Machine Learning Enabled Brain Segmentation for Small Animal Image Registration

Hendrik Klug[1] Horea-Ioan Ioanas[2] Markus Rudin[2]
[1]Department of Information Technology and Electrical Engineering, ETH
[2]Institute for Biomedical Engineering, ETH and University of Zurich

Abstract — Cross-subject and cross-study comparability of imaging data in general, and magnetic resonance imaging (MRI) data in particular, is contingent on the quality of registration to a standard reference space. Current methods rely on full image processing, with high varying intensities outside the Region Of Interest that interfere with registration. Applying the processing to a masked image improves the quality of the latter. Here, we present as an additional step in the SAMRI registration workflow, a deep learning enabled framework for segmentation of brain tissue in functional and structural MR images.

Background

In order to make meaningful comparisons across subjects inside a study, it is imperative that the images lie in a standard reference frame. Because of positioning imprecision and anatomical animal variations, this is not the case for the original MR acquired images. To solve this issue, the images need to be projected into the reference frame via registration [1, 2]. As reported by Ioanas et al. [3], the general approach for mouse-brain image registration is to use high-level functions designed and optimized for human brain images. This requires the mouse-data to be adapted to the processing function instead of vice-versa. To provide contrast, they compare two workflows, the Legacy workflow that adapts the data to the processing functions and an optimized Generic workflow, which is optimized to the data. While the Legacy workflow expands voxel size and deletes orientation information of the affine matrix in order to fit human brain data, the Generic workflow uses functions provided by the ANTs package, with spatial parameters adapted to the mouse brain. In a quality control, it is shown that the Generic workflow improves volume conservation, smoothness conservation and shows a reduction in variance.

Intensities outside the brain region of a mouse MR image present high variations and bias the registration process. To combat this, ANTs enables the user to define a mask to focus the optimization effort on the region of interest (ROI) [4]. The Generic workflow uses a template mask defined in the reference space to indicate the ROI. This template is not ad hoc to each subject and often includes the skull which contains high intensity variations. This can lead to stretching or skewing of the brain in order to fit unwanted intensities inside the mask region. As a remedy, it would be useful to extract the region of interest for each individual and continue the registration on an ad hoc region of interest. For this purpose, we propose a machine learning enabled brain extraction in an additional node to the Generic workflow presented by Ioanas et al. in [3]. The additional node creates a mask of the brain region using a classifier. The image is then masked such that only the region of interest remains. The Generic workflow then continues with the masked image.

Convolutional Neural Networks

In recent years it has been shown that convolutional neural network give the best results for semantic image segmentation in terms of precision and flexibility [5, 6]. Training a convolutional neural network into a classifier is a supervised method, meaning that the model needs to learn its parameters based on observations of data.

The training data set of a classifier is as important as the architecture of the model itself. To improve general-purpose application, training examples need to be drawn from a usually unknown probability distribution, that is expected to be representative of the space of occurrences. We define the space of occurrences as the space of which the data of interest is drawn from. In our case this consists of all the different mouse brain MRI data sets coming from multiple experiments, with their corresponding labels. Ideally experiment setups are uniform and the resulting data does not differ much, but small variations in the experiment setup and animal size are unavoidable. Based on an approximation of the occurrence space, the network has to build a general model that enables it to extrapolate and produce sufficiently accurate predictions in new cases. Manually creating annotations as required to train a deep-learning classi-

fier for high-resolution data is often infeasible, since it requires manual expert segmentation of vast amounts of slices.

Here we take as data set, images that were registered through the SAMRI workflow into a reference frame defined by the Toronto Hospital for Sick Children Mouse Imaging Center [**?**]. A mask that was made in the same reference frame was used as ground truth. Because the registration process is not perfect, the mask does not always align perfectly with the brain region of every slice. While our purpose was to create a workflow that generates better masks than the one from the template space, we showed that the latter could be used as training data for the deep-learning model, by applying small changes to them.

The Classifier

Model

As the architecture of the classifier, the U-Net from Ronneberger et al [6] was chosen based on its high performance in the field of biomedical image segmentation. This is a convolutional neural network that consists of a contracting path that captures context in addition to a symmetric expanding path that enables precise localisation. Localisation in this context means that a class label is assigned to each pixel. We used the U-Net implementation from zhixuhao [7], written in Keras. Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. It allows for a easily readable code and makes it thus easier to reproduce.

The implementation of the U-Net from zhixuhao has two drop-out layers in addition to the original one. A drop-out layer randomly sets a fraction of input units from the layer to 0 at each update during training time. The set fraction rate is 0.5. It is known that dropout helps prevent overfitting and greatly improves the performance of deep learning models [8].

Three losses were tested for the training of the model, namely the Dice-loss, the binary-cross-entropy loss and the sum of both.

The Dice-loss is computed from the Dice score. It calculates the similarity of two binary samples X and Y with

$$D_{coef} = \frac{2|X \cap Y|}{|X| + |Y|} \tag{1}$$

It is a quantity ranging from 0 to 1 that is to be maximised. The loss is then calculated with $1 - D_{coef}$. Because the Dice loss is not differentiable, small changes need to be made. In our case, the two samples to be compared are normalised, grey valued images and are thus not binary but have values between 0 and 1. Additionally, instead of using the logical operation *and*, element wise products are used to approximate the non-differentiable intersection operation. To avoid a division by zero, +1 is added on the numerator and denominator.

Because many more pixels in the masks are 0 than 1, there is a class imbalance problem. It is a problem, because in this case a false positive gives a much higher loss than a false negative. For example, predicting only black would give an acceptable loss, while predicting only white pixels would not. Using the Dice coefficient as a loss function for training should make it invariant to this class imbalance problem as stated by Fausto Milletari et al. in [9].

The binary cross-entropy loss, also called Log loss, is defined by:

$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(p(y_i)) + (1-y_i) \cdot log(1-p(y_i)) \tag{2}$$

For pixel values of 0 and 1, it adds $log(p(y))$ for each white pixel (y=1) and $log(1 - p(y))$ for every black pixel (y=0) to the loss.

We quickly realised that the Dice-loss gives the best results for our task and therefore used it to train the model.)TODO:compare results)

Data Set

The data set consists of 3D MR images taken from an aggregation of three studies; irsabi , opfvta [10], drlfom [11] and other unpublished data, acquired with similar parameters. TODO:cite

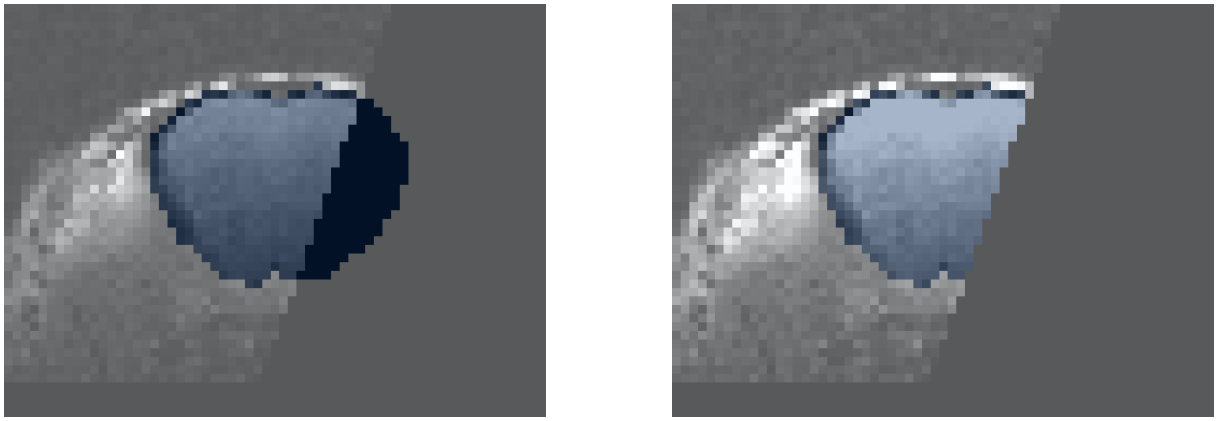
The images are transformed into a standard space with one defined mask via SAMRI [12] and are thus defined in the same affine space. SAMRI is a data analysis package of the ETH/UZH Institute for Biomedical Engineering. It is equipped with an optimized registration workflow and standard geometric space for small animal brain imaging [3].

Because of variance in mouse brain anatomy and in the experiment setup, some of the transformed data do not overlap perfectly with the reference template. To filter these images out, most of the incongruent slices were removed manually from the data set.

For the registration of the images, a padding was needed to make the originally not affine space affine. Due to this, the 3D volumes present many zero-valued slices, some of them overlapping with the mask.

Since it is not wanted for the model to predict a mask on black slices, the mask is set to zero where the image is as well. Because some pixels representing the brain tissue are zero-valued, holes result from this operation. To patch these, the function *binary_fill_holes* from scipy.ndimage.morphology [13] is used. An example of this can be seen in fig. 1.

In the coronal view, each slice of the transformed data is originally of shape (63, 48), matching the reference space resolution of 200 µm. It is then reshaped into (128, 128) by first zero-padding the smaller dimension to the same size of the bigger one and then reshaping the image into 128 using the function *cv2.resize* from the opencv python package [14].



(a) Example of an unpreprocessed slice.

(b) Example of a preprocessed slice.

Figure 1: The preprocessing removes the mask there, where the image-pixelvalues are 0. Plots of the same image, superposed with the template mask, with and without preprocessing.

Finally, the images are normalised by first clipping them from the minimum to the 99th percentile of the data in order to remove outliers and then divided by the maximum.

The data set is separated into Training, Validation and Test sets such that 90% of the total data are used for training and validation while 10% are used for testing. The Validation set is used for the optimisation of hyper parameters while the Test set is used as a measure of extrapolation capability. Images from the irsabi study are only used for quality control of the registration and are thus unknown to the classifier. This allows for a better estimation of the general performance of the workflow.

Data Augmentation

Because of diverse settings in the experiment setup, including animal manipulations causing artifacts, MR image quality can differ substantially between labs and even individual study populations. To account for these variations, we apply an extensive set of transformations to our data. This includes rotations of up to 90°, a width and height shift range of 30 pixels, a shear range of 5 pixels, zoom range of 0.2 and horizontal as well as vertical flips.

This not only increases the data set size but also makes it more representative of the general data distribution of Mice brain MR images and results in a model with a better generalisation capability.

Many more sophisticated methods have been tested, but it has been shown that one of the more successful data augmentation strategies is the simple transformations mentioned above [15].

Training

The model was trained slice wise, with the coronal view and 600 as the maximum number of epochs. The coronal view was chosen over the axial one, because the shapes of the masks are much simpler in the coronal view and thus easier to learn for the network. Additionally the coronal view has the advantage of higher resolution as the MR images were recorded coronally.

To improve the learning process of the network, two callbacks from Keras were used [16]. "*ReduceLROnPlateau*" reduces the learning rate when the validation loss has stopped improving and "*EarlyStopping*" stops the training when the validation loss has stopped improving for a number of epochs. The latter reduces computation time and prevents overfitting.

Masking

In order to improve the SAMRI registration workflow, an additional node is implemented where the images are masked, such that only the brain region remains. For this, the input image needs to first be resampled into the resolution of the template space, which has a voxel size of $0.2 \times 0.2 \times 0.2$. This is done with *Resample* command from the FSL library which is an analysis tool for FMRI, MRI and DTI brain imaging data [17]. Then, the image is preprocessed using the operations described in section 2.2. Since the classifier was trained to predict on images of shape (128, 128), the input needs to be reshaped. For each slice in the image, the classifier then predicts a segmentation of the brain, which is used to create a 3D mask. The latter is then reshaped into the original shape inverting the preprocessing step, either with the opencv resize method or by cropping. Additionally, the binary mask is resampled into its original affine space, before being multiplied with the brain image to extract the ROI. The workflow then continues with only the Region Of Interest as the image.

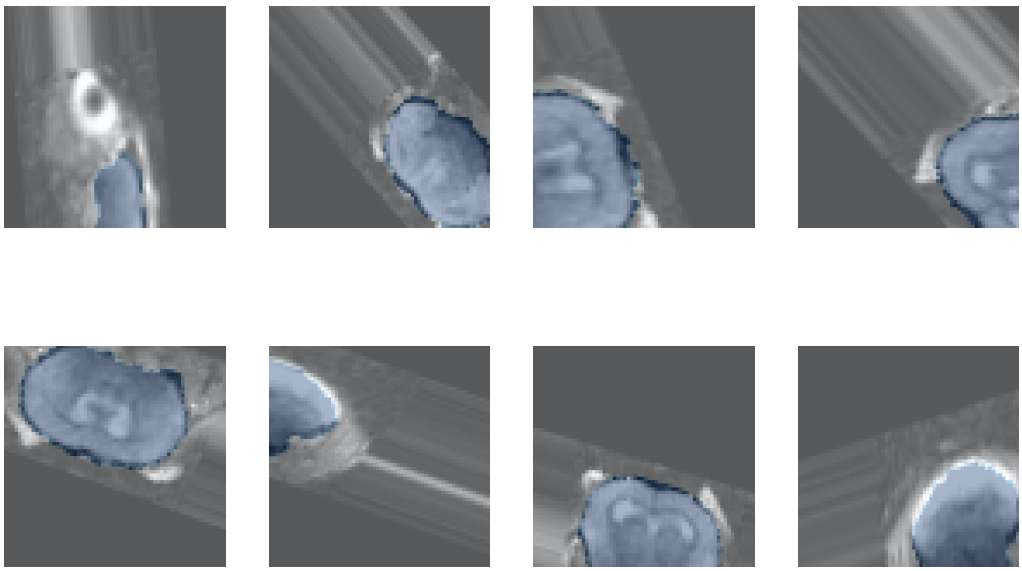


Figure 2: Augmented samples from the Training set.

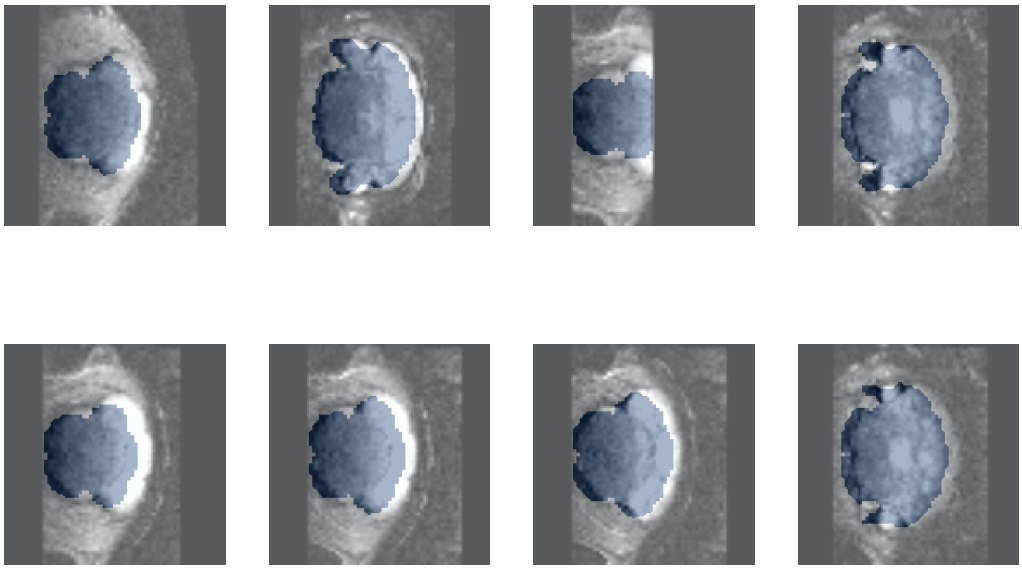


Figure 3: Slices where the mask includes too much outer-brain intensities are excluded from the data set. Examples from the slices that were excluded from the data set. The mask is shown in blue, on top of the brain image.

Evaluation

For the quality control of the workflow, we first evaluate the segmentation process and then the registration. The segmentations of the classifier should be as precise as possible for the masking process, which in return is then used to improve the registration.

As stated by Ioanas et al. in [3] a major challenge of registration QC is that a perfect mapping from the measured image to the template is undefined. To address this challenge, they developed four alternative evaluation metrics: volume conservation, smoothness conservation, functional analysis, and variance analysis. We will use these metrics to benchmark our workflow against theirs.

Segmentation

Quality control of our classifier is difficult in the sense that it should predict a better mask than the template. Nevertheless, it is use-full to verify if the output is similar, as it should be. As a similarity metric we have used the Dice score (see eq. (1)). The average Dice score taken on every slice of the test data set is $D_{coef} = 0.973 \sim 1$. The prediction thus have only minor changes in comparison with the template, which should represent small improvements.

As an evaluation of the registration, we make use of the quality control from [3]. We denote the original workflow as Generic and our improved version as Generic*.

Volume Conservation

Volume conservation is based on the assumption that the total volume of the scanned segment of the brain should remain roughly constant after preprocessing. Beyond just size differences between the acquired data and the target template, a volume increase may indicate that the brain was stretched to fill in template brain space not covered by the scan, while a volume decrease might indicate that non-brain voxels were introduced into the template brain space. For this analysis we compute a Volume Conservation Factor (VCF), whereby volume conservation is highest for a VCF equal to 1.

As seen in fig. 5a, we note that in the described dataset VCF is sensitive to the workflow ($F_{1,132} = 7.636$, $p = 0.0065$), but not the interaction of the workflow with the contrast ($F_{1,132} = 0.0015$, $p = 0.97$).

The performance of the Generic SAMRI workflow is significantly different from that of Generic*, yielding a two-tailed p-value of 1.2×10^{-5}. Moreover, the root mean squared error ratio favours the Generic* workflow ($\mathrm{RMSE}_{G*}/\mathrm{RMSE}_{G} \simeq 1.2$).

Descriptively, we observe that the Generic* level of the workflow variable introduces a volume gain (VCF of −0.04, 95%CI: −0.06 to −0.02). Further, we note that there is a very strong variance increase in all conditions for the Generic workflow (0.4661978357794804 0.47-fold).

With respect to the data break-up by contrast (CBV versus BOLD, fig. 5a), we see no notable main effect for the contrast variable (VCF of −0.02, 95%CI: −0.04 to 0.01). We do, however, report a notable effect for the contrast-workflow interaction, with the Generic* workflow and CBV contrast interaction level introducing a volume loss (VCF of 0.00, 95%CI: −0.03 to 0.03).

Smoothness Conservation

A further aspect of preprocessing quality is the resulting image smoothness. Although controlled smoothing is a valuable preprocessing tool used to increase the signal-to-noise ratio (SNR), uncontrolled smoothness limits operator discretion in the trade-off between SNR and feature granularity. Uncontrolled smoothness can thus lead to undocumented and implicit loss of spatial resolution and is therefore associated with inferior anatomical alignment [**?**]. We employ a Smoothness Conservation Factor (SCF), expressing the ratio between the smoothness of the preprocessed images and the smoothness of the original images.

With respect to the data shown in **??**, we note that SCF is sensitive to the workflow ($F_{1,132} = 2.019$, $p = 0.16$). (TODO: not really?)

The performance of the Generic SAMRI workflow is significantly different from that of the Generic* workflow, yielding a two-tailed p-value of 0.01.(TODO: not really?) In this comparison, the root mean squared error ratio favours the Generic* workflow ($\mathrm{RMSE}_{G*}/\mathrm{RMSE}_{G} \simeq 0.91$).

Descriptively, we observe that neither the Generic nor the Generic* workflow introduce a strong smoothing against (SCF of −0.02, 95%CI: −0.04 to 0.00).

Further, we note that there is a strong variance increase for the Generic workflow (0.8186943168066748 0.82 -fold)

Given the break-up by contrast shown in fig. 5b, we see only very weak effect sizes for the contrast variable (SCF of 0.05, 95%CI: 0.02 to 0.08) and the contrast-workflow interaction (SCF of −0.01, 95%CI: −0.03 to 0.02).

Functional Analysis

Functional analysis is a frequently used avenue for preprocessing QC. Its viability derives from the fact that the metric being maximized in the registration process is not the same output metric as that used for QC. The method is however primarily suited to examine workflow effects in light of higher-level applications, and less suited for wide-spread QC (as it is computationally intensive and only applicable to stimulus-evoked functional data). Additionally, functional analysis significance is documented to be sensitive to data smoothness [**?**], and thus an increased score on account of uncontrolled smoothing can be expected. For this analysis we compute the Mean Significance (MS), expressing the significance detected across all voxels of a scan.

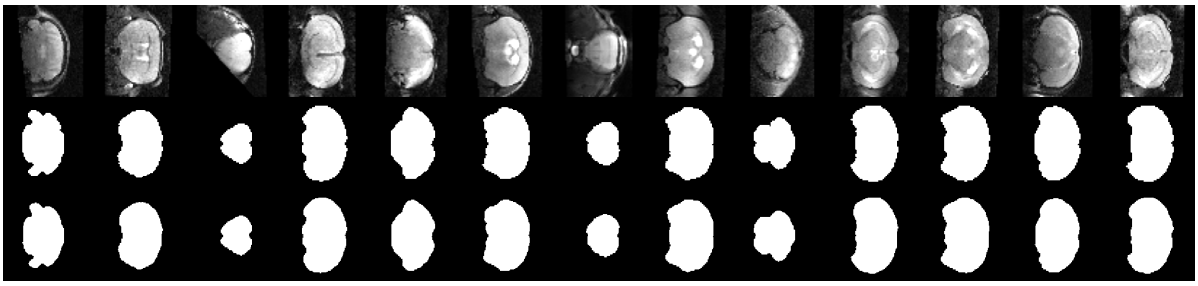


Figure 4: The Classifier predicts a similar mask to the ground truth. Random plots from the Test set illustrate the predictions of the classifier. The first row presents the input image, the second the ground truth and the third row shows the predictions of the classifier.

As seen in **??**, MS is sensitive to the workflow ($F_{1,132}=0.018$, $p=0.89$).

The performance of the SAMRI Generic workflow differs significantly from that of the Generic* workflow in terms of MS, yielding a two-tailed p-value of 2.7×10^{-4}.

Descriptively, we observe that the Generic* level of the workflow variable introduces a notable significance increase (MS of −0.05, 95%CI: −0.09 to −0.01), Furthermore, we again note a variance increase in all conditions for the Generic* workflow (0.9140659534334036 0.91-fold)

With respect to the data break-up by contrast (fig. 5c), we see no notable main effect for the contrast variable (MS of −0.08, 95%CI: −0.85 to 0.69).

Functional analysis effects can further be inspected by visualizing the statistic maps. Second-level t-statistic maps depicting the CBV and BOLD omnibus contrasts (common to all subjects and sessions) provide a succinct overview capturing both amplitude and directionality of the signal (fig. 6). Crucial to the examination of registration quality and its effects on functional read-outs is the differential coverage. We note that the Legacy workflow induces coverage overflow, extending to the cerebellum (**????** and figs. 6b and 6d), as well as to more rostral areas when used in conjunction with the Legacy template (figs. 6b and 6d). Separately from the Legacy workflow, the Legacy template causes acquisition slice misalignment (**????** and figs. 6b and 6d). Positive activation of the Raphe system, most clearly disambiguated from the surrounding tissue in the BOLD contrast, is notably displaced very far caudally by the joint effects of the Legacy workflow and the Legacy template (fig. 6d). We note that processing with the Generic template and workflow (figs. 6a and 6c), does not show issues with statistic coverage alignment and overflow.

Variance Analysis

An additional way to assess preprocessing quality focuses on the robustness to variability across repeated measurements, and whether this is attained without overfitting (i.e. compromising physiologically meaningful variability). The core assumption of this analysis of variance is that adult mouse brains in the absence of intervention retain size, shape, and implant position throughout the 8 week study period. Consequently, when examining similarity scores of preprocessed scans with respect to the target template, more variation should be found across levels of the subject variable rather than session variable. This comparison can be performed using a type 3 ANOVA, modelling both the subject and the session variables. For this assessment we select three metrics with maximal sensitivity to different features: Neighborhood Cross Correlation (CC, sensitive to localized correlation), Global Correlation (GC, sensitive to whole-image correlation), and Mutual Information (MI, sensitive to whole-image information similarity).

Figure 7 renders the similarity metric scores for both the SAMRI Generic and Legacy workflows (considering only the matching workflow-template combinations). The Legacy workflow produces results which show a higher F-statistic for the session than for the subject variable: CC (subject: $F_{10,19}=6.664$, $p=0.00021$, session: $F_{4,19}=3.461$, $p=0.028$), GC (subject: $F_{10,19}=1.052$, $p=0.44$, session: $F_{4,19}=0.65$, $p=0.63$), and MI (subject: $F_{10,19}=1.399$, $p=0.25$, session: $F_{4,19}=1.293$, $p=0.31$).

The Generic SAMRI workflow shows a reversing trend. Resulting data F-statistics are consistently higher for the subject variable than for the session variable: CC (subject: $F_{10,19}=6.221$, $p=0.00033$, session: $F_{4,19}=3.183$, $p=0.037$), GC (subject: $F_{10,19}=1.761$, $p=0.14$, session: $F_{4,19}=1.235$, $p=0.33$), and MI (subject: $F_{10,19}=0.78$, $p=0.65$, session: $F_{4,19}=1.753$, $p=0.18$).

Methods

The same methods that are described in the original paper have been applied in this work. A more detailed

description can be found there.

The slice-wise predictions of the model are reconstructed to a 3D mask via the command *Nifit1Image* from the neuroimaging python package nibabel [18]. This is done using the same affine space as the input image.

For the training of the classifier, the data are separated into a Training, Validation and Test set with the help of the function *train_test_split* from the package sklearn.model_selection [19].

For the quality control of the workflows, a dataset with an effective size of 102 scans is used. is included, with each animal scanned on up to 5 sessions (repeated at 14 day intervals). Each session contains an anatomical scan and two functional scans — with Blood-Oxygen Level Dependent (BOLD) [**?**] and Cerebral Blood Volume (CBV) [**?**] contrast, respectively (for a total of 68 functional scans).

The measured animals were fitted with an optic fiber implant ($\mathrm{l} = 3.2\,\mathrm{mm}$ $\mathrm{d} = 400\,\mu\mathrm{m}$) targeting the Dorsal Raphe (DR) nucleus in the brain stem.

All experimental procedures were approved by the Veterinary Office of the Canton of Zurich and done in accordance with the relevant regulations.

Metrics

For the current VCF implementation brain volume is defined as estimated by the 66^{th} voxel intensity percentile of the raw scan before any processing. The arbitrary unit equivalent of this percentile threshold is recorded for each scan and applied to all preprocessing workflow results for that particular scan, to obtain VCF estimates — eq. (3), where v is the voxel volume in the original space, v' the voxel volume in the transformed space, n the number of voxels in the original space, m the number of voxels in the transformed space, s a voxel value sampled from the vector S containing all values in the original data, and s' a voxel value sampled from the transformed data.

$$VCF = \frac{v' \sum_{i=1}^{m} [s'_i \geq P_{66}(S)]}{v \sum_{i=1}^{n} [s_i \geq P_{66}(S)]} = \frac{v' \sum_{i=1}^{m} [s'_i \geq P_{66}(S)]}{v \lceil 0.66n \rceil} \tag{3}$$

The SCF metric is based on the ratio of smoothness before and after processing. The smoothness measure is the full-width at half-maximum (FWHM) of the signal amplitude spatial autocorrelation function (ACF [**?**]). Since fMRI data usually do not have a Gaussian-shaped spatial ACF, we use AFNI [**?**] to fit the following function in order to compute the FWHM — eq. (4), where r is the distance of two amplitude distribution samples, a is the relative weight of the Gaussian term in the model, b is the width of the Gaussian and c the decay of the mono-exponential term [**?**].

$$ACF(r) = a * e^{-r^2/(2*b^2)} + (1-a) + e^{-r/c} \tag{4}$$

We examine statistical power as relevant for the MS metric via the negative logarithm of first-level p-value maps. This produces voxelwise statistical estimates for the probability that a time course could — by chance alone — be at least as well correlated with the stimulation regressor as the voxel time course measured. We compute the per-scan average of these values as seen in eq. (5), where n represents the number of statistical estimates in the scan, and p is a p-value.

$$MS = \frac{\sum_{i=1}^{n} -log(p_i)}{n} \tag{5}$$

Software

The workflow functions make use of the Nipype [**?**] package, which provides high-level workflow management and execution features. Via this package, functions provided by any other package can be encapsulated in a node (complete with error reporting and isolated re-execution support) and integrated into a directed workflow graph. Parallelization can also be managed via a number of execution plugins, allowing scalability. Most importantly, Nipype can generate graph descriptor language (DOT) summaries, as well as visual workflow representations suitable for operator inspection, graph theoretical analysis, and programmatic comparison between workflow variants.

Via Nipype, we utilize basic MRI preprocessing functions from the FSL package [17] and registration functions from the ANTs package [4]. The choice of the ANTs package (in addition to FSL, which also provides registration functions) owes to the package's functions being more highly parameterized. This feature allows us to avoid maladaptive optimization choices, and instead fine-tune the registration to the overarching characteristics of the brain type at hand. Additionally, we have implemented a number of functions in our workflow directly, e.g. to read BIDS [**?**] inputs, and perform dummy scans management.

For Quality Control we distribute as part of this publication additional workflows using the NumPy [20], SciPy [21], pandas [22], and matplotlib packages [23], as well as Seaborn [24] for plotting, and Statsmodels [25] for top-level statistics, using the HC3 heteroscedasticity consistent covariance matrix [**?**]. Specifically, distribution densities for plots are drawn using the Scott bandwidth density estimator [26].

Data and Code Availability

The data archive relevant for this article is freely available [**?**], and automatically accessible via the Gentoo Linux package manager. The code relevant for reproducing this document is also freely available [**?**], as are its dependencies, and most prominently, SAMRI [**?**], the package via which the herein described workflows are distributed.

References

[1] J B Antoine Maintz and Max A Viergever. An Overview of Medical Image Registration Methods. page 22.

[2] Aristeidis Sotiras, Christos Davatzikos, and Nikos Paragios. Deformable Medical Image Registration: A Survey. *IEEE Transactions on Medical Imaging*, 32(7):1153–1190, July 2013. ISSN 1558-254X. doi: 10.1109/TMI.2013.2265603.

[3] Horea-Ioan Ioanas, Markus Marks, Mehmet Fatih Yanik, and Markus Rudin. An Optimized Registration Workflow and Standard Geometric Space for Small Animal Brain Imaging. preprint, Neuroscience, April 2019. URL `http://biorxiv.org/lookup/doi/10.1101/619650`.

[4] Brian B Avants, Nick Tustison, and Gang Song. Advanced normalization tools (ants). *Insight j*, 2 (365):1–35, 2009.

[5] Qichuan Geng, Zhong Zhou, and Xiaochun Cao. Survey of recent progress in semantic image segmentation with CNNs. *Science China Information Sciences*, 61(5):051101, May 2018. ISSN 1674-733X, 1869-1919. doi: 10.1007/s11432-017-9189-6. URL `http://link.springer.com/10.1007/s11432-017-9189-6`.

[6] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. *arXiv:1505.04597 [cs]*, May 2015. URL `http://arxiv.org/abs/1505.04597`. arXiv: 1505.04597 version: 1.

[7] zhixuhao. zhixuhao/unet, January 2020. URL `https://github.com/zhixuhao/unet`. original-date: 2017-04-06T01:58:15Z.

[8] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1): 1929–1958, 2014.

[9] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation. *arXiv:1606.04797 [cs]*, June 2016. URL `http://arxiv.org/abs/1606.04797`. arXiv: 1606.04797.

[10] Horea-Ioan Ioanas, Bechara John Saab, and Markus Rudin. A Whole-Brain Map and Assay Parameter Analysis of Mouse VTA Dopaminergic Activation. page 19, .

[11] Horea-Ioan Ioanas, Bechara John Saab, and Markus Rudin. Effects of Acute and Chronic Reuptake Inhibition on Optogenetically Induced Serotonergic Activity. page 20, .

[12] IBT-FMI/SAMRI, December 2019. URL `https://github.com/IBT-FMI/SAMRI`. original-date: 2015-04-27T00:26:08Z.

[13] Multi-dimensional image processing (scipy.ndimage) — SciPy v1.4.1 Reference Guide, . URL `https://docs.scipy.org/doc/scipy/reference/ndimage.html#morphology`.

[14] opencv-python: Wrapper package for OpenCV python bindings., . URL `https://github.com/skvark/opencv-python`.

[15] Luis Perez and Jason Wang. The Effectiveness of Data Augmentation in Image Classification using Deep Learning. *arXiv:1712.04621 [cs]*, December 2017. URL `http://arxiv.org/abs/1712.04621`. arXiv: 1712.04621.

[16] Callbacks - Keras Documentation, . URL `https://keras.io/callbacks/`.

[17] Mark Jenkinson, Christian F Beckmann, Timothy EJ Behrens, Mark W Woolrich, and Stephen M Smith. Fsl. *Neuroimage*, 62(2):782–790, 2012.

[18] Neuroimaging in Python — NiBabel 2.5.0 documentation, . URL `https://nipy.org/nibabel/`.

[19] scikit-learn/scikit-learn, January 2020. URL `https://github.com/scikit-learn/scikit-learn`. original-date: 2010-08-17T09:43:38Z.

[20] Travis E. Oliphant. *Guide to NumPy*. Provo, UT, March 2006. URL `https://www.numpy.org/devdocs/contents.html`.

[21] K. Jarrod Millman and Michael Aivazis. Python for scientists and engineers. *Computing in Science & Engineering*, 13(2):9–12, March 2011. doi: 10.1109/mcse.2011.36. URL `https://doi.org/10.1109/mcse.2011.36`.

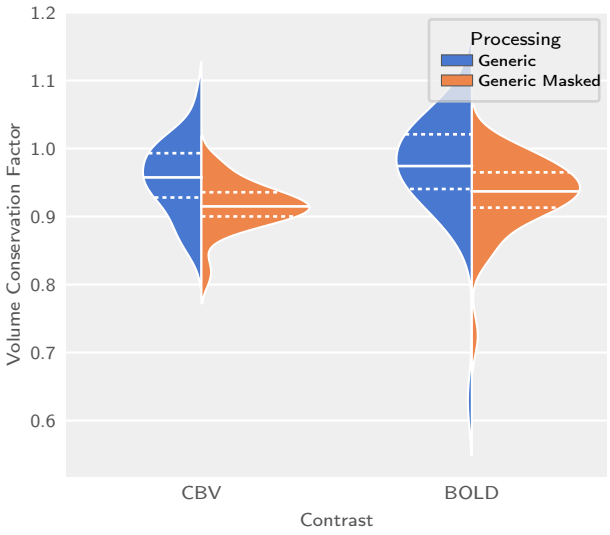
[22] Wes McKinney. Data Structures for Statistical Computing in Python. In Stéfan van der Walt and Jarrod Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 51 – 56, June 2010.

[23] John D. Hunter. Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering*, 9(3):90–95, June 2007. doi: 10.1109/mcse.2007.55. URL `https://doi.org/10.1109/mcse.2007.55`.

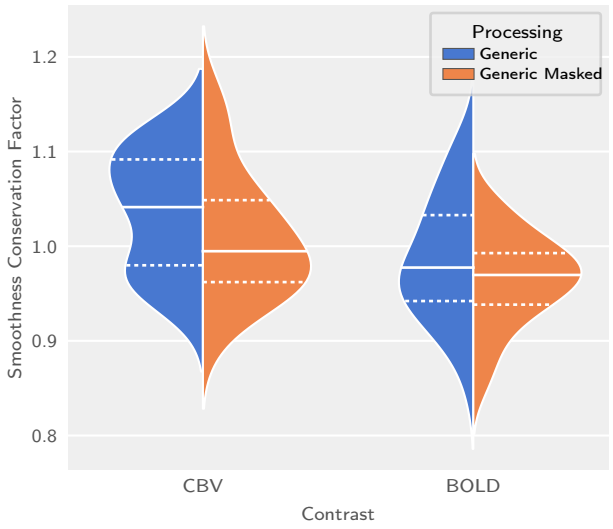
[24] Michael Waskom, Olga Botvinnik, Drew O'Kane, Paul Hobson, Saulius Lukauskas, David C Gemperline, Tom Augspurger, Yaroslav Halchenko, John B. Cole, Jordi Warmenhoven, Julian de Ruiter, Cameron Pye, Stephan Hoyer, Jake Vanderplas, Santi Villalba, Gero Kunter, Eric Quintero, Pete Bachant, Marcel Martin, Kyle Meyer, Alistair Miles, Yoav Ram, Tal Yarkoni, Mike Lee Williams, Constantine Evans, Clark Fitzgerald, Brian, Chris Fonnesbeck, Antony Lee, and Adel Qalieh. Seaborn: v0.8.1, September 2017. URL `https://doi.org/10.5281/zenodo.883859`.

[25] Skipper Seabold and Josef Perktold. Statsmodels: Econometric and statistical modeling with Python. In *9th Python in Science Conference*, June 2010. URL `http://conference.scipy.org/proceedings/scipy2010/pdfs/seabold.pdf`.

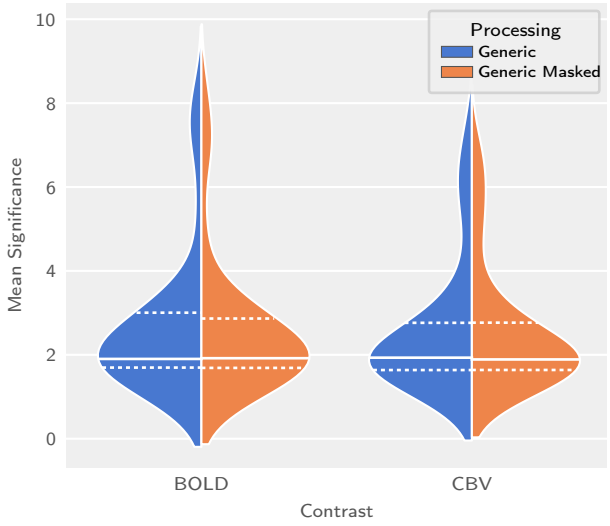
[26] David W. Scott. On optimal and data-based histograms. *Biometrika*, 66(3):605–610, December 1979. doi: 10.1093/biomet/66.3.605. URL `https://doi.org/10.1093/biomet/66.3.605`.



(a) Comparison across workflows and functional contrasts, considering only matching template-workflow combinations.

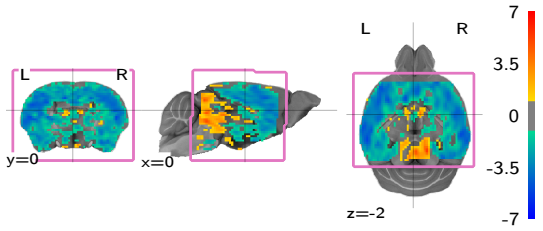


(b) Comparison across workflows and functional contrasts, considering only matching template-workflow combinations.

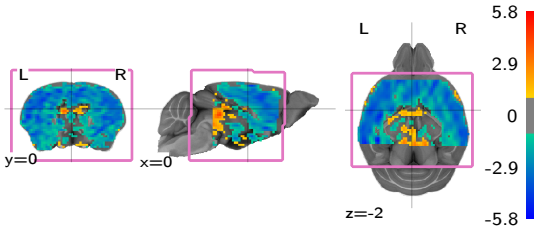


(c) Comparison across workflows and functional contrasts, considering only matching template-workflow combinations.

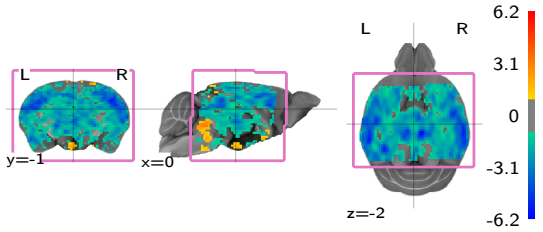
Figure 5: The SAMRI Generic workflow and template optimally and reliably conserve volume and smoothness — unlike the Legacy workflow and template. Plots of three target metrics, with coloured patch widths estimating distribution density, solid lines indicating the sample mean, and dashed lines indicate the inner quartiles.



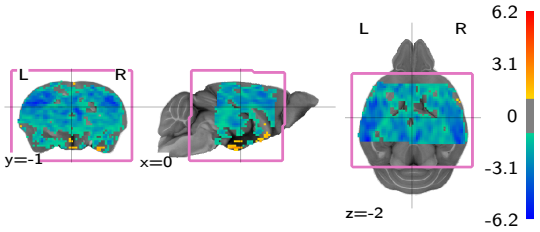
(a) Generic workflow with Generic template CBV map, showing correct slice orientation and coverage correctly bounded to the acquisition area.



(b) generic* workflow with Generic template CBV map, showing incorrect slice orientation and coverage both caudally and rostrally extending beyond acquisition area.



(c) Generic workflow with Generic template BOLD map, showing correct slice orientation and coverage correctly bounded to the acquisition area.



(d) Legacy workflow with Legacy template BOLD map, showing incorrect slice orientation and coverage both caudally and rostrally extending beyond acquisition area.

Figure 6: Legacy workflow processing leads to a problematic overflow of the statistic maps into adjacent anatomical regions, leaking beyond the acquistion area. SAMRI mitigates this effect as illustrated by multiplanar depictions of second-level omnibus statistic maps separately evaluating CBV and BOLD scans, and thresholded at $|t| \geq 2$. The acquisition area is bracketed in pink, and in comparing it to statistic coverage it is important to note that the latter is always underestimated, as the omnibus statistic contrast is only defined for voxels captured in *every* evaluated scan.

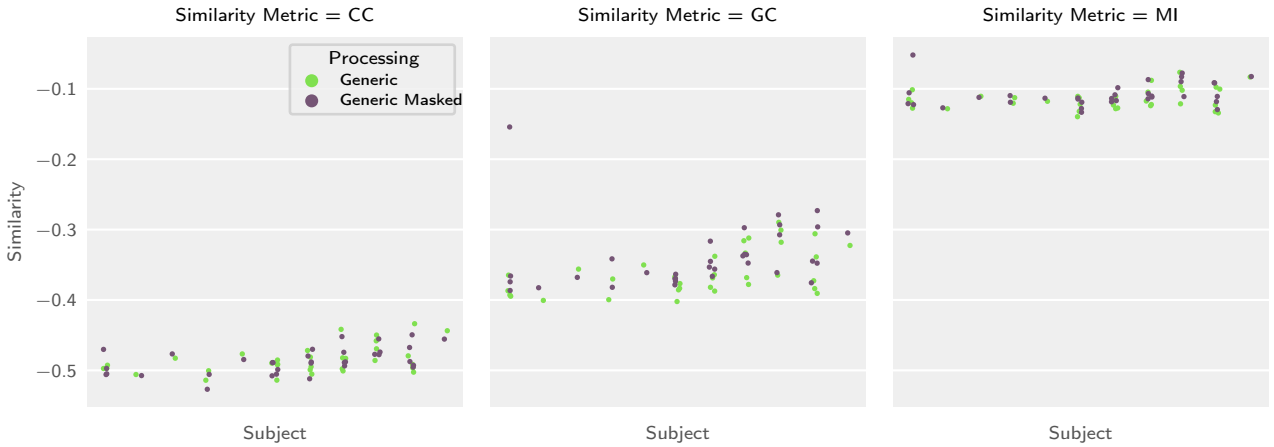
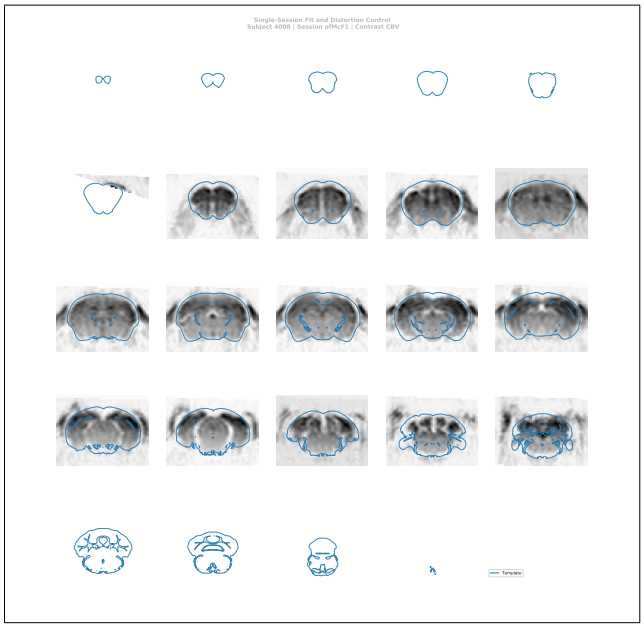
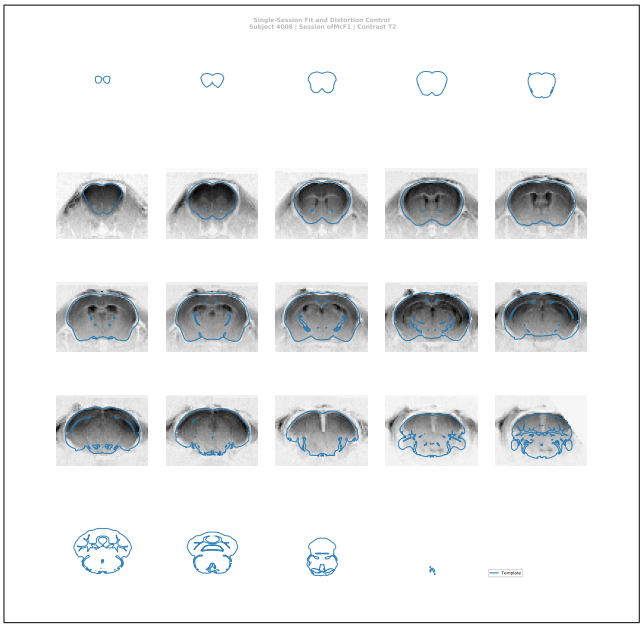


Figure 7: The SAMRI Generic workflow conserves subject-wise variability and minimizes trial-to-trial variability compared to the Legacy workflow. Swarmplots illustrate similarity metric scores of preprocessed images with respect to the corresponding workflow template, plotted across subjects (separated into x-axis bins) and sessions (individual points in each x-axis bin), for the CBV contrast.

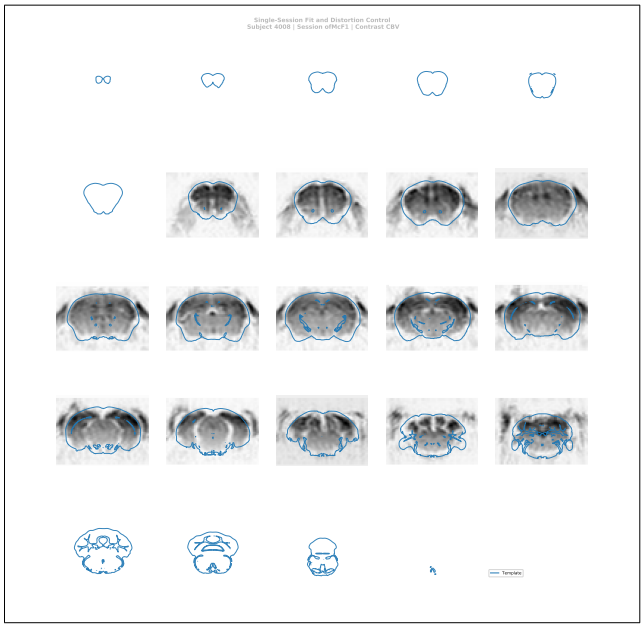
Supplementary Materials



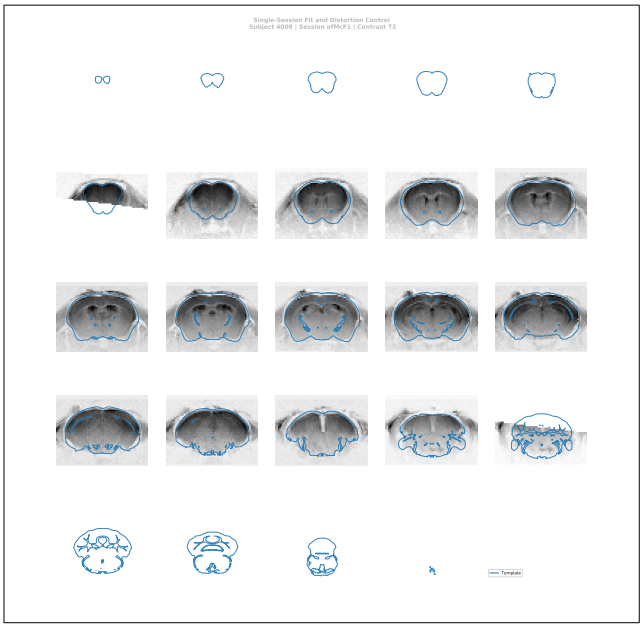
(a) SAMRI Generic workflow with Generic template, note the undistorted mapping and conservative smoothing.



(b) SAMRI Generic workflow with Generic template, inspecting the structural scan intermediary; note the undistorted mapping and conservative smoothing.



(c) SAMRI Generic Masked workflow, note the strong smoothing and the mapping distortion in the rostral and caudal areas.



(d) SAMRI Generic Masked workflow, note the strong smoothing and the mapping distortion in the rostral and caudal areas.

Figure S1: The SAMRI Generic workflow induces less smoothness, and provides more accurate coverage. Depicted are automatically created operator overview graphics, which allow a slice-by-slice (spacing analogous to acquisition) inspection of the registration fit. This representation affords a particularly detailed view of the preprocessed MRI data, and highly accurate template contours.

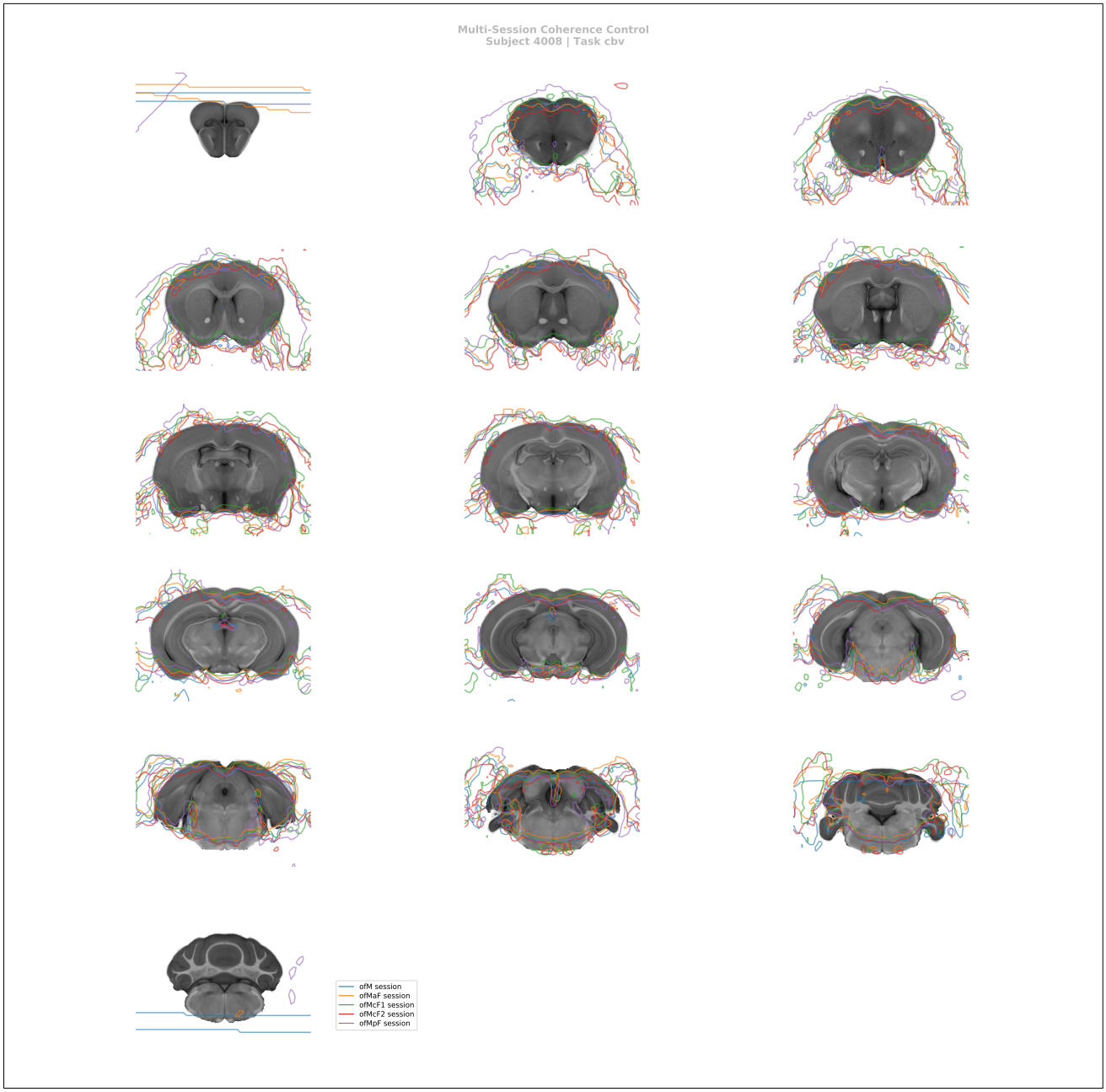


Figure S2: The SAMRI Generic workflow consistently maps high-salience features such as the implant site across sessions. Automatically created operator overview graphic, allowing a slice-by-slice (spacing analogous to acquisition) inspection of registration coherence. This representation permits a coarse assessment of registration consistency for multiple sessions — though at the cost of some clarity. Particularly, this visualization, allows an operator to track the position of high-amplitude fixed features across scans in order to ascertain coherence (similarly to what is automatically assessed by the Variance analysis of the session factor).

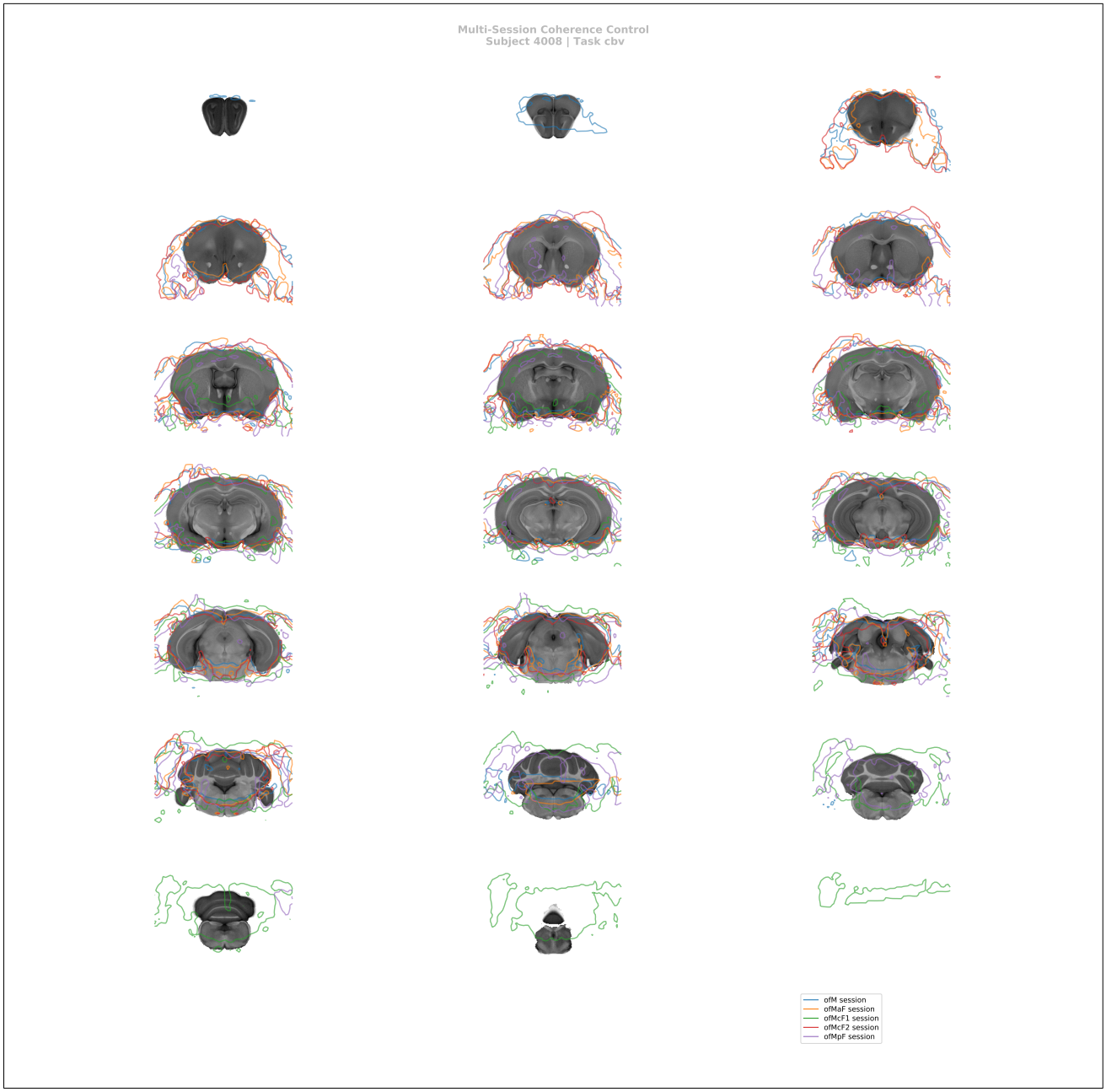
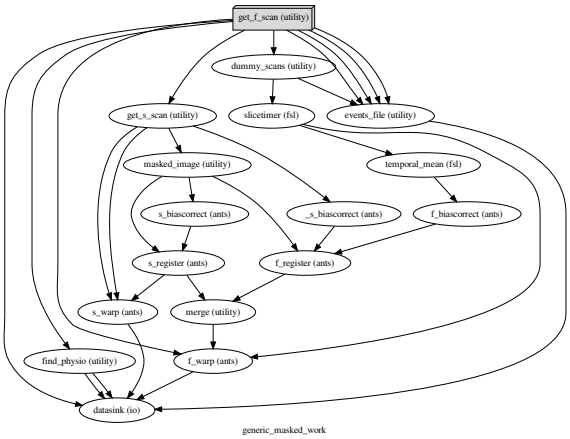
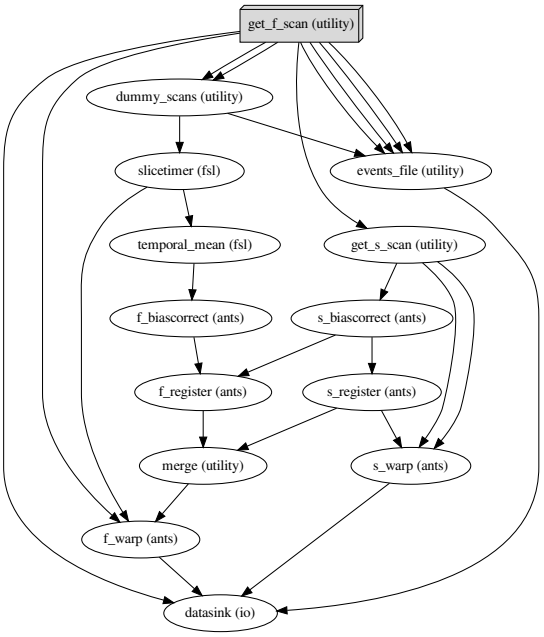


Figure S3: The SAMRI Generic Masked workflow consistently maps high-salience features such as the implant site across sessions. Automatically created operator overview graphic, allowing a slice-by-slice (spacing analogous to acquisition) inspection of registration coherence. This representation permits a coarse assessment of registration consistency for multiple sessions — though at the cost of some clarity. Particularly, this visualization, allows an operator to track the position of high-amplitude fixed features across scans in order to ascertain coherence (similarly to what is automatically assessed by the Variance analysis of the session factor).



(a) "SAMRI Generic Masked" workflow, which is based on the `antsIntroduction.sh` function (and other functions with hard-coded parameters optimized for human brain registration), and also performs destructive affine manipulations.



(b) "SAMRI Generic" workflow, based on the `antsRegistration` function. The pipeline uses a higher-resolution structural scan intermediary for registration (note the two processing streams), which facilitates differential handling of anatomical variation and susceptibility artefacts. The function used is highly parameterized, and both of its instances — "s_register" and "f_register" — are supplied in the workflow with defaults optimized for mouse brain registration.

Figure S4: Directed acyclic graphs detailing the precise node names (as seen in the SAMRI source code) for the two alternate MRI registration workflows. The package correspondence of each processing node is appended in parentheses to the node name. The "utility" indication corresponds to nodes based on Python functions specific to the workflow, distributed alongside it, and dynamically wrapped via Nipype. The "extra_interfaces" indication corresponds to nodes using explicitly defined Nipype-style interfaces, which are specific to the workflow and distributed alongside it.