Supplementary Materials and Methods

This supplementary materials and methods file contains the scripts and code used for the bioinformatic analyses of the RNA-seq data and for statistical analyses of the phenotypic data shown in Rodrigues et al. (2023; "Germline proliferation trades off with lipid metabolism in Drosophila", published Evolution Letters).'

A. Statistical analyses of phenotypic data

1. No effect of the fertile genotypes on fertility (Figure 1)

```
# Load the package
library(car)
# Load the data
dat1Fec <- read.table(file = "1.1.data_Fecundity_OnlyFertile.txt", header = T)
# Rank the data
dat1Fec$rankA <- rank(dat1Fec$Fecundity, ties.method = c("average"))
# Type II ANOVA
dat1Fec.lmRA <- lm(rankA ~ Genotype * Diet, data = dat1Fec)
Anova(dat1Fec.lmRA, type = 2)
```

2. Effects of treatments (germline ablation; dietary yeast) on female fecundity (Figure 1)

```
# Load the package
library(car)
# Load the data
dat12Fec <- read.table(file = "1.2.data_Fecundity_all.txt", header = T)
# Rank the data
dat12Fec$rank1 <- rank(dat12Fec$Fecundity, ties.method = c("average"))
# Type II ANOVA
dat12Fec.lmR1 <- lm(rank1 ~ Genotype * Diet, data = dat12Fec)
Anova(dat12Fec.lmR1, type = 2)
```

3. Effect of age and reproduction on fat content (Figure 8)

```
# Load the package
library(car)
# Load the data
dat2Fat <- read.table(file = "2.data_Fat.txt", header = T)
# Type II ANOVA
dat2Fat.lm <- lm(Fat ~ Genotype * Age, data = dat2Fat)
Anova(dat2Fat.lm, type = 2)
```

4. No effect of the cross directions on the fat content (ug fat / mg fly) of germline-less F1 flies (see Materials and Methods section)

```
# Load the package
library(car)
# Load the data for BN vs NB crosses
dat3Fat <- read.table(file = "3.data_fat_FemalesFed_BNNB.txt", header = T)
# Type II anova for BN vs NB crosses
dat3Fat.lm <- lm(Fat ~ Genotype, data = dat3Fat)
Anova(dat3Fat.lm, type = 2)

# Load the data for BNCyo vs NBCyo crosses
dat4Fat <- read.table(file = "4.data_fat_FemalesFed_BNCyoNBCyo.txt", header = T)
# Type II anova for BN vs NB crosses
dat4Fat.lm <- lm(Fat ~ Genotype, data = dat4Fat)
Anova(dat4Fat.lm, type = 2)
```

B. RNAseq analyses: Trimming, mapping and obtaining the gene count data file

1. Trim raw FASTQ reads with Q-score below 35 with cutadapt v1.15

```
cutadapt \
-q 35 \
-o /path/Trimming/SAMPLE_1_trimmed.fastq.gz \
/path/SAMPLE_1.fastq.gz
```

2. Run FastQC v0.11.7 in the trimmed data

```
fastqc /path/SAMPLE_1_trimmed.fastq.gz \
--outdir=/path/Trimmed_FastQC
```

3. Map the reads on the reference D. melanogaster transcriptome r6.17 with Kalisto v0.43.0

```
# Download the reference transcriptome
curl -O ftp://ftp.flybase.net/genomes/Drosophila_melanogaster/dmel_r6.17_FB2017_04/
fasta/dmel-all-CDS-r6.17.fasta.gz

# Download reference gtf file
curl -O ftp://ftp.flybase.net/genomes/Drosophila_melanogaster/dmel_r6.17_FB2017_04/
gtf/dmel-all-r6.17.gtf.gz

# Index the transcriptome
kallisto index \
 -i /path/reference/dmel-all-CDS-r6.17.idx \
/path/reference/dmel-all-CDS-r6.17.fasta.gz

# Map on the reference with Kallisto
kallisto quant \
-i /path/reference/dmel-all-CDS-r6.17.idx \
-o /path/Mapping/SAMPLE \
--fragment-length 100 \
--sd  \
--bootstrap-samples 100 \
/path/Trimming/SAMPLE_trimmed1.fastq.gz /path/Trimming/SAMPLE_trimmed2.fastq.gz\
-t 4
```

4. Merge the count data of all libraries obtained from Kallisto v0.43.0

```
python /path/scripts/mergeKallisto.py \
--directory /path/Mapping/ \
--gtf /path/reference//dmel-all-r6.17.gtf.gz \
--output /path/Counts/RawReadsCounts.txt
```

C. RNAseq analyses: Identifiying differentially expressed genes with Bioconductor R packages edgeR v3.20.8 and Limma-voom v3.34.7. Example with the data subset "fat body, young", the same procedure was applied for the four data subsets: (1) fat body, young; (2) fat body, old; (3) head, young; and (4) head, old.

1. Create DGE object using edgeR

```
# Load the package
library(limma)
```

```
library(edgeR)
# Load the data
rawALLCounts <- read.table("/path/Counts/RawReadsCounts.txt", header = T)
FullDGE_ALLdata <- DGEList(rawALLCounts, group = names(rawALLCounts))
```

2. Add factors to the DGE object

```
# Define the factors of the samples in the same order as in the counts table
Fecundity = factor(c("S", "S", "S", "S", "S", "S", "S", "S", "F", "F", "F", "F",
    "F", "F", "F", "F", "F", "S", "S", "S", "S", "S", "S", "S", "S", "F", "F", "F",
    "F", "F", "F", "F", "F", "F", "S", "S", "S", "S", "S", "S", "S", "S", "F", "F",
    "F", "F", "F", "F", "F", "F", "S", "S", "S", "S", "S", "S", "S", "S", "F", "F",
    "F", "F", "F", "F", "F", "F"))

Tissue = factor(c("Fb", "Fb", "Fb", "Fb", "Fb", "Fb", "Fb", "Fb", "Fb", "Fb", "Fb",
    "Fb", "Fb", "Fb", "Fb", "Fb", "Fb", "B", "B", "B", "B", "B", "B", "B", "B", "B",
    "B", "B", "B", "B", "B", "B", "B", "B", "Fb", "Fb", "Fb", "Fb", "Fb", "Fb", "Fb",
    "Fb", "Fb", "Fb", "Fb", "Fb", "Fb", "Fb", "Fb", "Fb", "Fb", "B", "B", "B", "B", "B",
    "B", "B", "B", "B", "B", "B", "B", "B", "B", "B", "B"))

Food = factor(c("2Y", "2Y", "4Y", "4Y", "8Y", "8Y", "12Y", "12Y", "2Y", "2Y", "2Y",
    "4Y", "4Y", "8Y", "8Y", "12Y", "12Y", "2Y", "2Y", "4Y", "4Y", "8Y", "8Y", "12Y",
    "12Y", "2Y", "2Y", "2Y", "4Y", "4Y", "8Y", "8Y", "12Y", "12Y", "2Y", "2Y", "4Y",
    "4Y", "8Y", "8Y", "12Y", "12Y", "2Y", "2Y", "4Y", "4Y", "8Y", "8Y", "12Y", "12Y",
    "2Y", "2Y", "4Y", "4Y", "8Y", "8Y", "12Y", "12Y", "2Y", "2Y", "4Y", "4Y", "8Y",
    "8Y", "12Y", "12Y"))

Age = factor(c("old", "old", "old", "old", "old", "old", "old", "old", "old", "old",
    "old", "old", "old", "old", "old", "old", "old", "old", "old", "old", "old", "old",
    "old", "old", "old", "old", "old", "old", "old", "old", "old", "old", "old", "old",
    "old", "old", "yng", "yng", "yng", "yng", "yng", "yng", "yng", "yng", "yng", "yng",
    "yng", "yng", "yng", "yng", "yng", "yng", "yng", "yng", "yng", "yng", "yng", "yng",
    "yng", "yng", "yng", "yng", "yng", "yng", "yng", "yng", "yng", "yng", "yng", "yng",
    "yng"))

Rep = factor(c("1", "2", "1", "2", "1", "2", "1", "2", "1", "2", "3", "1", "2", "1",
    "2", "1", "2", "1", "2", "1", "2", "1", "2", "1", "2", "1", "2", "3", "1", "2",
    "1", "2", "1", "2", "1", "2", "1", "2", "1", "2", "1", "2", "1", "2", "1", "2",
    "1", "2", "1", "2", "1", "2", "1", "2", "1", "2", "1", "2", "1", "2", "1", "2",
    "1", "2", "1", "2"))

# Add factors to the DGE object
FullDGE_ALLdata$samples$Fecundity <- Fecundity
FullDGE_ALLdata$samples$Age <- Age
FullDGE_ALLdata$samples$Tissue <- Tissue
FullDGE_ALLdata$samples$Food <- Food
FullDGE_ALLdata$samples$Rep <- Rep
```

3. Transform counts (Counts per million: CPM) and normalize the counts using the TMM method in edgeR v3.20.8

```
FullDGE_ALLdata <- calcNormFactors(FullDGE_ALLdata, method = "TMM")
cpmFullDGE_ALLdata <- cpm(FullDGE_ALLdata, log = T)
```

4. Remove genes covered by less than 2 read per million in at least twelve samples

```
keep <- rowSums(cpmFullDGE_ALLdata >= 2) >= 12
FullDGE.trimmedALL <- FullDGE_ALLdata[keep, ]
cpmFullDGE.trimmedALL <- cpmFullDGE_ALLdata[keep, ]
```

5. Principal Components Analysis (Figure 2)

```
# Load the package
library(RColorBrewer)
library(ggplot2)
library(ggfortify)
library(ggpubr)

# Transpose matrix
newcpm = t(cpmFullDGE.trimmedALL)
# Define colors, size and symbols
col.Food = cex = c(10, 2, 4, 8)[Food]
col.F = c("red", "blue")[Fecundity]
col.TA = pch = c(19, 15, 1, 0)[Tissue.Age]
# Produce and plot PCA
PCA <- prcomp(newcpm)
autoplot(PCA, col = col.F, shape = col.TA, size = col.Food, title = "Tissue") +
    theme_bw() + theme(panel.grid.major = element_blank()) +
    theme(panel.grid.minor = element_blank()) + theme(plot.margin = unit(c(0.6,
    5, 1.5, 0.6), "cm"))
```

6. Subsample “fat body, young”

```
FullDGE.trimmed_FB_yng <- FullDGE.trimmedALL[, grepl(".yng.Fb.",
    colnames(FullDGE.trimmedALL))]
SampleNames_FB_yng = interaction(Fecundity, Food, Rep)
```

7. Factorial analyses with the Bioconductor package ‘limma-voom’ v3.34.7

```
library("limma")

# Make design table
MR <- paste(Fecundity, Food, sep = ".")
MR <- factor(MR, levels = c("S.2Y", "S.4Y", "S.8Y", "S.12Y",
    "F.2Y", "F.4Y", "F.8Y", "F.12Y"))
design <- model.matrix(~0 + MR)
colnames(design) <- levels(MR)
rownames(design) <- SampleNames_FB_yng
design

# Run model
voomFullDGE_Fb_yng <- voom(FullDGE.trimmed_FB_yng, design, plot = T)
corfit <- duplicateCorrelation(voomFullDGE_Fb_yng, design, block = Rep)
fit <- lmFit(voomFullDGE_Fb_yng, design, block = Rep, correlation = corfit$consensus)

# Factorial analysis for genotype
cont.gen <- makeContrasts(SvsF = (S.2Y + S.4Y + S.8Y + S.12Y)/4 -
    (F.2Y + F.4Y + F.8Y + F.12Y)/4, levels = design)
# Global F-test across pairwise comparisons
fit.gen <- contrasts.fit(fit, cont.gen)
efit.gen <- eBayes(fit.gen)
# Extract F-statistic for genotype from eBayes with the
# Benjamini-Hochberg correction.
efit.gen.F = data.frame(F = efit.gen$F, Pval = efit.gen$F.p.value,
    Pval.BH = p.adjust(efit.gen$F.p.value, method = "BH"))
rownames(efit.gen.F) = rownames(efit.gen)
# Export only significant genes
efit.gen.F2 <- subset(efit.gen.F, Pval.BH < 0.05)

# Factorial analysis for diet
cont.diet <- makeContrasts(`2Yvs4Y` = ((S.2Y + F.2Y) - (S.4Y +
    F.4Y))/4, `2Yvs8Y` = ((S.2Y + F.2Y) - (S.8Y + F.8Y))/4, `2Yvs12Y` = ((S.2Y +
    F.2Y) - (S.12Y + F.12Y))/4, `4Yvs8Y` = ((S.4Y + F.4Y) - (S.8Y +
    F.8Y))/4, `4Yvs12Y` = ((S.4Y + F.4Y) - (S.12Y + F.12Y))/4,
    `8Yvs12` = ((S.8Y + F.8Y) - (S.12Y + F.12Y))/4, levels = design)


# Global F-test across pairwise comparisons
fit.diet <- contrasts.fit(fit, cont.diet)
efit.diet <- eBayes(fit.diet)
# Extract F-statistic for diet from eBayes with the
# Benjamini-Hochberg correction.
efit.diet.F = data.frame(F = efit.diet$F, Pval = efit.diet$F.p.value,
    Pval.BH = p.adjust(efit.diet$F.p.value, method = "BH"))
rownames(efit.diet.F) = rownames(efit.diet)
# Export only significant genes
efit.diet.F2 <- subset(efit.diet.F, Pval.BH < 0.05)

# Factorial analysis for the interaction genotype * diet
cont.int <- makeContrasts(S2vsF4 = (S.2Y - S.4Y) - (F.2Y - F.4Y),
    S2vsF8 = (S.2Y - S.8Y) - (F.2Y - F.8Y), S2vsF12 = (S.2Y -
        S.12Y) - (F.2Y - F.12Y), S4vsF8 = (S.4Y - S.8Y) - (F.4Y -
```

```
        F.8Y), S4vsF12 = (S.4Y - S.12Y) - (F.4Y - F.12Y), S8vsF12 = (S.8Y -
        S.12Y) - (F.8Y - F.12Y), levels = design)
# Global F-test across pairwise comparisons
fit.int <- contrasts.fit(fit, cont.int)
efit.int <- eBayes(fit.int)
# Extract F-statistic for the interaction from eBayes with
# the Benjamini-Hochberg correction.
efit.int.F = data.frame(F = efit.int$F, Pval = efit.int$F.p.value,
    Pval.BH = p.adjust(efit.int$F.p.value, method = "BH"))
rownames(efit.int.F) = rownames(efit.int)
# Export only significant genes
efit.int.F2 <- subset(efit.int.F, Pval.BH < 0.05)
```

D. RNAseq analyses: pathway and Go-term analyses. Example with the data subset "fat body, young", the same procedure was applied for the four data subsets: (1) fat body, young; (2) fat body, old; (3) head, young; and (4) head, old.

1. Pathway analyses with the Bioconductor package 'ReactomePA' v1.28.0

```
library(AnnotationDbi)
library(org.Dm.eg.db)
library(ReactomePA)

## Genotype Convert the gene names
efit.gen.F2.adj <- as.data.frame(efit.gen.F2)
efit.gen.F2.adj$symbol <- mapIds(org.Dm.eg.db, keys = row.names(efit.gen.F2.adj),
    column = "SYMBOL", keytype = "FLYBASE", multiVals = "first")

efit.gen.F2.adj$entrez <- mapIds(org.Dm.eg.db, keys = row.names(efit.gen.F2.adj),
    column = "ENTREZID", keytype = "FLYBASE", multiVals = "first")

efit.gen.F2.adj$name = mapIds(org.Dm.eg.db, keys = row.names(efit.gen.F2.adj),
    column = "GENENAME", keytype = "FLYBASE", multiVals = "first")

# Pathway analyses
Genelist.gen <- efit.gen.F2.adj$entrez
Fb_yng_GSE_gen <- gsePathway(Genelist.gen, organism = "fly",
    nPerm = 10000, pvalueCutoff = 0.2, pAdjustMethod = "BH",
    verbose = FALSE)

## Diet Convert the gene names
efit.diet.F2.adj <- as.data.frame(efit.diet.F2)
efit.diet.F2.adj$symbol <- mapIds(org.Dm.eg.db, keys = row.names(efit.diet.F2.adj),
    column = "SYMBOL", keytype = "FLYBASE", multiVals = "first")

efit.diet.F2.adj$entrez <- mapIds(org.Dm.eg.db, keys = row.names(efit.diet.F2.adj),
```

```
    column = "ENTREZID", keytype = "FLYBASE", multiVals = "first")

efit.diet.F2.adj$name = mapIds(org.Dm.eg.db, keys = row.names(efit.diet.F2.adj),
    column = "GENENAME", keytype = "FLYBASE", multiVals = "first")
# Pathway analyses
Genelist.diet <- efit.diet.F2.adj$entrez
Fb_yng_path_diet <- enrichPathway(gene = Genelist.diet, organism = "fly",
    pAdjustMethod = "BH", pvalueCutoff = 0.05, readable = TRUE)

## Interaction genotype * diet Convert the gene names
efit.int.F2.adj <- as.data.frame(efit.int.F2)
efit.int.F2.adj$symbol <- mapIds(org.Dm.eg.db, keys = row.names(efit.int.F2.adj),
    column = "SYMBOL", keytype = "FLYBASE", multiVals = "first")

efit.int.F2.adj$entrez <- mapIds(org.Dm.eg.db, keys = row.names(efit.int.F2.adj),
    column = "ENTREZID", keytype = "FLYBASE", multiVals = "first")

efit.int.F2.adj$name = mapIds(org.Dm.eg.db, keys = row.names(efit.int.F2.adj),
    column = "GENENAME", keytype = "FLYBASE", multiVals = "first")
# Pathway analyses
Genelist.int <- efit.int.F2.adj$entrez
Fb_yng_path_interaction <- enrichPathway(gene = Genelist.int,
    organism = "fly", pAdjustMethod = "BH", pvalueCutoff = 0.05,
    readable = TRUE)
```

2. Go-term analyses with the Bioconductor package 'topGo' v2.16.0.

```
library(topGO)

# Genotype
TotalAnnot <- readMappings(file = "Allgenes_AllAnnotatedGo.txt")
geneNames <- names(TotalAnnot)
FYG <- names(readMappings("Sign_Fb_Yng_Genotype.txt"))
FYGb <- factor(as.integer(geneNames %in% FYG))
names(FYGb) <- geneNames

GOdataFYG_BP <- new("topGOdata", ontology = "BP", allGenes = FYGb,
    nodeSize = 5, annot = annFUN.gene2GO, gene2GO = TotalAnnot)
GOdataFYG_BP1 <- runTest(GOdataFYG_BP, algorithm = "weight01",
    statistic = "fisher")
GOdataFYG_BP_orderWeight01F <- GenTable(GOdataFYG_BP, weight01Fisher = GOdataFYG_BP1,
    orderBy = "weight01Fisher", ranksOf = "weight01Fisher", topNodes = 3260)

# Diet
TotalAnnot <- readMappings(file = "Allgenes_AllAnnotatedGo.txt")
geneNames <- names(TotalAnnot)
FYD <- names(readMappings("Sign_Fb_Yng_Diet.txt"))
FYDb <- factor(as.integer(geneNames %in% FYD))
names(FYDb) <- geneNames

GOdataFYD_BP <- new("topGOdata", ontology = "BP", allGenes = FYDb,
```

```
    nodeSize = 5, annot = annFUN.gene2GO, gene2GO = TotalAnnot)
GOdataFYD_BP1 <- runTest(GOdataFYD_BP, algorithm = "weight01",
    statistic = "fisher")
GOdataFYD_BP_orderWeight01F <- GenTable(GOdataFYD_BP, weight01Fisher = GOdataFYD_BP1,
    orderBy = "weight01Fisher", ranksOf = "weight01Fisher", topNodes = 3260)

# Interaction genotype * diet
TotalAnnot <- readMappings(file = "Allgenes_AllAnnotatedGo.txt")
geneNames <- names(TotalAnnot)
FYI <- names(readMappings("Sign_Fb_Yng_Int.txt"))
FYIb <- factor(as.integer(geneNames %in% FYI))
names(FYIb) <- geneNames

GOdataFYI_BP <- new("topGOdata", ontology = "BP", allGenes = FYIb,
    nodeSize = 5, annot = annFUN.gene2GO, gene2GO = TotalAnnot)
GOdataFYI_BP1 <- runTest(GOdataFYI_BP, algorithm = "weight01",
    statistic = "fisher")
GOdataFYI_BP_orderWeight01F <- GenTable(GOdataFYI_BP, weight01Fisher = GOdataFYI_BP1,
    orderBy = "weight01Fisher", ranksOf = "weight01Fisher", topNodes = 3260)
```