Neuro
-endocrinology

DOI: 10.1159/000508735
Received: 11/14/2019
Accepted: 5/15/2020
Published(online): 5/18/2020

Lipid Droplets accumulate in the hypothalamus of mice and humans with and without metabolic diseases.

Maya-Monteiro C.M. Corrêa-da-Silva F. Hofmann S.S. Hesselink M.K. la Fleur S.E. Yi C.-X.

ISSN: 0028-3835 (Print), eISSN: 1423-0194 (Online)
https://www.karger.com/NEN
Neuroendocrinology

Disclaimer:
Accepted, unedited article not yet assigned to an issue. The statements, opinions and data contained in this publication are solely those of the individual authors and contributors and not of the publisher and the editor(s). The publisher and the editor(s) disclaim responsibility for any injury to persons or property resulting from any ideas, methods, instructions or products referred to in the content.

Neuroendocrinology

Manuscript:	NEN-2019-11-17/R1 RESUBMISSION
Title:	Lipid Droplets accumulate in the hypothalamus of mice and humans with and without metabolic diseases.
Authors(s):	Clarissa M Maya-Monteiro (Corresponding Author), Felipe Corrêa-da-Silva (Co-author), Susanna S Hofmann (Co-author), Matthijs K.C. Hesselink (Co-author), Susanne E La Fleur (Co-author), Chun-Xia Yi (Corresponding author)
Keywords:	brain, hypothalamus, insulin resistance, lipid droplet, PLIN2, Type 2 Diabetes Mellitus
Type:	Research Article

Lipid Droplets accumulate in the hypothalamus of mice and humans with and without metabolic diseases.

Clarissa M. Maya-Monteiro*[1,2,3], Felipe Corrêa-da-Silva [2], Susanna S. Hofmann[4], Matthijs K. C. Hesselink[5], Susanne E. la Fleur [2,3], Chun-Xia Yi*[2,]

[1]Laboratory of Imunofarmacology, Instituto Oswaldo Cruz, Fundação Oswaldo Cruz (FIOCRUZ), Rio de Janeiro, Brazil.
[2]Laboratory of Endocrinology and Department of Endocrinology and Metabolism, Amsterdam Neuroscience, Amsterdam University Medical Centers (UMC), University of Amsterdam, Amsterdam, the Netherlands.
[3]Netherlands Institute for Neuroscience, an Institute of the Royal Netherlands Academy of Arts and Sciences, Amsterdam, the Netherlands.
[4]Institute for Diabetes and Regeneration, Helmholtz Diabetes Center, Helmholtz Zentrum München, German Research Center for Environmental Health (GmbH), Neuherberg, Germany.
[5]Department of Nutrition and Movement Sciences, Maastricht University Medical Centre+ and NUTRIM School for Nutrition and Translational Research in Metabolism, Maastricht, the Netherlands

Running title: Lipid droplets in hypothalamus.

*corresponding authors
Clarissa Menezes Maya-Monteiro, M.D., Ph.D.
Laboratory of Imunofarmacology, Instituto Oswaldo Cruz, Fundação Oswaldo Cruz (FIOCRUZ). Av. Brasil 4365, Pavilhão 108, sala 48; Rio de Janeiro, Brasil. 21045-900, phone: 55-21-25621771.
Laboratory of Endocrinology and Department of Endocrinology and Metabolism, Amsterdam Neuroscience, Amsterdam University Medical Centers (UMC), University of Amsterdam, Amsterdam,. location AMC, K2- 283, Meibergdreef 9, 1105 AZ Amsterdam, the Netherlands; phone: +31-020-5662428.
Emails:
clarissa@ioc.fiocruz.br; c.maya@amsterdamumc.nl

Chun-Xia Yi, M.D., Ph.D.
Department of Endocrinology and Metabolism
Amsterdam University Medical Centers, location AMC, University of Amsterdam
Meibergdreef 9, 1105 AZ, Amsterdam, Netherlands
Tel: +31 20 5664807
Fax: +31 20 6917682
Email: c.yi@amsterdamumc.nl

Keywords: lipid droplet, PLIN2, insulin resistance, Type 2 Diabetes Mellitus, hypothalamus

Abstract

Background: In peripheral tissues, the lipid droplet (LD) organelle links lipid metabolism, inflammation and insulin resistance. Little is known about the brain LDs. **Objectives:** We hypothesised that hypothalamic LDs would be altered in metabolic diseases. **Methods:** We used immunofluorescence for the specific LD protein, PLIN2, as the marker to visualize and quantify LDs. **Results:** LDs were abundant in the hypothalamic third ventricle (3V) wall layer with similar heterogeneous distributions between control mice and humans. The LD content was enhanced by HFD in both WT and in low-density lipoprotein receptor deficient (*Ldlr* -/- HFD) mice. Strikingly, we observed a lower LD amount in T2DM patients when compared with non-T2DM, patients. **Conclusions:** LDs accumulate in the normal hypothalamus, with similar distributions in human and mouse. Moreover, metabolic diseases differently modify LD content in mouse and human. Our results point hypothalamic LDs accumulation as an important target to the study of metabolism.

Introduction

The development of metabolic diseases, such as obesity and Type 2 diabetes mellitus (T2DM), coincides with disturbed fatty acid metabolism and excess ectopic fat storage as lipid droplets (LD) in many peripheral tissues (1, 2). While LDs are linked to insulin resistance in peripheral tissues, little is known about the role of LDs in the brain and their putative role in the pathogenesis of metabolic disease and insulin resistance (2). The impairment of glucose uptake induced by insulin in peripheral tissues, is a core feature of T2DM, and insulin resistance also occurs in the central nervous system (3-5). Insulin resistance, induced by high fat diet (HFD) feeding, is accompanied by disruption of lipid metabolism with ectopic lipid deposition in intracellular LDs (6-9).

LDs are hydrophobic organelles formed in the majority of cells with a core of neutral lipids and a great variety of LD-proteins that are related to lipid metabolism, cell signalling and production of inflammatory mediators (10-12). The perilipin (PLIN) family comprise specific LD structural proteins (13), and the most studied proteins are the adipocyte PLIN1, and the PLIN2 (14). PLIN2 is present in all cells and tissues and it is exclusively present in LDs, being considered the standard marker for this organelle (15).

In Drosophila, LDs were identified in microglia and shown important for the neuronal energy supply (16). PLIN2 has also been described in LDs in isolated mammalian brain microglia under LPS stimuli (17, 18). Microglia in the hypothalamus has been shown to be involved in development of insulin resistance and lipid metabolism (19), however little is known about the role of other hypothalamic cell types such as ependimocytes and tanycytes, within he hypothalamic third ventricle (3V) region. These cells have barrier functions and are also important for the control of lipid and glucose metabolism/sensing (20-23), but whether LDs are present in the 3V wall cells remains unclear.

In the current study, we use the PLIN2 immunodetection as a marker for describing the physiological and pathological presence and distribution of LDs along the 3V in mouse and human with and without metabolic disorders.

Materials and Methods

Mice

C57BL/6 wild-type (WT) and low-density lipoprotein receptor deficient (Ldlr -/-) mice (Jackson Laboratories, Bar Harbor, ME) were housed in a pathogen-free environment, in conformity with institutional guidelines for animal experiments and approved by the Institutional Animal Care and Use Committee at the University of Cincinnati. For the description of the normal LD distribution we used C57BL/6 mice (age 8 weeks) fed with chow diet. The WT mice (age 8 weeks) were divided into two groups, one fed with chow diet, one with high fat diet (HFD) containing high carbohydrates (D12079B western diet, Research Diets, New Brunswick, NJ, containing 17 kcal% protein, 42% carbohydrate, and 41 kcal% fat as well as a high-fructose corn syrup sweetened beverage) for 32 weeks. Ldlr -/- mice (age 8 weeks) were fed with the HFD for 26 weeks. All mice were housed on a 12-hour-light/12-hour-dark cycle at 22 ± 2°C with food and water *ad libitum*. At the end of the study, all mice were decapitated and brains were immerse-fixed in 4% paraformaldehyde 0.1 M phosphate-buffered saline (PBS, pH 7.4) at 4°C for 48 hrs. Brains were then equilibrated 48 h in 30% sucrose in 0.1 M Tris-buffered saline (TBS, pH 7.2), and coronally cut by a cryostat into 30 μm sections.

Human brain tissue

Post-mortem hypothalamic tissues of 17 T2DM and 8 non-T2DM age matched controls, and 9 non-T2DM young subjects were obtained from the Netherlands Brain Bank, through autopsy approved by the Medical Ethic Committee of the VU Medical Center, The Netherlands. Subjects that had Braak stage V-VI or clinically diagnosed severe dementia were excluded. Subject details, including sex, age, post-mortem delay (PMD), clinical diagnosis and cause of death are provided in Table 1 and Supplemental Table 1. The tissues were immersion fixed in

10% phosphate-buffered formalin, then paraffin embedded and coronal sectioned in 6 µm slices. Every 100th section was used for Nissl staining to pre-determine the anatomical orientation of infundibular nucleus (IFN), which was further analysed by the range of NPY-positive neurons. For immunofluorescence, two sections of the IFN areas were mounted on SuperFrost Plus slides and dried at 37 ºC. Sections were deparaffinised and hydrated, rinsed in 50 mM TBS (pH 7.6), and were antigen retrieved for 10 min using microwave treatment at 700W in 0.1 M sodium citrate (pH 6.0).

Immunofluorescence

Samples were washed (2x) with TBS before incubations. All antibodies and streptavidin conjugates were diluted in TBS buffered solution containing 0.25 % gelatin and 0.5 % Triton X100, and all washes between incubations were done with TBS (4 x; 5 min). In all experiments control stainings were performed without the specific primary antibody. Samples were incubated with a guinea pig polyclonal anti-PLIN2 antibody (N-terminus aa 1-29, PROGEN, cat. GP40, 1:400 in mouse sections, 1:200 in human sections) and rabbit polyclonal anti-GFAP (DAKO, cat. Z0334, 1:100 in mouse sections, 1:500 in human sections) overnight at 4 ºC. Sections were then incubated for 1 hr with a secondary anti-guinea pig conjugated with biotin antibody (Vector, cat. BA7000, 1:400). After that, sections were incubated for 3 h with Streptavidin-Alexa 488 (Thermo-Fisher, cat. S32354, 1:300) in the human slides, or Streptavidin-Alexa 647 (Thermo-Fisher, cat. S32357, 1:500) in the mouse free floating slices. For the mouse sections, the Alexa-488 conjugated secondary anti-rabbit antibody (Thermo-Fisher, cat. A21206, 1:500) was added to the incubation, on the last hour. The sections were washed and incubated with DAPI (Thermo Fisher, cat. 62247, 1:2000) and NeuroTrace Red (when indicated) (Thermo Fisher, N21482) for 10 min. After final washing, all slides were then mounted with Vectashield Hard-Set mounting media (Vector).

Image acquisition and analysis

Images were acquired using a TCS SP8 Confocal in DMi8 inverted microscope (Leica Microsystems Wetzlar, Germany). Identical hardware and software settings were used for each of the mouse experiments and another fixed setting for the human samples.

The quantification of the mouse LDs by the PLIN2 immunoreactivity (PLIN2-ir) was performed with the Imaris software (Oxford Instruments, Abingdon, UK). Images for quantification (z-stack, 20 um) were taken with 63x lens. The DAPI and the PLIN2-Alexa647 images were used to generate surfaces corresponding to the total volume of nuclei and LDs respectively, with fixed settings of threshold. The amount of LDs per nuclei was obtained by dividing the volume of PLIN2 labelling per volume of DAPI labelling. The GFAP labelling was used as a marker of astrocytes and specific α-tanycytes.

The human PLIN2-ir LD quantification was done in images obtained with a 10x objective to cover half of the ventricle wall (left or right)on the slide, and 6 to 10 consecutive images were taken depending on the ventricle length. The analysis was made with the use of the ImageJ software and a macro developed by Ard Jonker (Cellular Imaging - Core Facility, Amsterdam UMC, location AMC). LD measurements were performed using an automatically generated stripe comprising the whole ependymal layer of the 3V half wall of each patient. The total length (from sulcus to infundibulus) was divided in 10 subsections to allow comparisons between patients. The region of interest to measure LD signal in the ependymal layer cells was determined by the nuclei (DAPI) staining, as detailed in the supplementary text file. The tissue fluorescence intensity outside the signal area was considered as the background signal and was subtracted from the signal fluorescence. The corrected signal of LD at the area of fluorescence was divided by the length of each subsection of the ventricle wall to give the LD fluorescence per μm of ventricle wall. The graphs and statistic calculations were performed with the GraphPad Prism software, and we used the Welsh’s t test for the comparison between specific groups.

Results

Heterogeneous distribution of LDs in mouse 3V

The presence and distribution of LDs along the 3V wall in the naïve mouse hypothalamus, was analysed by the presence of PLIN-ir in three regions determined by the distance to bregma along the rostral-caudal axis (24). We observe that the majority of LDs are concentrated along the ependymal layer of the 3V in three different levels along the rostral-caudal axis with a heterogeneous distribution (Fig.1). We compared the amount of LD at 3 different rostral-caudal levels, corresponding to specific surrounding hypothalamic nuclei, the suprachiasmatic (SCN; Fig.1A), the periventricular (PVN; Fig.1B) and arcuate nuclei (ARC; Fig.1C), and the 3 regions of the ventricle (i.e. top, middle and bottom). Astrocytes (GFAP-ir) can be observed outside the ependymal layer (Fig.1A2-7, B2-7, C2-7), but the majority of the astrocytes do not present LDs. At the level of bregma -1.8, the α tanycytes that project to the dorsal ARC and also express GFAP (25), coincide with an area with high amount of LDs (Fig.1C4, C5). There is no significant variation between the different bregma levels when we compare the top and the middle regions. Interestingly, in the bottom region with GFAP negative α2 tanycytes, known to project to the ventromedial ARC, we found negligible amounts of LD along the ventricle wall (Fig.1C6, C7, E).

LDs in the 3V ventricle in insulin resistant HFD mice

The ARC is a key brain region containing important neuron populations that are involved in development of insulin resistance, and the tanycytes in this area of the 3V are involved in the control of systemic energy metabolism (26). We investigated whether the LD heterogeneous distribution at the level of bregma -1.82 of the 3V, was associated to the presence of insulin resistance. We studied the alterations induced by a high fat diet (HFD) in the LDs distribution in both WT mice, and in the low-density lipoprotein receptor deficient (*Ldlr* -/-) mice. The *Ldlr* -/- mice on a HFD present a complete phenotype for the study of metabolic diseases with enhanced adipose tissue, insulin resistance and hypothalamic inflammation (7, 8). The top area of the ventricle, with a majority of ependimocytes, has enhanced LDs in both the HFD-WT and the HFD-fed *Ldlr* -/- mice (Fig.2A1, A2, B1, B2 and C1, C2, E). The GFAP-ir α tanycytes area presents high amount of LDs similarly to the top part of the ventricle in all the groups (Fig.2A3, A4, B3, B4 and C3, C4). The lower part of the ventricle wall with GFAP negative α and β tanycytes (as shown in Fig.1), has very low amounts of LDs in the WT under chow (Fig.2A5, A6, E). Moreover, the HFD in the WT mouse did not induce LDs in this area (Fig.2 B5, B6, E). In the HFD-fed *Ldlr* -/- mice, with a more severe metabolic disruption, we can observe enhanced LDs at the ARC level (Fig.2C5, C6, E).

LDs in the 3V in Control (non-diabetic) and T2DM subjects

Insulin resistance induced by HFD in animal models, do not fully mimic the human T2DM pathogenesis, and this is a major obstacle hampering translational studies on brain dysfunction in T2DM patients (19). We therefore investigated the distribution of LDs in the human 3V, comparing control (non-diabetic patients) with T2DM patients. We used hypothalamic samples from the Netherlands Brain Bank selected with age matched between the groups (Table.1). We show here that the human 3V wall contains significant amounts of LDs as evidenced by the PLIN2 staining (Fig.3A) and the LDs are mainly associated to the ependimocyte layer (Fig.3B). All the patients with different ages, distinct causes of death, different body mass index (BMI, only part of the subjects had BMI data) and different post-mortem delay (PMD) present a significant amount of LD in the 3V wall. Moreover the LD content in this group of subjects did not correlate with age, BMI or post mortem delay (PMD) (Suppl.Fig. 1). Important to point that we also show the presence of lipid droplets already in the 3V wall from a separate group of young brains (Supplement Figure 3; Supplement Table 1). To compare the control patients with the T2DM patients, we performed the quantification of the LDs along the ventricle as described in methods section (Fig.3C). In the control and T2DM subjects, we found the same pattern of distribution of LD as in mice, with higher amounts of LDs in the top and middle areas (PVN and DMH) and lower amounts in the bottom area (infundibular nucleus) (Fig. 3A, C, D). Within the T2DM patients this distribution is less pronounced with lower amounts of LDs in the ventricle wall cells (Fig.3C, D, E).

Discussion

Here we provide evidence for the existence of significant amounts of LD in normal hypothalami in both mouse and human, concentrated in the wall of the 3V. We further demonstrate that with high fat feeding and genetic deletion of LDL receptor in mice, the LD content is augmented, depending on degree of metabolic disturbances. Interestingly, in the human brain, the presence T2DM disease was accompanied by lower amounts of LDs in the hypothalamus. Taken together our data point to an important role for brain LDs in metabolism.

To our knowledge, we are the first to show a major LD accumulation within the ependymal layer of the 3V both in physiological and pathological conditions. Earlier studies have described the presence of LDs in the brain ventricle ependymal layer, however they only identified significant LDs in the diseased brain with Alzheimer disease (AD) or aging (27-29). These authors propose that the presence of LDs in the brain is correlated directly to a pathological condition. We could demonstrate the age correlation when group of younger subjects was analysed together with the older non-T2DM control group. Our findings point to a broader physiological role for the accumulation of LDs along the ventricle, that is also modified in metabolic diseases and aging. The heterogeneity of the distribution of LDs along the 3V is evident in the mouse and human brains and this pattern is specifically modified under pathogenic conditions. Interestingly, the changes in LDs in mouse brain with metabolic disruption either with HF feeding alone, or even more severe, through Ldlr deletion, was region specific. We found that the more metabolically disturbed the more markedly the increase of LDs in the top of the ventricle whereas there were discrete increases in LD in the ARC region of the 3V wall. Interestingly, in humans, with T2DM, we observed a reduction of LDs throughout the ventricle regions. We can speculate that the complexity and chronic conditions of human T2DM may lead to a reduction of LDs when compared to the control, non-diabetic patients.

LD organelle present a hydrophobic core of neutral lipids that can be partially or completely extracted during fixation procedures. This characteristic makes it difficult to fully assess the presence of LDs in the brain by using hydrophobic probes solely. PLIN2 is the standard and very specific marker of LDs in all cells and the protein is readily degraded when not associated with the organelle (30). LDs have distinct roles depending on the cell and stimuli. The most studied function is of lipid storage organelles, but LDs are also known sites of inflammatory mediator production and intracellular signalling, and vary in protein and lipid composition and size (31, 32). We observed a great diversity of LDs size (0.5-4 um) in the ependymal layer. According to literature, this suggests diverse roles for these LDs in lipid storage (larger), and in control of lipid metabolism, inflammation and signalling (smaller) (32, 33).

A previous study proposed that only the tanycytes, and not the ependymal cells, would accumulate LDs only under HFD (20). However, we show here that the majority of LDs in both mouse and in human brain is clearly associated with the ependimocytes, and there is a high content of LDs in areas with no tanycytes. These results show that ependimocytes and tanycytes accumulate LDs and may have a physiological role in brain lipid metabolism.

Recently, Geller *et al*. showed that mouse tanycytes can sense palmitate, in an oxidation dependent manner, and modulate neurons to indirectly inhibit the lipolysis in the periphery (34). Interestingly, these palmitate-sensing tanycytes are exactly in the region of the GFAP negative α tanycytes that we show here. Our suggestion is that this ARC region of the ventricle is a specialised lipid sensing area and, therefore, do not physiologically accumulate LDs. Additionally, we suggest that this is an important feature since the same pattern was present in both mouse and human hypothalamus.

Earlier reports showed LD accumulation in microglia and astrocytes being important in the control of neuronal metabolism (16, 35, 36). In line, we also observed LDs in both microglia and astrocytes however less abundant when compared to the ependymal layer (as marked by Iba1-ir and GFAP-ir; Supp. Fig. 2).

We conclude that the presence of hypothalamic LDs is region specific, and is modified depending on the metabolic conditions in both mice and humans. The characterization of the distribution of the LDs opens possibilities for better understanding of lipid metabolism and lipid sensing in the brain. We propose that the evaluation of the LD organelle within the brain tissue should always be considered when studying the hypothalamic control of metabolism under physiological and pathogenic conditions.

Acknowledgements

The authors want to express their gratitude to Martin Kalsbeek, Unga Unmehopa, Khalid Lamuadni and Nikita Korpel for laboratory technical assistance and to Ard Jonker for the development of the macro for ImageJ.

Statement of Ethics

Post-mortem hypothalamic human tissues were obtained from the Netherlands Brain Bank, through autopsy approved by the Medical Ethic Committee of the VU Medical Center, The Netherlands.

Disclosure Statement

The authors have no conflicts of interest to declare.

Funding

This work was supported by Fundação Oswaldo Cruz (FIOCRUZ), Instituto Oswaldo Cruz (IOC) and Conselho Nacional de Desenvolvimento Científico e Tecnológico (306290/2014-6 CNPq) (CMM), AMC fellowship (CXY, 2014, Amsterdam University Medical Center), the Dutch Diabetes Research Foundation (CXY, Diabetes Fonds, 2015.82.1826), by the Netherlands Organization of Scientific Research (NWO-VICI grant 016.160.617) and by the Deutsche Forschungsgemeinschaft (German Research Foundation) within the framework of the Munich Cluster for Systems Neurology (EXC 2145 SyNergy). All authors declare no competing interests.

Author Contributions

The conception of the study was done by CMM, SEF, MKH and CXY. The experimental design, execution, analysis and interpretation of the data was done by CMM, SEF, FCS, SSH and CXY. The manuscript was written by CMM and CXY, and the final version was critically revised and approved by all the authors.

Uncategorized References

1. Olofsson SO, Andersson L, Haversen L, Olsson C, Myhre S, Rutberg M, et al. The formation of lipid droplets: possible role in the development of insulin resistance/type 2 diabetes. Prostaglandins Leukot Essent Fatty Acids. 2011;85(5):215-8.

2. Gemmink A, Goodpaster BH, Schrauwen P, Hesselink MKC. Intramyocellular lipid droplets and insulin sensitivity, the human perspective. Biochim Biophys Acta Mol Cell Biol Lipids. 2017;1862(10 Pt B):1242-9.

3. Lewis GF, Carpentier A, Adeli K, Giacca A. Disordered fat storage and mobilization in the pathogenesis of insulin resistance and type 2 diabetes. Endocr Rev. 2002;23(2):201-29.

4. Kahn BB, Flier JS. Obesity and insulin resistance. J Clin Invest. 2000;106(4):473-81.

5. Steculorum SM, Solas M, Bruning JC. The paradox of neuronal insulin action and resistance in the development of aging-associated diseases. Alzheimers Dement. 2014;10(1 Suppl):S3-11.

6. Merat S, Casanada F, Sutphin M, Palinski W, Reaven PD. Western-type diets induce insulin resistance and hyperinsulinemia in LDL receptor-deficient mice but do not increase aortic atherosclerosis compared with normoinsulinemic mice in which similar plasma cholesterol levels are achieved by a fructose-rich diet. Arterioscler Thromb Vasc Biol. 1999;19(5):1223-30.

7. Schreyer SA, Vick C, Lystig TC, Mystkowski P, LeBoeuf RC. LDL receptor but not apolipoprotein E deficiency increases diet-induced obesity and diabetes in mice. Am J Physiol Endocrinol Metab. 2002;282(1):E207-14.

8. Yi CX, Al-Massadi O, Donelan E, Lehti M, Weber J, Ress C, et al. Exercise protects against high-fat diet-induced hypothalamic inflammation. Physiol Behav. 2012;106(4):485-90.

9. Gupte AA, Minze LJ, Reyes M, Ren Y, Wang X, Brunner G, et al. High-fat feeding-induced hyperinsulinemia increases cardiac glucose uptake and mitochondrial function despite peripheral insulin resistance. Endocrinology. 2013;154(8):2650-62.

10. Zhang C, Liu P. The New Face of the Lipid Droplet: Lipid Droplet Proteins. Proteomics. 2019;19(10):e1700223.

11. Welte MA, Gould AP. Lipid droplet functions beyond energy storage. Biochim Biophys Acta Mol Cell Biol Lipids. 2017;1862(10 Pt B):1260-72.

12. Maya-Monteiro CM, Bozza PT. Leptin and mTOR: partners in metabolism and inflammation. Cell Cycle. 2008;7(12):1713-7.

13. Brasaemle DL. Thematic review series: adipocyte biology. The perilipin family of structural lipid droplet proteins: stabilization of lipid droplets and control of lipolysis. J Lipid Res. 2007;48(12):2547-59.

14. Blanchette-Mackie EJ, Dwyer NK, Barber T, Coxey RA, Takeda T, Rondinone CM, et al. Perilipin is located on the surface layer of intracellular lipid droplets in adipocytes. J Lipid Res. 1995;36(6):1211-26.

15. Sztalryd C, Brasaemle DL. The perilipin family of lipid droplet proteins: Gatekeepers of intracellular lipolysis. Biochim Biophys Acta Mol Cell Biol Lipids. 2017;1862(10 Pt B):1221-32.

16. Liu L, MacKenzie KR, Putluri N, Maletic-Savatic M, Bellen HJ. The Glia-Neuron Lactate Shuttle and Elevated ROS Promote Lipid Synthesis in Neurons and Lipid Droplet Accumulation in Glia via APOE/D. Cell Metab. 2017;26(5):719-37 e6.

17. Fan B, Dun SH, Gu JQ, Guo Y, Ikuyama S. Pycnogenol Attenuates the Release of Proinflammatory Cytokines and Expression of Perilipin 2 in Lipopolysaccharide-Stimulated Microglia in Part via Inhibition of NF-kappaB and AP-1 Activation. PLoS One. 2015;10(9):e0137837.

18. Khatchadourian A, Bourque SD, Richard VR, Titorenko VI, Maysinger D. Dynamics and regulation of lipid droplet formation in lipopolysaccharide (LPS)-stimulated microglia. Biochim Biophys Acta. 2012;1821(4):607-17.

19. Hogenboom R, Kalsbeek MJ, Korpel NL, de Goede P, Koenen M, Buijs RM, et al. Loss of arginine vasopressin- and vasoactive intestinal polypeptide-containing neurons and glial cells in the suprachiasmatic nucleus of individuals with type 2 diabetes. Diabetologia. 2019.

20. Hofmann K, Lamberz C, Piotrowitz K, Offermann N, But D, Scheller A, et al. Tanycytes and a differential fatty acid metabolism in the hypothalamus. Glia. 2017;65(2):231-49.
21. Frayling C, Britton R, Dale N. ATP-mediated glucosensing by hypothalamic tanycytes. J Physiol. 2011;589(Pt 9):2275-86.
22. Gao Y, Tschop MH, Luquet S. Hypothalamic tanycytes: gatekeepers to metabolic control. Cell Metab. 2014;19(2):173-5.
23. Garcia-Caceres C, Balland E, Prevot V, Luquet S, Woods SC, Koch M, et al. Role of astrocytes, microglia, and tanycytes in brain control of systemic metabolism. Nat Neurosci. 2019;22(1):7-14.
24. Paxinos G, Franklin KBJ, Academic P, Elsevier. Paxinos and Franklin's the mouse brain in stereotaxic coordinates. Amsterdam [etc.]: Elsevier/Academic Press; 2013.
25. Robins SC, Stewart I, McNay DE, Taylor V, Giachino C, Goetz M, et al. alpha-Tanycytes of the adult hypothalamic third ventricle include distinct populations of FGF-responsive neural progenitors. Nat Commun. 2013;4:2049.
26. Goodman T, Hajihosseini MK. Hypothalamic tanycytes-masters and servants of metabolic, neuroendocrine, and neurogenic functions. Front Neurosci. 2015;9:387.
27. Hamilton LK, Fernandes KJL. Neural stem cells and adult brain fatty acid metabolism: Lessons from the 3xTg model of Alzheimer's disease. Biol Cell. 2018;110(1):6-25.
28. Ogrodnik M, Zhu Y, Langhi LGP, Tchkonia T, Kruger P, Fielder E, et al. Obesity-Induced Cellular Senescence Drives Anxiety and Impairs Neurogenesis. Cell Metab. 2019;29(5):1061-77 e8.
29. Shimabukuro MK, Langhi LG, Cordeiro I, Brito JM, Batista CM, Mattson MP, et al. Lipid-laden cells differentially distributed in the aging brain are functionally active and correspond to distinct phenotypes. Sci Rep. 2016;6:23795.
30. Bickel PE, Tansey JT, Welte MA. PAT proteins, an ancient family of lipid droplet proteins that regulate cellular lipid stores. Biochim Biophys Acta. 2009;1791(6):419-40.
31. Welte MA. Expanding roles for lipid droplets. Curr Biol. 2015;25(11):R470-81.
32. Pereira-Dutra FS, Teixeira L, de Souza Costa MF, Bozza PT. Fat, fight, and beyond: The multiple roles of lipid droplets in infections and inflammation. J Leukoc Biol. 2019;106(3):563-80.
33. Fujimoto T, Parton RG. Not just fat: the structure and function of the lipid droplet. Cold Spring Harb Perspect Biol. 2011;3(3).
34. Geller S, Arribat Y, Netzahualcoyotzi C, Lagarrigue S, Carneiro L, Zhang L, et al. Tanycytes Regulate Lipid Homeostasis by Sensing Free Fatty Acids and Signaling to Key Hypothalamic Neuronal Populations via FGF21 Secretion. Cell Metab. 2019.
35. Hu X, Xu B, Ge W. The Role of Lipid Bodies in the Microglial Aging Process and Related Diseases. Neurochem Res. 2017;42(11):3140-8.
36. Belanger M, Allaman I, Magistretti PJ. Brain energy metabolism: focus on astrocyte-neuron metabolic cooperation. Cell Metab. 2011;14(6):724-38.

Figure legends

Figure.1 Distribution of LDs in normal mouse hypothalamus. Representative immunofluorescence images of the 3 different Bregma levels of hypothalamus with labelled neurons (NeuroTrace, in cyan, evidencing the hypothalamic nuclei), and LDs (LD; perilipin 2 immunoreactive, PLIN2-ir, in red) (A1, B1, C1). Higher magnification z-stack images were obtained at the marked areas, with the additional label for cell nuclei (DAPI, in grey), glial fibrillary acidic protein-ir (GFAP-ir) astrocytes and GFAP-ir α-tanycytes (in green), and LD (PLIN2-ir, in red) (panels 2-7 in A, B and C). LDs were observed at the GFAP-ir α-tanycytes region (C4). Quantification of the volume of LDs and nuclei are performed by generating the 3D surfaces, from the high magnification z-stack images, with the IMARIS software (D). Representative image from the outlined area in A4 (original image; D1, D2; nuclei in grey; LD in red) (surface image; D3, D4; nuclei in cyan, LD in yellow). The volume of LDs was normalized by the volume of nuclei in order to compare the different areas (E). Scale bar: 400 µm in A1, B1, C1; 40 µm in A2-A7, B2-B7, C2-C7; 10 µm in D1-D4. III: third ventricle. Data are presented as mean ±SEM of a minimum n=3. * $p<0.05$; ** $p<0.01$ according to the Welch´s t test.

Figure.2 LD content is altered in the hypothalamus of insulin resistant mice. Representative immunofluorescence images of the mouse hypothalamus (bregma -1.82) from wild type (WT) (A1-A6), high fat diet (HFD) fed- WT mice (WT-HFD) (B1-B6), and HFD-fed low-density lipoprotein receptor knock out mice (*Ldl*-KO-HFD, C1-C6). Cell nuclei (DAPI; in grey), glial fibrillary acidic protein immunoreactive (GFAP-ir) astrocytes and GFAP-ir α-tanycytes (GFAP-ir; in green), and LD (PLIN2-ir, in red) (A, B, C). Quantification of the volume of LDs and nuclei are performed by generating the 3D surfaces with the IMARIS software (D). Representative image from the outlined area in A1 (original image; D1, D2; nuclei in grey; LD in red) (surface image; D3, D4; nuclei in cyan, LD in yellow). The volume of LDs was normalized by the volume of nuclei in order to compare the different areas (E). Scale bar: 40 µm in A1-C6; 10 µm in D1-D4. III: third ventricle. Data is presented as mean ±SEM of a minimum n=4. * $p<0.05$; ** $p<0.01$ according to the Welch´s t test.

Figure.3 Characterisation of LD distribution and content in the third ventricle of non-diabetic controls and T2DM individuals.

The anatomical orientation of different hypothalamic areas along the third cerebral ventricle and the distribution of perilipin 2 immunoreactive (PLIN2-ir, in green) LDs (LD) and cell nuclei (DAPI, in grey) (A1). Higher magnification images from three regions outlined in A1 are shown in A2-A4. The morphology of the LDs within the ependimocytes is demonstrated by images in B1 and B2 from the area selected in D1. The PLIN2-ir LD content was quantified as described in the methods section from the images as in A1 and averaged for the 3 different regions of the ventricle wall according to the hypothalamic nuclei in the adjacent parenchyma (C). Representative images with higher magnification, showing the nuclei (DAPI, grey) and LD (PLIN2-ir, green), to evidence the variation of LDs between the control and the T2DM patients (D, E). III, third ventricle lumen; DMH, dorsomedial hypothalamic nucleus; VMH, ventromedial hypothalamic nucleus; IFN, infundibular hypothalamic nucleus. Scale bar: 1 mm in A1; 200 µm in A2-A4; 10 µm in B1-B2; 40 µm D1-D6, E1-E6. Data is presented as mean ±SEM of n=8 (controls) and n= 17 (T2DM). * p<0.05 according to the Welch´s t test.

Supplementary Figure.1 Graphs showing the absence of correlation between the amount of LDs in the human third ventricle and the parameters: Body Mass Index (BMI) Post Mortem Delay (PMD) and Age.

Supplementary Figure.2 LD in glial fibrillary acidic protein immunoreactive (GFAP-ir) astrocyte (A1, higher magnification of the arrow-pointed cell in A2); and Ionized calcium binding adaptor molecule 1-ir (Iba1-ir) microglial cells (B1, higher magnification of the arrow-pointed cell in B2) in the area next to third ventricle in post-mortem human brain tissue. Scale bar: 35 µm in A1 and B1, 10 µm in A2 and B2.

Supplementary Figure.3 LDs distribution and quantification in the 3V wall of younger subjects.

The PLIN2-ir LD was performed and quantified as described in the methods section and Figure 3. Representative images with higher magnification, showing the nuclei (DAPI, grey) and LD (PLIN2-ir, green), within different ages and regions of the third ventricle (A, B, C). Graphs showing the quantification along the ventricle in this younger group (D1), and the comparison with the older group (Control group from the study in figure 3) (D2 and D3). III, third ventricle lumen; DMH, dorsomedial hypothalamic nucleus; IFN, infundibular hypothalamic nucleus; y, years. Scale bar: 40 µm. Data is presented as mean ±SEM of n=9 (21-50 y) and n= 8 (65-95 y). * p<0.05 according to the Welch´s t test.

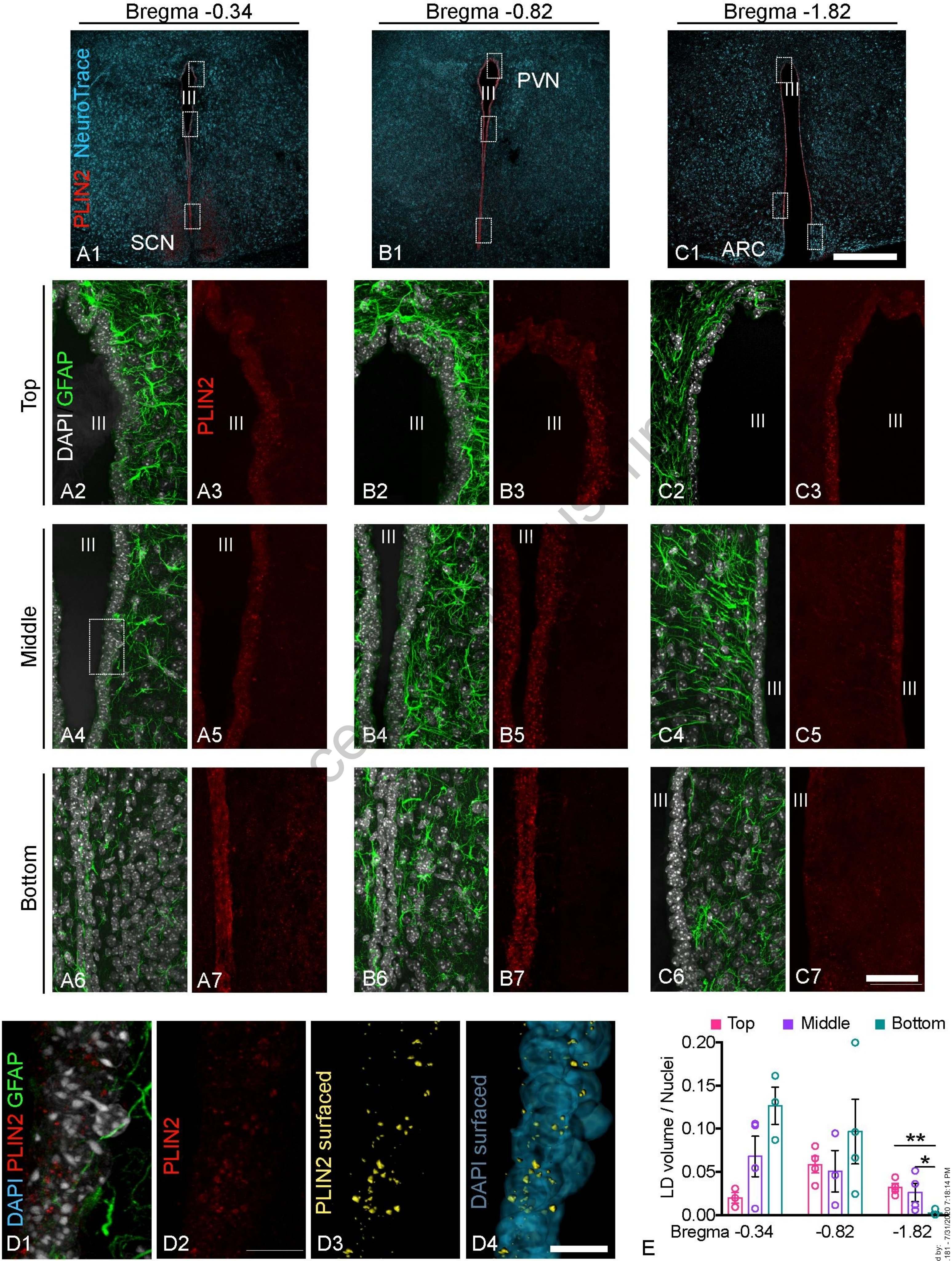
Supplementary text File Full description of the 4 macros used to quantify the human tissue LDs.

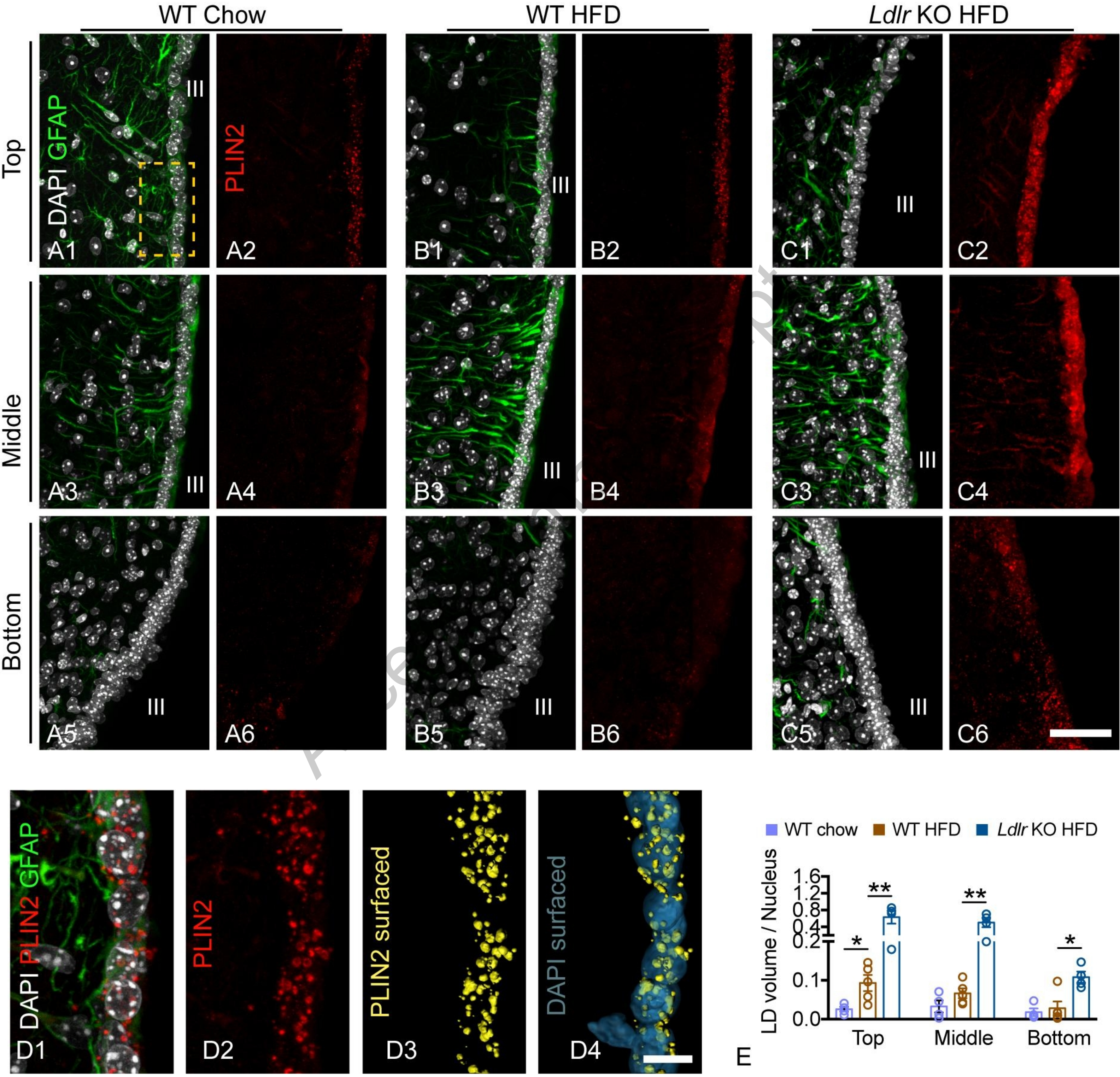
Supplementary Table Clinico-pathological data of younger subjects

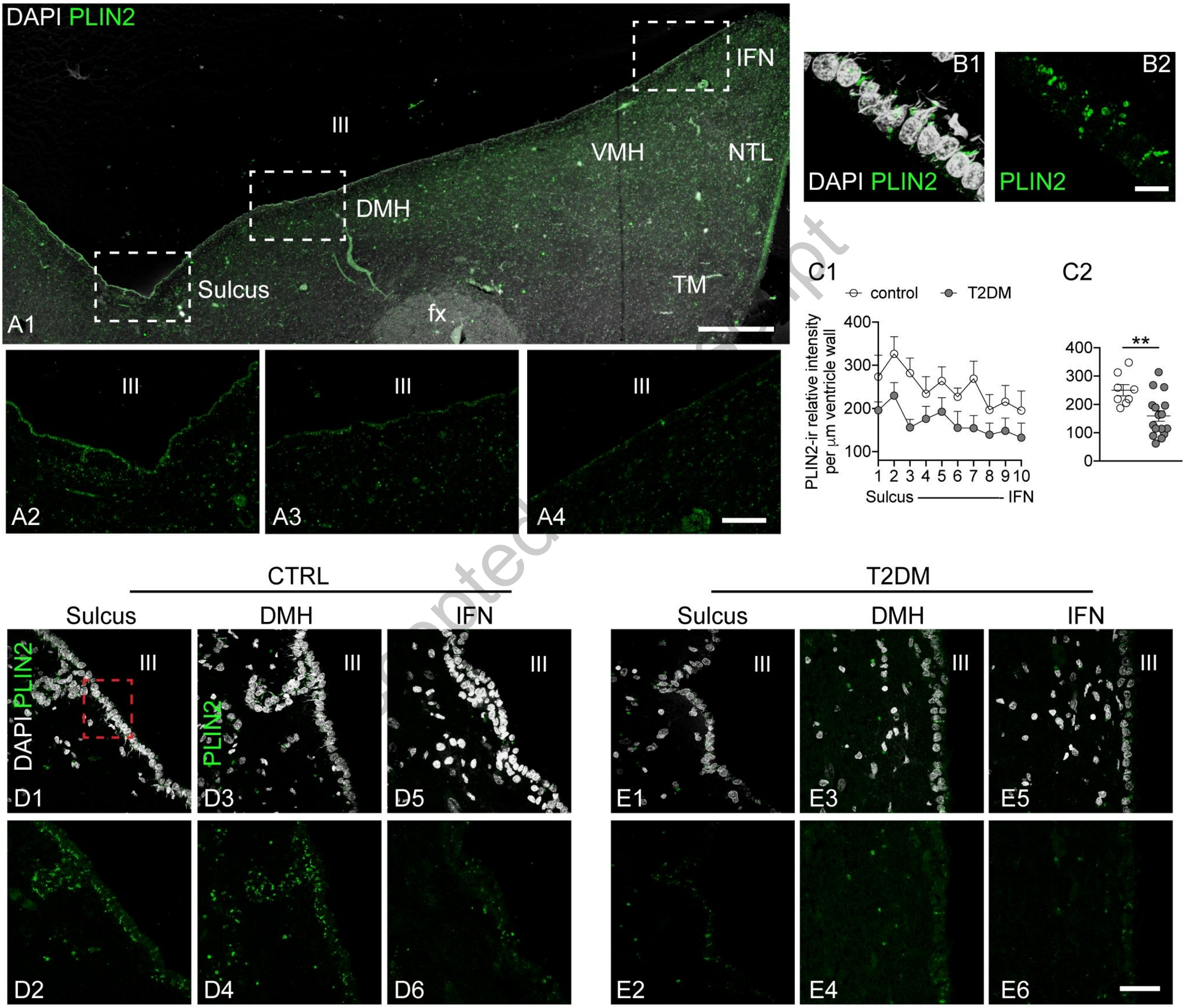
Table.1 Clinico-pathological data of patients and matched controls.

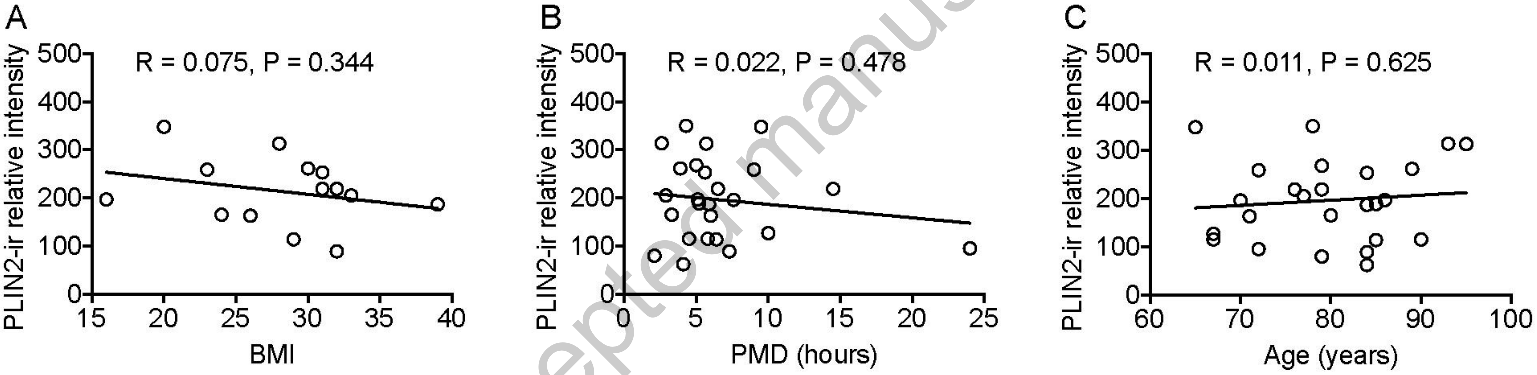
NBB nr.	S	Ag	Brk	PM	BMI	Cause of death and clinical diagnosis
Controls						
1980-009	f	72	/	9.0	23	Epithelial carcinoma.
1991-205	f	65	/	9.5	20	Breast tumor, cardiac failure
2007-088	f	82	3	5.2	22	Cachexia, cardiac failure, encephalopathy, mitral valve insufficiency.
2009-022	f	77	1	2.9	33	Pulmonary metastasis of vulva carcinoma.
2011-082	f	84	2	5.9	39	Respiratory failure, angina pectoris, mitral valve insufficiency.
2012-005	f	84	2	5.6	31	Heart failure, metastatic breast cancer, scoliosis.
2012-033	f	95	3	5.7	28	Heart failure, cachexia and dehydration, pulmonary disease.
2012-104	m	79	2	6.5	31	Legal euthanasia, Ischemic colitis, heart failure with dyspnea.
T2DM						
1989-032	m	84	/	4.1	/	Heart failure, intestinal tumor.
1995-008	f	79	3	2.1	/	Dehydration, endometrium carcinoma.
1997-088	f	78	1	4.3	/	Pneumonia, seizures, CVA, cortical dysarthria syndrome.
1998-080	f	72	3	24.	/	Respiratory failure, pulmonary hypertension.
1998-126	m	71	2	6.0	26	Respiratory insufficiency, lung carcinoma.
1999-015	f	93	3	2.6	/	Pneumonia, dehydration, breast cancer.
2001-003	m	67	/	10.	/	Heart failure, urinary tract infection.
2003-054	m	67	1	4.5	/	Cardiac shock CVA.
2006-033	m	79	1	5.0	/	Pneumonia, dehydration, CVA, choledochus carcinoma.
2008-105	f	89	3	3.9	30	Pneumonia, coronary artery bypass, atrial fibrillation,
2009-091	m	84	1	7.3	32	Anaemia, colon and prostate carcinoma metastasis.
2010-092	m	86	3	5.1	16	Dehydration and cachexia, CVA.
2011-027	m	80	1	3.3	24	Pneumonia, CVA, ischemic attack.
2012-049	f	70	2	7.6	/	Cachexia, pancreatic carcinoma.
2012-088	f	85	3	6.4	29	Legal euthanasia, hypoparathyroidism.
2012-092	m	90	2	5.8	/	Prostate carcinoma, CVA.
2014-040	m	85	3	5.2	/	Pulmonary carcinoma, pneumonia, arteritis temporalis.

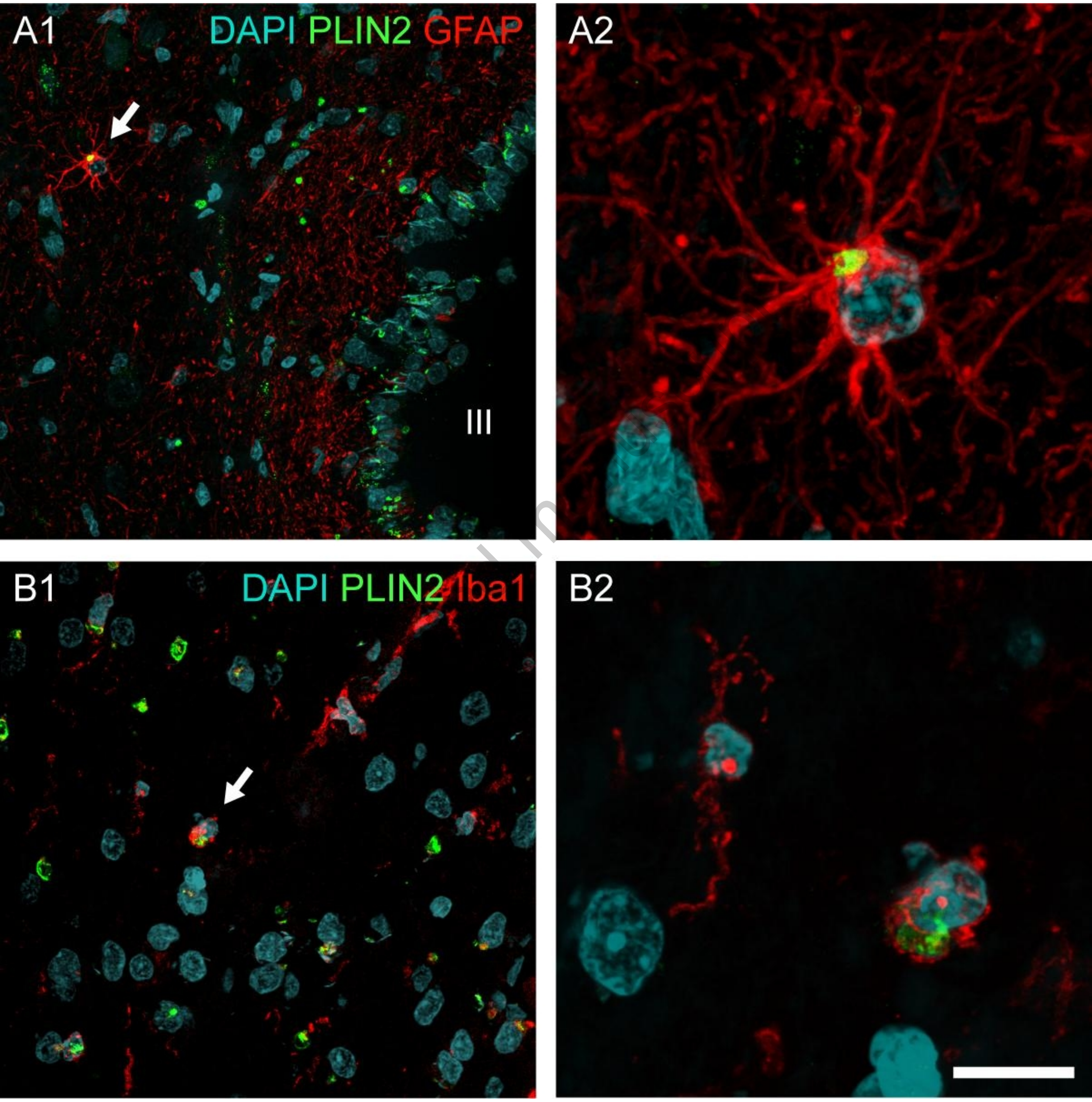
Brk: Braak stage, PMD: post-mortem delay between time of death and time of autopsy (hours), BMI: body mass index, CVA: cerebrovascular accident.

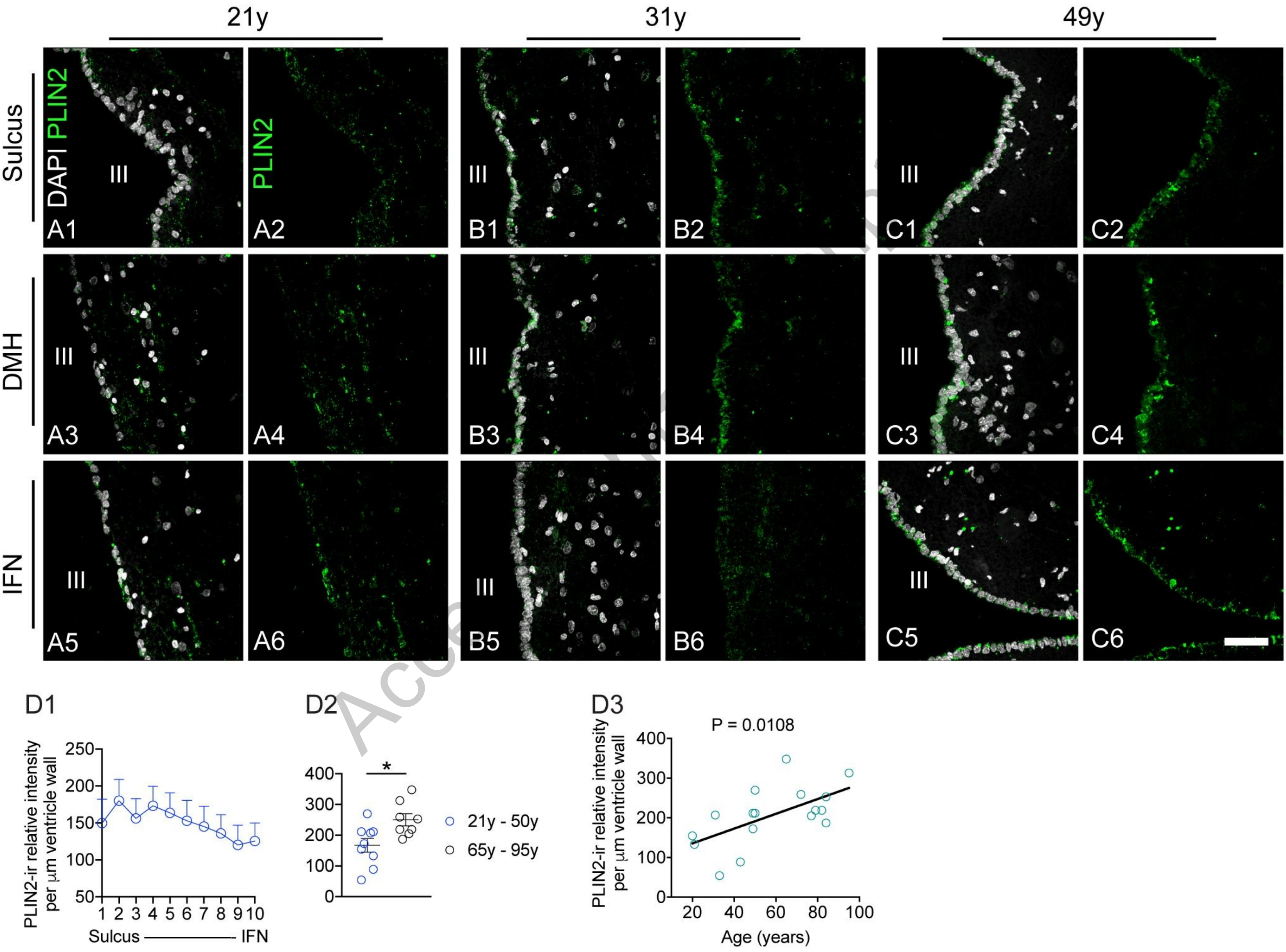

Bregma -0.34
Bregma -0.82
Bregma -1.82
PLIN2 NeuroTrace
III
SCN
A1
PVN
B1
ARC
C1
Top
DAPI GFAP
PLIN2
A2
A3
B2
B3
C2
C3
Middle
A4
A5
B4
B5
C4
C5
Bottom
A6
A7
B6
B7
C6
C7
DAPI PLIN2 GFAP
D1
PLIN2
D2
PLIN2 surfaced
D3
DAPI surfaced
D4
Top
Middle
Bottom
LD volume / Nuclei
0.20
0.15
0.10
0.05
0.00
Bregma -0.34
-0.82
-1.82
**
*
E


WT Chow
WT HFD
Ldlr KO HFD
Top
Middle
Bottom
DAPI GFAP
PLIN2
III
A1
A2
A3
A4
A5
A6
B1
B2
B3
B4
B5
B6
C1
C2
C3
C4
C5
C6
DAPI PLIN2 GFAP
PLIN2
PLIN2 surfaced
DAPI surfaced
D1
D2
D3
D4
WT chow
WT HFD
Ldlr KO HFD
LD volume / Nucleus
1.6
1.2
0.8
0.4
0.2
0.1
0.0
*
**
**
*
Top
Middle
Bottom
E


DAPI PLIN2
III
IFN
VMH
NTL
DMH
Sulcus
fx
TM
A1
A2
A3
A4
B1
B2
DAPI PLIN2
PLIN2
C1
control
T2DM
PLIN2-ir relative intensity per μm ventricle wall
400
300
200
100
1 2 3 4 5 6 7 8 9 10
Sulcus ———— IFN
C2
**
400
300
200
100
0
CTRL
T2DM
Sulcus
DMH
IFN
DAPI PLIN2
PLIN2
D1
D2
D3
D4
D5
D6
E1
E2
E3
E4
E5
E6


A
R = 0.075, P = 0.344
PLIN2-ir relative intensity
0
100
200
300
400
500
15
20
25
30
35
40
BMI
B
R = 0.022, P = 0.478
PLIN2-ir relative intensity
0
100
200
300
400
500
0
5
10
15
20
25
PMD (hours)
C
R = 0.011, P = 0.625
PLIN2-ir relative intensity
0
100
200
300
400
500
60
70
80
90
100
Age (years)


A1
DAPI PLIN2 GFAP
III
A2
B1
DAPI PLIN2 Iba1
B2


21y
31y
49y
Sulcus
DMH
IFN
DAPI PLIN2
PLIN2
III
A1
A2
A3
A4
A5
A6
B1
B2
B3
B4
B5
B6
C1
C2
C3
C4
C5
C6
D1
D2
D3
PLIN2-ir relative intensity per μm ventricle wall
250
200
150
100
50
1 2 3 4 5 6 7 8 9 10
Sulcus
IFN
400
300
200
100
0
*
21y - 50y
65y - 95y
P = 0.0108
400
300
200
100
0
20 40 60 80 100
Age (years)

Supplementary Table. Clinico-pathological data of younger subjects.

NBB Number	Sex	Age	Braak	PM	BMI	Cause of death and clinical diagnosis
1994-040	m	20	0	8.0	25	Heart failure, B cell lymphoma, viral pneumonia
2001-009	f	21	/	19.5	/	Myocard infart
1991-005	m	31	/	34	23	Necrotic tumor, adenocarcinoma
1984-026	f	33	/	24.75	22	Anoxia, lung oedema, tracheobronchitis, brain death
1980-006	f	43	/	50	/	Renal metastatic carcinoma
1994-118	m	49	0	/	20	Sepsis (E.coli), coloncarcinoma with metastasis
2011-069	m	49	0	6.5	24	Legal euthanasia, Hodgkin`s lymphoma, urolithiasis
1982-008	f	50	/	52.75	23	Coma; bronchopneumonia, carcinoma
1998-091	f	50	0	41	19	Legal euthanasia, metastatic Pancoust tumor

Macro files
Each file needs to be saved separately by the name mentioned in the first line of the macro

```
//file "1. lif2tif.txt"
/*
 * Extract images from a lif file
 * Images are RGB but for the current analysis, only the green channel is saved
 * New insights also require the blue channel.
 * Enter starting folder, process recursively all lif files, creating a subfolder for each
 * Name of subfolder reflects the name of the lif file
 * Name of the image files are created from series names in lif files.

 * Blue is in the first channel, green in the second and red in the third channel
 * (as opposed to the usual RGB ordering of pixel data).
 */
//last changes (for publication) 20191105 13:55
//by a.jonker@amsterdamumc.nl

//modified version of "lif2tif 20190418"

macro "Extract Timelapse from Lif as Tif"{
//avoiding the Ext functions
startTime = getTime();
close("*");
print("\\Clear");
run("Bio-Formats Macro Extensions");
startPath = getDirectory("Choose a Directory");
setBatchMode(true);//change to true for speeding up, should be false for debugging or you won't see images!
nLifFiles = processFolder(startPath);
setBatchMode(false);
close("*");
endTime = getTime();deltaTime=floor((endTime-startTime)/60);
showMessage("Processed "+nLifFiles+" Lif files in " + deltaTime + " seconds");
}

function processFolder(inPath){
processedFiles=0;
arrayOfNames = getFileList( inPath );
for(i=0;i<arrayOfNames.length;i++){
thisName=arrayOfNames[i];
if(File.isDirectory(inPath+thisName)){
processedFiles = processedFiles + processFolder(inPath + thisName);
}
}
for(i=0;i<arrayOfNames.length;i++){
thisName = arrayOfNames[i];//shows up in Debug window
if(endsWith(thisName,".lif")){
thisFolder=replace(arrayOfNames[i],".lif","");
outPath = inPath+thisFolder+File.separator;
bluePath = outPath+"blue"+File.separator;
//having a symmetrical solution in bluePath and greenPath would have been better
File.makeDirectory(outPath);
File.makeDirectory(bluePath);
if (!File.exists(outPath) || !File.exists(bluePath)){
errorMessage="Unable to create directory " + outPath;
exit(errorMessage);
}
processedFiles = processedFiles + extractGreen(thisFolder,inPath,outPath);
}
}
return processedFiles;
}

function extractGreen(fileName,inpath, outpath){//fileName has no extension, fileName is same as tail of outpath
close("*");//remove remaining windows
pf=0;//number of processed files
print("processing "+fileName+ " into "+outpath);//keep track of progress in big batches
run("Bio-Formats Macro Extensions");
//import headers in Original Metadata Window
//get an array with series names
//for each series
//could use FluoCubeName property as indicator of wavelength used
id = inpath + fileName+".lif";
Ext.setId(id);//use EXT functions of Bio-Formats extensions to read .lif files
Ext.getCurrentFile(file);
Ext.getImageCount(n);
Ext.getSeriesCount(seriesCount);

Ext.getFormat(inpath+fileName+".lif", format);
Ext.getSeriesCount(seriesCount);
for(s=0; s < seriesCount; s++){
Ext.setSeries(s);st=s+1;
Ext.getSeriesName(seriesName);
//string, format " Series X Name\t StringValue"
//string, format "Image name\tStringValue"
//a slash is sometimes found in the seriesName, as subfolder indicator
//slashes are a directory indicator that we don't want to handle; replace slash by undersocre
if(indexOf(seriesName,"/")!=-1){seriesName=replace(seriesName,"/","_");}
destFile = fileName + "/" + seriesName;
Ext.getImageCount(imageCount);//if 0 then this is a text file
if(imageCount > 0){
ext = ".tif";
}else{
ext="";
}
testname = seriesName + ext;
if(!File.exists(outpath + testname) || true){//force re-saving of blue channel; save time by removing ||true and skip old files.
print("\\Update:trying to load "+testname);
run("Bio-Formats Importer", "open=["+inpath+fileName+".lif"+"] use_virtual_stack autoscale color_mode=Default rois_import=[ROI
manager] view=Hyperstack stack_order=XYCZT series_"+st);
print("\\Update:Successfully loaded "+testname);
outFileName = outpath + testname;
print("\\Update:Saving "+outFileName);
if(ext == ".tif"){
Stack.setDisplayMode("color");
run("Stack to Images");
tws=getList("image.titles");
for(i=tws.length;i>0;i--){//close images that are not needed, save other images
thisTitle=tws[i-1];
selectImage(thisTitle);
foundPos=indexOf(thisTitle,"3/3");//image containing red (remember: order is blue, green, red)
if(foundPos!=-1){
close;//run("Close");
}

foundPos=indexOf(thisTitle,"2/3");
if(foundPos!=-1){
saveAs("tiff", outpath+seriesName+".tif");//save the green image. Not a symmetrical solution (would be
outpath/green/seriesName.tif).
close;//run("Close");
}
foundPos=indexOf(thisTitle,"1/3");
if(foundPos!=-1){
saveAs("tiff", outpath+"blue/"+seriesName+".tif");
close;//run("Close");
}
}
}
print("\\Update:Saving "+outFileName );
pf++;
}else{
print ("skipping already existing "+testname);
}
}
return pf;
}
//file "2. generate and clean up rois.txt"

/*
 * Generate a polyline region of interest that follows the edge of tissue in the image
 * The edge starts at an image's border, crosses the point farthest away from any border and exits on another border.
 *
 * These ROIs are stored in a zip file within the folder where tiff files are found
 * The macro recursively follows any folder from the chosen start path
 *
 * Last change (for publication) dd 20191105 12:36
 * By a.jonker@amsterdamumc.nl
 *
 * Outline of the macro:
 * Tissue is considered land, background is considered water. Try to find the shoreline.
 * Assumption is that the edge of the tissue is near the middle of the image
 * Therefore, any cracks, holes and other artifacts should mostly be distregarded when finding the shore
 *
 * If shoreline cannot be detected for any reason, draw a diagonal ROI, as a visual alert
 *
 * Failures occur and users must be aware of, or manually corrected if
 * -foreground and background are too similar (noisy background, sporadic signal in foreground)
 * -tissue has cracks touching the edge (shoreline enters and exits through the same image edge)
 * -tissue has cracks, which will be followed, but which do not represent edge tissue
 * -multiple parallel areas of low signal are present (the wrong edge may be chosen)
 *
 * Because of these shortcomings, resulting ROIs should be inspected manually.
 * To this end, a built-in switch can be used, the functions can be called in two ways:
 * -function argument = 1 -> tissue edge ROIs will be generated without user intervention
 *  Input: a folder with images containing tissue that touches at least one image edge
 *  Output: a zip file containing one ROI for each image

 * -function argument = 0 -> ROIs will be loaded onto images for inspecition and correction
 *  Input: a folder with a zip file containing named ROIs, and correspondingly named image files
 * (possible user interaction to amend or re-create the proper tissue edge)
 *  Output: a zip file containing one ROI for each image, indicating the tissue edge
 *
 * User intervention can take place in two ways after ROIs are loaded onto images
 * -Rois can be re-drawn/re-created using the (already active) broken-line tool
 * -Rois can be amended by deleting (alt-click) or relocating equidistant handles on the ROI
 *
 * If the user accepts the ROI, the file remains as is
 * If the user amends a ROI the roi is resampled and the original ROI.zip overwritten
 *
 * User inspection and intervention is slow.
 * Re-examining is prevented, and inspection can be sped up by skipping folders
 * Any folder name containing an asterix (*) will be skipped.
 * Bug fix 20191018, identified potential bug in 'restore selection'
 */
requires("1.52p");
generate=1;//can be changed to 0 for inspection
inspect=1-generate;//complementary boolean value

var logLineNr = 0;//vertical indent in log file allows for concise log
print("\\Clear");
if(nResults!=0){
IJ.deleteRows(0,nResults); //any previous result should be wiped
}
startPath = getDirectory("Choose a Directory");
close("*");
pf = processFolder(startPath,generate);
//pf = processFolder(startPath,inspect);
print ("analysed "+pf+" files");

function processFolder(inPath,generateOrInspect){
if(indexOf(inPath,"*")!= -1){//skip completed folders; the asterix can be, upon completion, manually added to a folder name
return 0;
}
generateload=generateOrInspect;
//(=1)-> generate without user intervention
//(=0)-> load, inspect, alter, accept and save
processedFiles=0;
arrayOfNames = getFileList( inPath );
for(i=0;i<arrayOfNames.length;i++){
thisName=arrayOfNames[i];//using a separate variable allows for inspection in debugger
if(File.isDirectory(inPath+thisName)){ //enter a depth-first search
print("\\Update"+logLineNr+": processing "+thisName);//logLineNr allows selective clearing
processedFiles = processedFiles + processFolder(inPath + thisName,generateload);
}
}
folderName=File.getName(inPath);
logLineNr++;
print("\\Update"+logLineNr+": processing "+folderName);//logLineNr allows selective clearing
if(generateload ==1){ //generate outlines without further intervention
processedFiles = processedFiles + generateRois(inPath);
nrois=roiManager("count");
if(checkProperRoiNames()!=0){ //Weird cases create exclamation marks in the ROI name
debug;
}
if(nrois>0){
if(File.exists(inPath+"roiSet.zip")){
//do not overwrite but move to date stamped copy
prevFile=inPath+"roiSet_before_"+ymdhs()+".zip";
File.rename(outFile, prevFile);
//or the more rude way: File.delete(inPath+"roiSet.zip");
}
if(checkProperRoiNames()!=0){
//sometimes an exclamation mark is preceding the roi/file name
debug;
}
roiManager("save",inPath+"roiSet.zip");
}

}else{ //load images and display for inspection with ROIs active and image contrast maximised
processedFiles = processedFiles + loadRois(inPath);
if (nImages() > 1){ //images have been found
run("Tile");
for(i=1;i<=nImages;i++){
selectImage(i);
run("Enhance Contrast", "saturated=0.35");
}
//ask user to accept the ROIs (with numbered begin-end points?) and store ROI's
updateDisplay();
accept= getBoolean("Do you need to change these outlines?\n"+thisName,"accept as is","change");

if (accept == 1) {
print("\\Update"+logLineNr+":accepted " + inPath); // no  need to re-save the roi
logLineNr++;
updateDisplay() ;
}else{
changeRois();//wait for the user to have modified ROIs and clicked OK
nrois=roiManager("count");//if rois were found
if(nrois>0){//overwrite the old ones
outFile = inPath+"roiSet.zip";
if(File.exists(outFile)){
prevFile=inPath+"roiSet_before_"+ymdhs()+".zip";
File.rename(outFile, prevFile);
// or the more rude way: File.delete(outFileName);
}
roiManager("save",outFileName);
}else{//no rois were found, delete the old roi set
if(File.exists(inPath+"roiSet.zip"))File.delete(inPath+"roiSet.zip");

}
}
}
}
if(nResults!=0)IJ.deleteRows(0,nResults);
close("*");
return processedFiles;
}

function loadRois(path){
//open the first encountered zip file in a folder and use that as ROI set
nRois = 0;
```

```
zipfiles=getFileList(path);
if(zipfiles.length == 0){
return 0;
}
i=0;found = 0;
do{
thisname=zipfiles[i];
fpn=path+zipfiles[i];
if(endsWith(fpn,"zip")){
found = 1;
}else{
i++;
}
}while (i<zipfiles.length&&found==0);
if(!endsWith(fpn,"zip")) return;
print("\\Clear");
if(nResults!=0)IJ.deleteRows(0,nResults);
close("*");
if(roiManager("count")>0){
roiManager("Deselect");
roiManager("Delete");
}
//path = File.openDialog("Select a roi (zip) File");
roiZip=File.getName(fpn);
dir = replace(path,roiZip,"");//this leaves the directory separator in place

roiManager("Open", fpn);
for(iRoi=0;iRoi<roiManager("count");iRoi++){
//get the name of the ROI, this is the same as the name of the image file the ROI belongs to
imName=call("ij.plugin.frame.RoiManager.getName", iRoi);
//test for names of images that we generated ourselves; exclude these usual suspects
if((!isOpen(imName)
&& indexOf(imName, "montage")==-1
&& indexOf(imName, "strip")==-1
&& indexOf(imName, "normalised")==-1
) !=0)
{
//the image is not (yet) open so attempt to importimage by that name
importImageName=dir+imName;
if(File.exists(importImageName)){
open(importImageName);
selectImage(imName);
roiManager("select",iRoi);
Roi.getCoordinates(xpoints,ypoints);
makeSelection("freeline",xpoints,ypoints);
run("Interpolate", "interval=10  adjust"); //put in adjustment handles
nRois++;
}else{
print("\\Update"+logLineNr+": Failed to locate image " + importImageName);
logLineNr++;
}

}
}
return nRois;
}
function generateRois(path){
//debug;
if(roiManager("count")>0){
roiManager("Deselect");
roiManager("Delete");
}
nRois = 0;
files=getFileList( path );
for(i=0;i<files.length;i++){
thisName=files[i];
if(indexOf(files[i],"tif")!=-1
&&indexOf(files[i],"Snapshot")==-1
&&indexOf(files[i],"strip")==-1
&&indexOf(files[i],"montage")==-1
){//only process Mark_and_find files
open(path+files[i]);
inImgID=getImageID();
findCoastline(inImgID);
nRois=roiManager("count");
print(nRois+" rois found ; i= "+i);
}
}
if (nRois == 0 ) {
//sometimes things go wrong, then offer debug option or fail silently
//debug;
}
return nRois;
}
function changeRois(){
//for every ROI in manager, get the image name, set roi, set ROI as editable
// this has been done already if changeROIs() is called after loading images and ROIs
setTool("polyline");
//wait for changes to be complete
waitForUser("Change ROIs to your satisfaction\nRemember to work from bio top to bio bottom\nThen click OK");
imagesTitlesList = getList("image.titles");
nImgs=imagesTitlesList.length; //now walk all open images
for(iImg=0;iImg<nImgs;iImg++){
imTitle=imagesTitlesList[iImg];
selectImage(imTitle);//get their title
st=selectionType() ;
if(st==6 || st == 7){
for(iRoi=0;iRoi<roiManager("count");iRoi++){
imName=call("ij.plugin.frame.RoiManager.getName", iRoi);
if(imName==imTitle){
//you can't just call 'update' as the roi manager is not aware of the new coordinates
//first harvest the current roi (this differs from the roiManager's current Roi!!) from the image
Roi.getCoordinates(xpoints,ypoints);
//select iRoi in roimanager to amend
roiManager("select",iRoi);
// re-create roi in image so roiManager is aware of changes
makeSelection("freeline",xpoints,ypoints);
//then do update the roi in the roiManager's list
roiManager("update");
}
}
}
}
return;
}
function findCoastline(imgID){
getDimensions(width, height, channels, slices, frames);
inTitle=getTitle();
//Tissue in the image covers at least 30% of the area.
aPart = 0.3 * width * height;
run("Select All");
run("Duplicate...", "title=coastline");//keep origial intensity data and work on copy
dupID=getImageID();
run ("Enhance Contrast", "saturated=0.35");//some images have very low signal
run("Apply LUT");//now image is bright, also to the eye. Data is no longer quantitative
run("Set Scale...", "distance=0 known=0 pixel=1 unit=pixel");//remove µm scale, eventually we want pixels anyway
rotateNeed=0;//might be changed to 1 by code further down
run("Gaussian Blur...", "sigma=2");//get a global intensity so small features disappear
setAutoThreshold("Default dark");
setOption("BlackBackground", true);
run("Convert to Mask");
run("Fill Holes");//remove any blobs and blebs in both the tissue...
run("Invert");
run("Fill Holes");//...and in the background
run("Invert");

run("Set Measurements...", "area mean standard modal min centroid center bounding shape integrated skewness area_fraction limit
redirect=None decimal=5");
setAutoThreshold("Default dark");
List.setMeasurements;
```

```
xm=List.getValue("XM");//get the center of mass of what is hopefully the tissue
ym=List.getValue("YM");
doWand(xm,ym);//create a ROI around this
run("Measure");
//the following code tries to figure out if selection is background or tissue
//and swaps found outlines if necessary
amean=getResult("Mean");//the current selection's mean
aarea=getResult("Area");
selectImage(imgID);//mesure in the original image
run("Restore Selection");//re-create the same area in the original image
run("Measure");
m1=getResult("Mean");
run("Make Inverse");
run("Measure");
m2=getResult("Mean");
if(m2<m1){//try alternative
run("Make Inverse");
}
selectImage(dupID);//continue to work with the duplicate.

//we already have a selection present here //run("Restore Selection");
if(m2<m1){//so also have the inverse in the duplicate, if necessary.
run("Make Inverse");
}

//aarea should now delineate tissue and run along the 'beach'
//amean should now be 255
//we are not interested in the entire ROI enclosing the tissue
//but only the area that touches the outer edge of the tissue
//
//find the roi coordinates farthest away from edge as
//this is a good starting point to find the shore line-part of the entire ROI
run("Interpolate", "interval=1");//walk pixel by pixel
Roi.getCoordinates(xpoints, ypoints);
dmax=0;//maximum distance to all edges
nmax=0;//ordinal of coordinate that is farthest away from all edges

for(i=0;i<xpoints.length;i++){
x=xpoints[i];
y=ypoints[i];
makePoint(x, y, "cross");//wait(delay); //if you want to SEE it happening, remove first two slashes
dx=min(x,width-x);//x-distance away from edges
dy=min(y,height-y);//y-distance away from edges
farthest=min(dx,dy);//should also check for convexity here?
if(farthest>dmax){//the current pixel is further away, keep this info
nmax=i;
dmax=farthest;
}
}
makeOval(xpoints[nmax]-2,ypoints[nmax]-2,5,5);//indicate the point found as the farthest from any image side
//waitForUser("Farthest");//uncomment if you want to SEE it happening
nmid=nmax;
i=nmid+1;//mod xpoints.length; go and inspect the next point of the (closed contour) ROI.
if(i==xpoints.length){i=0;}
do{//ordinally walk forward in coordinates, starting at the coordinate farthest away from any image edge,
//till we reach the image border
x=xpoints[i];
y=ypoints[i];
makePoint(x, y, "cross");//wait(delay);
i=(i+1);
if(i==xpoints.length){//oops, the beach is not [first..start______end..last] but [first___end...start___last]
i=0;
rotateNeed=1;//rotating is pushing out values on one side, moving them in on the other side of the array
}
}while(i<xpoints.length && x>0 && y>0 && x<width-1 && y<height-1 && i!=nmax);//rounding gives x=-0.00 which is false for x==0
// that is: wile we have not reached the end of the array,
//do not hit a wall left or right(x=0 or x=width-1)
//or top or bottom (y=0 or y=height-1)
//makeOval(x-2,y-2,5,5);//uncomment if you want to SEE it happening
//waitForUser("Forward");//uncomment if you want to SEE it happening
```

```
end=i;
if(rotateNeed!=0){
//from the polygon ROI we have selected part of it that describes the coastline
//index i now indicates how far the coastline runs; from 0-i and from ?? to nmax
//we rotate the array so the last point of the coastline is the last point of the array.
//elements that are 'rotated out' at the end, enter in the head of the array
xpoints=Array.rotate(xpoints,-i);
ypoints=Array.rotate(ypoints,-i);
nmax=nmax-i;
end=xpoints.length-1;
}else{
end=i;
}
i=nmax;
//
do{//walk backward in coordinates till we reach the image border
i=(i-1);
if(i==-1){//oops, we backwar we ran out of the front of the array
i=xpoints.length -1;//so we enter in the end of the array
}
x=xpoints[i];
y=ypoints[i];
print("\\Update"+logLineNr+":x = "+x+" y = "+y+" i = "+i);
makePoint(x, y, "cross");//wait(2);
//if(i==0) debug;//complex expression; save in variable for debugging purposes:
finished=!(i<xpoints.length && x>0 && y>0 && x<width-1 && y<height-1 && i!=nmid);
}while(!finished);

makeOval(x-2,y-2,5,5);
//waitForUser("Backward");//if you want to SEE it happening halt until the user has viewed the image

start=i;

//nmin is first coordinate nearest to edge of image and at start of coastline, nmax the last of the coastline
if(start>end){
ror=xpoints.length-start;//rotate over right distance
Array.rotate(xpoints,ror);//elements that shift out right, shift in left.
Array.rotate(ypoints,ror);
start=0; end=end+ror;
}
s=min(start,end)+1;
e=max(start,end)-1;
makeArrow(xpoints[s],ypoints[s],xpoints[e],ypoints[e],"notched"); //quicklook visualisation
//waitForUser("Direction");//uncomment if you want to SEE the direction the beach was travelled
xpoints=Array.slice(xpoints,s,e);
ypoints=Array.slice(ypoints,s,e);
makeSelection("freeline",xpoints,ypoints);//the actual visualisation of the beach
//
if(xpoints.length < 2){//generate diagonal if no roi has been found.
makeLine(0, 0, width/2, height/2,width,height);
//waitForUser("No beach");//uncomment if you want to SEE in case no beach was found

}
//store the shoreline ROI in the roiManager and name it after the image it was obtained from
run("Interpolate", "interval=5 smooth adjust");//iron out small deviations
roiManager("add");
roiManager("select",roiManager("count")-1);
roiManager("rename",inTitle);
Roi.getCoordinates(xpoints,ypoints);
close;
makeSelection("freeline",xpoints,ypoints);//draw the outline in the original image
return;
}
function checkProperRoiNames(){
//if an exclamation mark or a dash is present in any ROI name
nRois=roiManager("count");
OK=0;
for(i=0;i<nRois;i++){
imName=call("ij.plugin.frame.RoiManager.getName", i);
if(indexOf(imName,"!")!=-1 || indexOf(imName,"-")!=-1){
```

```
print("Roi name of roi number "+i+" contains ! or - ");
return 1;
}
}
return 0;
}

function min(a,b){
if(a>b)return b; else return a;
}
function max(a,b){
if(a<b) return b;else return a;
}
function abs(a){
if(a<0)return a*-1;else return a;
}
function ymdhs(){
getDateAndTime(year, month, dayOfWeek, dayOfMonth, hour, minute, second, msec);
yes=IJ.pad(year,4);
mos=IJ.pad(month,2);
das=IJ.pad(dayOfMonth,2);
hos=IJ.pad(hour,2);
mis=IJ.pad(minute,2);
ses=IJ.pad(second,2);
ymdhsString=yes+mos+das+"_"+hos+"."+mis+"."+ses;
return ymdhsString;
}
// file "3. straighten ROIs.txt"
/* The current macro is the third (3.) step in the process of measuring brain tissue
 * Input:
 * -a roiSet.zip file containing ROIs
 * -image files containing quantitative fluorescent signal
 * -a text file containing information on the relative image locations
 *
 * Output:
 * - strip files, one strip per image, containing the straightened tissue edge stripe
 * - a montage, containing all pixels in a stripHeight-pixel wide neighbourhood of the tissue edge
 * - a scaled back image of 1000 and of 100 pixel wide, stripHeight pixels high
 * - numeric measurements representing the total fluorescence in the 0-100% length of the tissue edge
 *
 * Method
 * Non-overlapping mosaïc images of brain tissue were recorded, for details see macro (1.)
 * Foreground and background of each recorded image was found using a separate macro (2.)
 *
 * The placement of the tissue section on the object glass necessitated a certain direction of recording
 * The edge of the tissue was followed either from Top to bottom or Bottom to top
 * Likewise, Left to right or Right to left indicates direction of recording
 * The directionality of the set of tiles was indicated by one of these capital four letters.
 * The directionality string contains the appropriate set from the letters BLRT, max length 4.
 * The letters indicating the order of borders to inspect in the image for a border-crossing of tissue edge
 *
 * For each image of the mosaïc the tissue border was outlined by a Region Of Interest (ROI)
 * this was done in the macro in file "2. generate and clean up rois"
 * In the current macro, these ROIs are straightened into strips, one strip per image
 * Strips are annealed to one long strip, the montage image
 * Pixels in the montage image representing 0-100% of the tissue edge found in the images.
 * Perpendicular, on either side of the tissue edge, 40 pixels were taken into consideration
 *
 * From the resulting arbitrary wide and stripHeight pixels high montage, fluorescence was measured
 *
 * Last change 20190930 16:59
 *
 */

var debugger=false;//set to true for inspection of code execution
var stripHeight = 80;//column of pixels perpendicular to the tissue edge, area that is extracted into strips
//strip is divided into four sections, each stripHeight/4 high. Two of them do not contain tissue, the other two do.
//Because cells are polarised, the nucleii will be far furhter from tissue edge, so the Blue channel is sampled 40 pixel-wide
//(green) lipid droplets are closer to edge, there only 20 pixel wide would do. For now, we consider a 40-wide pixel column.
```

```
//Select the root folder of files to process
directionsFile=File.openDialog("Select the file sampledirections.txt");
//get folderlist
//open directionality file
//debugging runs into the Main function automatically.
macro "Main"{
//setBatchMode(true);//for faster execution, images will not be updated during processing
g=Main("");
setBatchMode(false);//we want to see the execution of the application for blue channel images
b=Main("blue/");
setBatchMode(false);
if(g!=b){//g should be equal to b
showMessage("Found "+g+" green edges and "+b+" blue edges");
}
}
macro "Open Zipped Rois"{
close("*");print("\\Clear");
if(roiManager("count")!=0){roiManager("deselect");roiManager("delete");}
path = File.openDialog("Select a roi (zip) File");
openRoiZipImages(path,"",true);
run("Tile");

}

function Main(blue){
fn=File.getName(directionsFile);
path=replace(directionsFile,fn,"");
if(!File.exists(directionsFile)){
return;//without information on the direction of the border we can't reconstruct
}

directions = File.openAsString(directionsFile);
lines = split(directions,"\n");

//loop through directionality list
for(lineNo=0;lineNo<lines.length;lineNo++){
line = lines[lineNo];
chunks = split(line,"\t");
fileNamePart = chunks[0];
direction = chunks[1];
//loop through folders in folder list
arrayOfNames = getFileList(path );

for(i=0;i<arrayOfNames.length;i++){
thisName=arrayOfNames[i];
print("Analysing "+ thisName);
//if directionality name is in folder name
partInFile=indexOf(thisName,fileNamePart);
if(File.isDirectory(path+thisName) && partInFile>-1){
if(nResults!=0)IJ.deleteRows(0,nResults);//start with clean Results slat
close("*");//close any remaining open images
if(roiManager("count")>0){//remove any remaining ROIs in the manager
roiManager("Deselect");
roiManager("Delete");
}
//let zip file open all associated images and activate their rois
//will disable the selection of images later on so do not use setBatchMode(true);
pf=processZipfile(path + thisName ,blue );
if(nImages()!=0){
run("Tile");//show an overview of all images with their ROI
processedFiles = processedFiles + pf;
//loop through roi names (they equal (now opened) image window names!)
totalWidth = 0;
nRois=roiManager("count");
stripIDs=newArray(nRois);//bookkeeping for fast access of images
stripWidths = newArray(nRois)//bookkeeping for total width and starting positions
for(roiNr = 0; roiNr<nRois;roiNr++){
imgName=call("ij.plugin.frame.RoiManager.getName", roiNr);//get image title from ROI name
selectImage(imgName);//select the appropriate image
roiManager("select",roiNr);//get its ROI into the image display
run("Interpolate", "interval=1");//also with sparse ROIs we need to pick up all points.
myFlip=flipNeeded(direction,roiNr); //see function for explanation
//straighten curved ROIs into a straight strip(e) of pixels
stripName = "strip_"+IJ.pad(stripHeight,2) +"_"+ IJ.pad(roiNr+1,2)+".tif";
run("Properties... ", "  width="+stripHeight);//set line width to desired size
run("Straighten...", "line="+stripHeight+" title="+stripName);
//flip if necessary
if(myFlip!=0){
run("Flip Horizontally");
}
stripIDs[roiNr]=getImageID();//for numeric reference
//save roi at full size with name stripX, where X is [01,..]
setOption("ScaleConversions", false);//we don't want scaling of grey values
run("8-bit");//we want raw 8-bit values!
fn=path+thisName+blue+stripName;
saveAs("Tiff", fn);
getDimensions(width, height, channels, slices, frames);
//(keep roi open)
stripWidths[roiNr] = width;
totalWidth = totalWidth + width;
}
//anneal all rois to one big strip @height=stripHeight

newImage("strip", "8-bit black", totalWidth, stripHeight, 1);
montageID=getImageID();
leftPos=0;
for(stripNr = 0;stripNr<nRois;stripNr++){
//select strip
selectImage(stripIDs[stripNr]);
run("Select All");
run("Copy");
//select montage
selectImage(montageID);
//create paste rectangle
sw=stripWidths[stripNr];
makeRectangle(leftPos,0,sw,height);
run("Paste");
leftPos=leftPos + sw;
}
//save annealed image
thisUName = replace(thisName,"/","_");
fn = thisName +blue+ thisUName + "montage_"+IJ.pad(stripHeight,2) + ".tif";
fn = path + fn;
saveAs("Tiff",fn);
}
}
}
}
close("*");
return 0;
}
function processZipfile(inPath,blue){
processedFiles=0;
arrayOfNames = getFileList( inPath );
for(i=0;i<arrayOfNames.length;i++){
thisName = arrayOfNames[i];
if(endsWith(thisName,"roiSet.zip")){
//make total intensity fluorescence in the column
//save as 1000 values
// use the sum of fluorescence per 1/100 and per 1/1000 of the total length of the edges.
processName = inPath + thisName; //shows up in Debug window
processedFiles = processedFiles + openRoiZipImages(processName,blue,0);
a=5;//debugger decoy, useless assignment needed for variable inspection during single stepping
//now all images belonging to this roiManager zip file are open with the ROIs selected
}
}
return 1;
}

function openRoiZipImages(path,blue,makeHandles){
```

```
//open a zip file (point at the zip file using Finder dialog)
//zip file contains rois, one roi per image, with roi name is image(file) name
//open every image by name, in same folder as zip file, name taken from roiManager entry

roiZip=File.getName(path);
dir = replace(path,roiZip,"");//this leaves the directory separator in place
roiManager("Open", path);
for(iRoi=0;iRoi<roiManager("count");iRoi++){
imName=call("ij.plugin.frame.RoiManager.getName", iRoi);
if(!isOpen(imName)){
//attempt to importimage by that name
importImageName=dir+blue+imName;
if(File.exists(importImageName)){
open(importImageName);
selectImage(imName);
roiManager("select",iRoi);
if(makeHandles){//convert the possible fine grained, or very coarse ROI to smooth handles
Roi.getCoordinates(xpoints,ypoints);
makeSelection("freeline",xpoints,ypoints);
run("Interpolate", "interval=10  adjust"); //put in adjustment handles
}
run("Enhance Contrast", "saturated=0.35");//in case of faint images
}
}
}
return 1; //one zip file processed
}
function flipNeeded(direction,roiNr){
Roi.getCoordinates(xpts,ypts);
getDimensions(width, height, channels, slices, frames);
dtet = 10;//distance-to-edge tolerance
l=xpts.length-1;//last coordinate
determined = false;
index = 0;
sc=0;
toofar=false;
direction = toUpperCase(direction);
do{
pd = substring(direction,index,index+1);//primary direction
if( pd == "T"){
if(ypts[0]<ypts[l] ){ //y[0] is nearest tot top
if(ypts[0] < dtet) { //y[0] is close enough to image's top edge
sc = 0;
determined = true;
}else{//not close enough, try again
toofar=true;
}
}else{//ypts[l] is close to the top
if(ypts[l] < dtet) { //y[l] is close enough to image's top edge
sc = l;
determined = true;
}else{//not close enough, try again
toofar=true;
}
}
}
if(pd == "B"){
if(ypts[0]>ypts[l] ){ //y[0] is nearest tot bottom
if(ypts[0] > (height - dtet)) { //y[0] is close enough to image's bottom edgee
sc = 0;
determined = true;
}else{//not close enough, try again
toofar=true;
}
}else{//ypts[l] is close to the bottom
if(ypts[l] > (height - dtet)) { //y[l] is close enough to image's bottom edge
sc = l;
determined = true;
}else{//not close enough, try again
toofar=true;
}
}
}
if(pd == "L"){
if(xpts[0] < xpts[l] ){ //x[0] is nearest tot left edge
if(xpts[0] < dtet) { //y[0] is close enough to image's left edgee
sc = 0;
determined = true;
}else{//not close enough, try again
toofar=true;
}
}else{//xpts[l] is nearest to the left edge
if(xpts[l] < dtet) { //x[l] is close enough to image's left edge
sc = l;
determined = true;
}else{//not close enough, try again
toofar=true;
}
}
}
if(pd == "R"){
if(xpts[0] > xpts[l] ){ //x[0] is nearest tot right edge
if(xpts[0] > (width - dtet)) { //y[0] is close enough to image's right edge
sc = 0;
determined = true;
}else{//not close enough, try again
toofar=true;
}
}else{//xpts[l] is nearest to the left edge
if(xpts[l] > (width - dtet)) { //x[l] is close enough to image's rightt edge
sc = l;
determined = true;
}else{//not close enough, try again
toofar=true;
}
}
}
index++;
}while(index < lengthOf(direction) && determined == false);
if(!determined && toofar){
//use [0] or [l] based on first character of direction, disregard distance to edge
pd = substring(direction,0,1);flipstring = "flip the direction";flip=0;
if(pd == "T"){if(ypts[0]<ypts[l])flip = 0;else flip = 1;}
if(pd == "B"){if(ypts[0]>ypts[l])flip = 0;else flip = 1;}
if(pd == "L"){if(xpts[0]<ypts[l])flip = 0;else flip = 1;}
if(pd == "R"){if(xpts[0]>ypts[l])flip = 0;else flip = 1;}
if(flip ==1)flipstring = "not "+flipstring;
//waitForUser("far away","based on "+pd+" we chose to "+flipstring);
return flip * l;
}
return sc != 0;
}

//analyse strips.txt
requires("1.52p");
var debugger=true;

/* Last change 20190930 11:25
 * written by a.jonker@amsterdamumc.nl
 *
 * Input:
 * image representing (green) fluorescent lipid droplets
 * image representing (blue) nuclear signal
 * Output: csv file
 *
 * Images contain straightened ROIs with 4 zones of equal size (usually 10 pixels high, arbitrary wide)
 * Images blue and green contain in 4 zones respectively tissue background, droplet signal, background: 1 near and 1 far from
tissue edge
 * Analyse Strips macro detects if tissue is in upper or lower part.
 * Green signal in tissue background is considered background, mean is measured and used as lower threshold in tissue edge.
 * Same procedure is followed for measuring blue tissue background and tissue edge.
 * Results of threshold value, net signal, and area (expressed in pixels) are collected for green and blue.
 * Measurement results are saved to Excel-readable comma separated file
 *
 * For visual inspection, the variable 'debugger' can be set to 'true', after which the macro is executed slowly
 * If debugger is set to false, processing takes a few minutes.
 *
 *
 *
 */



macro "Analyse Strips"{
close("*");
pattern = "montage_80.tif";//later: montage_"+IJ.pad(stripHeight,2);
normaliseTo=100;
startPath = getDirectory("Choose a Directory");
nFiles=analyseStrips(startPath,pattern);
print("Processed "+nFiles+" files");
}


function analyseStrips(path,pat){
arrayOfNames = getFileList(path );
nSubstrips=4;
processedFiles=0;
substripVals=newArray(nSubstrips);
run("Set Measurements...", "mean modal min integrated limit redirect=None decimal=0");
for(i=0;i<arrayOfNames.length;i++){
thisName=arrayOfNames[i];
if(File.isDirectory(path+thisName) && indexOf(thisName,"blue")==-1){
processedFiles = processedFiles + analyseStrips(path + thisName,pat);
}
}
for(iNames=0;iNames<arrayOfNames.length;iNames++){
thisName=arrayOfNames[iNames];
updateResults();
blueDaughter=path+"blue/"+thisName;
parentName=File.getName(path);
if(indexOf(thisName,pat)!= -1 && indexOf(thisName,".tif")!=-1 && File.exists(blueDaughter)){
print("\\Clear");
print("\\Update0:processing " + parentName);
run("Clear Results");
close("*");
//open strip
pn="blue/"+thisName;
pn=path+pn;
open(pn);
blueImageID=getImageID();
rename("Blue");
run("Select All");;
run("Duplicate...", "title=bluegauss"); //if we can't make nice outlines
dupGauss=getImageID();
getDimensions(width, height, channels, slices, frames);
//these arrays hold coordinates for a polyLine ROI that separates cells in foreground from tissue background
var edgeArrayY = newArray(width);//add 2 for corners of enclosed area
var edgeArrayX = newArray(width);

var sigAreaArray = newArray(width); //area for blue and green is the same.
var blueSigValueArray = newArray(width);//background area is heigt/2 -SigArea
var bgAreaArray = newArray(width); //is not strictly necessary, can be calculated from sigAreaArray
var blueBgValueArray = newArray(width);

var greenSigValueArray = newArray(width);//area is same as blue
var greenBgValueArray = newArray(width); //area is same as blue


open(path+thisName);
greenImageID=getImageID();
```

```
rename("Green");
run("Tile");
substripHeight = height / nSubstrips;
//measure fat droplet strip
selectImage(greenImageID);//best find tissue using green channel
for(iStrips=0;iStrips<nSubstrips;iStrips++){
makeRectangle(0,substripHeight*iStrips,width,substripHeight);
meanVal=getValue("Mean");
substripVals[iStrips]=meanVal;

}
//four substrips: either upper 2 or lower 2 represent the tissue
topVal = substripVals[0]+substripVals[1] ;
bottomVal =  substripVals[2]+substripVals[3];
halfHeight = height/2;
yStart=halfHeight;
//decide if tissue is in the upper half or lower half of the image
//if in the lower half,  flip lower and top half
if(topVal < bottomVal){ //tissue is in the lower part, flip that part vertical, so tissue is in upper part
selectImage(greenImageID);
run("Select All");
run("Flip Vertically");

selectImage(blueImageID);
run("Select All");
run("Flip Vertically");

selectImage(dupGauss);
run("Select All");
run("Flip Vertically");
}
//the tissue edge is near the middle of the image.
//flip top half (containing tissue now) so tissue edge is at top of image; easier for for-loops.

selectImage(greenImageID);
makeRectangle(0,0,width,height/2);
run("Flip Vertically");

selectImage(blueImageID);
makeRectangle(0,0,width,height/2);
run("Flip Vertically");

selectImage(dupGauss);
makeRectangle(0,0,width,height/2);
run("Flip Vertically");

//first find the division line based on the blue signal. (Then measure foreground and background signals in both original images.)
//as selectImageID is very slow, don't measure whiledetecting the division line.
selectImage(dupGauss);
setBatchMode(true);
for(x = 0; x < width; x++){//walk the strip
makeRectangle(x,0,1,halfHeight);
max=getValue("Max");//highest value of the current strip
halfMax=max/2;
pv=0; //pixel value
y=-1; //want increment in loop, want exit value to be pointing at half max location
do{//find max position of column, there might be more?
y++;
pv = getPixel(x,y);

}while(y < halfHeight && pv < max);
while(y < halfHeight && pv > halfMax){ //walk downhill till half max in column
y++;
pv = getPixel(x,y);
}
//y points to the first position of the background
//as gathered from the blue channel (= DNA intensity)
edgeArrayY[x] = y-1; //the foreground was left already, y points to first pixel of background, so edge should be one bak

while (y < halfHeight){//while loop needs to be here, if y == halfHeight skip this loop
//walk down hill until end of image
pv = getPixel(x,y);
y++;
}
edgeArrayX[x]=x;
}


for(x = 0; x < width; x++){//set ROI to shape in original image
sigAreaArray[x] = edgeArrayY[x];//easiest moment to extract signal area
if(topVal > bottomVal){//flip  coordinates of signal rectangle wrt. top half
edgeArrayY[x] = halfHeight - edgeArrayY[x];
}else{//flip  coordinates of signal rectangle wrt. bottom half
edgeArrayY[x] = halfHeight + edgeArrayY[x];
}
edgeArrayX[x]=x;
}

selectImage(blueImageID);
//warning: using sigArea!
makeSelection("polyline",edgeArrayX,sigAreaArray);
//warning: used sigArea!
for(x = 0; x < width; x++){//set ROI to shape of original image
makeRectangle(x,sigAreaArray[x],1,halfHeight-sigAreaArray[x]);
blueBgValue=getValue("IntDen");
blueBgValueArray[x] = blueBgValue;

makeRectangle(x,0,1,sigAreaArray[x]);
blueSigValue=getValue("IntDen");
blueSigValueArray [x] = blueSigValue;
}

selectImage(greenImageID);
for(x = 0; x < width; x++){//set ROI to shape of original image
makeRectangle(x,sigAreaArray[x],1,halfHeight-sigAreaArray[x]);
greenBgValue=getValue("IntDen");
greenBgValueArray[x] = greenBgValue;

makeRectangle(x,0,1,sigAreaArray[x]);
greenSigVal=getValue("IntDen");
greenSigValueArray[x] = greenSigVal;
}
if(debugger){setBatchMode(false);debug;}

//flip back tissue edge to middle of image, so it resembles original again
selectImage(greenImageID);
makeRectangle(0,0,width,height/2);
run("Flip Vertically");

selectImage(blueImageID);
makeRectangle(0,0,width,height/2);
run("Flip Vertically");

selectImage(dupGauss);
makeRectangle(0,0,width,height/2);
run("Flip Vertically");
//if tissue was in lower half of image, restore by flipping entire image vertically
if(topVal < bottomVal){ //tissue is in the lower part, flip that part vertical, so tissue is in upper part
selectImage(greenImageID);
run("Select All");
run("Flip Vertically");

selectImage(blueImageID);
run("Select All");
run("Flip Vertically");

selectImage(dupGauss);
run("Select All");
run("Flip Vertically");
}
```

```
//draw edge of signal
selectImage(blueImageID);
makeSelection("polyline",edgeArrayX,edgeArrayY);
selectImage(greenImageID);
makeSelection("polyline",edgeArrayX,edgeArrayY);
if(roiManager("count")>0){
roiManager("Deselect");
roiManager("Delete");
}
parent = File.getName(path);
roiManager("add");
roiManager("select",roiManager("count")-1);
roiManager("rename",parent);

roiManager("save",path+parent+"edge.roi");


fn = parent + "_normalised_"+greenSigValueArray.length+".csv";fn =  path + fn;
l=greenSigValueArray.length;
reportResults(fn,greenSigValueArray,greenBgValueArray,blueSigValueArray,blueBgValueArray,sigAreaArray,l);
fn = parent + "_normalised_"+"100"+".csv";fn =  path + fn;
reportResults(fn,greenSigValueArray,greenBgValueArray,blueSigValueArray,blueBgValueArray,sigAreaArray,100);
fn = parent + "_normalised_"+"10"+".csv";fn =  path + fn;
reportResults(fn,greenSigValueArray,greenBgValueArray,blueSigValueArray,blueBgValueArray,sigAreaArray,10);
close("*");
processedFiles++;
}
}
return processedFiles;
}
function reportResults(pfn,gs,gb,bs,bb,sa,normaliser){
chunk = floor(gs.length/normaliser);counter = 0;
gsx=0;gbx=0;bbx=0;sax=0;bsx=0;//running counters for normalising to normaliser values in output
run("Clear Results");
row = 0;
for(x=0;x<gs.length;x++){
gsx=gsx+gs[x];
gbx=gbx+gb[x];
bbx=bbx+bb[x];
sax=sax+sa[x];
bsx=bsx+bs[x];
counter ++;
if(counter >= chunk){
setResult("Green Signal",row,gsx);
setResult("Green Background",row,gbx);
setResult("Blue Background",row,bbx);
setResult("Signal Area",row,sax);
setResult("Blue Signal",row,bsx);
updateResults();
counter = 0;gsx=0;gbx=0;bbx=0;sax=0;bsx=0;
row++;
}
}
//export results
saveAs("results", pfn);
```