salmo_trutta_fario",
    "Salmo_trutta_lacustris",
    "Salmo_trutta_macrostigma",
    "Salmo_trutta_trutta",
    "Salmo_trutta_marmoratus",
    "Salvelinus_fontinalis",
    "Salvelinus_namaycush",
    "Salvelinus_umbla",
    "Thymallus_thymallus",
    "Oncorhynchus_mykiss",
    "Lota_lota",
    "Salmo_trutta",
    "Leuciscus_souffia",
    "Barbus_peloponnesius",
    "Pseudophoxinus_stymphalicus",
    "Oncorhynchus_kisutch",
    "Leuciscus_souffia",
    "Eudontomyzon_danfordi",
    "Salvelinus_alpinus",
    "Barbus_haasi",
    "Barbus_petenyi",
    "Rutilus_pigus",
    "Zingel_asper",
    "Romanogobio_vladykovi",
    "Gobio_kesslerii",
    "Gobio_albipinnatus",
    "Aspius_aspius",
    "Romanogobio_uranoscopus",
    "Proterorhinus_marmoratus",
    "Parachondrostoma_turiense",
    "Zingel_zingel",
    "Gobio_uranoscopus",
    "Zingel_streber",
    "Pelecus_cultratus",
    "Coregonus_albula"
    )
```

```
occas$Obs_O2INTOL <- sapply(occas$Site_code, function (site) {
  tbl <-catch[catch$Site_code == site &
            catch$Species %in% o2IntorerantFish  ,"Run1_number_all"]
  tbl<- as.data.frame(tbl)
  ifelse(nrow(tbl) > 0, sum(tbl), 0)
  })
```

```
calcTbl<-occas[, c(
               "Site_code",
               "Country_code",
               "E_water_source_type",
               "E_floodplain",
               "E_geomorphological_type",
               "E_natural_sediment",
               "lDR",
               "E_distsource",
                "E_slope",
                "E_temp_jul",
                "captures",
                "totSpecies",
                "E_fishedarea",
                "Obs_O2INTOL",
                "Obs_RHPAR"
                )]

newNames <- c("Site_code",
              "country", # this is actually not needed for calculations
              "watersource",
              "fldpl",
              "geomorph",
              "natsed",
              "lDR",
              "ldist",
              "lslope",
              "Tjul",
              "captures",
              "richesse",
              "fished_area", # this is actually not needed for calculations
              "Obs_O2INTOL",
              "Obs_RHPAR")

names(calcTbl) <- newNames
```

```
#these are factors
calcTbl$fldpl <- as.factor(calcTbl$fldpl)
calcTbl$geomorph <- as.factor(calcTbl$geomorph)
calcTbl$watersource <- as.factor(calcTbl$watersource)
calcTbl$natsed <- as.factor(calcTbl$natsed)

#the distance from source should be logged
calcTbl$ldist <- log(calcTbl$ldist)
calcTbl$lslope <- log(calcTbl$lslope)
```

Calculating the predicted values

Calculate model parameters

```
# calculate syngeomorph1 and syngeomorph2
s1<-3.3026419 -(0.2646899*calcTbl$lDR) -(0.4524346*calcTbl$ldist)
s1<-ifelse(calcTbl$geomorph=="braided",s1,s1)
s1<-ifelse(calcTbl$geomorph=="constraint",(s1+0.8452244),s1)
s1<-ifelse(calcTbl$geomorph=="meandering",(s1+0.2598678),s1)
s1<-ifelse(calcTbl$geomorph=="sinuous",(s1+0.1084985),s1)
s1<-ifelse(calcTbl$fldpl=="yes",(s1-0.9023082),s1)
s1<-ifelse(calcTbl$watersource=="pluvial",(s1-0.2881176),s1)
#####
s2<-1.6153586 +(0.1075679*calcTbl$lDR) +(0.1682376*calcTbl$ldist)
s2<-ifelse(calcTbl$geomorph=="braided",s2,s2)
```

```r
s2<-ifelse(calcTbl$geomorph=="constraint",(s2-0.3548232),s2)
s2<-ifelse(calcTbl$geomorph=="meandering",(s2-1.8143746),s2)
s2<-ifelse(calcTbl$geomorph=="sinuous",(s2-1.5475224),s2)
s2<-ifelse(calcTbl$fldpl=="yes",(s2-0.3103629),s2)
s2<-ifelse(calcTbl$watersource=="pluvial",(s2-1.8082852),s2)
#### Add to  the table
calcTbl<-cbind(calcTbl,s1,s2)
names(calcTbl)[(dim(calcTbl)[2]-1):dim(calcTbl)[2]] <- c("syngeomorph1",
                                                          "syngeomorph2")
```

```r
#cleanup
rm(s1,s2)
```

Predict the values

The models are created from reference data provided by Oliver Delaigue

```r
#O2INTOL
modO2ONTOL <- readRDS(paste0(datadir,"/models/","modO2ONTOL.rds"))
```

```r
#RHPAR
modRHPAR <- readRDS(paste0(datadir,"/models/","modRHPAR.rds"))
```

```r
calcTbl$Pred_O2INTOL <- predict.glm(modO2ONTOL,calcTbl,type="response")
calcTbl$Pred_RHPAR <- predict.glm(modRHPAR,calcTbl,type="response")
```

Calculate Metrics

```r
#these mean etc. numbers were provided by Oliver Delaigue
#TODO find out from what data these means medians min and max originate from
metrics = c("Met_O2INTOL", "Met_RHPAR")
means <- data.frame(
  Metric = metrics,
  Mean = c(-0.206689319885814, -0.026072847438523),
  SD = c(0.870460729624591, 0.244495124493742),
  Median = c(0.262810554075668, 0.238432763254169),
  Min = c(-19.4047062031834, -14.6337791936192),
  Max= c(2.63049443400437, 4.51366010189352),
  row.names = metrics
)
```

helper functions

```r
#substract logs of 2 values
logDiff<-function(a,b){
  log(a + 1) - log(b+1)
}

# clip vector values at predefined minimum and maximum levels
clipExtreams <- function(val, means){
  val <- ifelse(val < means$Min, means$Min, val)
  res <- ifelse(val > means$Max, means$Max, val)
  return(res)
}

# standardize a value by using custom mean, median and SD value
standardize <- function(val, meansdf, clipMinMax = T){
  if(!nrow(meansdf)==1){stop("the means must have one row!")}
  means <- as.list(meansdf)
  #todo check if means have all the needed cols
  res <- (val - means$Mean) / means$SD
  res <- res / means$Median

  if(clipMinMax)  res <- clipExtreams(res, means)

  return(res)
}

# value scaling using a breakpoint at 0.8
customScale <-function(val, meansdf, treshold = 0.8){
  if(!nrow(meansdf)==1){stop("the means must have one row!")}
  means <- as.list(meansdf)
  #todo check if means have all the needed cols
  t1<- treshold
  t2 <- 1 - t1
  res <- ifelse(val < 1,
                (val - means$Min) * t1 / (1 - means$Min),
                (val - 1) * t2 / (means$Max-1) +t1)

  return(res)
}
```

```
calcTbl$Met_O2INTOL <- logDiff(calcTbl$Obs_O2INTOL, calcTbl$Pred_O2INTOL)
calcTbl$Met_O2INTOL <- standardize(calcTbl$Met_O2INTOL, means["Met_O2INTOL",])
calcTbl$Met_O2INTOL <- customScale(calcTbl$Met_O2INTOL, means["Met_O2INTOL",])
```

```
calcTbl$Met_RHPAR <- logDiff(calcTbl$Obs_RHPAR, calcTbl$Pred_RHPAR)
calcTbl$Met_RHPAR <- standardize(calcTbl$Met_RHPAR, means["Met_RHPAR",])
calcTbl$Met_RHPAR <- customScale(calcTbl$Met_RHPAR, means["Met_RHPAR",])
```

```
calcTbl$IndexAll_Mean <- (calcTbl$Met_RHPAR + calcTbl$Met_O2INTOL) / 2
```

validate results

Compare the calculated results with the values in the common metric table from the CrossGIG report.

import data

```
#Common_Metrics
metr <- readRDS(paste0(datadir,"tables/","Common_Metrics.rda"))
```

load helper functions

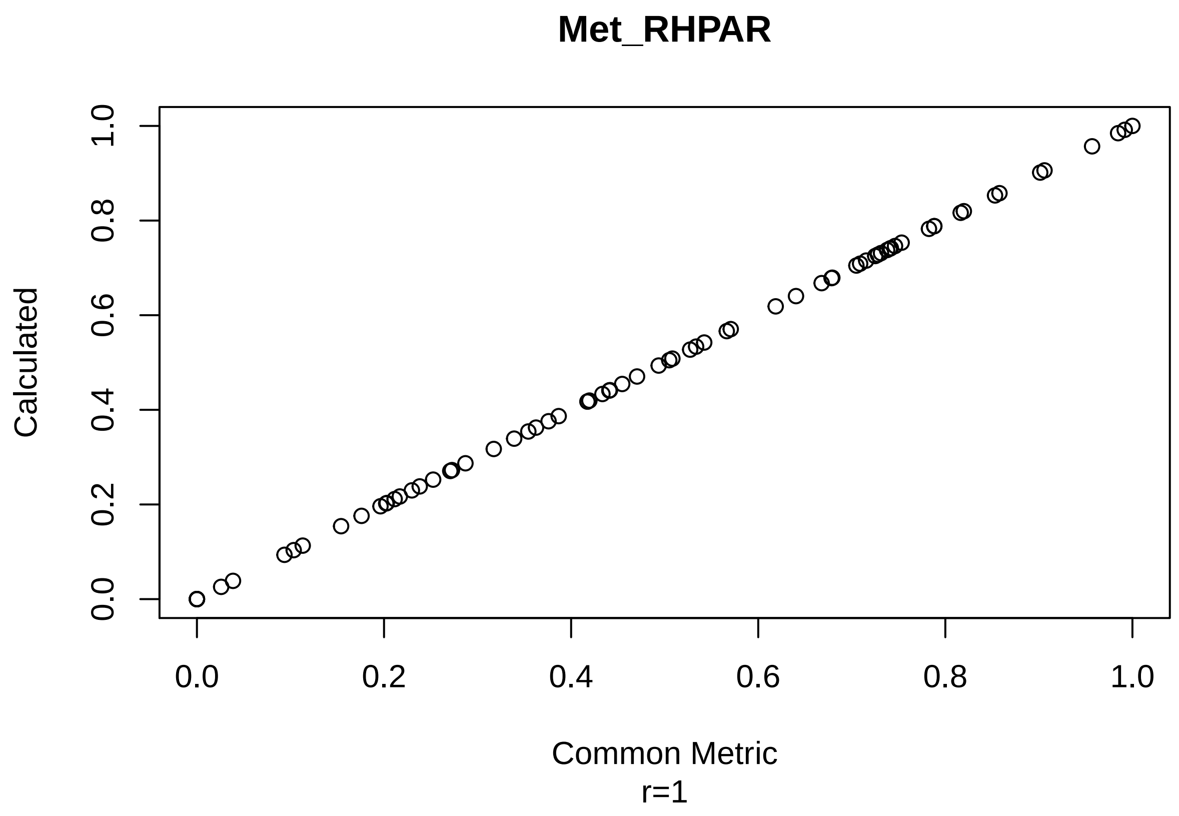
```r
#create plot fun
plotCor <- function(val, metr, calcTbl){
  test<- merge(metr[,c("Site_code",val)],
               calcTbl[,c("Site_code",val)],
               by="Site_code")
plot(test[,paste0(val,".x")], test[,paste0(val,".y")],
     main = val,
     sub = paste0("r=",
                  cor(test[,paste0(val,".x")],
                      test[,paste0(val,".y")],
                      use = "pairwise.complete.obs")
                  ),
     xlab = "Common Metric",
     ylab = "Calculated")
}

#show differing sites at a rounding treshold
different_sites <- function(val, metr, calcTbl, digits = 6){
  test<- merge(metr[,c("Site_code",val)],
               calcTbl[,c("Site_code",val)],
               by="Site_code",
               suffixes = c(".metr", ".calc"))
  v1 <- paste0(val,".metr")
  v2 <- paste0(val,".calc")
  test$equal <- round(test[, v1], digits) == round(test[,v2], digits)
  test[test$equal == F, ]
}
```

Check Calculations

RHPAR metric

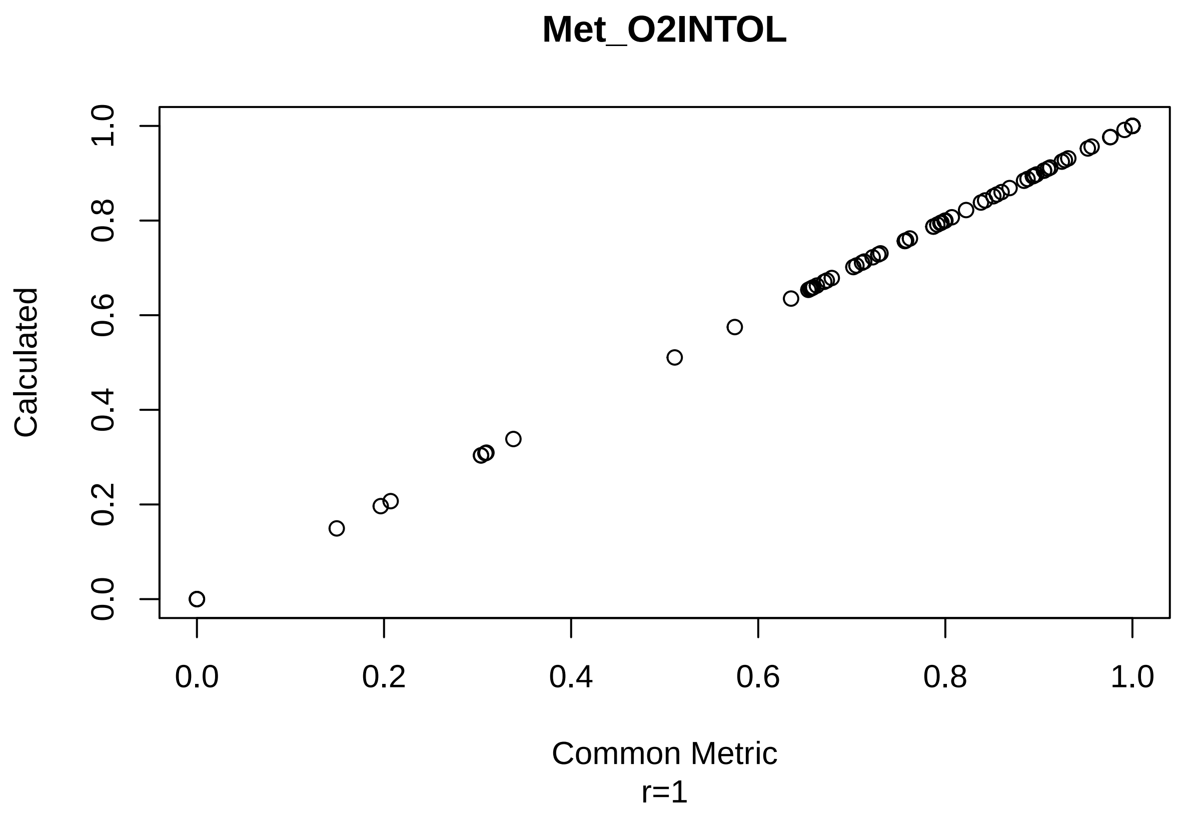
```r
plotCor("Met_RHPAR", metr, calcTbl)
```



The observed and predicted values also match perfectly (data not shown).

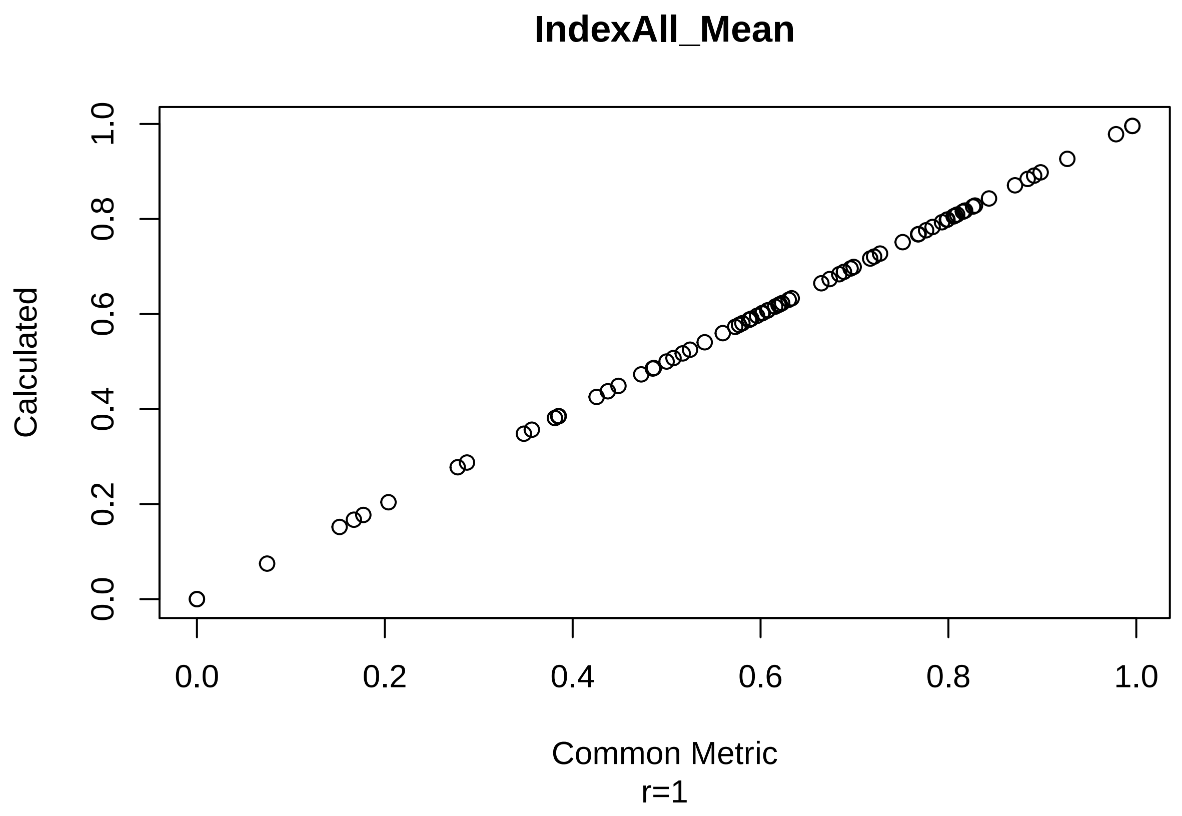
O2Intol metric

```
plotCor("Met_O2INTOL", metr, calcTbl)
```



The ICM metric

```
plotCor("IndexAll_Mean", metr, calcTbl)
```



So the correlation between the calculated and original table values is perfect (i.e calculation steps are correct).

Conclusions

There is a considerable ammount of data in the CrossGIG database. The above scripts shows what is actually needed to to perform the common metric calcuations (calculate the new fish index). Below it is highlighted what fishing/site data is absolutely needed for the calcualtions and what extra data is needed.

Required field data

To do these calcualations for any site (even those not present in the dataset) 2 tables are needed. The first one is the **Fishing_Occasion** table with following columns:

- **Site_code** Unique identifier of the site's fissing occasion
- **E_catchsize** Size of catchment upstream of the sampling site (km^2)
- **E_water_source_type** Glacial-nival dominant or pluvial dominant, based on the hydrograph of the river close to the sampling site.
- **E_floodplain** Presence of a former floodplain: yes, no (e.g. significant area of adjacent landscape flooded at least every 10 years), (data source: old maps, reports, expert judgement). Situation before any major human control of river
- **E_geomorphological_type** 4 categories: naturally constraint without mobility (riverbed is fixed), braided, sinuous and meandering. Situation before any major human control of river bed!

- **E_natural_sediment** 3 categories: fine (organic-silt-sand), medium (gravel-pebble-cobble), large (boulder-rock). Situation before major changes of sediment conditions, always for the dominating substrate! For large rivers, consider dominant sediment in the potamic zone with weak to medium water depths.
- **E_distsource** Distance from source (km) to the sampling site measured along the river. In the case of multiple sources, measurement shall be made to the most distant upstream source
- **E_slope** Given as slope of streambed along stream (m/km; ‰). The slope is the drop of altitude divided by stream segment length. The stream segment should be as close as possible to 1 km for small streams, 5 km for intermediate streams and at least 10 km for large streams.
- **E_temp_jul** Mean July air temperature at the site (measured for at least 10 years). Given in degrees Celsius (°C).

It would be also sugessed to have a **SiteID** and or additional fields (i.e River_name, Site_name , Date) to make time series data for each site, but it is not strictly needed for metric calcualtions.

For some strange reason the calcualtions are not using the **E_fishedarea** (Area of the section that has been definitely sampled given in m^2.) I would suggest to add this as well to the table just in case allthough it is not used in the current calcualtions.

The second needed table is the **Catch** table. It contains with following columns:

- **Site_code** unique identifier of the site's fissing occasion
- **Species** Scientific name of species caught (**NB!** *for the species that have multipe name types the names must be in the form used in this script or the script should be modified to include the other name formats.*)
- **Run1_number_all** number of all caught individuals of this species (including 0+) in run 1

Required extra data

In addition to the tables the predicted metric calculation needs this script and glm models in the directory "data/models"