SUPPLEMENTARY METHODS

The following R code snippets were used for re-analyzing scRNA-Seq data. All the scripts are meant to be run from a terminal in a *nix system. We have tested them only on Debian 10. The workflow is divided in four self-contained steps to allow for QC checking and manual analysis of cluster markers. Directory trees and graphs are produced automatically, but the user should become familiar with the scripts to understand their use and the arguments needed.

Pre-processing. This step is implemented using parallel computation, assuming >100 Gb of RAM are available. Parallelization can be turned off by setting “samples.number” to 1.

```
#!/usr/bin/env Rscript

#-------------------------Check requirements-----------------

req_packs <- c("parallel", "Rsubread", "scPipe")

for (i in req_packs){
        if( ! require( i, character.only = TRUE ) ){
                BiocManager::install(i)
                library(i)
        }
}

#--------------------------Definitions----------------------

arg <- commandArgs(TRUE)

#[1] define chemistry for 10x; 1 > v2        2 > v3
#[2] index location
#[3] GTF file to use
#[4] plain text file with a list of samples

index_folder <- c(arg[2]) # Folder containing the genome index for Rsubread
gtf_file <- c(arg[3])     # GTF annotation file
sample.list <- c(arg[4])  # simple list of sample names in a plain text file

## Define read structure

if (arg[1] == 1) {
print("Using 10X v2")
read_struc <- list(
        bs1 = -1,   # barcode start position in fq_R1, -1 indicates no barcode
        bl1 = 0,    # barcode length in fq_R1, 0 since no barcode present
        bs2 = 0,    # barcode start position in fq_R2
        bl2 = 16,   # barcode length in fq_R2
        us = 16,    # UMI start position in fq_R2
        ul = 10     # UMI length
```

```
        )
} else if (arg[1] == 2) {
print("Using 10X v3")
read_struc <- list(
        bs1 = -1,
        bl1 = 0,
        bs2 = 0,
        bl2 = 16,
        us = 16,
        ul = 12
        )
} else {
print ("ERROR. Please define 10X Chromium version")
}

# output folder where files will be generated
out_dir = "output"

# output file names
trimmed_fastq = file.path(out_dir, "trimmed.fastq.gz")
aligned_bam = file.path(out_dir, "aligned.bam")
mapped_bam = file.path(out_dir, "aligned.mapped.bam")

# filter sets for trimming barcodes
filt_set <- list(
        rmlow = T,     #DEFAULT IS T
        rmN = T,       #DEFAULY IS T
        minq = 20,     #DEFAULT IS 20
        numbq = 2      #DEFAULT IS 2
        )

# set number of threads to be used
sample.list <- read.table(sample.list, header = F, sep = "\t")
sample.list <- as.character(sample.list$V1)
if (length(sample.list) > 5) {
        samples.number <- c(5)
} else {
        samples.number <- length(sample.list)
}
# Aligning more than 5 samples simultaneously will use > 90 Gb of memory.
# Rsubread does not support sharing the index in memory for all the child processes.
# Each process needs to load its own index.

#--------------------Function-------------------------

preprocess <- function(x) {
        setwd(paste0("./", x))
        name_files <- Sys.glob("*1.fastq.gz")
        name_files <- gsub("_1.fastq.gz", "", name_files)
```

```
        ## trim barcodes
        sc_trim_barcode(
                outfq = trimmed_fastq,
                r1 = paste0(name_files, "_2.fastq.gz"),
                r2 = paste0(name_files, "_1.fastq.gz"),
                read_structure = read_struc,
                filter_settings = filt_set
                )
        ## align
        align(
                index = paste0(index_folder, "/", "mouse_index"),
                readfile1 = trimmed_fastq,
                output_file = aligned_bam,
                type = "rna",
                nthreads = floor(30/samples.number)
                )
        ## detect barcodes
        sc_detect_bc(
                infq = trimmed_fastq,
                outcsv = "barcode_anno.csv",   ## barcode annotation output file name
                bc_len = read_struc$bl2,
                max_reads = 5000000,        # only process first 5 million reads
                min_count = 100             # discard barcodes with fewer than 100 hits
                )
        ## produce count matrix
        sc_count_aligned_bam(
                inbam = aligned_bam,
                outbam = mapped_bam,
                annofn = gtf_file,
                bc_len = read_struc$bl2,
                UMI_len = read_struc$ul,
                outdir = out_dir,
                bc_anno = "barcode_anno.csv"
                )
}

#----------------------Pre-process-----------------------

mclapply(sample.list,
        preprocess,
        mc.preschedule = F,
        mc.cores = samples.number)

#--------------------EPILOGUE--------------------------

#Output the version of the software used.
sink("session_info.pre")
sessionInfo()
sink()
```

Post-process. This will remove outlier cells and construct Seurat objects for each sample in parallel. Then, datasets are integrated using the SCTransform procedure. Low-dimensional projections are generated in UMAP and PHATE space. Finally, clustering and marker detection are performed.

```
#!/usr/bin/env Rscript

#-------------------------Check requirements-----------------

req_packs <- c("SingleCellExperiment", "scPipe", "Seurat", "sctransform", "phateR", "viridis",
"MAST", "AnnotationDbi", "EnsDb.Mmusculus.v79", "dplyr")

for (i in req_packs){
        if( ! require( i, character.only = TRUE ) ){
                BiocManager::install(i)
                library(i)
        }
}

#-------------------------Variables----------------------

arg <- commandArgs(TRUE)

sample.table <- c(arg[1]) # matrix with sample names and groups

sample.table <- read.table(sample.table, header = T, sep = "\t")
# rename rows to obtain a sample's group by calling the sample
row.names(sample.table) <- sample.table$sample
sample.list <- as.character(sample.table$sample)
group.list <- as.character(sample.table$group)

if (length(sample.list) > 20) {
        samples.number == 20
} else {
        samples.number <- length(sample.list)
}

#-------------------------Functions------------------------

# Function to capitalize first letter of vector elements, FROM R TUTORIAL
capwords <- function(s, strict = FALSE) {
   cap <- function(s) paste(toupper(substring(s, 1, 1)),
               {s <- substring(s, 2); if(strict) tolower(s) else s},
                        sep = "", collapse = " " )
   sapply(strsplit(s, split = " "), cap, USE.NAMES = !is.null(names(s)))
}

# Function to wrap the creation of normalized count tables for each sample with Seurat
```

```
createNorm <- function(x) {
	setwd(paste0("./", x))
	print('Creating SCE object')
	sce <- create_sce_by_dir( "./output" )
	print('SCE object created')

	dir.create("./qc")
	sce <- calculate_QC_metrics(sce)
	plot_demultiplex(sce)
	ggsave( paste0( "./qc/", x, "_", "barcode_demultiplex.pdf") )

	plot_UMI_dup(sce)
	ggsave( paste0( "./qc/", x, "_", "UMI_duplication.pdf" ) )

	sce <- detect_outlier(sce, comp = 2, type = "low")

	qcPairs <- plot_QC_pairs(sce)
	ggsave( paste0( "./qc/", x, "_", "QC_pair.pdf" ), plot = qcPairs )

	sce <- remove_outliers(sce)
	print("Outliers removed")
	## The count table is stored in counts(sce)

	seu <- CreateSeuratObject( counts = counts(sce) )
	print("Created Seurat Object")
	rm(sce)
	seu <- AddMetaData(seu, x, col.name = "sample")
	print("Added sample metadata")
	seu <- AddMetaData(seu, sample.table[x, 'group'], col.name = "treatment")
	print("Added group metadata")

	# normalize data with SCTransform()
	seu <- SCTransform(
		seu,
		assay = 'RNA',
		new.assay.name = 'SCT',
		verbose = T)

	# Perform cell cycle analysis
	seu <- CellCycleScoring(
		seu,
		s.features = s.genes$id,
		g2m.features = g2m.genes$id,
		assay = 'SCT',
		set.ident = TRUE)

	# Normalize again but this time including also the cell cycle scores,
	# notice the use of RNA assay to normalize the original count data
	seu <- SCTransform(
```

```
                seu,
                assay = 'RNA',
                new.assay.name = 'SCT',
                vars.to.regress = c('S.Score', 'G2M.Score'),
                verbose = T)
        ## Normalized data (Pearson residuals) are stored in seu[["SCT"]]@scale.data

        dir.create("./normalized")
        top10 <- head(VariableFeatures(seu), 10)
        plot1 <- VariableFeaturePlot(seu)
        plot2 <- LabelPoints(plot = plot1, points = top10, repel = TRUE)
        ggsave( paste0( "./normalized/", x, "_", "variable_genes.pdf" ) )

        return(seu)
}

#---------------------------Post-process-------------------------

# Get cell cycle genes in lowcase, transform to mouse symbols, map to GeneIDs >:-/
s.genes <- tolower(cc.genes$s.genes)
s.genes <- capwords(s.genes)
s.genes$id <- mapIds(
        EnsDb.Mmusculus.v79,
        keys=s.genes,
        column="GENEID",
        keytype="SYMBOL",
        multiVals="first")

g2m.genes <- tolower(cc.genes$g2m.genes)
g2m.genes <- capwords(g2m.genes)
g2m.genes$id <- mapIds(
        EnsDb.Mmusculus.v79,
        keys=g2m.genes,
        column="GENEID",
        keytype="SYMBOL",
        multiVals="first")

# Create list of seurat objects for each sample
seuList <- mclapply(sample.list,
                createNorm,
                mc.preschedule = F,# set to T if too many samples
                mc.cores = samples.number)

#---------------------Integrate datasets-----------------------

features <- SelectIntegrationFeatures(object.list = seuList, nfeatures = 3000)

seuList <- PrepSCTIntegration(
        object.list = seuList,
```

```
        anchor.features = features,
        verbose = T)

anchors <- FindIntegrationAnchors(
        object.list = seuList,
        normalization.method = "SCT",
        anchor.features = features,
        verbose = T)

seu_integrated <- IntegrateData(
        anchorset = anchors,
        normalization.method = "SCT",
        verbose = T)

#----------------------------Visualize----------------------------

## Use UMAP for further processing

seu_integrated <- RunPCA(
        seu_integrated,
        features = VariableFeatures(object = seu_integrated) )

seu_integrated <- FindNeighbors(seu_integrated, dims = 1:50)
seu_integrated <- FindClusters(seu_integrated, resolution = 1.5)

seu_integrated <- RunUMAP(seu_integrated, dims = 1:50)

dir.create("./clusters")
DimPlot(seu_integrated, reduction = "umap")
ggsave("./clusters/UMAP_clusters.pdf")

DimPlot(seu_integrated, reduction = "umap", group.by = "treatment")
ggsave("./clusters/UMAP_treatment.pdf")

## Next we will use phateR for dimensionality reduction; this is complementary to UMAP
## Phate uses normalized counts matrix

norm_count <- list()
norm_count$data <- t(seu_integrated[["SCT"]]@scale.data)
norm_count$treatment <- seu_integrated$treatment
norm_count$seurat_clusters <- seu_integrated$seurat_clusters
ph <- phate(norm_count$data, n.jobs = 20)

ggplot(ph) +
        geom_point(aes(PHATE1, PHATE2, color = norm_count$treatment)) +
        labs(color = norm_count$treatment)
ggsave("./clusters/phate_treatment.pdf")

ggplot(ph) +
```

```
        geom_point(aes(PHATE1, PHATE2, color = norm_count$seurat_clusters)) +
        labs(color = norm_count$seurat_clusters)
ggsave("./clusters/phate_clusters.pdf")

#----------------Get cluster markers----------------------

## Find markers

markers <- FindAllMarkers(
        seu_integrated,
        only.pos = T,
        min.pct = 0.5,
        logfc.threshold = 0.5,
        test.use = "MAST") #MAST has good FDR control and is faster than DESeq2

markers$symbol <- mapIds(
        EnsDb.Mmusculus.v79,
        keys=row.names(markers),
        column="SYMBOL",
        keytype="GENEID",
        multiVals="first")

seu_symbol <- seu_integrated
symbols <- mapIds(
        EnsDb.Mmusculus.v79,
        keys=row.names(seu_symbol[["RNA"]]@data),
        column="SYMBOL",
        keytype="GENEID",
        multiVals="first")

row.names(seu_symbol[["RNA"]]@data) <- symbols

## Draw marker plots

top1 <- markers %>% group_by(cluster) %>% top_n(n = 1, wt = avg_logFC)
FeaturePlot(seu_integrated, features = top1$gene)
ggsave("./clusters/top1.pdf", width = 20, height = 20, units = "cm", dpi = 100)

top5 <- markers %>% group_by(cluster) %>% top_n(n = 5, wt = avg_logFC)
DoHeatmap(seu_integrated, features = top5$gene) + NoLegend()
ggsave("./clusters/top5.pdf", width = 20, height = 20, units = "cm", dpi = 100)

write.table(markers, file = "./clusters/cluster_markers", sep = "\t")

#------------------Epilogue------------------

saveRDS(seu_symbol, file = "./Seurat_UMAP.rds", compress = T)
sink("session_info.post")
sessionInfo()
```

```
sink()
```

Isolate clusters of interest. A list of clusters to be further analyzed is provided and those clusters are isolated. New low-dimensional projections are generated.

```
#!/usr/bin/env Rscript

#-------------------------Check requirements-----------------

req_packs <- c("Seurat", "sctransform", "phateR", "viridis", "MAST", "AnnotationDbi",
"EnsDb.Mmusculus.v79", "dplyr", "ggplot2")

for (i in req_packs){
        if( ! require( i, character.only = TRUE ) ){
                BiocManager::install(i)
                library(i)
        }
}

#--------------------------Variables----------------------

arg <- commandArgs(TRUE)

RDSfile <- arg[1] # RDS file output by previous steps
clustOI <- arg[2] # A file with a list of Cluster Of Interest

#------------------------Re-cluster-------------------------

seu <- readRDS( file = RDSfile )

clustOI <- read.table(clustOI, header = F)
clustOI <- as.character(clustOI$V1)

seu <- subset(seu, idents = clustOI)

seu <- RunPCA(seu, features = VariableFeatures(object = seu))

seu <- FindNeighbors(seu, dims = 1:50)
seu <- FindClusters(seu, resolution = 1.0)

seu <- RunUMAP(seu, dims = 1:50)
DimPlot(seu, reduction = "umap")
ggsave("./umap_clust.pdf")
DimPlot(seu, reduction = "umap", group.by = "treatment")
ggsave("./umap_clust_treatment.pdf")

norm_count <- list()
norm_count$data <- t(seu[["SCT"]]@scale.data)
norm_count$treatment <- seu$treatment
```

```
norm_count$seurat_clusters <- seu$seurat_clusters
ph <- phate(norm_count$data, n.jobs = 10)

ggplot(ph) +
        geom_point(aes(PHATE1, PHATE2, color = norm_count$treatment)) +
        labs(color = norm_count$treatment)
ggsave("./phate_treatment.pdf")

ggplot(ph) +
        geom_point(aes(PHATE1, PHATE2, color = norm_count$seurat_clusters)) +
        labs(color = norm_count$seurat_clusters)
ggsave("./phate_clusters.pdf")

## Find markers

markers <- FindAllMarkers(
        seu,
        only.pos = T,
        min.pct = 0.5,
        logfc.threshold = 0.5,
        test.use = "MAST") #MAST has good FDR control and is faster than DESeq2

markers$symbol <- mapIds(
        EnsDb.Mmusculus.v79,
        keys=row.names(markers),
        column="SYMBOL",
        keytype="GENEID",
        multiVals="first")

## Draw marker plots

top1 <- markers %>% group_by(cluster) %>% top_n(n = 3, wt = avg_logFC)
VlnPlot(seu, features = top1$gene)
ggsave("./top3.pdf")

top5 <- markers %>% group_by(cluster) %>% top_n(n = 5, wt = avg_logFC)
DoHeatmap(seu, features = top5$gene) + NoLegend()
ggsave("./top5.pdf")

write.table(markers, file = "./cluster_markers", sep = "\t")

#------------------Epilogue------------------

saveRDS(seu, file = "./Seurat_clust_interest.rds", compress = T)
sink("session_info.clusters")
sessionInfo()
sink()
```

Graph genes of interest. In this step a list gene symbols is provided. Expression levels are graphed as violin plots, and color coded in UMAP and PHATE space. MAGIC is used for imputation before projecting into PHATE space.

```
#!/usr/bin/env Rscript

#-------------------------Check requirements-----------------

req_packs <- c("Seurat", "sctransform", "AnnotationDbi", "EnsDb.Mmusculus.v79", "dplyr",
"ggpubr", "phateR", "viridis", "Rmagic")

for (i in req_packs){
        if( ! require( i, character.only = TRUE ) ){
                BiocManager::install(i)
                library(i)
        }
}

#--------------------------Variables----------------------

arg <- commandArgs(TRUE)

RDSfile <- c(arg[1]) # A Seurat object with experiment data

gene.list <- c(arg[2]) # A plain text file with symbols of genes of interest
gene.list <- read.table(gene.list, header = T)
gene.list <- as.data.frame(gene.list)
row.names(gene.list) <- gene.list$symbol
gene.list$id <- mapIds(
        EnsDb.Mmusculus.v79,
        keys=row.names(gene.list),
        column="GENEID",
        keytype="SYMBOL",
        multiVals="first")
gene.list <- na.omit(gene.list)
row.names(gene.list) <- gene.list$id

if (length(gene.list$id) > 20) {
        samples.number <- c(20)
} else {
        samples.number <- length(gene.list$id)
}

#------------------------Function--------------------------

graphUMAP <- function(x) {
        symbol <- gene.list[x, 'symbol']
        graph <- FeaturePlot(seu, features = x)
        graph <- ggpar(graph, title = symbol)
```

```
	ggsave(
		paste0("./UMAP/", x, ".pdf"),
		width = 20,
		height = 20,
		units = "cm",
		dpi = 100)
}

graphViolin <- function(x) {
	symbol <- gene.list[x, 'symbol']
	graph <- VlnPlot(seu, features = x, split.by = "treatment")
	graph <- ggpar(graph, title = symbol)
	ggsave(
		paste0("./Violin/", x, ".pdf"),
		width = 20,
		height = 20,
		units = "cm",
		dpi = 100)
}

graphPHATE <- function(x) {
	symbol <- gene.list[x, 'symbol']
	mag <- magic(seu, genes = x)
	mag.mtx <- t(mag[['MAGIC_SCT']][x, ])
	ggplot(ph) +
		geom_point(aes(PHATE1, PHATE2, color = mag.mtx[, x])) +
		labs(color = x) +
		scale_color_viridis(option = "B") +
		ggtitle(symbol)
	ggsave(
		paste0("./PHATE/", x, ".pdf"),
		width = 20,
		height = 20,
		units = "cm",
		dpi = 100)
	rm(mag)
	rm(mag.mtx)
}

#----------------------Graph expression levels---------------

seu <- readRDS( file = RDSfile )

## extract data for PHATE
norm_count <- list()
norm_count$data <- t(seu[["SCT"]]@scale.data)
norm_count$treatment <- seu$treatment
norm_count$seurat_clusters <- seu$seurat_clusters
```

```
DefaultAssay(seu) <- "SCT"

dir.create("./UMAP")
dir.create("./Violin")
dir.create("./PHATE")

lapply(gene.list$id, graphUMAP)

lapply(gene.list$id, graphViolin)

ph <- phate(norm_count$data, n.jobs = 30)

ggplot(ph) +
        geom_point(aes(PHATE1, PHATE2, color = norm_count$treatment)) +
        labs(color = norm_count$treatment)
ggsave("./PHATE/treatment.pdf")

ggplot(ph) +
        geom_point(aes(PHATE1, PHATE2, color = norm_count$seurat_clusters)) +
        labs(color = norm_count$seurat_clusters)
ggsave("./PHATE/clusters.pdf")

mclapply(gene.list$id,
        graphPHATE,
        mc.preschedule = F,# set to T if too many samples
        mc.cores = samples.number)

#------------------Epilogue------------------

sink("session_info.graphing")
sessionInfo()
sink()
```

SUPPLEMENTARY TABLES

Table S1. Primers used for genotyping and qPCR.

Locus	Primer	Sequence
Flvcr2^{GFP}	1	TGCATATGTATGGGCTCATGTTG
	2	TGCCTAGCATCCATGCATGAG
	3	TCCCGCCAACTTGAGAAGGTC
Cre	1	GCCTGCATTACGGGTCGATGCAACCAG
	2	CTGGCAATTTCGGCTATACGTAACAGGGTG

Table S2. Antibodies used in this study.

Antibody	Host	Source	Catalog	Conc IF	Conc WB
anti aSMA	Rabbit	Abcam	ab5694	1:300	NA
anti AXIN2	Rabbit	Abcam	ab32197	NA	1:1000
anti beta-Actin	Mouse	R & D Systems	MAB8929	NA	1:5000
anti cCASP3	Rabbit	Cell Signaling Technology	9664	1:300	NA
anti CD13	Rat	Biorad	MCA2183	1:300	NA
anti CD31	Goat	R & D Systems	AF3628	1:300	NA
anti CLDN5-488	Mouse	Invitrogen	352588	1:100	NA
anti DLL4	Goat	R & D Systems	AF1389	1:300	1:1000
anti ERG	Rabbit	Abcam	ab195514	1:300	NA
anti ESM1	Goat	R & D Systems	AF1999	1:300	1:1000
anti GFP	Chicken	Abcam	ab13970	1:1000	NA
anti Goat IgG-488	Donkey	Invitrogen	A11055	1:300	NA
anti Goat IgG-555	Donkey	Invitrogen	A21432	1:300	NA
anti Goat IgG-647	Donkey	Invitrogen	A21447	1:300	NA
anti Goat IgG-HRP	Donkey	Invitrogen	A15999	NA	1:20000
anti HIF1a	Rabbit	Novus	NB100-479SS	1:300	NA
anti IBA1	Goat	Novus	NB100-1028	1:300	NA
anti KI67	Rat	eBiosciences	14-5698-82	1:300	NA
anti KLF4	Goat	R & D Systems	AF3158	NA	1:1000
anti Mouse IgG-488	Donkey	Abcam	ab150105	1:300	NA
anti Mouse IgG-HRP	Donkey	Invitrogen	A16011	NA	1:20000
anti NG2	Rabbit	Gift from Dr. William Stallcup	NA	1:300	NA
anti NRP1	Goat	R & D Systems	AF566	1:300	1:1000
anti Pimonidazole-594	Mouse	Hypoxyprobe	Mab549	1:100	NA
anti PLVAP	Rat	Serotec	MCA2539	1:300	NA
anti PU.1	Rabbit	Cell Signaling Technology	2266S	1:300	NA
anti Rabbit IgG-488	Donkey	Abcam	ab150065	1:300	NA
anti Rabbit IgG-555	Donkey	Abcam	ab150062	1:300	NA
anti Rabbit IgG-647	Donkey	Abcam	ab150063	1:300	NA
anti Rabbit IgG-HRP	Donkey	Invitrogen	A16029	NA	1:20000
anti Rat IgG-488	Donkey	Abcam	ab15013	1:300	NA
anti Rat IgG-555	Donkey	Abcam	ab150154	1:300	NA
anti Rat IgG-647	Donkey	Abcam	ab150155	1:300	NA
anti pSmad3	Rabbit	Cell Signaling Technology	9520	NA	1:1000
anti pSmad3	Rabbit	Abcam	ab52903	1:150	NA
anti Smad3	Rabbit	Epitomics	1735-1	NA	1:1000
anti SOX9	Goat	R & D Systems	AF3075	1:300	NA
anti TER119	Rat	R & D Systems	MAB1125	1:300	NA
anti VEGF	Goat	R & D Systems	AF-493-NA	1:300	NA
anti VEGFR2	Rabbit	Cell Signaling Technology	2479	1:300	1:1000
anti VEGFR3	Goat	R & D Systems	AF743	1:300	1:1000
Isolectin-B4-488	NA	Invitrogen	I21411	1:100	NA
Streptavidin-647	NA	Invitrogen	S32357	1:100	NA

SUPPLEMENTARY FIGURES

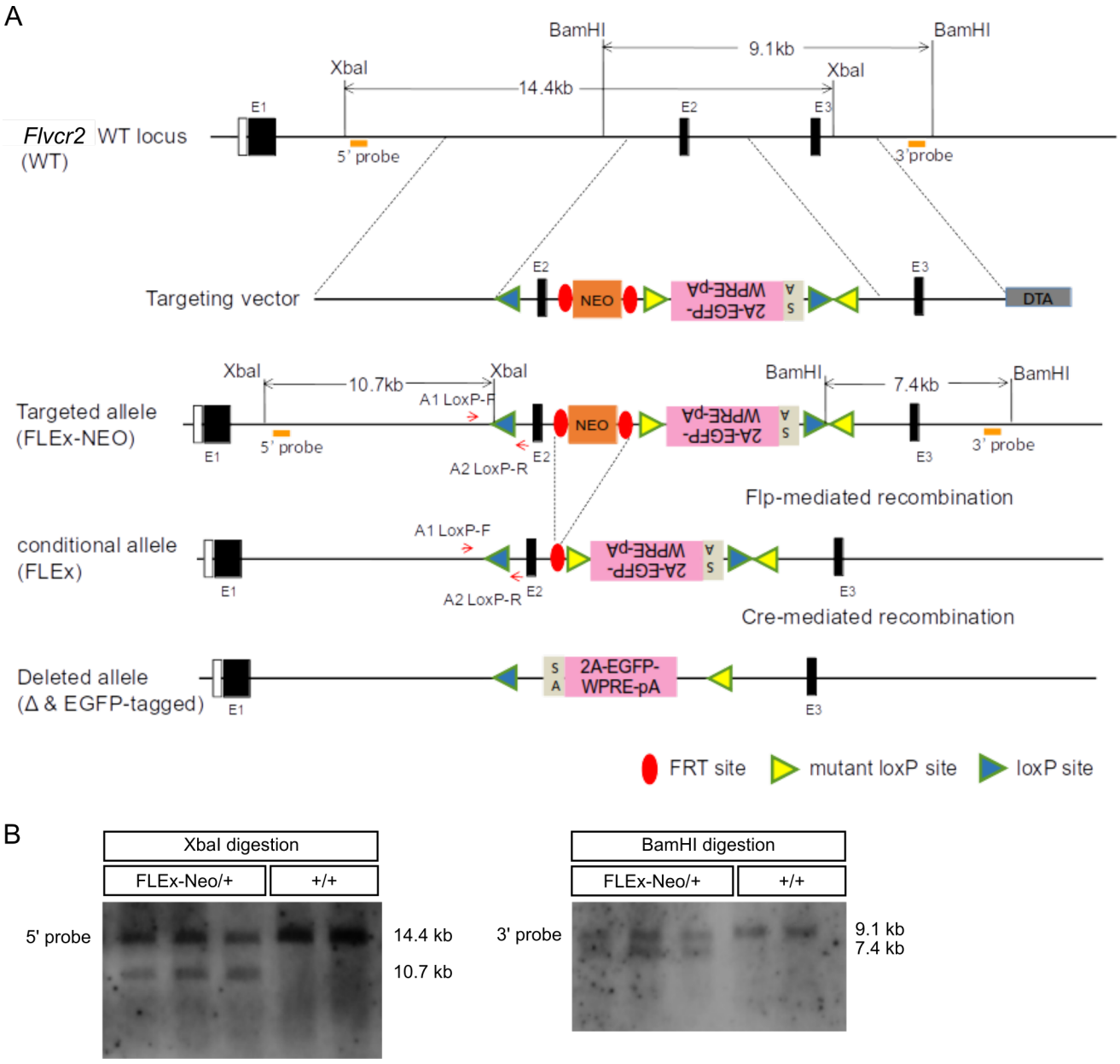

Figure S1. Targeting allele for the *Flvcr2* locus. A. The *Flvc2*^{GFP} allele was generated by homologous recombination in embryonic stem cells. Two sets of inverted loxP sites (one regular set and one mutated set) direct recombination to remove the second exon and flip eGFP in-frame. **B.** Validation of the cloned construct by Southern blot using two probes and two restriction enzymes.

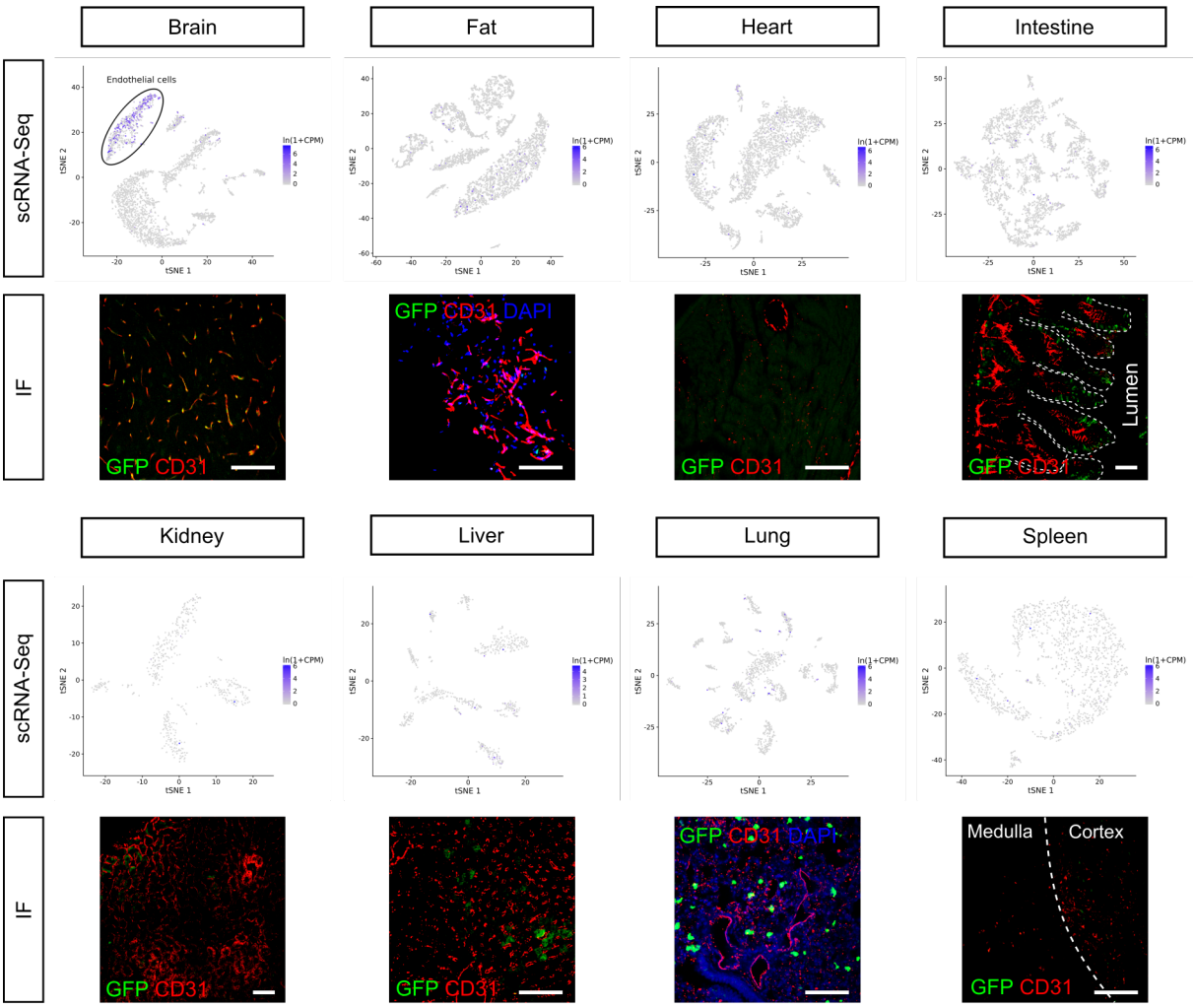

Figure S2. Expression pattern of the *Flvcr2* locus in adult mice. Expression of *Flvcr2* in different tissues of adult mice is shown by single cell RNA-Seq using data from the Tabula muris (1), and by immunodetection of GFP in adult *Flvcr2*$^{+/GFP}$ mice. N=3 for immunofluorescence. Bars: 100 µm.

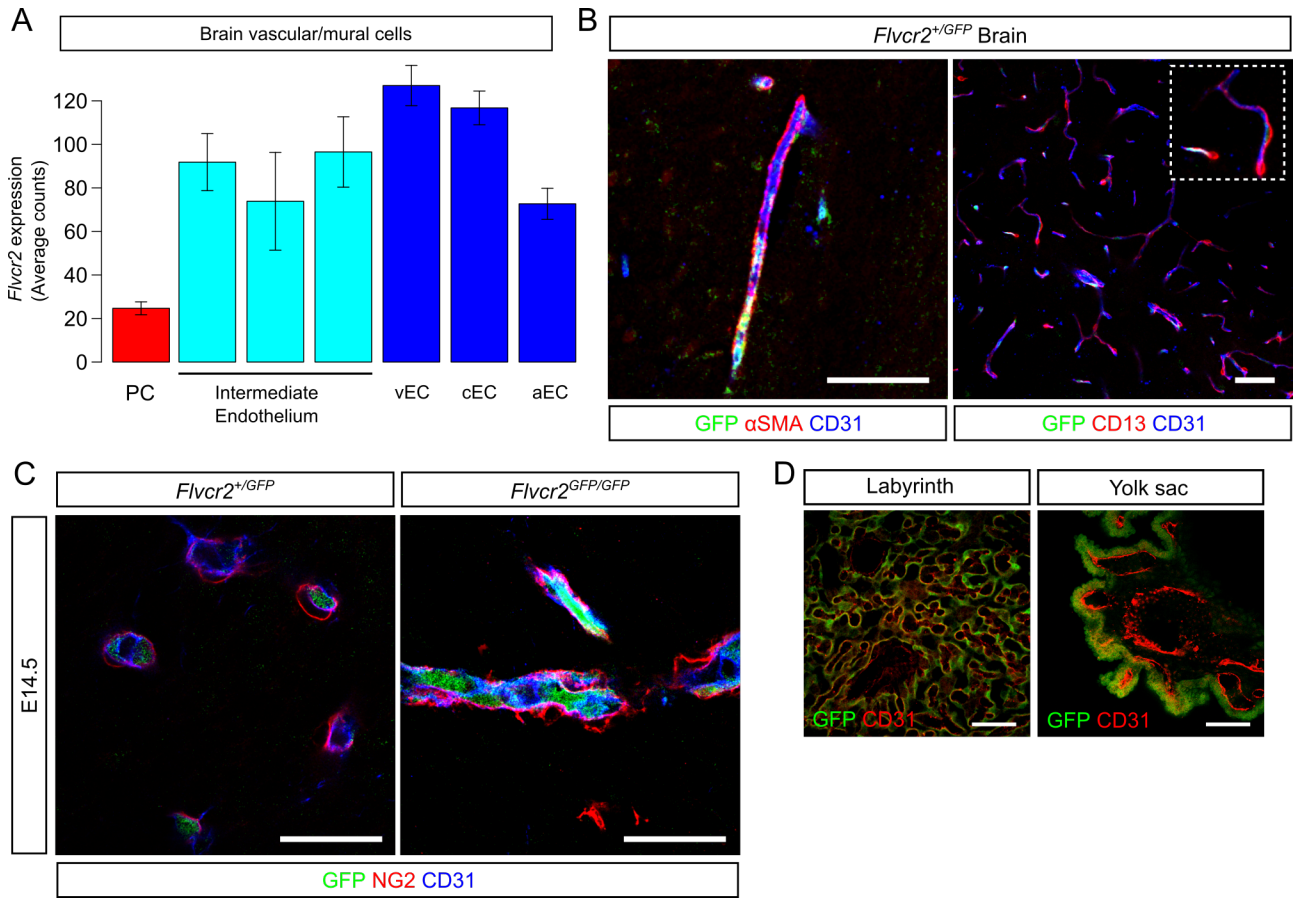

Figure S3. *Flvcr2* distribution in vascular segments and extraembryonic tissues. A. *Flvcr2* expression data in brain vascular and mural cells was obtained from http://betsholtzlab.org/VascularSingleCells/database.html (2). **B.** GFP was detected in brains from adult mice in arteries (vessels surrounded by αSMA), but not in CD13+ pericytes. N=3. Bar: 50 µm. **C.** GFP staining was not detected in NG2 positive cells in embryonic brains at E14.5. N=3 per genotype. **D.** GFP localization in the placental labyrinth and yolk sac from *Flvcr2*$^{+/GFP}$ embryos at E14.5. N=3. Bars: 100 µm.

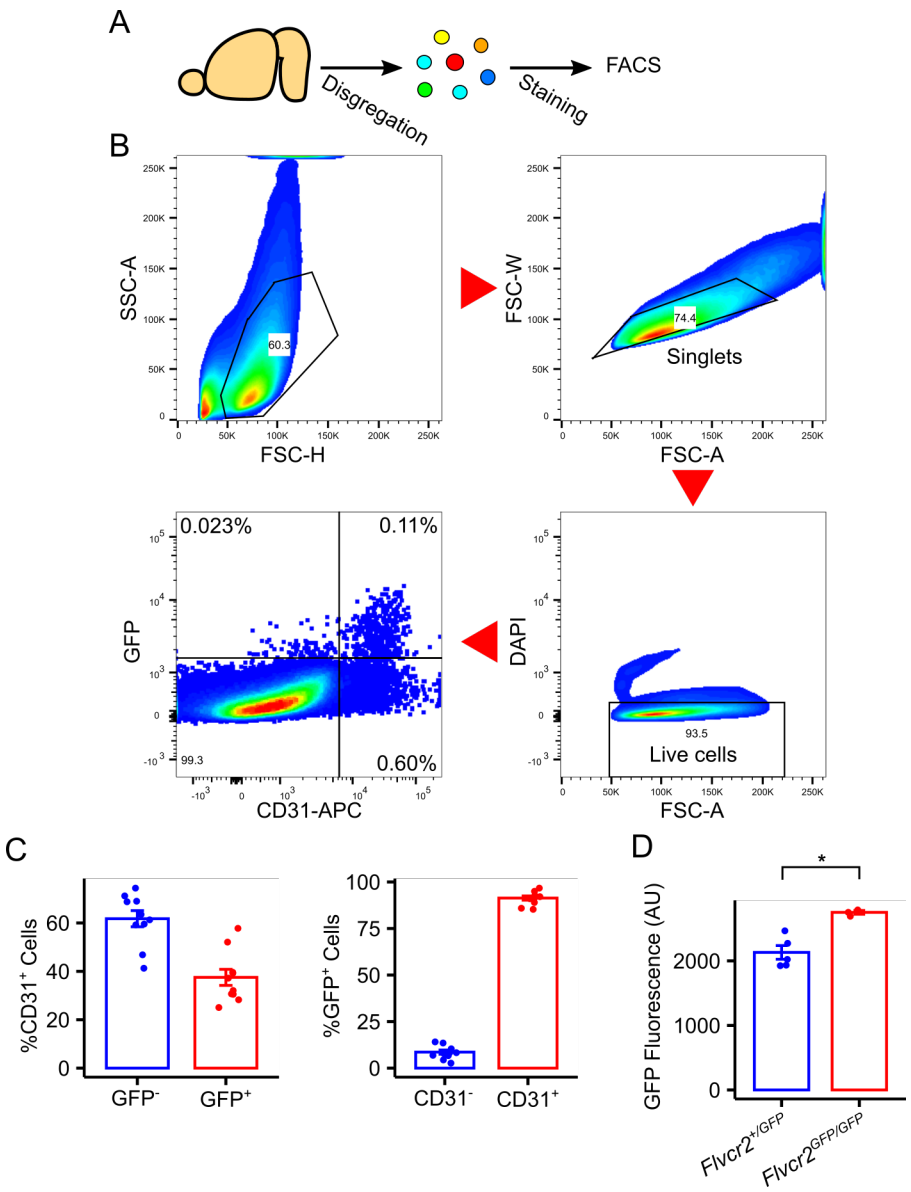

Figure S4. Cellular expression of *Flvcr2* in embryonic brain. A. Cell suspension from whole brains of *Flvcr2*$^{+/GFP}$ embryos were analyzed by flow cytometry at E15.5. **B.** The gating strategies to eliminate cell debris, capture singlets, and capture live cells are shown. This population was analyzed for GFP fluorescence and CD31 content. **C.** Proportion of endothelial cells expressing *Flvcr2* is shown in the graph on the left. The proportion of cells expressing *Flvcr2* that are endothelial cells or not are shown on the graph on the right. N=10. **D.** GFP fluorescence intensity was compared in *Flvcr2*$^{+/GFP}$ and *Flvcr2*$^{GFP/GFP}$ endothelial cells by flow cytometry using the same strategy as above. *p<0.05; Wilcoxon ranked-sum test. N: *Flvcr2*$^{+/GFP}$ = 5, *Flvcr2*$^{GFP/GFP}$ = 3.

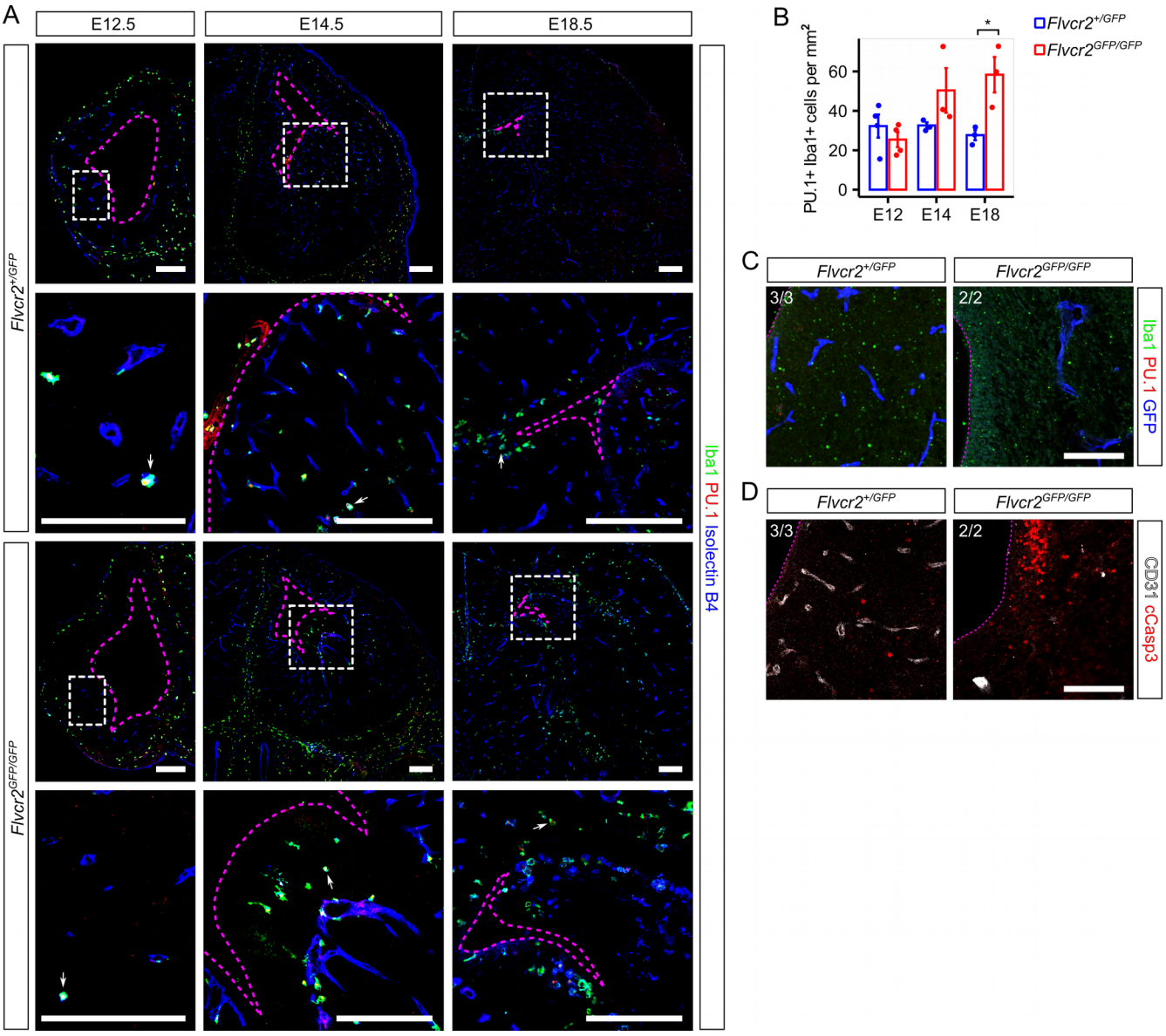

Figure S5. Role of myeloid cells in vascular defects in *Flvcr2*$^{GFP/GFP}$ embryos. A. Brain macrophages were detected in embryonic brains at different stages with the markers IBA1, PU.1, and IB4 reactivity. Bar: 200 µm. **B.** Macrophage numbers were quantified in the whole hemisphere. *p<0.05; t-test. E12.5 N=4; E14.5 N=3; E18.5 N=3. **C.** One pregnant dam was fed PLX5622-containing diet to ablate myeloid cells in embryos. Presence of macrophages in embryonic brains at E18.5 was evaluated by immunodetecting Iba1+ PU.1+ cells. Numbers represent embryos with the shown phenotype. Dotted violet lines indicate the ventricle wall. Bar: 100 µm. **D.** Cell death was revealed with an anti cleaved Caspase 3 antibody (cCasp3). Bar: 100 µm.

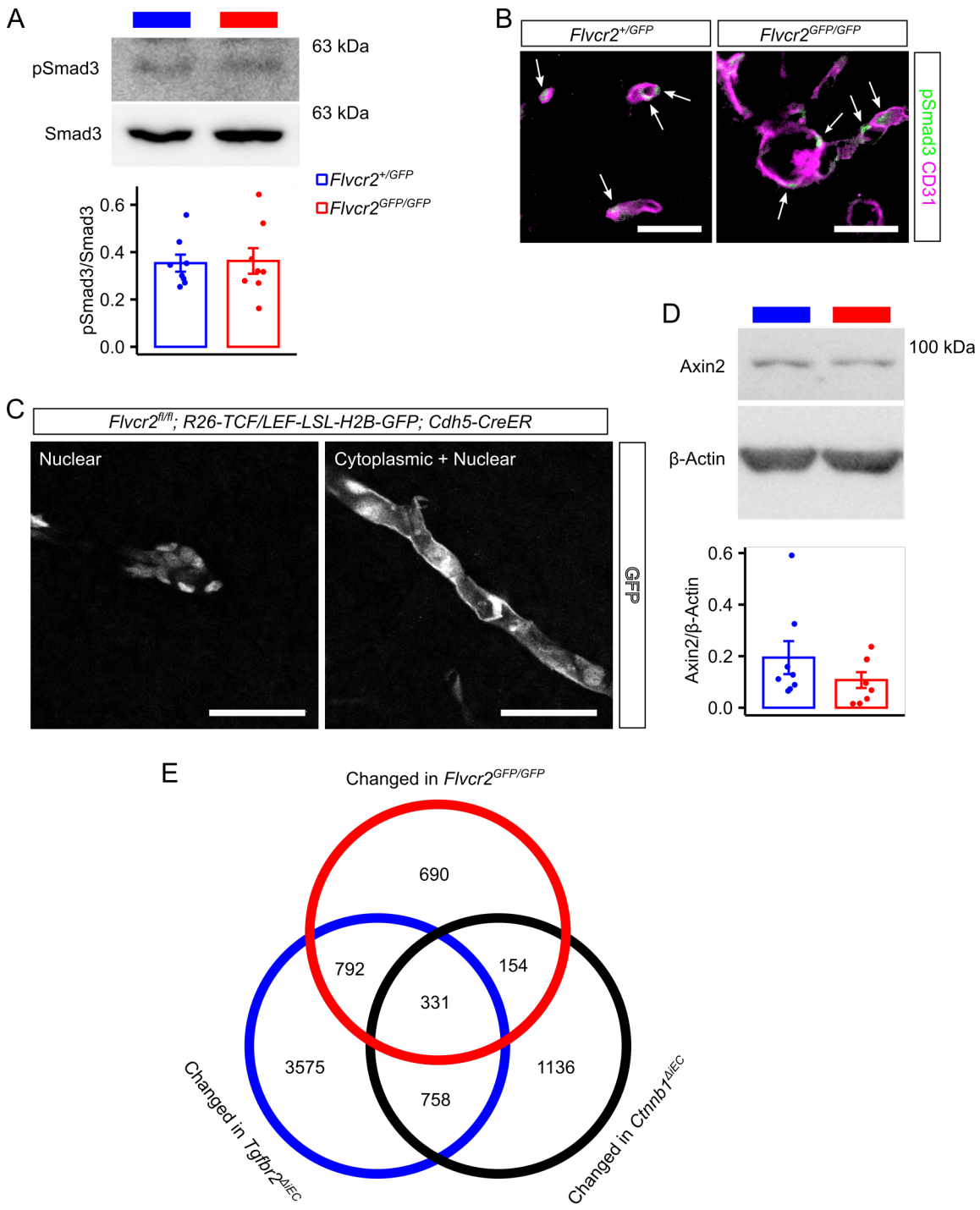

Figure S6. Effect of *Flvcr2* inactivation on TGF-β and WNT/β-catenin signaling. A. Phosphorylated and total SMAD3 were detected by Western blot in lysates from endothelial cells. N=8 per genotype **B.** pSMAD3 positive nuclei (arrows) were detected using immunofluorescence in brain sections through the LGE at E14.5. For clarity, the signal in the green channel outside of vessels was eliminated. N=3 per genotype. Bar: 50 µm. **C.** WNT signaling pathway activation was evaluated in endothelial cells lacking *Flvcr2*. Nuclear GFP indicates WNT signaling activation, while cytoplasmic GFP reports recombination in the *Flvcr2* locus. N=2. Bar: 50 µm. **D.** AXIN2 protein was detected by Western blot in vascular fragments from embryonic brains. N=8 per genotype. **E**. Transcriptional profiles from

endothelial cells lacking either *Flvcr2*, *Tgfbr2* (*Tgfbr2*$^{\Delta iEC}$), or *Ctbnn1* (*Ctnnb1*$^{\Delta iEC}$) were compared to assess similarity.

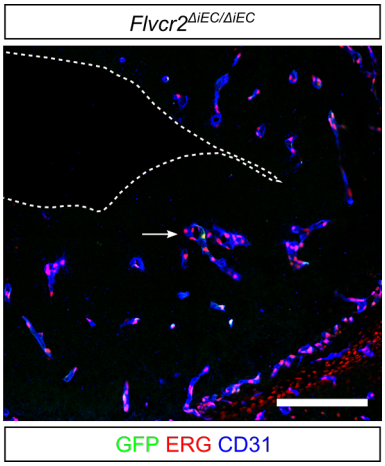

Figure S7. Mosaic inactivation of *Flvcr2* in endothelial cells leads to vascular malformations. *Flvcr2* inactivation was induced in embryonic endothelial cells by tamoxifen injection at E10.5, E11.5, and E12.5. Vascular malformations were observed in the GE despite lack of recombination in all endothelial cells. Representative picture of 3/11 *Flvcr2*$^{\Delta iEC/\Delta iEC}$ embryos. Bar: 200 µm.

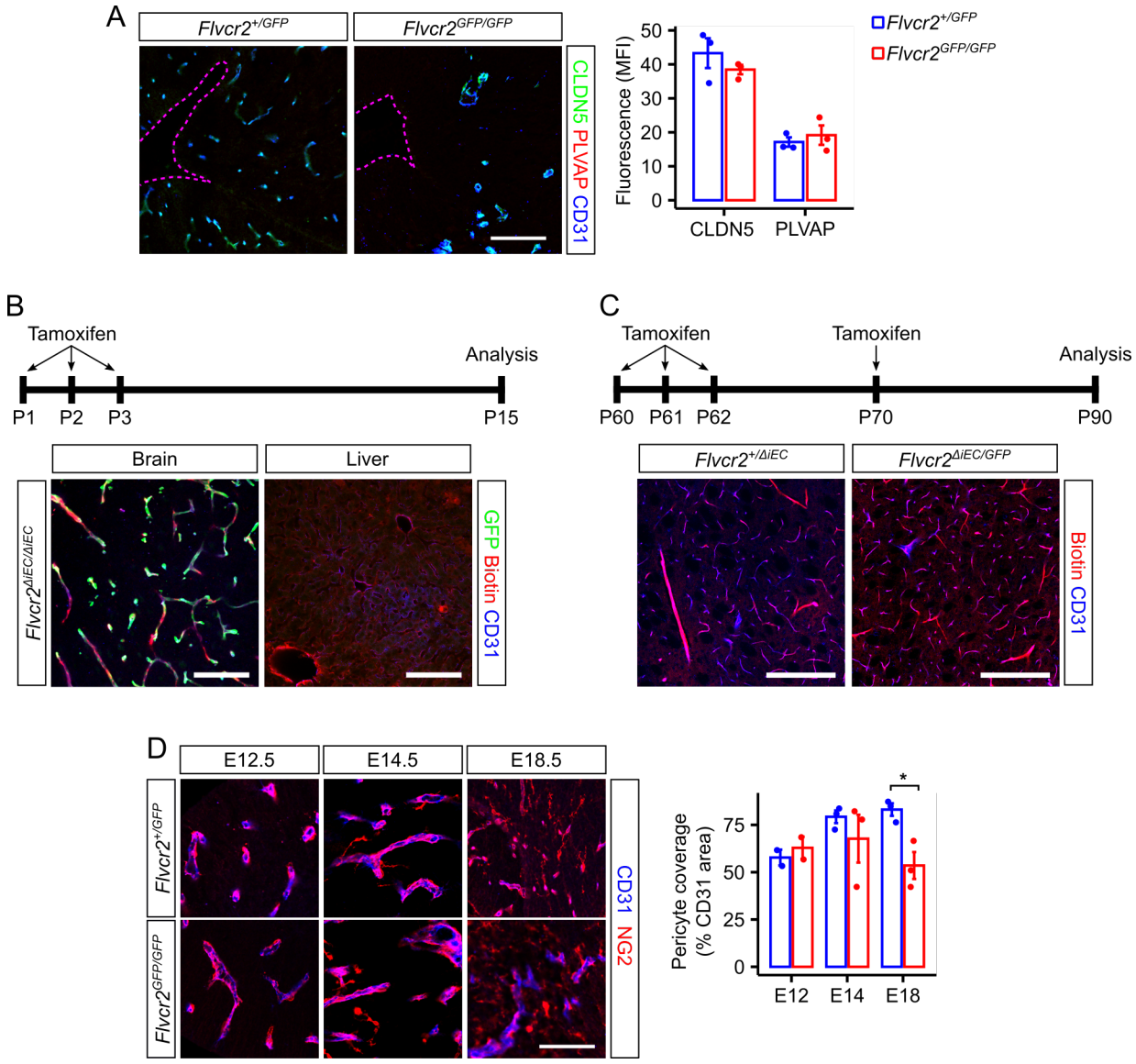

Figure S8. BBB is largely normal in mice lacking *Flvcr2*. A. Protein levels of a positive marker of BBB formation (CLDN5) and a marker of fenestrations (PLVAP). Quantifications indicate the mean fluorescence intensity (MFI) of each protein in CD31+ cells. T-test. N=3. Bar: 100 µm. **B.** *Flvcr2* inactivation was induced in pups at P1, P2, and P3. At P15, mice were perfused with biotin and the brains were evaluated for leakage into the parenchyma. The liver is shown as a positive control of endothelial cells without a BBB. N=3. Bar: 100 µm. **C.** *Flvcr2$^{fl/fl}$; Cdh5-CreER* were crossed to *Flvcr2$^{+/GFP}$* mice to generate mice with endothelial cell-specific deletion of the gene in the null background. Twenty days after four doses of tamoxifen, animals were perfused with biotin and analyzed for brain parechymal leakage of the tracer. N: *Flvcr2$^{+/\Delta iEC}$* = 3; *Flvcr2$^{\Delta iEC/GFP}$* = 2. Bar: 250 µm. **D.** Pericytes were detected in brain sections with an antibody against NG2. The quantification show the percentage of vessel area covered by pericytes. *p<0.05; t-test. E12.5 N=2; E14.5 N=3; E18.5 N=3. Bar: 100 µm.

REFERENCES

1. Tabula Muris Consortium et al. Single-cell transcriptomics of 20 mouse organs creates a Tabula Muris. *Nature.* 2018;562(7727):367–372.

2. Vanlandewijck M et al. A molecular atlas of cell types and zonation in the brain vasculature. *Nature.* 2018;554(7693):475–480.