

## Supplementary Information for the Fetal Tissue Annotation 2022 Challenge Results

### Table of Contents

1	Methods Description.....	7
1.1	Tajoshiusc .....	7
1.2	Blackbean .....	7
1.3	BlueBrune .....	8
1.4	Deepsynth .....	12
1.5	Dolphins: Coarse-to-Fine Models for FeTA2022 Segmentation .....	14
1.6	FeTA-Imperial-TUM Team (FIT_1) – FIT-nnU-Net .....	17
1.7	FeTA-Imperial-TUM Team (FIT_2) – FIT-SwinUNETR; .....	21
1.8	FMRSK.....	24
1.9	fudan_zmic.....	26
1.10	hilab.....	27
1.11	Neurophet .....	29
1.12	NVAUTO .....	31
1.13	Pasteur DBC.....	33
1.14	Sano.....	34
1.15	symsense.....	35
1.16	UNIANDES .....	39
1.17	xinlab-scut-iai-ahu.....	41
2	References.....	46
3	Benchmarking report for multiTaskChallengeDice_global.....	50
3.1	Ranking.....	50
3.2	Visualization of raw assessment data.....	51
3.3	References.....	57
4	Benchmarking report for multiTaskChallengeHD_global.....	58
4.1	Ranking.....	58
4.2	Visualization of raw assessment data.....	59
4.3	References.....	65

<b>FIT_1 0.8052332 1</b>	<b>5</b>	Benchmarking report for multiTaskChallengeVolSim_global .....	66
<b>symsense 0.8046716 2</b>	5.1	Ranking.....	66
<b>NVAUTO 0.8041635 3</b>	5.2	Visualization of raw assessment data.....	67
	5.3	References.....	73
<b>FIT_2 2.310282 1</b>	<b>6</b>	Benchmarking report for Dice Metrics – In Domain .....	74
<b>Institut_Pasteur_DBC 2.403996 7</b>	6.1	Ranking:.....	74
<b>NVAUTO 2.463725 3</b>	6.2	Visualization of raw assessment data.....	75
	6.3	References.....	81
<b>FMR SK 0.8276876 1</b>	<b>7</b>	Benchmarking report for Hausdorff Metrics – In Domain .....	82
<b>FIT_1 0.8268333 2</b>	7.1	Ranking.....	82
<b>BlueBrune 0.8224908 3</b>	7.2	Visualization of raw assessment data.....	83
	7.3	References.....	89
<b>FMR SK 2.050589 1</b>	<b>8</b>	Benchmarking report for Volume Similarity Metrics – In Domain .....	90
<b>FIT_1 2.172403 2</b>	8.1	Ranking.....	90
<b>BlueBrune 2.282161 3</b>	8.2	Visualization of raw assessment data.....	91
	8.3	References.....	97
<b>FMR SK 0.8276876 1</b>	<b>9</b>	Benchmarking report for Dice Metrics – Out of Domain.....	98
<b>FIT_1 0.8268333 2</b>	9.1	Ranking.....	98
<b>BlueBrune 0.8224908 3</b>	9.2	Visualization of raw assessment data.....	99
	9.3	References.....	105
<b>FMR SK 2.050589 1</b>	<b>10</b>	Benchmarking report for Hausdorff Metrics – Out of Domain.....	106
<b>FIT_1 2.172403 2</b>	10.1	Ranking.....	106
<b>BlueBrune 2.282161 3</b>	10.2	Visualization of raw assessment data.....	107
	10.3	References.....	113
<b>FMR SK 0.8276876 1</b>	<b>11</b>	Benchmarking report for Volume Similarity Metrics – Out of Domain.....	114
<b>FIT_1 0.8268333 2</b>	11.1	Ranking.....	114
<b>BlueBrune 0.8224908 3</b>	11.2	Visualization of raw assessment data.....	115
	11.3	References.....	121
<b>FMR SK 2.050589 1</b>	<b>12</b>	Evaluation Metrics per Label .....	122
<b>FIT_1 2.172403 2</b>	12.1	Global Evaluation Metrics per Label .....	123
<b>BlueBrune 2.282161 3</b>	12.2	In-Domain Evaluation Metrics per Label.....	144
	12.3	Out-of-Domain Evaluation Metrics per Label.....	165

13	Benchmarking report for Dice Metrics – Excellent Quality Reconstructions	
186		
13.1	Ranking.....	186
13.2	Visualization of raw assessment data.....	187
13.3	References.....	194
14	Benchmarking report for Hausdorff Metrics – Excellent Quality Reconstructions.....	195
14.1	Ranking.....	195
14.2	Visualization of raw assessment data.....	196
14.3	References.....	203
15	Benchmarking report for Volume Similarity Metrics – Excellent Quality Reconstructions.....	204
15.1	Ranking.....	204
15.2	Visualization of raw assessment data.....	205
15.3	References.....	212
16	Benchmarking report for Dice Metrics – Good Quality Reconstructions....	212
16.1	Ranking.....	212
16.2	Visualization of raw assessment data.....	214
16.3	References.....	221
17	Benchmarking report for Hausdorff Metrics – Good Quality Reconstructions	
222		
17.1	Ranking.....	222
17.2	Visualization of raw assessment data.....	223
17.3	References.....	230
18	Benchmarking report for Volume Similarity Metrics – Good Quality Reconstructions.....	231
18.1	Ranking.....	231
18.2	Visualization of raw assessment data.....	232
18.3	References.....	239
19	Benchmarking report for Dice Metrics – Poor Quality Reconstructions....	240
19.1	Ranking.....	240
19.2	Visualization of raw assessment data.....	241
19.3	References.....	248

20	Benchmarking report for Hausdorff Metrics – Poor Quality Reconstructions	249
20.1	Ranking.....	249
20.2	Visualization of raw assessment data.....	250
20.3	Visualization of ranking stability .....	253
20.4	References.....	258
21	Benchmarking report for Volume Similarity Metrics – Poor Quality Reconstructions.....	259
21.1	Ranking.....	259
21.2	Visualization of raw assessment data.....	260
21.3	References.....	267
22	Benchmarking report for Dice Metrics – Neurotypical Brains .....	268
22.1	Ranking.....	268
22.2	Visualization of raw assessment data.....	269
22.3	References.....	276
23	Benchmarking report for Hausdorff Metrics – Neurotypical Brains.....	277
23.1	Ranking.....	277
23.2	Visualization of raw assessment data.....	278
23.3	References.....	285
24	Benchmarking report for Volume Similarity Metrics – Neurotypical Brains	286
24.1	Ranking.....	286
24.2	Visualization of raw assessment data.....	287
24.3	References.....	294
25	Benchmarking report for Dice Metrics – Pathological Brains .....	295
25.1	Ranking.....	295
25.2	Visualization of raw assessment data.....	296
25.3	References.....	303
26	Benchmarking report for Hausdorff Metrics – Pathological Brains.....	304
26.1	Ranking.....	304
26.2	Visualization of raw assessment data.....	305
26.3	References.....	312

27	Benchmarking report for Volume Similarity Metrics – Pathological Brains 313	
27.1	Ranking.....	313
27.2	Visualization of raw assessment data.....	314
27.3	References.....	321
28	Benchmarking report for Dice Metrics – irtkSimple Reconstruction Method 322	
28.1	Ranking.....	322
28.2	Visualization of raw assessment data.....	323
28.3	References.....	330
29	Benchmarking report for Hausdorff Metrics – irtkSimple Reconstruction Method 331	
29.1	Ranking.....	331
29.2	Visualization of raw assessment data.....	332
29.3	References.....	339
30	Benchmarking report for Volume Similarity Metrics – irtkSimple Reconstruction Method.....	340
30.1	Ranking.....	340
30.2	Visualization of raw assessment data.....	341
30.3	References.....	348
31	Benchmarking report for Dice Metrics – mial-srtk Reconstruction Method 349	
31.1	Ranking.....	349
31.2	Visualization of raw assessment data.....	350
31.3	References.....	357
32	Benchmarking report for Hausdorff Metrics – mial-srtk Reconstruction Method 358	
32.1	Ranking.....	358
32.2	Visualization of raw assessment data.....	359
32.3	References.....	366
33	Benchmarking report for Volume Similarity Metrics – mial-srtk Reconstruction Method.....	367
33.1	Ranking.....	367
33.2	Visualization of raw assessment data.....	368
33.3	References.....	375

34	Benchmarking report for Dice Metrics – NiftyMIC Reconstruction Method	
	376	
34.1	Ranking.....	376
34.2	Visualization of raw assessment data.....	377
34.3	References.....	384
35	Benchmarking report for Hausdorff Metrics – NiftyMIC Reconstruction Method	385
35.1	Ranking.....	385
35.2	Visualization of raw assessment data.....	386
35.3	References.....	393
36	Benchmarking report for Volume Similarity Metrics – NiftyMIC Reconstruction Method.....	394
36.1	Ranking.....	394
36.2	Visualization of raw assessment data.....	395
36.3	References.....	402

## 1 Methods Description

Here we present the methods descriptions for all teams who took place in the Fetal Tissue Annotation (FeTA) 2022 Challenge.

### 1.1 ajoshiusc

**Team Members:** Anand A Joshi\*, Haleh Akrami, Wenhui Cui, John C Wood, Krishna N Nayak\*, Richard M. Leahy\*

\*authors included in paper

**GPU training was performed on.** NVIDIA 2060, P100.

**Software used.** Pytorch (1.10.2), SimpleITK (2.1.1), nilearn (0.9.1)

**Model Architecture.** We adopted a 3-dimensional CNN called TransUNet [1] using 2D slices as input as our backbone model. The 2D slices for training were extracted from 3D image scans and resized to 256\*256 and normalized to unit magnitude. This model combines U-Net [2] and Transformer [3] networks. TransUNet is based on an encoder-decoder structure and takes advantage of Transformer to learn not only local context information but also global semantic correlations.

The specific model architecture we used is a combination of ResNet-50 [4] and ViT [5], denoted as “R50-ViT” [1]. The loss function we used was cross entropy. Based on our previous work, we also used robust cross entropy based on beta divergence as the loss function, so the model can be trained in the presence of errors in the training data [6]–[8]. We trained the model with SGD optimizer with learning rate 0.01, momentum 0.9 and weight decay 1e-4 and the batch size set to 4. The training takes 5-6 hours on Nvidia 2060 GPU.

We used data from only 1 collection (University Children's Hospital Zurich (Kispi)). The training set was 80 scans. We used 75 scans for training and 5 scans for validation during development. We performed 150 epochs of training of TransUNet and evaluated the model performance on the 5 validation scans. The average dice coefficient was computed for the validation data and the epoch with best performance on the validation data was chosen for deployment in the docker.

The source code for our implementation is available at <https://github.com/ajoshiusc/brainseg/tree/main/feta2022>

### 1.2 Blackbean

**Team Members:** Haoyu Wang\*, Ziyuan Huang\*, Jin Ye\*, Zhongying Deng, Chenglong Ma, Can Tu, Junjun He, Yuncheng Yang, Shiyi Du

\*authors included in paper

Training was performed on Tesla A100 GPU using Pytorch 1.12.

**Network Architecture.** We trained two networks ([a modified U-Net and a ViT-Adaptor](#)) on all the training cases for the final submission.

*Model 1:* Our modified U-Net has an encoder-decoder architecture with five layers like the default U-Net. The number of feature channels for the first layer are set 32 instead of 64. At each step, the number of channels is doubled.

*Model 2:* We trained a 3D version of ViT-Adaptor, a state-of-the-art transformer architecture proposed by [9]. The ViT-Adaptor contains a vanilla 3D Vision Transformer (ViT) for general feature extraction and some additional vision-specific modules to improve the performance via introducing inductive biases.

**Training Settings.** For most settings, we follow the default settings of nnUNet. Additionally, we adjust the gamma range of the random gamma transform from (0.7,1.5) to (0.1, 3.0). For the training of ViT-Adaptor, we adopt SGD optimizer with the default learning rate 0.01 and weight decay 0.0001. The batch size is 2 and loss function is the average of Dice loss and BCE loss. We conducted only the 3D input. The models submitted in the docker file are all trained on the full training set (all training cases). No extra post-processing steps are used.

**Test-time Augmentation (TTA).** Instead of the flipping augmentation for TTA (8x inference time), we adopt a multi-scale TTA strategy (1.5x inference time) in our docker file. During the inference, we resample the 3D images into two different spacings: 1.0 and 1.2 times the default statistical spacing from nnUNet. Predictions for the two volumes are resized to the original shape and ensembled by simply averaging the softmax.

### 1.3 BlueBrune

**Team Members:** Niccolò McConnell\*, Mark Nchongmaje, Alina Miron\*, Yongmin Li\*

\*authors included in paper

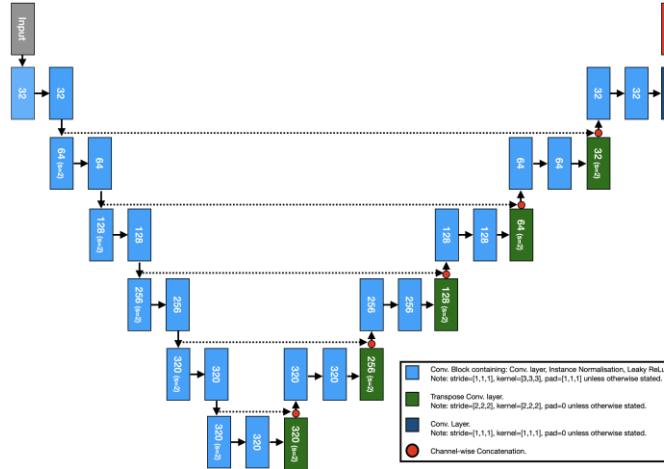
**GPU training was performed on.** NVIDIA A6000

**Software used.** Pytorch 11.12

**Description of Method.** The key novelty of our method lies in utilizing a domain adversarial approach [10] for the training of the network component of the nnUNet framework [11]. We therefore newly integrate a domain adversarial approach into nnUNet which utilizes both a 3D UNet architecture [2] as the segmentation network and a vanilla convolutional neural network as the domain discriminator network. The segmentation network outputs the segmentation maps, while the discriminator is tasked with recognising from which domain, hospital 1 or hospital 2, its input originates from, with the two networks trained in an adversarial fashion.

The aim of this approach is to train the UNet to learn features which are domain invariant between the two hospitals in the training set, and which would then allow improved performance on hospitals not included in the training dataset.

**Segmentation Network Architecture.** The adopted UNet inspired architecture is illustrated in Fig. 1, with the network having a depth of six. The nnUNet utilizes a 3D encoder-decoder UNet inspired network. 3D convolutions with kernel size 3x3x3 are utilized for feature extraction, upsampling is performed via transposed convolutions while downsampling is performed via strided convolutions with stride 2. The network's convolutional blocks consist of a convolutional layer followed by instance normalisation, and finally a LeakyReLU activation function is applied with gradient 0.01. We also note that deep supervision is utilized [12].



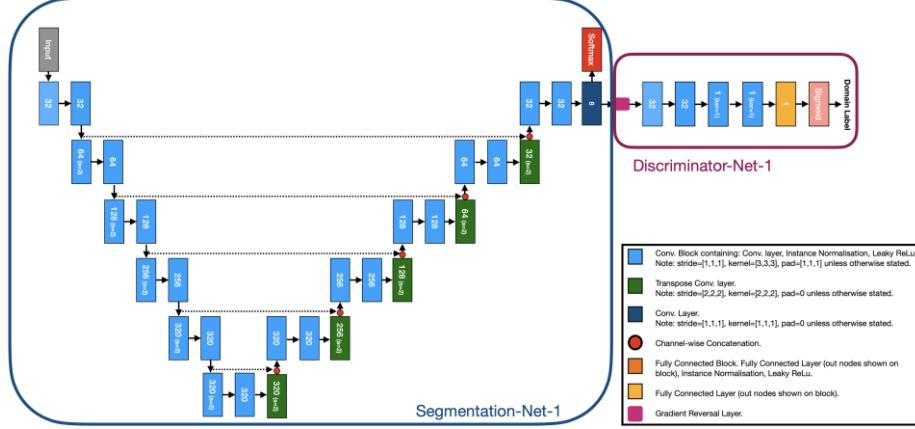
**Fig. 1.** Utilised UNet Inspired Segmentation Network.

**Domain Adversarial Approach.** The discriminator aims to recognize from which domain the input originates from i.e. from hospital 1 or hospital 2. We utilized two different models with the key difference being the location from which the feature maps from the segmentation network are inputted to the discriminator network.

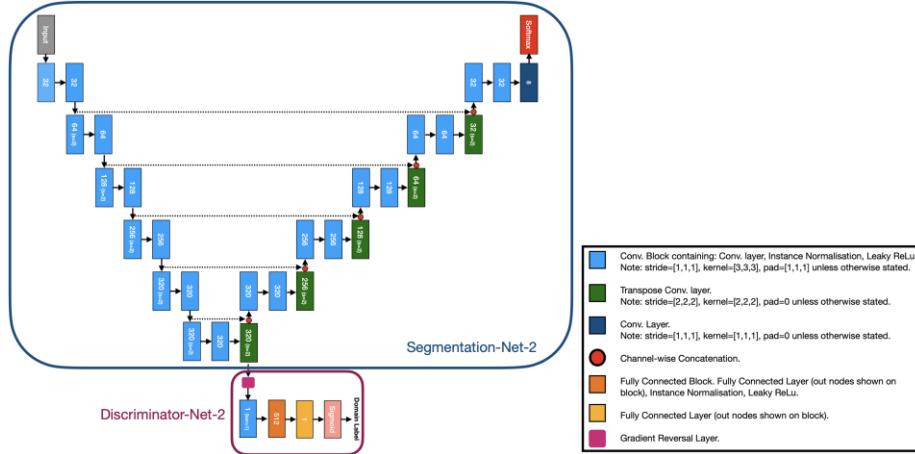
In model 1 the input to the Discriminator-Net-1 is the outputted feature map from the Segmentation-Net-1 just before the soft-max layer. The architecture for Discriminator-Net-1 is illustrated in Fig. 2. Model 1 aims to allow Segmentation-Net-1 to learn domain invariant features throughout the UNet.

In model 2 the input to the Discriminator-Net-2 is outputted feature map from the bottleneck layer (just before first upsampling block) of Segmentation-Net-2. The architecture for Discriminator-Net-2 is illustrated in Fig. 3; we note that Discriminator-Net-2 is shallower than Discriminator-Net-1 as the inputted feature map originates

from deeper in the UNet and is hence at a coarser scale. Model 2 aims to allow Segmentation-Net-2 to learn domain invariant features in the UNet's encoder.



**Fig. 2.** Illustration of Model 1. Discriminator-Net-1 takes as input feature maps outputted just before softmax layer of Segmentation-Net-1.



**Fig. 3.** Illustration of Model 2. Discriminator-Net-2 takes as input feature maps outputted from bottleneck layer of Segmentation-Net-2.

**Training and Loss Function.** The two models are trained separately using a training procedure similar to what is described by Ganin et al [10]. The main difference is that instead of training the classification network (in our case a segmentation network) using cases where only a single domain has class labels, we utilize the ground-truth labels in all cases from both domains. We hence aim to train the segmentation net-

work to perform optimally on both provided hospitals while also training it to learn domain invariant features.

The segmentation network utilizes a cross entropy and dice loss function as shown in Eqn.1. The discriminator network utilizes a custom loss function which is based on the weighted cross entropy function – we used a weight of 2 due to hospital 1 containing twice as many cases as hospital 2. The key difference of our custom loss function is the replacement of log function with a function inspired by the Witch of Agnesi function as shown in Eqn. 2. The overall objective function is shown in Eqn.3, where  $\lambda$  varies with training according to Eqn.4.

$$\mathcal{L}_{segmentation} = \mathcal{L}_{crossentropy} - \mathcal{L}_{dice} \quad (1)$$

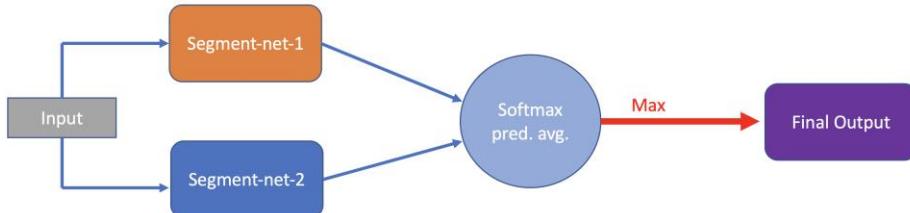
$$f(X) = \frac{8 \cdot (0.73279)^3}{(x)^2 + 4 \cdot (0.73279)^2} - 1 \quad (2)$$

$$\min \mathcal{L}_{objective} = \min [\mathcal{L}_{segmentation} - \lambda \cdot \mathcal{L}_{discriminator}] \quad (3)$$

$$\lambda = \frac{2}{1 + \exp(-10 \cdot \frac{current\_epoch}{1000})} - 1 \quad (4)$$

Effectively, during training the input image/patch is passed to the segmentation-net which will output the segmentation map. The discriminator-net will take as input a feature map from the segmentation-net and will output domain class labels. Important note: there is a gradient reversal layer inserted just before the discriminator-net (shown in Fig. 2 and Fig. 3) which will make the gradient passing to the segmentation-net negative during backpropagation – this ensures that the networks train adversarially. The outputs from both Discriminator-Net and Segmentation-Net are passed to the loss function shown in Eqn.3, and then backpropagation is executed.

**Inference and Ensemble Approach.** During Inference only the segmentation networks will be utilised, with the discriminators being disregarded i.e. only Segmentation-Net-1 and Segmentation-Net-2 utilised. Our overall method uses an ensemble approach to combine methods 1 and 2 described earlier. Segmentation-Net-1 and Segmentation-Net-2 will each output softmax predictions, which are then averaged in order to produce the final output as illustrated in Fig. 4.



**Fig. 4.** Visualisation of overall approach. Adverserially trained segmentation networks each predict the output which is then ensembled using a softmax averaging approach.

**Summary of Framework Details.** We utilized the nnUNet framework [11] for our submission to FeTA2022. For the segmentation network, we utilized the 3D UNet exclusively with all inputs being 3D. We did not do any hyperparameter optimization beyond what is automatically done by nnUNet and did not use cross-validation. The following are some of the key requested details:

- Preprocessing: We maintain the preprocessing steps conducted by nnUNet framework automatically. This includes use of intensity normalization via z-scoring (subtract mean and divide by stdev). A input patch size of 128x160x128 was utilised. For the resampling, third-order spline interpolation was utilized with the spacing was kept the same at 0.5039x0.5039x0.5039.
- Data Augmentation: We maintained the standard data augmentation utilized by the nnUNet framework. Augmentation hence includes: Rotations, scaling, Gaussian noise, Gaussian blur, brightness, contrast, simulation of low resolution, gamma correction and mirroring.
- Post-Processing: As discussed in the *Inference and Ensemble Approach* section, we train 2 networks Model 1 and Model 2 and use and ensemble approach for final prediction. The ensemble approach consists of averaging outputted softmax probabilities and then outputting the max. We do not utilize post-processing.
- Initialization: Kaiming He.
- Optimizer: Stochastic Gradient Descent with Nesterov momentum ( $\mu = 0.99$ ).
- Number of epochs: 1000 (250 mini-batch runs per epoch).
- Number of trainable parameters: Approximately 30,500,000 per model; therefore, approximately 61,000,000 in total.
- Learning Rate and Schedule: Initially set to 0.1 and decays according to following schedule:  $(1 - \frac{\text{epoch}}{1000})^{0.9}$
- Batch Size: Two.
- External Dataset: No External dataset used.
- Training time: Approximately 72 hours per model.
- Training/validation/testing data splits: 96/24/0 (we did not utilize test set for final prediction).
- Cases included: We included all cases and tried to roughly maintain same ratio (hospital 1 : hospital 2 ) in both training and validation set.

#### 1.4 Deepsynth

**Team Members:** Romain Valabregue\*

\*authors included in paper

We use, a standard 3D unet with residual connections implemented in pytorch (source <https://github.com/romainVala/unet> which is a modified forked of <https://github.com/fepegar/unet>).

We used 5 blocks of 3 convolutions with increasing number of output channels [24, 48, 96, 192, 384] for the encoder, and the symmetric decreasing order for the decoder. We used residual connection and a 3D kernel size of 2 with a dropout of 10% after each convolution layers. The input size for the training were 3D patches of dimension 128x128x128. This leads to 21 684 000 trainable parameters. We trained from random initialized weights with the Adam optimizer and a learning rate of 1e-4. The loss function is the mean dice score (average over the dice of each tissue label).

The training strategy follow the proposition of [13] to train on synthetic data. We did not add any original work to it, our contribution is more how to adapt the method for the FeTA Challenge. We re-implement the method within the torchio environment, and thus did different choice for the generative model.

We use the label from the dHCP dataset, with only the 80 youngest subjects. We used the desc-drawem9\_dseg, segmentation files, that contains 9 tissues classes, the same that the Feta challenge plus the hippocampi and Amygdala. (this label will be merge to Cortical Gray matter, for inference on FeTA data).

For the generative data synthesis, was perform with torchio [14] transform we use the following step:

- Random contrast: mean tissue intensity is chosen from  $U([0.1 \ 0.9])$  and a standard deviation randomly chosen from  $U([0 \ 0.001])$ .
- Random Affine (translation [-10 10] scales [0.9 1.1] rotation [-20 20]
- Random Anisotropy (with a probability of 0.5) the resolution in one of the third direction is re-slice to a slice thickness randomly chosen for  $U([1 \ 6])$
- Random BiasField (default torchio parameters)
- Random Noise global gaussian noise is added, with 0 mean, and a standard deviation from  $U([0.01 \ 0.1])$
- Rescale Intensity: min and max value are set to [0 1]

Because we did not have any labels for the body part outside the brain, we could not generate non brain tissue. We then adapt the synthetic generation to take the background from the T2 images. Doing so we lose the random contrast (for non-brain tissue only) but we had non-brain tissue (with T2 contrast).

For the training, each label generates a specific synthetic data with previously describe steps. Then 8 random patches of size 128x128x128 are extracted and used (after shuffle) for training with a batch size of 4

We first train with 200 000 iterations (with an iteration containing a batch size of 4). this took around 6 days. Then we fine-tune the model with a training on real T2 volume from the 80 first HCP subject. We perform only  $80*8=640$  iterations with a batch size of 1. Finally, we also fine tune the model with a training on the 80 subjects of the feta data set. Again, we perform only  $80*8=640$  iterations with a batch size of 1.

For the inference, all volumes (256x256x256) are processed at once, with a unique pre-processing step to rescale intensity between 0 and 1. Since we learned on the 9 dHCP labels, we further merged the hippocampi and amygdala with cortical gray matter

### 1.5 Dolphins: Coarse-to-Fine Models for FeTA2022 Segmentation

**Team Members:** Moona Mazher\*, Abdul Qayyum\*, Domènec Puig, Mohamed Abdell-Nasser

\*authors included in paper

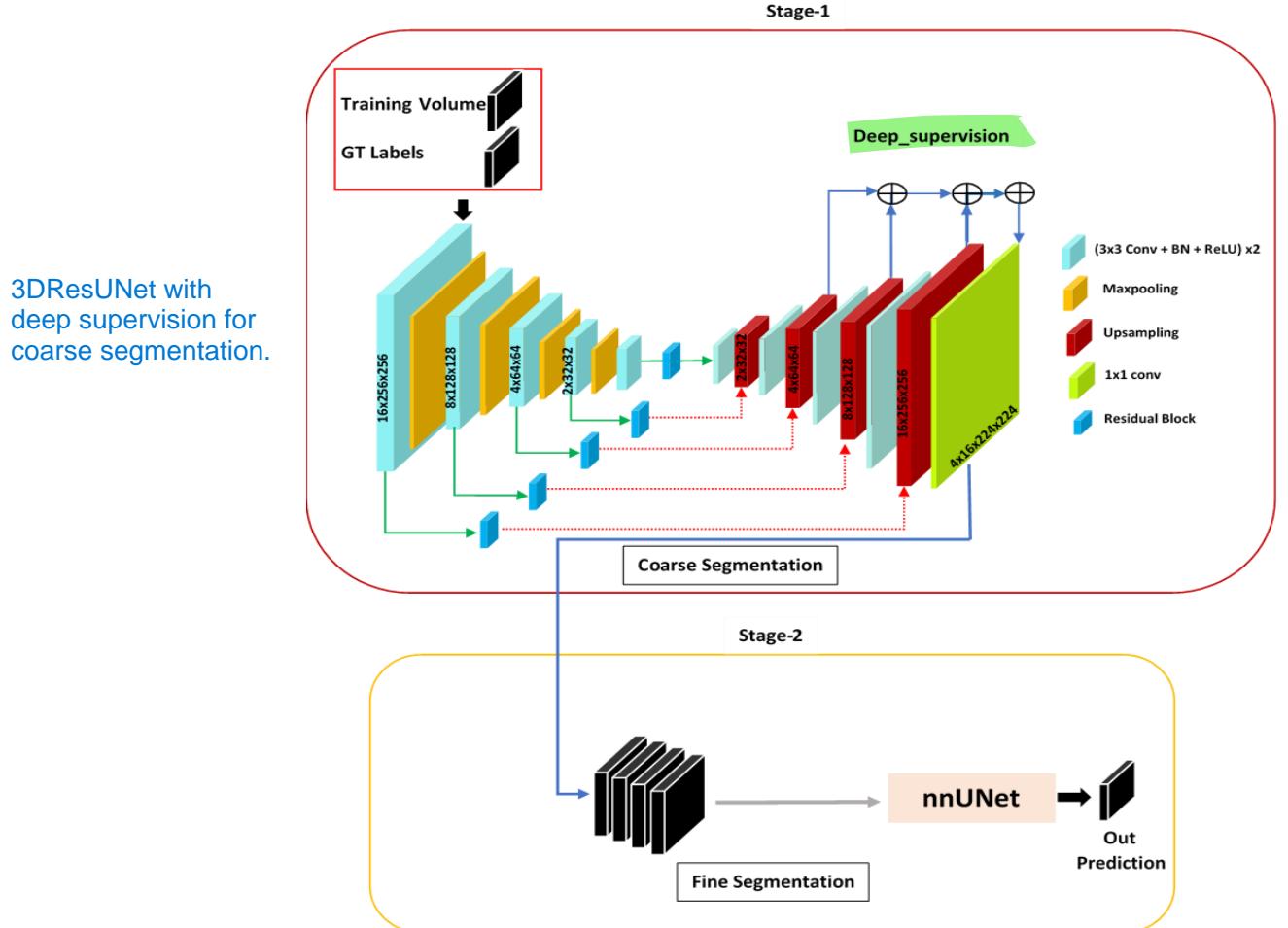
The main finding of the challenge is

1. Developed 3DResUNet with deep supervision for coarse segmentation
2. Use nnUNet for fine segmentation
3. Proposed 3DResUNet with deep supervision used for coarse segmentation and concatenated the output of coarse segmentation with nnUNet to get the fine segmentation output.
4. Two institutions dataset were used for training and validation of our proposed approach

The detailed description is shown in Fig. 5. The proposed solution consisted of two models, the first model in Stage 1 provided the coarse segmentation output and the second model used this coarse segmentation output coming from the Model 1.

**3D-ResUnet with Deep Supervision:** A framework of the proposed model is presented as an encoder, a decoder, and a baseline module. The 1x1 convolutional layer with softmax function has been used at the end of the proposed model. The 3D strides convolutional layer has been used to reduce the input image spatial size. The convolutional block consists of convolutional layers with Batch-Normalization and ReLU activation function to extract the different feature maps from each block on the encoder side. In the encoder block, the spatial input size has been reduced with an increasing number of feature maps and on the decoder side, the input image spatial size will increase using a 3D Conv-Transpose layer. The input features' maps that are obtained from every encoder block are concatenated with every decoder block feature map to reconstruct the semantic information. The convolutional (3x3x3conv-BN-ReLu) layer used the input feature maps extracted from every convolutional block on the encoder side and further passed these feature maps into the proposed residual module. The spatial size doubled at every decoder block and feature maps are halved at each decoder stage of the proposed model. The residual block has been inserted at each encoder block with skip connection. The feature concatenation has been done at every encoder and decoder block except the last 1x1 convolutional layer. The three-level deep-supervision technique is applied to get the aggregate loss between ground truth and prediction. We have used nnUNet with cross-validation and selected the best fold for FeTA 2022 segmentation, we have modified training and optimization pa-

rameters as compared to the original nnUNet. The batch size in nnUNet was 128x128x128 using 500 epochs.



**Fig. 5.** The proposed solution for the feta2022 segmentation model.

#### Model Details

- Model architecture: we have used coarse and fine segmentation approaches, in course segmentation, the proposed 3DResUNet model is trained and for fine segmentation, the nnUNet has been used for feta segmentation
- Number of layers: 10 numbers blocks were used and each block consisted of 3 layers (Conv-batch-relu)  $10 \times 3 = 30$  layers for each encoder and decoder and 15 layers for residual blocks, hence the total number of layers is 45
- Convolution kernel size: 3x3 and 5x5

- Initialization: “he” normal initialization
- Optimizer: Adam optimizer
- Cross-validation used: 5k fold cross validation and select the best one based on validation score
- Number of epochs: 200 epochs for coarse and 500 for fine segmentation model
- Number of trainable parameters: 23,889,221
- Learning Rate and schedule: None
- Loss Function: BCE+Dice for coarse and fine model
- Dimensionality of input/output (3D) 3D/3D
- Batch Size: 2 for coarse and fine models
- Preprocessing steps used: normalized data between 0 and 1, patch-size 128x128x128 for coarse and 128x128x128 for fine segmentation models
- Data Augmentation steps: HorizontalFlip (p=0.5), VerticalFlip (p=0.5), RandomGamma (p=0.8)
- The external dataset used? No
- Framework: nnUNet
- Number of models trained for final submission: two models
- Post-Processing Steps: None
- Training/validation/testing data splits: 80 % for training and 20% for validation
- Hyperparameter tuning performed: No
- Training time: 4 hours for coarse and 15 hours for fine model

We have trained our proposed 3DResUNet from scratch for coarse segmentation and used existing nnUNet (<https://github.com/MIC-DKFZ/nnUNet>) for fine segmentation. We have used all cases belonging to both institutions for training.

**Implementation Details.** Environments and requirements: The proposed deep learning model is implemented in PyTorch and other libraries based on python are used for preprocessing and analysis of the datasets. The SimpleITK is used for reading and writing the nifty data volume. The ITK-SNAP is used for data visualization. The environments and requirements of the proposed method are shown in **Error! Reference source not found..**

**Table 1.** Environments and requirements.

CPU	Intel(R) Core (TM) i9-7900X CPU@3.30GHz
RAM	16x2GB
GPU	Nvidia V100
CUDA version	11
Programming language	Python3.7
Deep learning framework	Pytorch (Torch 1.7.0, torchvision 0.2.2)
Specification of dependencies	SimpleITK, Numpy, Skimage, Scipy, Nibabel, ITK-SNAP

*Training protocols.* The learning rate of 0.0004 with Adam optimizer has been for training the proposed model. The binary cross-entropy function is used as a loss function between the output of the model and the ground-truth sample. 48 batch-size with 200 epochs has been used with 20 early stopping steps. The best model weights have been saved for prediction in the validation phase. The 256x256x96 input image size was used for training and prediction resample with the original input size at prediction using the nearest-neighbor interpolation method. The Pytorch library is used for model development, training, optimization, and testing. The V100 tesla NVidia-GPU machine is used for training and testing the proposed model. The data augmentation methods mentioned in Table.2. are used to further improve the results. The dataset cases have different intensity ranges. The dataset is normalized between 0 and 1 using the max and min intensity normalization method. The detail of the training protocol is shown in Table.2

**Table 2.** Training protocols.

Data augmentation methods	HorizontalFlip (p=0.5), VerticalFlip (p=0.5), RandomGamma (p=0.8)
Initialization of the network	“he” normal initialization
Patch sampling strategy	None
Batch size	2
Patch size	128x128x128
Total epochs	200
Optimizer	Adam
Initial learning rate	0.0001
Learning rate decay schedule	None
Stopping criteria, and optimal model selection criteria	The stopping criterion is reaching the maximum number of epochs (1000).
Training time	10 hours
Initialization of the network	“he” normal initialization

*Testing protocols.* The same preprocessing has been applied at testing time. The training size of each image is fixed (128x128x128) and used linear interpolation method to resample the prediction mask to the original shape for each validation volume. The prediction mask produced by our proposed model has been resampled such that it has the same size and spacing as the original image and copies all of the meta-data, i.e., origin, direction, orientation, etc.

## 1.6 FeTA-Imperial-TUM Team (FIT\_1) – FIT-nnU-Net

**Team Members:** Liu Li\*, Maik Dannecker, Chen Chen\*, Cheng Ouyang\*, Zeju Li, Benjamin Hou, Qingjie Meng, Bernhard Kainz, Daniel Rueckert

\*authors included in paper

**Model architecture.** In this submission, we employ 1) data augmentation-based domain generalization for improving model robustness on test images from unseen domains; 2) a network ensemble mechanism that combines the predictions from different segmentation models. These models are trained on aforementioned data augmentation strategies; 3) an output-level denoising autoencoder (DAE) [15] that corrects implausible predicted segmentations.

Specifically, we first trained 5 models separately with different data augmentation strategies to cover much as possible the distributions of potential target domain datasets. The detailed data augmentation strategies we used are described in the *Preprocessing* section. All of the models implemented are based on nnU-Net pipeline [11] with patch-based input. After that, we ensemble the segmentation results from different models by averaging the logit predictions and then choosing the category with the largest value as the ensemble prediction. In addition, to ensure the quality of final prediction when confronted with hard testing samples, we further employ a DAE-based post-processing strategy for further rectifying implausible target prediction results. Whether the predictions post-processed by DAE are adopted or not is dependent on the similarity of the predictions before and after DAE. Details about this post-processing strategy is discussed in the *Postprocessing* Section.

**Number of layers.** The final ensemble model comprises of 6 individual nnU-Net models, as highlighted in Table 3. Models 1 to 5 are used for segmentation, whereas model 6 is used for post-processing. All models share the same U-Net-like structure, with 5 stages of down sampling and 5 stages of up sampling. However, models 2-6 adopt the default nnU-Net architecture (two convolutions per resolution), whereas model 1 has three convolutions per resolution instead.

**Convolution kernel size.** All kernel sizes are set as  $3 \times 3 \times 3$  (default nnU-Net setting [11]).

**Initialization.** All models use random Kaiming initialization for weights (default nnU-Net setting [11], [16]).

**Optimizer.** All the models are optimized using stochastic gradient descent (SGD) with Nesterov momentum ( $\mu = 0.99$ ), same as the default nnU-Net setting [11].

**Cross-validation.** Instead of cross-validation, we split 20% samples (24 samples) for validation at the hyper-parameter tuning stage. Note, to fairly evaluate the generalization ability of our model, we further generate three challenging synthetic out-of domain validation sets based on these 24 samples. To this end we adopt nnUNet default data augmentation [11], random style augmentation [17] and random bias-field augmentation [18], respectively.

**Number of epochs.** The default setting in nnU-Net is 1000 epochs per model [11]. However, with strong data augmentation such as random-network-based augmentation [19] and style augmentation [17], these model cannot fully converge within 1000 epochs and thus the maximum epoch is set to 2000 epochs for models involving these augmentations, as shown in Table 3.

**Number of trainable parameters.** The numbers of trainable parameters for each models are shown in Table 3. As shown, model 2-6 that share default nnU-Net structure have 31.2M parameters, while model 1 that uses 3 convolutions per resolution has more parameters than model 2-6.

**Learning Rate and schedule.** All models are trained with a polynomial learning rate schedule, with an initial value of 0.01 [11].

**Table 3.** Model settings.

Model	Stage	Data Augmentation	No. Trainable Parameters (M)	Epochs	Time (hours)
1	Segmentation	default [11]	44.2	2000	74.3
2	Segmentation	default [11] + random bias [18]	31.2	1000	51.8
3	Segmentation	default [11] + random bias [18] + random style [17]	31.2	2000	100.4
4	Segmentation	default [11] + random network [19]	31.2	2000	72.2
5	Segmentation	default [11] + random motion [14]	31.2	1000	62.5
6	Post-Processing	None	31.2	1000	30.3

**Loss Function.** All models are trained using a combination of cross-entropy and soft Dice loss, as per the default setting in nnU-Net [11].

**Dimensionality.** All the models are trained in 3D, with an input patch size of 128×128×128.

**Batch Size.** Batch size is set to 5 for each model to satisfy memory constraints of a 24GB GPU.

**Preprocessing.** We follow the default preprocessing process of nnU-Net with intensity normalization, voxel resampling, (using a resampling factor based on the heuristics of all volumes in the training data set) and foreground-focused patch extraction [11].

**Data Augmentation.** As shown in Table 3, we trained 5 models with different data augmentation strategies. For models 1-5 (segmentation networks), we employ default data augmentations as those in nnU-Net, including rotations, scaling, additive Gaussi-

an noise, Gaussian blurring, brightness, contrast, simulation of low resolution, gamma correction and mirroring [11]. For model 2, we additionally include random bias-field augmentation [18]. Here we only use the basic random bias field augmentation function instead of the adversarial augmentation due to unfavourable cost-benefit ratio of hyper-parameter searching. The hyperparameter that controls the range of value for the multiplicative bias field is  $\epsilon = 0.8$ . From our validation results, we find this augmentation will preserve the performance on source domain while benefit the performance on our synthetic out-of-domain validation sets.

For model 3, we further include style augmentation [17] in addition to default and random bias-field augmentation. Here we use a style generator that is pretrained on ImageNet as an offline augmentation method.

For model 4, we utilize random networks [19] for photometric augmentation. From our synthetic out-of-domain validation set, we find that this method is most robust to the out-of-domain data, although with a slight side-effect on the source domain.

For model 5, we cover MRI-specific motion artifacts from the moving subjects. Here the motion artifacts are simulated by TorchIO [14], with hyperparameter set as translation=20, number transforms=2.

All 5 trained models are ensembled, based on average logit predictions (see the *Postprocessing* Section for details).

**External dataset.** No external dataset is used in our training.

**Framework.** We used nnU-Net as our framework [11].

**Number of models trained for final submission.** We trained 6 models for fetal brain segmentation, i.e., 5 basic segmentation models with different data augmentations, and 1 model for DAE post-processing, as shown in Table 3.

**Post-Processing.** In this submission, we have two stages of post-processing. **Model ensemble:** The first stage is model ensemble for 5 segmentation networks. Given the 5 pretrained segmentation models, we first run inference individually and got 5 logit predictions. Then, all the logit predictions are averaged and further passed by an argmax layer to get the final discrete segmentation result. **Rule-based post-processing with DAE:** The second stage is a rule-based post-processing using a denoising autoencoder (DAE). The DAE is designed to correct implausible predicted segmentation. The input to the DAE is the output prediction of the first stage, in the form of 8-channel one hot variables, and the output is a refined segmentation prediction. In order to train the DAE while avoid learning identity mappings only, we first generated a dataset with noisy segmentation, by running an inference of model 1 to 5 and randomly dropping out features from the encoders and the bottleneck layer of U-Nets. Empirically, we set the dropout ratio equals to 0.90 to get a visually noisy but not too damaged segmentation.

We noticed that this DAE would only improve the segmentation performance when the input prediction has low visual quality, by making well-observable corrections to implausible segmentations. However, it may slightly hurt the performance on

high-quality predictions that are already sufficiently accurate, and the changes in prediction before and after the DAE are slight. Based on the amount of changes in predictions before and after DAE, to achieve desirable accuracy on both high-quality and low-quality predictions, we apply the following empirical rule: We only trust predictions after DAE when large changes in predictions have been made, compared to the original predictions, i.e., the Dice similarity between pre-DAE prediction and post-DAE prediction is lower than  $0.70 \text{ } DICE_{SIM} <= 0.70$ . Otherwise, the original prediction (pre-DAE) from the first stage is trusted.

**Original work.** Our originality are two-folds: 1) we proposed data-augmentation based model ensemble for domain generalization, as discussed in the *Preprocessing* Section and 2) we further included a rule-based post-processing method for correcting implausible predictions, as discussed in the *Post-processing* Section.

**Citations and packages.** In this submission, the basic develop is based on the official release of nnUNet [11] (<https://github.com/MIC-DKFZ/nnUNet>). Besides, we used the data augmentation codes from [18] (<https://github.com/cherise215/AdvBias>), [19] (<https://github.com/cheng-01037/Causality-Medical-Image-Domain-Generalization>) , [17] and TorchIO [14] (<https://github.com/fepegar/torchio>) for random bias-field, random-network-based, style and motion augmentation, respectively.

**Which FeTA cases were included in the training and testing.** In the first hyperparameters tuning stage, we randomly split 24 samples (20%) from FeTA and Vienna datasets as validation set. Both pathological and normal cases are in the training and validation set.

**Training/validation/testing data splits.** After hyperparameter tuning and selecting the augmentation strategy, we use the full data (80 from FeTA and 40 from Vienna) for training.

**Hyperparameter tuning.** To adapt nnU-Net to our task, we changed several hyperparameters of nnU-Net, including number of epochs (from 1000 to 2000), number of convolutions per resolution stage (from 2 to 3), number of base features (from 32 to 48). Also, we tried different combination of data augmentation techniques and tuned their hyperparameters to balance the performance in both source domain and out-of-domain validation sets.

**Training time.** In our submission, we have 5 models for ensemble and 1 model for post-processing. Our experiment is conducted based on a single NVIDIA RTX A5000 GPU. The training time of the 6 models are listed in Table 3.

## 1.7 FeTA-Imperial-TUM Team (FIT\_2) – FIT-SwinUNETR;

**Team Members:** Liu Li, Maik Dannecker\*, Chen Chen\*, Cheng Ouyang\*, Zeju Li, Benjamin Hou, Qingjie Meng, Bernhard Kainz, Daniel Rueckert

\*authors included in paper

**GPU training was performed on:** Nvidia A5000/ A6000/ GTX 3080

**Model architecture.** In this submission, we proposed a transformer-based segmentation model, using the Swin UNETR architecture [20], [21] with MRI specific augmentation techniques from the library TorchIO [14]. Swin UNETR is a U-shaped segmentation model with a transformer-based encoder acting on multiple resolutions. The multiresolution outputs are fed to a Fully Convolutional Neural Network (FCNN) decoder. Additionally, we used a skull stripping model, based on the SynthStrip model [22], to separate the fetal brain from irrelevant background structure, such as the mother’s womb. This helped the segmentation model to focus on the region of interest, and furthermore, to drastically reduce the required training time. Since the provided pre-trained SynthStrip model did not achieve satisfying results on fetal brain MRI, we fine-tuned the model on the data provided by the FeTA2022 Challenge, using cross entropy loss and soft Dice loss.

**Number of layers.**

*Swin UNETR:* The transformer consists of 4x2 layers with 3, 6, 12, and 24 attention heads, respectively. The FCNN consists of 5x2 convolutional layers, 5 upsampling layers and a segmentation head.

*Synthstrip:* 7x2 layers for encoder and 6x2 layer for decoder + 1 output layer.

**Convolution kernel size.** For both models, all kernel sizes are set to  $3 \times 3 \times 3$ .

**Initialization.** We use random initialization for the Swin UNETR model and Xavier initialization for the Synthstrip model.

**Optimizer.** All the models use AdamW as optimizer.

**Cross-validation.** We didn’t do cross validation in our setting.

**Number of epochs.** We trained the Swin UNETR model for 2000 epochs, and the SynthStrip model for 200 epochs.

**Number of trainable parameters.** The Swin UNETR model has around 97M trainable parameters, whereas the SynthStrip model has around 2M trainable parameters.

**Learning Rate and schedule.** The learning rate is set to 1e-4 for both models and we used linear warmup and cosine annealing for 100 epochs.

**Loss Function.** We used the default loss function for Swin UNETR, a combination of weighted cross entropy loss and soft Dice loss. For the SynthStrip model we also used a combined cross entropy and soft Dice loss for the fine tuning on MRI of fetal brains.

**Dimensionality.** Both models used 3D input. Swin UNETR uses an input patch size of 64x64x64.

**Batch Size.** Batch size was set to 2 for both models.

**Pre-processing.** We reoriented the input image to RAS coding, resampled the image to an isotropic spacing of 1 mm, and applied intensity normalization.

**Data Augmentation.** Applied data augmentation included flipping, rotation and TorchIO [14] augmentations (affine+elastic transformation, noise, blur, gamma, ghosting, spike, motion, bias, blur, anisotropy).

**External dataset.** For training, neonate subjects of the dHCP data [23] (<http://www.developingconnectome.org/>) was used.

**Framework.** We used Swin UNETR from MONAI (<https://monai.io>) as framework.

**Number of models trained for final submission:** For final submission, we trained one Swin UNETR model and one skull stripping model.

**Post-Processing.** We resampled the image to the original spacing and orientation.

**Original work.** Existing work: SynthStrip [22] model and Swin UNETR [20], [21] model are publicly available Original work: the constructed pipeline of skull stripping + MRI specific data-augmentation, using the TorchIO [14] library, to segment fetal brains.

#### Citations and packages.

- Swin UNETR [20], [21]
- SynthStrip [22]
- TorchIO [14]

**Which FeTA cases were included in the training and testing (i.e. – all cases, only pathological, only 1 institution, etc.).** All cases were used for training and validation.

**Training/validation/testing data splits.** 98 for training and 22 for validation, uniformly sampled.

**Hyperparameter tuning performed.** We tuned feature size, warm-up epochs, and data-augmentation intensity/probability.

**Training time:** 5h training of skull stripping network, 24h pre-training transformer network, 24h training transformer network.

## 1.8 FMR SK

**Team Members:** Maria Deprez, Alena Uus\*, Irina Grigorescu\*, Paula Ramirez Gil-liland\*

\*authors included in paper

**GPU training was performed on.** NVIDIA Titan XP, NVIDIA GeForce RTX 3090

**Software used.** Pytorch v1.10.2, MONAI v0.9.0, TorchIO v0.18.73 [14]

**Model architecture.** Attention UNet based on Otkay et al. "Attention U-Net: Learning Where to Look for the Pancreas" <https://arxiv.org/abs/1804.03999>, the MONAI implementation [24]

**Number of layers.** 5 layers with 32, 64, 128, 256, and 512 channels respectively

**Convolution kernel size.** 3x3x3

**Initialization.** He initialization

**Optimizer.** AdamW with default parameters and a weight decay of 0.00001

**Cross-validation used.** No

**Number of epochs.** 300

**Number of trainable parameters:** 23.6 M

**Learning Rate and scheduler:** lr=0.001 with a linearly decaying scheduler

**Loss Function:** a combination of Dice and Cross Entropy loss (DiceCELoss from MONAI)

**Dimensionality of input/output.** 3D

**Batch Size.** 2

**Preprocessing steps used.**

- We trained a standard 3D UNet from MONAI (<https://github.com/Project-MONAI/MONAI>) to perform brain extraction on all the training data and we use it as a preprocessing step in the Docker.
- All image and label volumes are resampled on the same 128x128x128 grid, and transformed to the standard atlas space
- Intensity normalisation between 0 and 1

### **Data Augmentation.**

- Motion artifacts (-6 -> +6 degrees rotations, -6 -> +6 voxels translations)
- MR Spike
- Bias field
- Affine transforms (scaling between 0.7 and 1.2, rotations between -60 degrees to +60 degrees)
- Noise (mean = 0.0, std between 0.001 and 0.05)
- Blurring (std between 0.2 and 1.0)
- Gamma (log\_gamma between -0.4 and 0.4)
- Random intensity shifts

**External dataset used.** dHCP neonates: 19 dHCP neonates with 23.7 -- 30.7 weeks gestational age at birth, and 26.6 -- 32.4 weeks postmenstrual age at scan (dHCP public release <http://www.developingconnectome.org/> [25]); Spina bifida atlases [26]

**Framework.** Attention UNet part of MONAI v0.9.0

**Number of models trained for final submission.** 2 models

**Post-Processing Steps.** We average the predictions of our 2 models

**Original work.** We used a semi-supervised approach to training our networks.

As a first step, we manually checked the training data provided and scored each volume based on the quality of the labels. We then used only the high-quality fetal label data to train an initial Attention UNet (MONAI, same as for our final model) to produce labels for the remaining datasets. The predicted labels were then manually corrected, and the process was repeated 3 times until we were satisfied with the quality of labels for the entire training dataset.

For the training of the final two models, on top of the on-the-fly augmentation explained above, we augmented the training dataset in two ways: 1) we smoothed the brain masks to create two types of images of the same subject: one with the original brain extraction, and one with an enlarged mask that encompasses more of the surrounding structures; and 2) we flipped the images and labels along the left-right direction.

**Which FeTA cases were included in the training and testing.** All cases except for low quality labels (sub-007, sub-022, sub-029, sub-035, sub-108, sub-119, sub-120, sub-134), and one subject (sub-125) which did not have a label; in total 111 FeTA cases

**Training/validation/testing data splits.** The final models are trained on all cases

**Hyperparameter tuning performed.** No

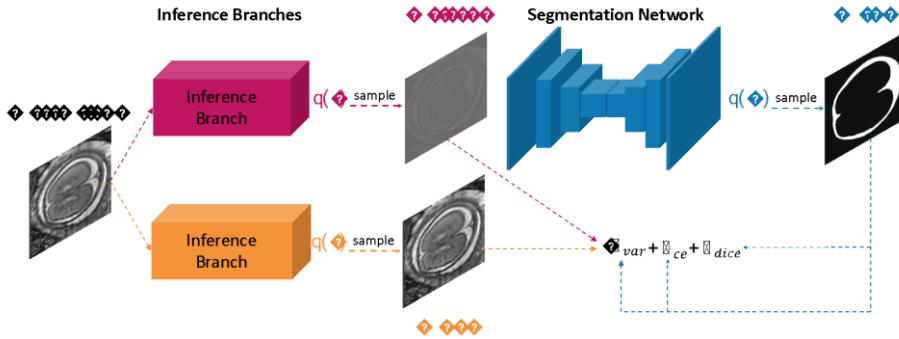
**Training time.** 23 hours

### 1.9 fudan\_zmic

**Team Members:** Yibo Gao\*, Hangqi Zhou\*, Shangqi Gao\*

\*authors included in paper

**Model Description.** Our method implements BayeSeg [27] based on nnUNet [11] framework. The code is published at <https://github.com/o00O00o/nBayeSeg>.



**Fig. 6.** Network Architecture of BayeSeg

BayeSeg is composed of two stages as shown in Fig. 6. In the inference stage, two inference branches are implemented to infer the distribution of contour and basis (i.e.,  $q(c)$  and  $q(b)$ ) respectively. Then, we sample  $c$  and  $b$  from the inferred distributions. In the segmentation stage, the sampled  $c$  is fed to a segmentation network to obtain the predicted mask. It is worth noting that the output of the segmentation network is also a distribution (i.e.,  $q(m)$ ). The final prediction is a random sample from  $q(m)$ . The framework is optimized by minimizing cross-entropy, dice loss and the weighted variational loss. The weight  $\lambda$  is set to 50 and the variational loss is elaborated in [27].

The contour inference branch consists of 10 residual blocks, and each block has a structure of “Conv + ReLU + Conv”. The output of this branch has two channels. One is the element-wise mean of the contour, and the other is its element-wise variance. The contour  $c$  in the figure denotes a random sample from  $q(c)$ . The basis inference branch consists of 6 residual blocks, and each block has a structure of “Conv + BN + ReLU + Conv + BN”. Similarly, this branch will output the mean and variance of the basis, and the basis  $n$  is randomly sampled from its variational posterior distribution.

At the segmentation stage, the segmentation network in the figure is automatically generated by nnUNet framework. It is a UNet with instance normalization and leaky ReLU following every convolution layer. The UNet infers the variational posterior of the label  $z$ , i.e.,  $q(z)$ . The output of this U-Net has 2K channels. The first K channels denote the element-wise mean of the label, and the left channels represent its element-wise variance. The label  $q(z)$  in Fig. 6 is a random sample from the resulting posterior distribution, and it will be taken as a stochastic segmentation for training.

**Training Method.** Our framework is based on nnUNet [11], which is publicly available at <https://github.com/MIC-DKFZ/nnUNet>. Training was done on TITAN RTX with Pytorch 1.12.0.

Training inputs are 2D images which are randomly sampled from training cases and the patch size of 224x192 is selected, with 24 batch size. Before fed into the network, all the training images need to be cropped according to the foreground. We also use Z score (mean subtraction and division by standard deviation) per image followed by cropping for data normalization. During training, randomly initialized network is optimized by Adam optimizer with 0.0001 initial learning rate is used. The learning rate annealing strategy is the default one in nnUNet. The maximum training epoch is 1000, and one epoch is defined as 250 iterations. Each epoch costs about 160s. We use all cases of FeTA dataset and an extra cardiac ACDC dataset for cutmix. We split the dataset into 5 folds so that we can run a 5 folds cross validation.

During validation, we found that when training on institution 1 dataset and testing on institution 2, the segmentation results degraded a lot. We supposed that the reason might be that the background of institution 2 images is much more complicated than that of institution 1 images. Thus, we added an extra cardiac cutmix augmentation to enrich the background. We randomly cut out the center of FeTA images with a random side length ratio sampled from uniform distribution between 0.6-0.9, and then paste it in the same position in ACDC cardiac images. The probability of cardiac cutmix is 0.5. Except for cardiac cutmix augmentation, other default augmentation techniques in nnUNet are also implemented. More details can be seen in [11].

For FeTA challenge, we have trained 5 models via a 5 folds cross validation. And all the 5 models were selected to run ensembling.

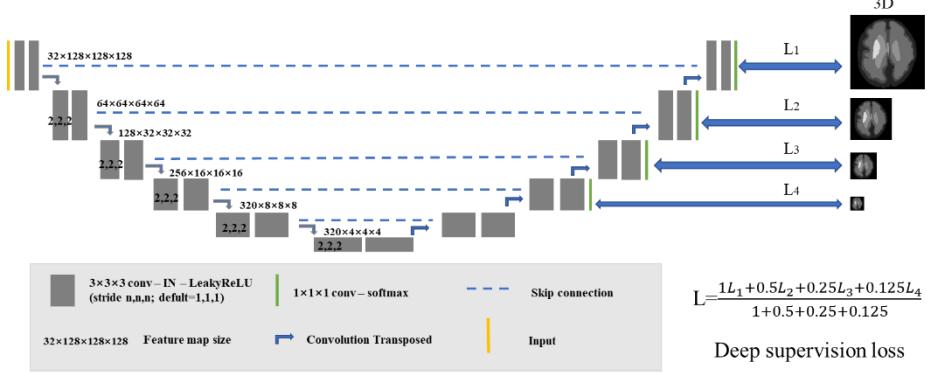
## 1.10 hilab

**Team Members:** Jia Fu\*, Guiming Dong\*, Guotai Wang\*

\*authors included in paper

**Model Description:** Our approach is based on the nnUNet (<https://github.com/MIC-DKFZ/nnUNet>) [11], which is implemented with Pytorch. To segment the fetal brain tissue, we propose a coarse-to-fine framework inspired by [28], which divides the segmentation process into two stages. In the first stage, we use 3D U-Net [2] to segment the seven tissues simultaneously. The regions of interest (ROIs) for each tissue are generated by the coarse segmentation results from the first stage. In the second stage, we train seven residual 3D U-Net models [4] separately based on the ROIs to achieve fine segmentation of brain tissues.

In the coarse and fine stages, the 3D U-Net and residual 3D U-Net share a similar architecture. The detailed structure of 3D U-Net in the first stage is shown in Fig. 7. The residual 3D U-Net uses residual blocks in the encoder to substitute the convolution blocks of 3D U-Net. The residual block is implemented as conv-IN-LeakyReLU-conv-IN-LeakyReLU, where the residuals are added before the last LeakyReLU activation [29].



**Fig. 7.** The network architecture of the 3D U-Net for coarse stage

**Data Preprocessing and augmentation.** In the first stage, we first cropped the non-zero region of each reconstructed fetal brain image. Then, each volume was normalized by Z-score and used as the input of 3D U-Net. Standard data augmentation strategies were used, including rotation and scaling, Gaussian noise, Gaussian blur, brightness and contrast adjustment, simulation of low resolution, gamma augmentation, and mirroring. The patch size was 128×128×128.

In the second stage, we cropped the ROIs of each tissue according to the coarse segmentation result in the first stage. Specifically, the whole non-zero regions were selected as the ROIs of external cerebrospinal fluid, grey matter, and white matter without further cropping. For the ventricles, cerebellum, deep grey matter, and brain-stem, the ROIs were determined by the segmentation results of 3D U-Net with a bias of 10. The data augmentation methods used in the second stage were the same as in the first stage.

**Implementation Details.** We used all provided data of the FeTA 2022 dataset and did not use any external dataset. We split all the training data randomly into 96 training cases and 24 test cases for all models.

All the model parameters were initialized randomly. In the two stages, we used the combination of cross-entropy and dice loss to train the segmentation networks. We used the Adam optimizer with an initial learning rate of 1e-3 and decreased it to zero at the end of the final epoch using Nesterov momentum ( $\mu = 0.99$ ). The weight decay was set as 3e-5, and the batch size was 2.

We trained the networks for 400 epochs at most, and we only saved the models with the highest dice coefficient on the test dataset in the two stages. In the first stage, each epoch cost about 120 s. In the second stage, each epoch cost about 140 s for the segmentation task of ventricles, while each epoch cost 50~80 s for the segmentation task of other tissues.

Our algorithm was implemented in Python 3.8.13 using Pytorch 1.12.0 framework. Experiments were performed on one NVIDIA GeForce RTX 2080 Ti or NVIDIA GeForce RTX 3090.

**Inference and Post-processing.** In the inference stage, the segmentation result is produced by the ensemble of the saved model of the second stage. If there is an overlap of segmentation masks for different tissues, the result produced by the saved model of the first stage is treated as the predicted label for the overlapping region.

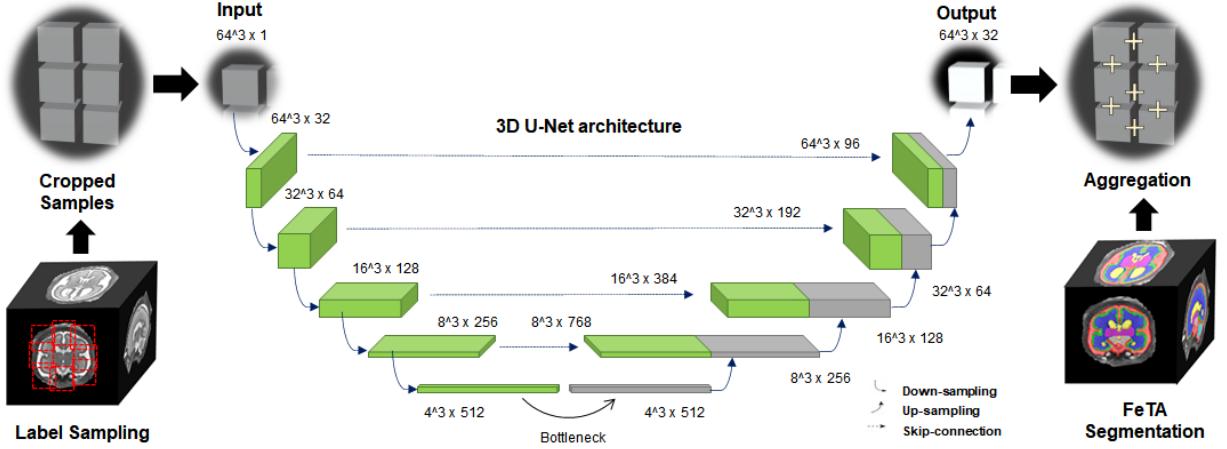
### 1.11 Neurophet

**Team Members:** ZunHyan Rieu\*, HyeonSik Yang\*, Minwoo Lee, Jimin Kang, Minho Lee\*

\*authors included in paper

**Model Description.** We used three-dimensional (3D) U-Net [30] as the baseline architecture of our model. Since the actual image data with fetal tissue have various resolutions depending on the institutions, we chose patch-based training with the input voxel size of 64x64x64. Our model consists of 5 encoding blocks with residual blocks. We applied a probability-based sampling method to make our model focus on the fetal tissue regions instead of the zero-padded regions.

**Training Method.** The entire training was conducted using Python 3.8 [31] using PyTorch 1.8 [32] as our main deep learning framework. For data augmentation process, we utilized torchIO [14], MONAI [33] and Kornia [34]. We strictly used the provided dataset from the challenge, which means neither an additional dataset nor pre-trained model weights for our training process. Since the quality of MRI was unidentified, we initially performed the visual inspection to define the poor and good quality of MRI from the dataset and distributed them equally to the training and validation dataset. For the pre-processing, we performed intensity normalization from 0 to 1 with a percentile cutoff of (0, 100). For data augmentation, we performed the spatial augmentations (horizontal flip, rotation, and affine transformation) and intensity augmentation (Gaussian blur).

**Fig. 8.** Model Architecture

All models were trained on GPUs (RTX3090 24GB x 4 / CUDA 11.1) with the following parameters:

- Optimizer: AdamW [35]
- Number of epochs: 300 (weights saved on minimum validation loss)
- Number of trainable parameters: 314,999,688
- Learning Rate and schedule: 2e-4
- Loss Function: DiceCELoss (custom weight)
- Batch Size: 24
- Samples per volume: 64

We trained the following 3 individual models:

1. Head only model (used the dataset of Institution 1 ONLY)
2. Head with body model (used the dataset of Institution 2 ONLY)
3. Combined model (Institution 1 & Institution 2)

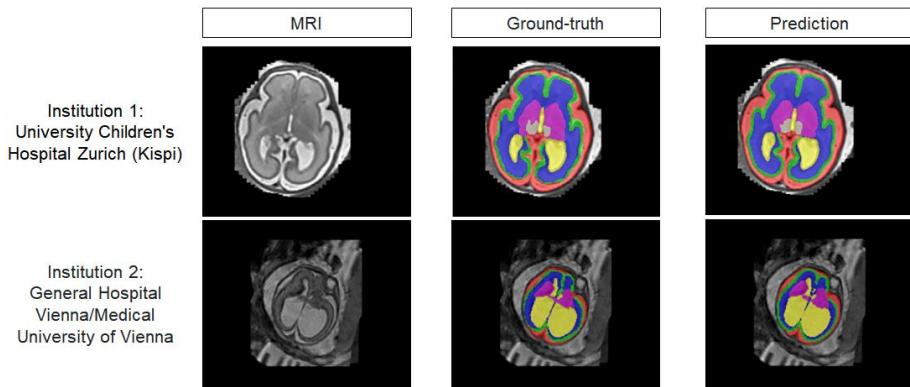
**Inference Method.** We initially found that models A and B performed the best on their trained institution dataset yet couldn't work vice-versa. However, model C, trained with institutions 1 and 2 datasets, could work vice-versa with mediocre performance. Therefore, to find the best yet safe segmentation result for the unknown input MRI, we designed a method to decide the best-performed model with the segmented volume.

From the input MRI, we perform inferences for all 3 models with the patch overlap size of 48x48x48. After that, we measure the non-zero voxels for the segmented results for all labels, which works like the intracranial volume (ICV). Once the non-zero voxels for all models are measured, we compare the volumes for Case 1 ( $1 - (A/C)$ ) and Case 2 ( $1 - (B/C)$ ). For example, if the absolute difference of either case 1 or case 2

is less than 0.07, we set that case as True, or else False. At last, the algorithm decides the best-performed model using the following decision matrix and outputs the result.

**Table 4.** Results

	Case 1	Case 2	Output
Decision Algorithm	True	False	Model A
	False	True	Model B
	True	True	Model C
	False	False	Model C



**Fig. 9.** Inference Examples

## 1.12 NVAUTO

**Team Members:** Md Mahfuzur Rahman Siddiquee\*, Dong Yang, Yufan He, Da-guang Xu\*, Andriy Myronenko\*

\*authors included in paper

**GPU training was performed on.** V100 16GB (each model was trained on 8 GPUs)

**Software used.** Pytorch

### Method.

**The Network.** We implemented our approach with MONAI (<https://github.com/Project-MONAI/MONAI>) [33]. We use the encoder-decoder backbone based on [36] with an asymmetrically larger encoder to extract image features and a smaller decoder to reconstruct the segmentation mask [2], [37], [38].

**Encoder part:** The encoder part uses ResNet [39] blocks. We have used 5 stages of down-sampling, each stage has 1, 2, 2, 4, and 4 convolutional blocks, respectively. We have used batch normalization and ReLU. Each block's output is followed by an additive identity skip connection. We follow a common CNN approach to progressively downsize image dimensions by 2 and simultaneously increase feature size by 2. For downsizing, we use strided convolutions. All convolutions are  $3 \times 3 \times 3$  with an initial number of filters equal to 32. The encoder is trained with  $224 \times 224 \times 144$  input region.

**Decoder part:** The decoder structure is similar to the encoder one, but with a single block per each spatial level. Each decoder level begins with upsizing with transposed convolution: reducing the number of features by a factor of 2 and doubling the spatial dimension, followed by the addition of encoder output of the equivalent spatial level. The end of the decoder has the same spatial size as the original image, and the number of features equal to the initial input feature size, followed by a  $1 \times 1 \times 1$  convolution into 8 channels and a softmax.

### **Training Method.**

*Dataset.* We have used the FeTA dataset [40] only for training the model. We have randomly split the entire dataset into 5-folds and trained a model for each.

*Loss.* We have used Dice Focal loss for training.

*Optimization.* We use the AdamW optimizer with an initial learning rate of  $2e-4$  and decrease it to zero at the end of the final epoch using the Cosine annealing scheduler. We have used a batch size of 8. The model is trained of 8 GPUs, each GPU optimizing for a batch size of 1. However, we have calculated batch normalization across all the GPUs. We have ensembled 15 models for submission. All the models were trained for 1000 epochs. All of these models were trained with deep supervision.

*Regularization.* We use L2 norm regularization on the convolutional kernel parameters with a weight of  $1e-5$ .

*Data preprocessing and augmentation.* We normalize all input images to have zero mean and unit std (based on nonzero voxels only). We have applied random rotation, random zoom on each axis, random Gaussian smoothing, and random Gaussian noise with a probability of 0.2. We have also applied random flip on each axis and random contrast adjustment with a probability of 0.5.

**Initialization.** Kaiming uniform

**Number of trainable parameters.** 87 million

**Training/validation/testing data splits.** 80%/20% (training/validation) split

**Hyperparameter tuning performed.** Manually

**Training time.** 8 hours

**Results on Cross-Validation.** Our cross-validation results on the 5-folds can be found in Table 5.

**Table 5.** Average DICE among classes using 5-fold cross-validation.

Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Average
0.8453	0.8532	0.8362	0.8301	0.8568	0.8431

### 1.13 Pasteur DBC

**Team Members:** Jean-Baptiste Masson\*, Astrid Nilsson\*, Charlotte Godard\*

\*authors included in paper

All the required information is listed below:

- GPU training was performed on: 4 tesla V100 GPUS (32GB of VRAM) with CUDA version 11.4
- Software used: PyTorch (1.12), cudatoolkit (11.3), numpy (1.22), nibabel (4.0)
- Model architecture: ensemble of 3 two-dimensional UNets fed respectively with image along the coronal, facial and sagittal axis + majority vote on the models predictions
- Number of layers: 10 convolutional blocks + 10 deconvolutional blocks per model
- Convolution kernel size: 3 x 3
- Initialization: Random without bias
- Optimizer: Adam algorithm
- Cross-validation used? No
- Number of epochs: 1000
- Number of trainable parameters: 166K x3
- Learning Rate and schedule: 0.001, no scheduler
- Parameters of the Adam optimizer: momentum = 0.9, weight\_decay = 1e-6
- Loss Function: CrossEntropy
- Dimensionality of input/output (ie: 2D, 3D, 2D+, etc.) : inputs are 3D images split in slices and fed to the models. Outputs are 2D slices stacked together to form a 3D segmentation
- Batch Size: 32
- Preprocessing steps used: Images normalisation + contrast adjustment
- Data Augmentation steps: noise + blur + 2D rotations + 2D translations + 2D horizontal flip + zoom
- External dataset used? Yes: fetal-brain-atlas-serag, fetal\_brain\_atlas Gholipour (2017) were used for self-supervised tasks
- Framework (ie – MONAI, nnUNet, etc.) Pytorch
- Number of models trained for final submission: 3

- Post-Processing Steps (ie – ensemble network, voting, label fusion) : majority voting on models prediction maps
- Clearly state which aspects are original work (if any) or already existing work: using self-supervised learning (SSL) for model pretraining
- Which FeTA cases were included: all cases were included in the training set
- Training/validation/testing data splits: 80/20/0
- Hyperparameter tuning performed: No
- Training time: 3 days

### 1.14 Sano

**Team Members:** Szymon Płotka\*, Michał K. Grzeszczyk\*, Arkadiusz Sitek\*  
 \*authors included in paper

**Model description.** We used Swin UNETR [20] as our base model. We used an original implementation from the official MONAI repository (<https://monai.io/>). We used Swin UNETR configuration as follows. As input, we fed  $128 \times 128 \times 128$  patch size to the network. Embedding dimension is set to 768, feature size to 60, number of blocks = [2, 2, 2, 2], window size = [7, 7, 7], and number of heads = [3, 6, 12, 24].

**Training method.** *Dataset:* For training, we used only FeTA 2022 dataset. The dataset consists of 120 T2-weighted fetal brain reconstructions from two different institutions: University Children’s Hospital Zurich and General Hospital Vienna/Medical University of Vienna with a corresponding label map that was manually segmented into 7 different tissues/labels: 1. External Cerebrospinal Fluid, 2. Grey Matter, 3. White Matter, 4. Ventricles, 5. Cerebellum, 6. Deep Grey Matter, 7. Brainstem, and background.

*Optimization:* We used the AdamW optimizer with an initial learning rate of  $2e-4$  and decrease it to zero at the end of the final epoch using the Cosine annealing scheduler. As regularization, we used L2 with a weight of  $1e-5$ . We used a batch size of 2. We implemented our solution using PyTorch and MONAI (<https://monai.io/>). We trained our method using  $2 \times$  NVIDIA A100 80GB GPUs. We have ensembled 5 models for the submission. All the models were trained for 800 epochs.

*Loss function:* We used a sum of Cross Entropy Loss and Dice Loss for training defined as:

$$\mathcal{L}_{sum} = \lambda_1 \mathcal{L}_{CE} + \lambda_2 \mathcal{L}_{Dice} \quad (1)$$

where  $\lambda_1 = \lambda_2 = 1$ .

*Data pre-processing and augmentation:* First of all, we normalize the input data. Then, we scale each input to the same spacing:  $0.8 \times 0.8 \times 0.8$ . Finally, we apply the data augmentation as follows: Crop Foreground (p=1.0), Random spatial crop

( $p=1.0$ ), Random zoom ( $p=0.3$ ), Random rotate ( $p=0.3$ ), Random Gaussian Noise ( $p=0.2$ ), Random Adjust Contrast ( $p=0.3$ ), Random Flip on each axis ( $p=0.5$ ).

**Results.** We used 5-fold Cross-Validation (CV) to validate the effectiveness and efficiency of our solution. The results are presented in Table 6.

**Table 6.** Average Dice among classes on FeTA 2022 challenge dataset

Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Average
0.867	0.845	0.863	0.845	0.860	0.856

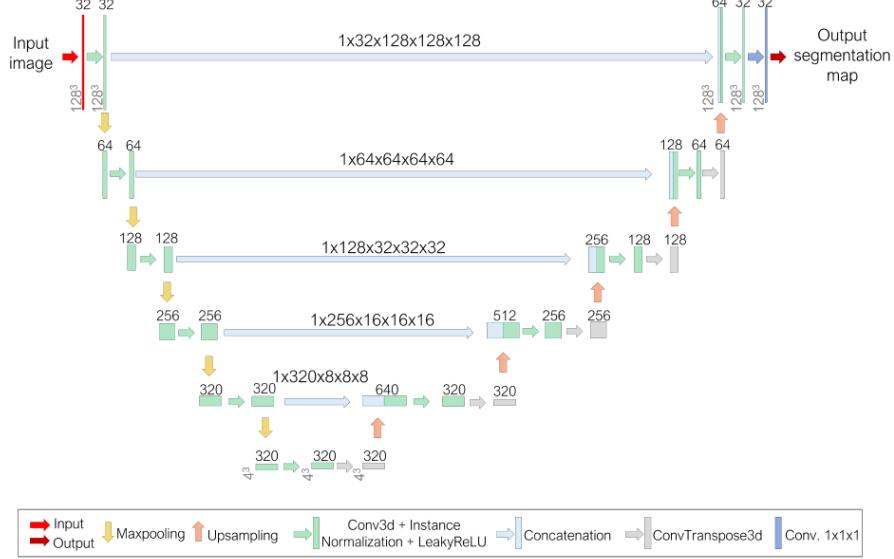
### 1.15 symsense

**Team Members:** Valentin Comte\*, Mireia Alenyà\*, Oscar Camara\*, Miguel Angel Gonzalez Ballester, Gemma Piella

\*authors included in paper

**Model description.** The baseline segmentation model we used for this challenge is the well-known nnU-Net [11]. The overall architecture of the model has not been changed, but we implemented a generalized Dice Loss proposed by [41] which accounts for the volume of segmented structures (Section Loss Function). We also performed data augmentation techniques to increase the size of our data set and make our model more robust to domain changes (Sections Data Augmentation: warping; Data Augmentation: GIN-IPA).

*Model architecture:* The model architecture is based on U-Net, the most popular CNN for image segmentation, which was introduced by [2]. It is an autoencoder-like CNN, made of a “contracting path” and an “expansive path”, or analysis path and synthesis path, respectively. The analysis path is composed by a succession of convolutional layers followed by ReLU activations and max pooling layers, this set of stacked layers being repeated four times. The synthesis path uses up-sampling layers followed by up-convolution layers, the output is then concatenated with the corresponding skipped feature map from the contracting path, and two successive convolutional layers, each followed by a ReLU activation.



**Fig. 10.** Architecture of the nnUNet.

*Training Method:* To train the network all cases provided (80 from Zurich and 40 from Vienna) were used. The dataset was enlarged by warping these cases with the method described in Section 1.4.2) and adding the 40 early-neonatal cases from the Development Human Connectome Project (dHCP - <https://data.developingconnectome.org/>) (Section Additional Datasets).

#### Model details

- The used network is 3D-nnU-Net with full resolution
- All five folds were run a total of 900 epochs
- The number of trainable parameters of the model is: 31200448
- Inference is done through five-fold cross-validation
- Dimensionality of inputs/outputs: 3D with one extra channel (See Section 1.4.3)
- Preprocessing: cropping, resampling and data normalization (z-normed)
- Batch size: 2
- Patch size: [128 128 128]
- Pool kernel sizes: [[2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2], [2, 2, 2]],
- Convolution kernel sizes: [[3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3], [3, 3, 3]]
- Stride [1, 1, 1] or [2, 2, 2] and padding [1, 1, 1]
- Initial loss rate: 0.01
- Optimizer: Stochastic gradient descent (SGD) with Nesterov momentum ( $\mu = 0.99$ )
- Activation function: LeakyReLU with negative slope 0.01
- Instance normalization with parameters: eps=10<sup>-5</sup>, momentum=0.1, affine=True
- No Post-Processing Steps are performed

*Initialization of model parameters:* The initialization of model parameters is fully optimized by the network after extracting the dataset fingerprint (a set of dataset-specific properties such as image sizes, voxel spacings, intensity information, etc).

*Data Augmentation Strategies:* The following data augmentation features have been applied:

- Rotation along each axis, range (-15°, 15°)
- Elastic deformation
- Scaling, range (0.85, 1.25)
- Add Gaussian noise, range (0, 0.1)
- Add Gaussian blur, range (0.5, 1)
- Gamma Transform, range (0.7, 1.5)
- Mirror along all axes
- Additive brightness transform, range (0.75, 1.25)
- Contrast transform, range (0.75, 1.25)
- Simulate low resolution transform, zoom range (0.5, 1)
- Warping of images (See Section 1.4.2)
- GIN-IPA (See Section 1.4.3)

#### 1.2.4 Software and training time

- Python 3.8
- CUDA 11.4
- Pytorch 1.11
- batchgenerators 0.24
- SimpleITK 2.1.1
- GPU: Quadro RTX 6000 (Turing), 24 GB
- Training time: 12-13h per fold /(each epoch 150s)

*Additional data sets:* 40 early neonatal cases from the publicly available dHCP database were included in the training dataset.

**Our contribution.** *Loss function: Generalized dice + cross-entropy.* The submitted model presents a modification in the computation of the loss function. It still uses Dice and cross entropy terms as the original network does, but the former has been modified to be a generalized Dice as presented in [41]. This new Dice metric assesses multi-label segmentations with a unique score, assigning a different weight for each structure according to its volume, and can be expressed as follows:

$$DICE_{ml} = \frac{2TC_{ml}}{TC_{ml} + 1}, TC_{ml} = \frac{\sum_l \alpha_l \sum_i \min(GT_{l,i}, X_{l,i})}{\sum_l \alpha_l \sum_i \max(GT_{l,i}, X_{l,i})}, \alpha_l = \frac{1}{V_l}$$

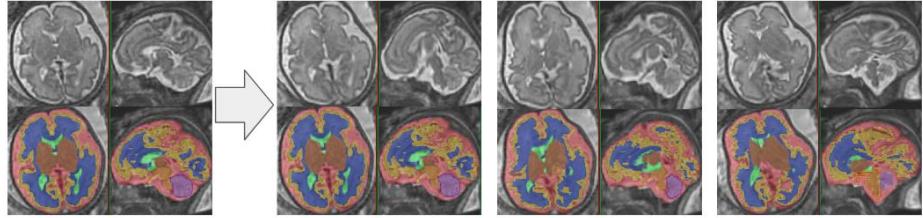
where  $TC_{ml}$  is the multilabel Tanimoto coefficient [11],  $GT_{l,i}$  is the value of the ground-truth segmentation of voxel  $i$  for label  $l$ ,  $X_{l,i}$  is the analogous for the predicted

one,  $\alpha_l = \frac{1}{V_l}$  is the label-specific weighting factor that affects how much each structure  $l$  contributes to the overlap accumulated over all labels; and  $V_l$  is the volume of each label  $l$ .

*Data augmentation: warping:* The first data augmentation step that we performed was to warp the input images and labels using a mixture of random 3D Gaussians, as described by:

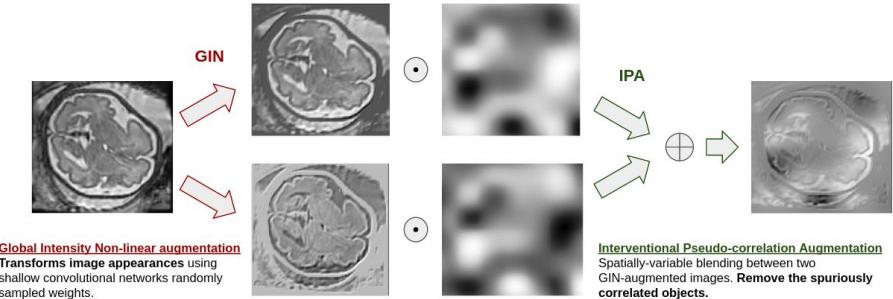
$$G(x, y, z) = \sum_i^N A_i \exp \left( \frac{(x - x_{0,i})^2}{2\sigma_{x,i}^2} + \frac{(y - y_{0,i})^2}{2\sigma_{y,i}^2} + \frac{(z - z_{0,i})^2}{2\sigma_{z,i}^2} \right)$$

where  $N$  is the number of Gaussians,  $A_i$  the amplitudes,  $x_{0,i}, y_{0,i}, z_{0,i}$  are the center positions of the peaks, and  $\sigma_{x,i}, \sigma_{y,i}, \sigma_{z,i}$  are the standard variations. Given a input of size  $(D, H, W)$ , the deformation field formed by the Gaussian mixture with size  $(D, H, W)$  is applied to the input to generate a warped image, as shown on Fig. 11.



**Fig. 11.** Example of an input image and labels (left), warped by three random Gaussian deformation fields (right).

*Data augmentation: GIN-IPA:* The other data augmentation step that we performed is inspired by [19]. It consists of Global Intensity Non-linear (GIN) transformation of the input by a convolutional network with randomly sample weights, which randomly alters the textures and intensities of the input. Next, the Hadamard products of two random GIN-augmented images and pseudo-correlation maps are added together (Interventional Pseudo-correlation Augmentation) to form an image cleared of its domain specific spurious correlations (See Fig. 12). The original algorithm proposed by [19] was extended to 3D images and implemented as a preprocessing step, such that it forms an additional “modality” of the input image.



**Fig. 12.** The GIN-IPA augmentation.

## 1.16 UNIANDES

**Team Members:** Santiago Usma\*, María Fernanda Peñuela, Luisa Vargas Daza\*,  
Maria Camila Escobar, Angela Castillo, Pablo Arbelaez\*  
\*authors included in paper

**Model architecture.** We use a model termed ROG proposed for the medical segmentation decathlon (MSD). ROG has an initial module with four convolutions, the main lattice of processing nodes, and a segmentation head with two convolutions. They organize the nodes in a triangular lattice with four scales, but unlike UNet++ it connects each node with both upper and lower-level nodes and removes the dense connections.

**GPU training was performed on.** Quadro RTX 8000

**Software used.** Pytorch 1.11.0 cuda11.3

**Number of layers.** 8 layers

**Convolution kernel size.** 3x3x3

**Initialization.** Random

**Optimizer.** Adam – weight decay 1e-5

**Cross-validation used.** Yes. We used 2-fold cross-validation

**Number of epochs.** 398 epochs for fold 0, 615 epochs for fold 1

**Number of trainable parameters.** 2596507 parameters

**Learning Rate and schedule:** We used a ReduceLROnPlateau scheduler with a learning rate of 1e-3 and a patience of 50

**Loss Function.** A combination of the Dice Loss and Cross-Entropy Loss

**Dimensionality of input/output (ie: 2D,3D, 2D+, etc.).** 3D

**Batch Size.** 2

**Preprocessing steps used (ie data normalization, creation of patches, etc.).** Clipping the intensities to the [0.5, 99.5] percentiles of the foreground values and perform z-score normalization.

**Data Augmentation steps (ie – rotation, flipping, scaling, blur, noise, etc.).** Spatial Transform (random rotation and scaling), Mirror Transform and gamma correction.

**External dataset used?** No

**Framework (ie – MONAI, nnUNet, etc.).** ROG

**Number of models trained for final submission.** 7 experiments with ROG, and 15 experiments with a 2D and 2D+ approach with Mask2Former.

**Post-Processing Steps.** We perform a closure and then an opening of grays in the segmentation map with a structuring element with 1s of dimension [4,4,2]

**Original work/Existing work.** This work was based on a previous work that proposed a single-architecture model for RObust Generic segmentation (ROG) [42] (<https://github.com/BCV-Uniandes/ROG>). We experimented by varying some parameters until we found the best performance.

**Which FeTA cases were included:** We split the data making sure to maintain similar distributions in both folds on the variables of gestational age, institution and pathology.

*Fold 0.* 60 volumes for training and 60 volumes for validation

Neurotypical: 0.53 Pathological 0.47

Zurich: 0.63, Vienna: 0.37

G1: 0.23, G2: 0.22, G3: 0.27, G4: 0.28

*Fold 1.* 60 volumes for training and 60 volumes for validation

Neurotypical: 0.53 Pathological 0.47

Zurich: 0.7, Vienna: 0.3

G1: 0.27, G2: 0.28, G3: 0.23, G4: 0.22

**Training/validation/testing data splits.** In each fold we use a 50/50 split between training and validation.

**Hyperparameter tuning performed.** No.

**Training time.** 6 hours per fold approx.

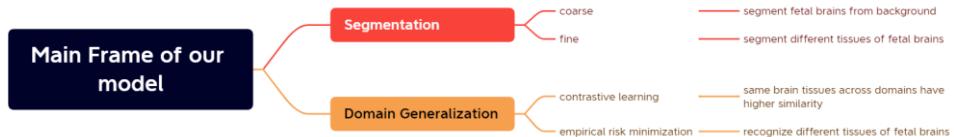
### 1.17 xinlab-scut-iai-ahu

**Team Members:** Wenying Lu\*, Wenhao Zhang\*, Jing Liang\*, JiaHui Wang, Chao Yang, Hao Mei, A/Prof. Xin Zhang, A/Prof. Qin Xu, A/Prof. Xiaofen Xing, Prof. Xiangming Xu

\*authors included in paper

GPU training was performed on NVIDIA GTX 1080ti. Software used: Pytorch 1.11.0+cu102, MONAI 0.9.0.

**Model architecture.** The constructed model is dedicated in solving problems from the following two aspects: One is to do segmentation on fetal brains, another is to generalize the segmentation model to different domains such as data collected from different institutions. The framework of our model is shown in Fig. 13.



**Fig. 13.** Model Framework

For segmentation, we apply a two-stages training method named coarse-to-fine to achieve accurate segmentation. In the coarse stage, we try to make the model learn whether a voxel belongs to a fetal brain or background. For a unique voxel, it is a binary-classification task, then before the fine stage. We multiply the output of the coarse stage and the corresponding original image to remove noise from the background. In the coarse stage, we make our efforts to teach the model to categorize an unique voxel to different brain tissues.

For domain generalization, we use a method named Domain Generalization using Causal Matching (MatchDG) [43], which aims at making the similarity of the same kind of brain tissues across domains as high as possible. There are also two stages in this method. On the one hand, we need a contrastive learning stage to give higher similarity between same kind of brain tissues, and we can acquire a match matrix from this stage according to the similarity, through this matrix, there is a base domain which contains the most number of samples, and samples from other domains are matched with the samples from the base domain. In other words, it is a many-to-one

match. On the other hand, we need not only to classify the samples from base domain accurately, but also to classify the samples from other domains in an accurate way, so while we use samples from the base domain to train the model. We also sample the samples of other domains from the match matrix to train the model to achieve a more accurate classification. The process of domain generalization is shown in Table 7.

**Table 7.** Domain Generalization

---

**Algorithm** MatchDG

---

**In:**Dataset( $d_i x_i, y_i$ ) $_{i=1}^n$  from m domains, $\tau, t$

**Out:**Function  $f: \chi \rightarrow \gamma$

Create random pairs  $\Omega_Y$

Build a p\*q data matrix  $M$

**Phase I**

**While** not converged **do**

**For** batch  $\sim M$  **do**

        Minimize contrastive loss

**End for**

**If** epoch % t == 0 **then**

        Update match pairs using  $\Phi_{epoch}$

**End if**

**End while**

**Phase II**

Compute matching based on  $\Phi$

Minimize the loss with learnt match function  $\Phi$  to obtain  $f$

---

We insert the domain generalization between the coarse stage and the fine stage to let the model in the fine stage learn anatomical features of fetal brains to perform better generalization on segmentation. More concretely speaking, we train the encoder of the fine stage in the stage of domain generalization. Furthermore, as for the input of domain generalization stage, we set the background that does not contain a concrete fetal brain tissue to 0, and one image only contains one kind of brain tissues. We achieve this process by the annotation of segmentation.

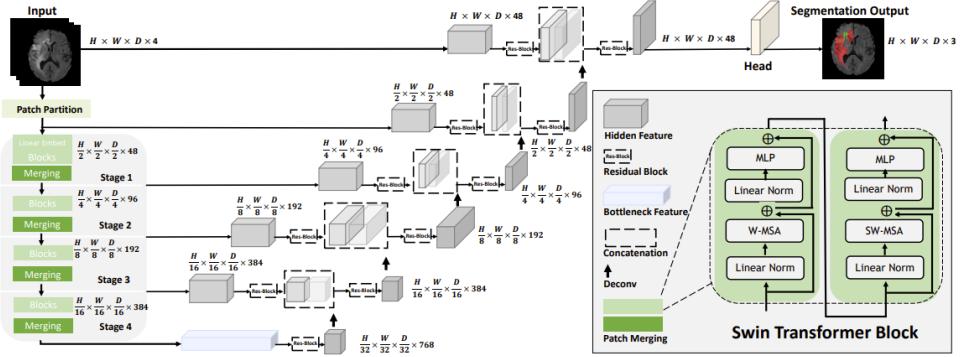
During the coarse stage, we use DynUNet [44] to achieve our primary segmentation. The input size of our model is (1,64,64,64), where 1 denotes the input channel number of the model and 64 denotes the size of the image of our input, noting that the input spatial dimensions of our model are 3. We set our convolutional kernel size as 3\*3\*3 during the down-sampling stage while the kernel size of the transposed convolution during the up-sampling is set to 2\*2\*2. As the conventional UNet architecture, our DynUNet contains four down-sampling blocks whose output channel are (128,256,512,1024) with stride 2, and the output channel of the input block is 64 before down-sampling.

During the stages of domain generalization and fine-segmentation, we apply Swin Transformer [45] as our backbone to extract abundant semantic information. The

main idea of the domain generalization stage is to train the Swin Transformer encoder to seize the domain-invariant features better.

Finally, we apply SwinUNETR [20], [21] to achieve finer segmentation, the parameters trained during the domain generalization stage will remain unchanged while other parameters will be updated during the training stage of the fine segmentation.

The architecture of SwinUNETR is shown in Fig. 14.



**Fig. 1.** Overview of the Swin UNETR architecture. The input to our model is 3D multi-modal MRI images with 4 channels. The Swin UNETR creates non-overlapping patches of the input data and uses a patch partition layer to create windows with a desired size for computing the self-attention. The encoded feature representations in the Swin transformer are fed to a CNN-decoder via skip connection at multiple resolutions. Final segmentation output consists of 3 output channels corresponding to ET,WT and TC sub-regions.

**Fig. 14.** This picture and caption are derived from [20].

The configurations of our SwinUNETR remains the same as the original paper as shown in Fig. 15.

Embed Dimension	Feature Size	Number of Blocks	Window Size	Number of Heads	Parameters	FLOPs
768	48	[2,2,2,2]	[7,7,7]	[3,6,12,24]	61.98M	394.84G

**Table 1.** Swin UNETR configurations.

**Fig. 15.** This table of configurations taken from [20]

**Data Processing.** There is no extra open-source data used for our training. The data we have used is those provided by FeTA challenge, and we use all of the data cases in order to find a result as satisfying as possible.

In order to acquire more diversity from the given data, some efforts of preprocessing and augmentation have been made (Table 8).

**Table 8.** Data Augmentation

Measures	Functionality
Orientation	Change the input image's orientation into the specified axis
Spacing	Change the resolution of the input image
CropForeground	Crop only the foreground object of the expected images
RandCropByPosNegLabel	Crop random fixed sized regions with the center being a foreground or background voxel based on the Pos Neg Ratio.
RandFlipd	Randomly flip the image along a specified axis
RandRotate90	Randomly rotate the input image by 90 degrees along a specified axis
ToTensor	Convert the data to pytorch tensor

**The process of optimization.** The settings of the optimization process in our experiment are in Table 9.

**Table 9.** Optimization Process

Stage	Optimizer	Learning Rate	Learning Rate Scheduler
Coarse stage	AdamW	1e-4	Warmup cosine
Domain generalization stage	SGD	1e-4	None
Fine stage	AdamW	1e-4	Warmup cosine

As for warmup cosine scheduler, there is a parameter called warmup steps. The formula of the learning rate is as follows:

$$\begin{aligned}
 \text{learning rate} = & \{ \text{current step} \times \frac{1}{\max(1, \text{warmup steps})}, \text{current step} \\
 & < \text{warmup steps} \max(1, 0.5 \times \\
 & \cos \cos \left( 2\omega\pi \frac{\text{current step} - \text{warmup step}}{(1, \text{epochs} - \text{warmup step})} \right)), \text{current step} \\
 & > \text{warmup steps}
 \end{aligned}$$

The number of the parameters we have trained is listed in Table 10.

**Table 10.** Number of parameters

Model	Number of Parameters
DynUNet	90285890
Swin Transformer Encoder(contrastive learning stage)	2172234
Swin Transformer Encoder(empirical risk minimization)	2172234
Swin UNETR(exclude the encoder)	54124951

The loss functions we try to optimize and epochs we have run to train the model are in Table 11.

**Table 11.** Loss Functions and Epochs

Stage	Loss Function	Epochs
Coarse segmentation	Dice loss	2000
Domain generalization(contrastive learning stage)	Contrastive loss	50
Domain generalization(empirical risk minimization )	Cross Entropy loss and 12	50
Fine segmentation	Dice loss	3000

**Training strategy.** In order to make our training process stable and to ensure the GPU can work smoothly, we set the batch size 4 among all of our stages.

The training strategy we apply to train our models is 5-fold cross validation, and our data splits are in Table 12.

**Table 12.** Training Strategy

Split	Institution 1 - Train	Institution 1 – Validation	Institution 2 - Train	Institution 2 – Validation
0	Sub001-sub064	Sub065-sub080	Sub101-sub0132	Sub133-sub140
1	Sub001- sub048,sub065-sub080	Sub049-sub064	Sub101- sub124,sub133-sub140	Sub125-sub132
2	Sub001- sub032,sub049-sub080	Sub033-sub048	Sub101- sub116,sub125-sub140	Sub117-sub124
3	Sub001- sub016,sub033-sub080	Sub017-sub032	Sub101- sub108,sub033-sub080	Sub109-sub116
4	Sub017-sub080	Sub001-sub016	Sub109-sub140	Sub101-sub108

**Implementation.** After the preparation referred above done, we need to achieve our ideas, and we use the framework MONAI to implement our experiments.

In the aspect of codes, we used the matchDG algorithm from Domain Generalization using Causal Matching and the APIs of DynUNet, SwinUNETR and some other functions developed by MONAI.

**Some measure and ideas after training the models.** We didn't do any post-processing after the training process finished.

Our work aims at guiding the model to learn domain-invariant features and no extra prior knowledge required during the input stage.

The original work of MatchDG is used to solve classification problems among multi-domains, and we use it to train an encoder that can extract proper semantic information to perform better in segmentation among multi-domains. The original paper didn't include training of a Swin Transformer or a 3D model, and we make efforts to combine all of them to create better domain-invariant model that can solve 3D imaging problems. In our design, the model of coarse stage is to remove the influence of the noise of the background, and the model trained under the guidance of the domain generalization stage to achieve finer segmentation on the basis of the coarse stage.

## 2 References

- [1] J. Chen, Y. Lu, Q. Yu, X. Luo, E. Adeli, Y. Wang, L. Lu, A. L. Yuille, and Y. Zhou, "TransUNet: Transformers Make Strong Encoders for Medical Image Segmentation." arXiv, 08-Feb-2021.
- [2] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, Cham, 2015, pp. 234–241.
- [3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is All you Need," in *Advances in Neural Information Processing Systems*, 2017, vol. 30.
- [4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [5] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale." arXiv, 03-Jun-2021.
- [6] H. Akrami, W. Cui, A. A. Joshi, and R. M. Leahy, "Learning from imperfect training data using a robust loss function: application to brain image segmentation." arXiv, 08-Aug-2022.
- [7] H. Akrami, A. A. Joshi, J. Li, S. Aydöre, and R. M. Leahy, "A robust variational autoencoder using beta divergence," *Knowledge-Based Systems*, vol. 238, p. 107886, Feb. 2022.
- [8] H. Akrami, S. Aydöre, R. M. Leahy, and A. A. Joshi, "Robust Variational Autoencoder for Tabular Data with Beta Divergence." arXiv, 15-Jun-2020.

- [9] Z. Chen, Y. Duan, W. Wang, J. He, T. Lu, J. Dai, and Y. Qiao, “Vision Transformer Adapter for Dense Predictions.” arXiv, 23-Oct-2022.
- [10] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, “Domain-adversarial training of neural networks,” *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2096–2030, Jan. 2016.
- [11] F. Isensee, P. F. Jaeger, S. A. A. Kohl, J. Petersen, and K. H. Maier-Hein, “nnUNet: a self-configuring method for deep learning-based biomedical image segmentation,” *Nat Methods*, vol. 18, no. 2, pp. 203–211, Feb. 2021.
- [12] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu, “Deeply-Supervised Nets,” in *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, 2015, pp. 562–570.
- [13] B. Billot, D. N. Greve, O. Puonti, A. Thielscher, K. Van Leemput, B. Fischl, A. V. Dalca, and J. E. Iglesias, “SynthSeg: Domain Randomisation for Segmentation of Brain Scans of any Contrast and Resolution.” arXiv, 21-Dec-2021.
- [14] F. Pérez-García, R. Sparks, and S. Ourselin, “TorchIO: A Python library for efficient loading, preprocessing, augmentation and patch-based sampling of medical images in deep learning,” *Computer Methods and Programs in Biomedicine*, vol. 208, p. 106236, Sep. 2021.
- [15] L. Gondara, “Medical Image Denoising Using Convolutional Denoising Autoencoders,” in *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, 2016, pp. 241–246.
- [16] K. He, X. Zhang, S. Ren, and J. Sun, “Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification,” in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1026–1034.
- [17] P. T. G. Jackson, A. A. Abarghouei, S. Bonner, T. P. Breckon, and B. Obara, “Style Augmentation: Data Augmentation via Style Randomization,” in *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2019, Long Beach, CA, USA, June 16-20, 2019*, 2019, pp. 83–92.
- [18] C. Chen, C. Qin, H. Qiu, C. Ouyang, S. Wang, L. Chen, G. Tarroni, W. Bai, and D. Rueckert, “Realistic Adversarial Data Augmentation for MR Image Segmentation,” in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2020*, Cham, 2020, pp. 667–677.
- [19] C. Ouyang, C. Chen, S. Li, Z. Li, C. Qin, W. Bai, and D. Rueckert, “Causality-inspired Single-source Domain Generalization for Medical Image Segmentation.” arXiv, 06-Dec-2021.
- [20] A. Hatamizadeh, V. Nath, Y. Tang, D. Yang, H. R. Roth, and D. Xu, “Swin UNETR: Swin Transformers for Semantic Segmentation of Brain Tumors in MRI Images,” in *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries: 7th International Workshop, BrainLes 2021, Held in Conjunction with MICCAI 2021, Virtual Event, September 27, 2021, Revised Selected Papers, Part I*, Berlin, Heidelberg, 2021, pp. 272–284.
- [21] Y. Tang, D. Yang, W. Li, H. R. Roth, B. Landman, D. Xu, V. Nath, and A. Hatamizadeh, “Self-Supervised Pre-Training of Swin Transformers for 3D Medical Image Analysis,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 20730–20740.

- [22] A. Hoopes, J. S. Mora, A. V. Dalca, B. Fischl, and M. Hoffmann, “SynthStrip: skull-stripping for any brain image,” *NeuroImage*, vol. 260, p. 119474, Oct. 2022.
- [23] A. Makropoulos, I. S. Gousias, C. Ledig, P. Aljabar, A. Serag, J. V. Hajnal, A. D. Edwards, S. J. Counsell, and D. Rueckert, “Automatic whole brain MRI segmentation of the developing neonatal brain,” *IEEE Trans Med Imaging*, vol. 33, no. 9, pp. 1818–1831, Sep. 2014.
- [24] O. Oktay, J. Schlemper, L. L. Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N. Y. Hammerla, B. Kainz, B. Glocker, and D. Rueckert, “Attention U-Net: Learning Where to Look for the Pancreas.” arXiv, 20-May-2018.
- [25] A. D. Edwards, D. Rueckert, S. M. Smith, S. Abo Seada, A. Alansary, J. Almalbis, J. Allsop, J. Andersson, T. Arichi, S. Arulkumaran, M. Bastiani, D. Bataille, L. Baxter, J. Bozek, E. Braithwaite, J. Brandon, O. Carney, A. Chew, D. Christiaens, R. Chung, K. Colford, L. Cordero-Grande, S. J. Counsell, H. Cullen, J. Cupitt, C. Curtis, A. Davidson, M. Deprez, L. Dillon, K. Dimitrakopoulou, R. Dimitrova, E. Duff, S. Falconer, S.-R. Farahibozorg, S. P. Fitzgibbon, J. Gao, A. Gaspar, N. Harper, S. J. Harrison, E. J. Hughes, J. Hutter, M. Jenkinson, S. Jbabdi, E. Jones, V. Karolis, V. Kyriakopoulou, G. Lenz, A. Makropoulos, S. Malik, L. Mason, F. Mortari, C. Nosarti, R. G. Nunes, C. O’Keeffe, J. O’Muircheartaigh, H. Patel, J. Passerat-Palmbach, M. Pietsch, A. N. Price, E. C. Robinson, M. A. Rutherford, A. Schuh, S. Sotiropoulos, J. Steinweg, R. P. A. G. Teixeira, T. Tenev, J.-D. Tournier, N. Tusor, A. Uus, K. Vecchiato, L. Z. J. Williams, R. Wright, J. Wurie, and J. V. Hajnal, “The Developing Human Connectome Project Neonatal Data Release,” *Frontiers in Neuroscience*, vol. 16, 2022.
- [26] L. Fidon, E. Viola, N. Mufti, A. L. David, A. Melbourne, P. Demaerel, S. Ourselin, T. Vercauteren, J. Deprest, and M. Aertsen, “A spatio-temporal atlas of the developing fetal brain with spina bifida aperta,” *Open Research Europe*, vol. 1, Oct. 2021.
- [27] S. Gao, H. Zhou, Y. Gao, and X. Zhuang, “Joint Modeling of Image and Label Statistics for Enhancing Model Generalizability of Medical Image Segmentation.” arXiv, 09-Jun-2022.
- [28] K. Payette, H. Li, P. de Dumast, R. Licandro, H. Ji, M. M. R. Siddiquee, D. Xu, A. Myronenko, H. Liu, Y. Pei, L. Wang, Y. Peng, J. Xie, H. Zhang, G. Dong, H. Fu, G. Wang, Z. Rieu, D. Kim, H. G. Kim, D. Karimi, A. Gholipour, H. R. Torres, B. Oliveira, J. L. Vilaça, Y. Lin, N. Avisdris, O. Ben-Zvi, D. B. Bashat, L. Fidon, M. Aertsen, T. Vercauteren, D. Sobotka, G. Langs, M. Alenyà, M. I. Villanueva, O. Camara, B. S. Fadida, L. Joskowicz, L. Weibin, L. Yi, L. Xuesong, M. Mazher, A. Qayyum, D. Puig, H. Kebiri, Z. Zhang, X. Xu, D. Wu, K. Liao, Y. Wu, J. Chen, Y. Xu, L. Zhao, L. Vasung, B. Menze, M. B. Cuadra, and A. Jakab, “Fetal Brain Tissue Annotation and Segmentation Challenge Results.” arXiv, 20-Apr-2022.
- [29] F. Isensee and K. H. Maier-Hein, “An attempt at beating the 3D U-Net.” arXiv, 04-Oct-2019.
- [30] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, “3D U-Net: learning dense volumetric segmentation from sparse annotation,” in *Internation-*

- tional conference on medical image computing and computer-assisted intervention*, 2016, pp. 424–432.
- [31] G. van Rossum (Guido), “Python reference manual,” *Department of Computer Science [CS]*, no. R 9525. CWI, 01-Jan-1995.
  - [32] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “PyTorch: An Imperative Style, High-Performance Deep Learning Library,” in *Advances in Neural Information Processing Systems*, 2019, vol. 32.
  - [33] MONAI Consortium, “MONAI: Medical Open Network for AI.” Mar-2020.
  - [34] E. Riba, D. Mishkin, D. Ponsa, E. Rublee, and G. Bradski, “Kornia: an Open Source Differentiable Computer Vision Library for PyTorch,” in *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2020, pp. 3663–3672.
  - [35] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization.” arXiv, 29-Jan-2017.
  - [36] A. Myronenko, “3D MRI Brain Tumor Segmentation Using Autoencoder Regularization,” in *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries*, Cham, 2019, pp. 311–320.
  - [37] Z. Zhou, M. M. Rahman Siddiquee, N. Tajbakhsh, and J. Liang, “UNet++: A Nested U-Net Architecture for Medical Image Segmentation,” in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, Cham, 2018, pp. 3–11.
  - [38] Z. Zhou, M. M. Rahman Siddiquee, N. Tajbakhsh, and J. Liang, “UNet++: Redesigning Skip Connections to Exploit Multiscale Features in Image Segmentation,” *IEEE Trans Med Imaging*, vol. 39, no. 6, pp. 1856–1867, Jun. 2020.
  - [39] K. He, X. Zhang, S. Ren, and J. Sun, “Identity Mappings in Deep Residual Networks,” in *Computer Vision – ECCV 2016*, Cham, 2016, pp. 630–645.
  - [40] K. Payette, P. de Dumast, H. Kebiri, I. Ezhov, J. C. Paetzold, S. Shit, A. Iqbal, R. Khan, R. Kottke, P. Grehten, H. Ji, L. Lanczi, M. Nagy, M. Beresova, T. D. Nguyen, G. Natalucci, T. Karayannis, B. Menze, M. Bach Cuadra, and A. Jakab, “An automatic multi-tissue human fetal brain segmentation benchmark using the Fetal Tissue Annotation Dataset,” *Sci Data*, vol. 8, no. 1, p. 167, Jul. 2021.
  - [41] W. R. Crum, O. Camara, and D. L. G. Hill, “Generalized overlap measures for evaluation and validation in medical image analysis,” *IEEE Trans Med Imaging*, vol. 25, no. 11, pp. 1451–1461, Nov. 2006.
  - [42] L. Daza, J. C. Pérez, and P. Arbeláez, “Towards Robust General Medical Image Segmentation,” in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part III*, Berlin, Heidelberg, 2021, pp. 3–13.
  - [43] D. Mahajan, S. Tople, and A. Sharma, “Domain Generalization using Causal Matching,” in *Proceedings of the 38th International Conference on Machine Learning*, 2021, pp. 7313–7324.

- [44] M. Futrega, A. Milesi, M. Marcinkiewicz, and P. Ribalta, “Optimized U-Net for Brain Tumor Segmentation.” arXiv, 24-Dec-2021.
- [45] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, “Swin Transformer: Hierarchical Vision Transformer using Shifted Windows,” in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 9992–10002.

### 3 Benchmarking report for multiTaskChallengeDice\_global

created by challengeR v1.0.2  
15 September, 2022

This document presents a systematic report on the benchmark study “multiTaskChallengeDice\_global”. Input data comprises raw metric values for all algorithms and cases. Generated plots are:

- Visualization of assessment data: Dot- and boxplot, podium plot and ranking heatmap
- Visualization of ranking stability: Blob plot, violin plot and significance map, line plot

Details can be found in Wiesenfarth et al. (2021).

#### 3.1 Ranking

Algorithms within a task are ranked according to the following ranking scheme:  
aggregate using function (“mean”) then rank

The analysis is based on 17 algorithms and 1120 cases. 0 missing cases have been found in the data set.

Ranking:

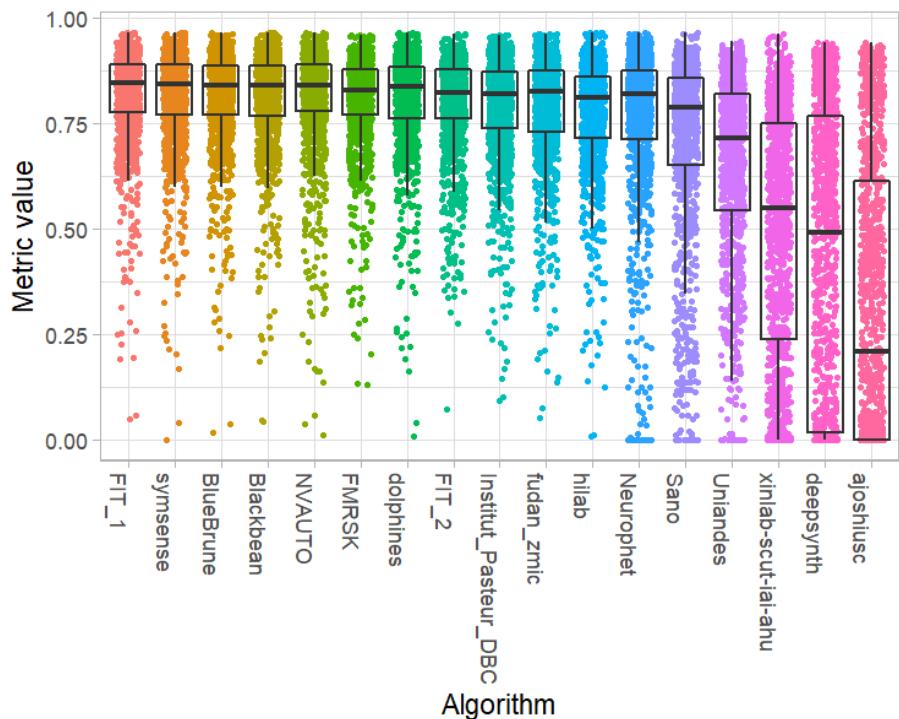
	Dice_mean	rank
FIT_1	0.8160333	1
symsense	0.8133240	2
BlueBrune	0.8120447	3
Blackbean	0.8120351	4
NVAUTO	0.8101497	5
FMRSK	0.8082458	6
dolphines	0.8059784	7
FIT_2	0.7980703	8
Institut_Pasteur_DBC	0.7886576	9

fudan_zmic	0.7881991	10
hilab	0.7735566	11
Neurophet	0.7393895	12
Sano	0.7094107	13
Uniandes	0.6521017	14
xinlab-scut-iai-ahu	0.4940798	15
deepsynth	0.4334837	16
ajoshiusc	0.3185059	17

### 3.2 Visualization of raw assessment data

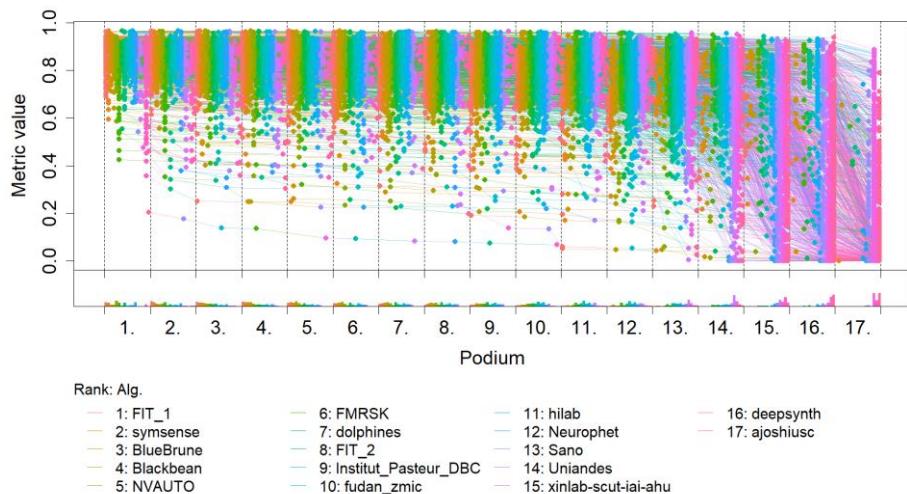
#### Dot- and boxplot

*Dot- and boxplots* for visualizing raw assessment data separately for each algorithm. Boxplots representing descriptive statistics over all cases (median, quartiles and outliers) are combined with horizontally jittered dots representing individual cases.



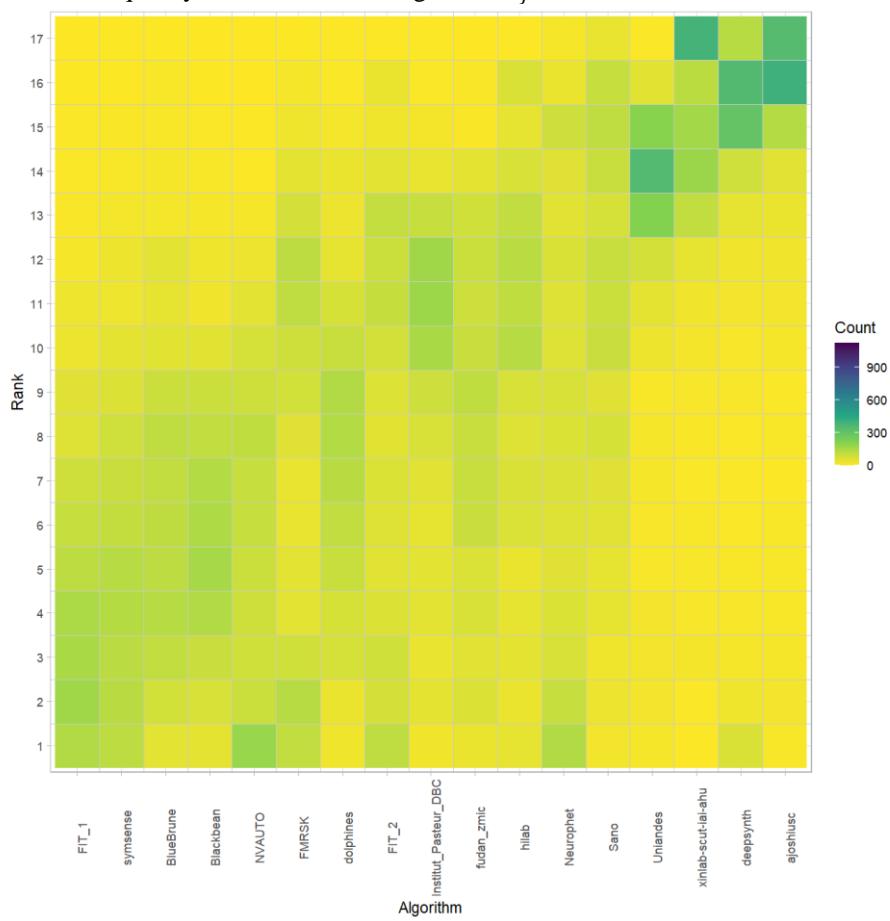
### Podium plot

*Podium plots* (see also Eugster et al., 2008) for visualizing raw assessment data. Upper part (spaghetti plot): Participating algorithms are color-coded, and each colored dot in the plot represents a metric value achieved with the respective algorithm. The actual metric value is encoded by the y-axis. Each podium (here:  $p=17$ ) represents one possible rank, ordered from best (1) to last (here: 17). The assignment of metric values (i.e. colored dots) to one of the podiums is based on the rank that the respective algorithm achieved on the corresponding case. Note that the plot part above each podium place is further subdivided into  $p$  “columns”, where each column represents one participating algorithm (here:  $p = 17$ ). Dots corresponding to identical cases are connected by a line, leading to the shown spaghetti structure. Lower part: Bar charts represent the relative frequency for each algorithm to achieve the rank encoded by the podium place.



## Ranking heatmap

**Ranking heatmaps** *Ranking heatmaps* for visualizing raw assessment data. Each cell  $(i, A_j)$  shows the absolute frequency of cases in which algorithm  $A_j$  achieved rank  $i$ .

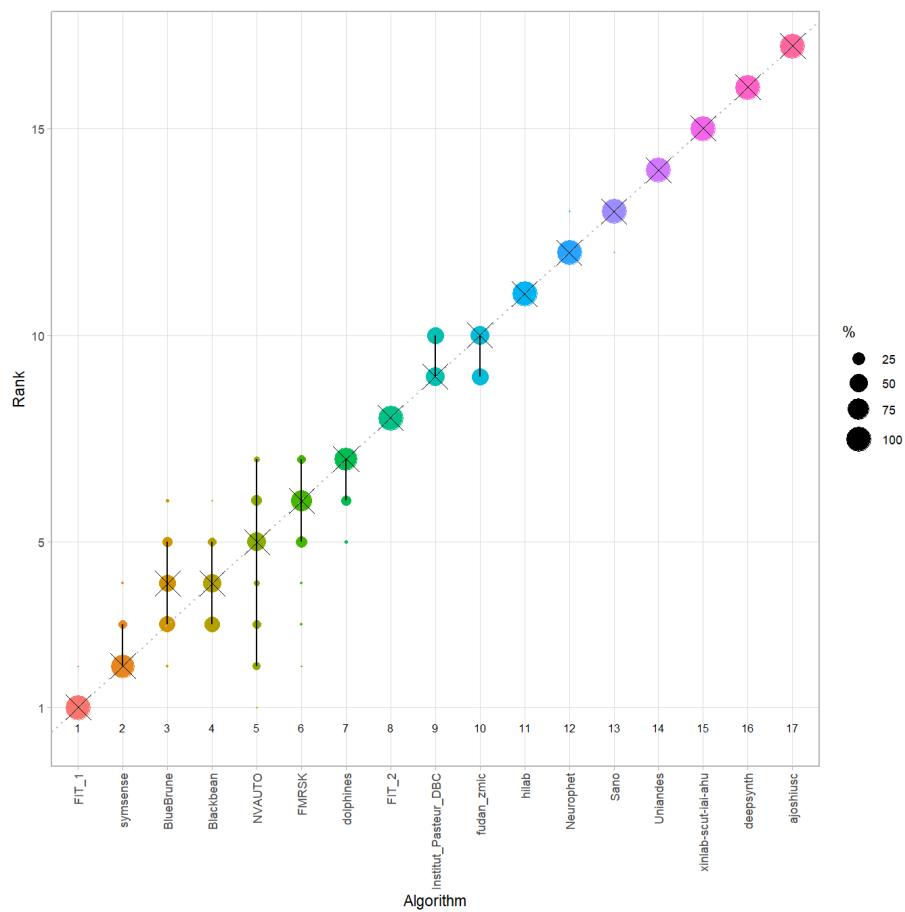


### Visualization of ranking stability

*Blob plot* for visualizing ranking stability based on bootstrap sampling

Algorithms are color-coded, and the area of each blob at position  $(A_i, \text{rank } j)$  is proportional to the relative frequency  $A_i$  achieved rank  $j$  across  $b = 1000$  bootstrap samples. The median rank for each algorithm is indicated by a black cross. 95% bootstrap intervals across bootstrap samples are indicated by black lines.

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.
```

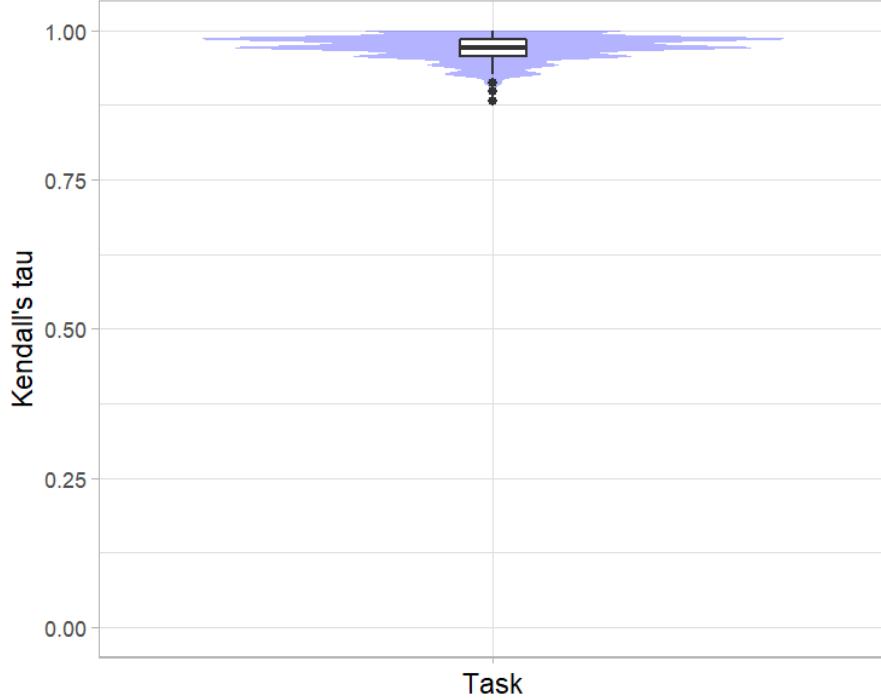


### **Violin plot for visualizing ranking stability based on bootstrapping**

The ranking list based on the full assessment data is pairwise compared with the ranking lists based on the individual bootstrap samples (here  $b = 1000$  samples). For each pair of rankings, Kendall's  $\tau$  correlation is computed. Kendall's  $\tau$  is a scaled index determining the correlation between the lists. It is computed by evaluating the number of pairwise concordances and discordances between ranking lists and produces values between  $-1$  (for inverted order) and  $1$  (for identical order). A violin plot, which simultaneously depicts a boxplot and a density plot, is generated from the results.

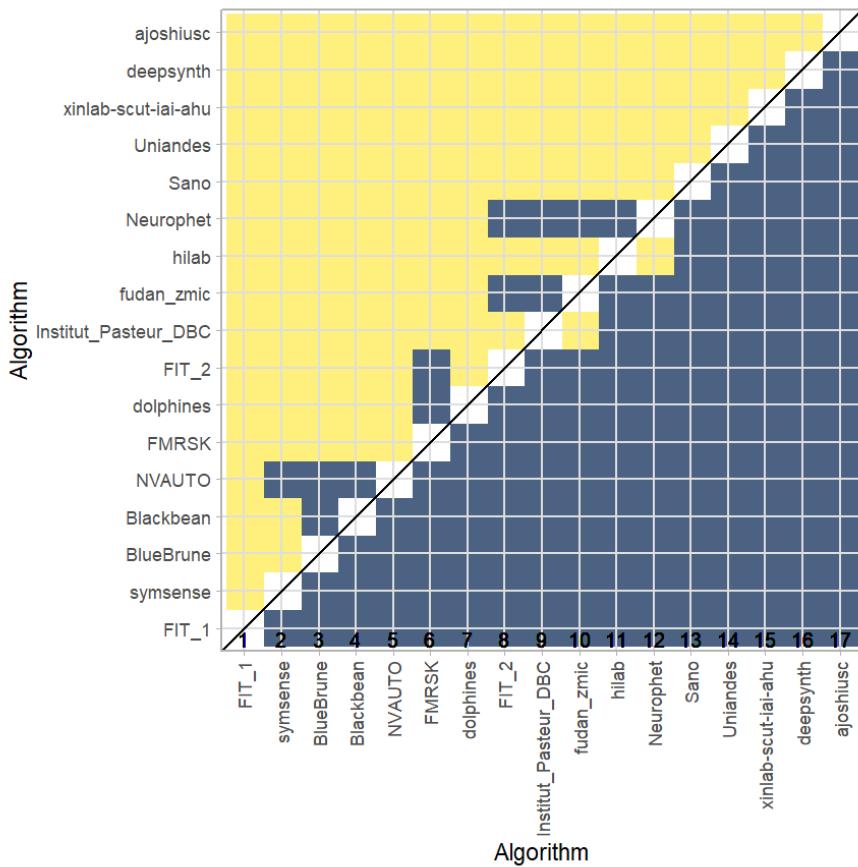
Summary Kendall's tau:

Task	mean	median	q25	q75
dummyTask	0.972	0.9705882	0.9558824	0.9852941



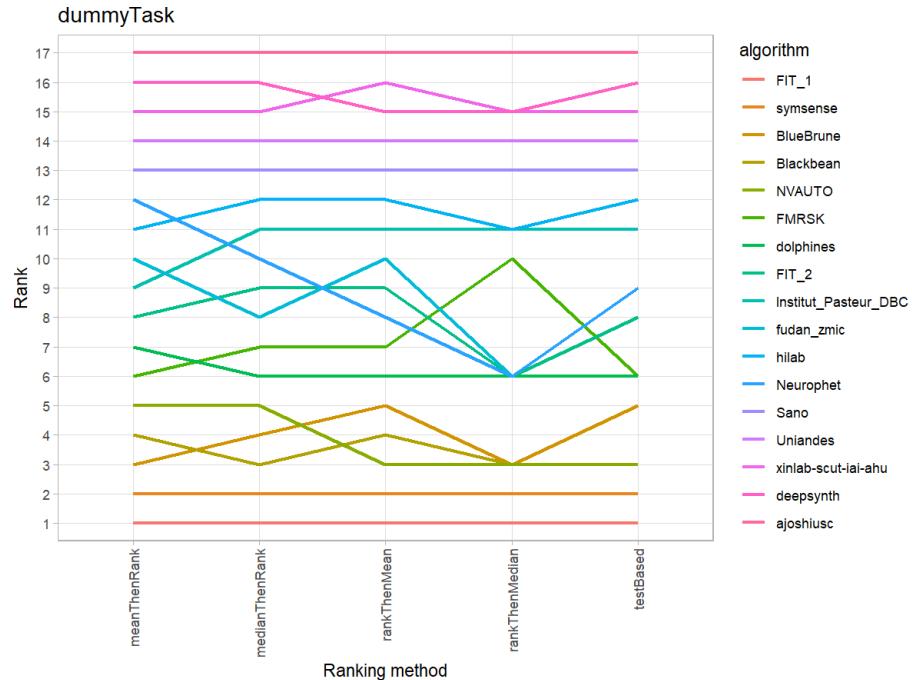
**Significance maps for visualizing ranking stability based on statistical significance**

*Significance maps* depict incidence matrices of pairwise significant test results for the one-sided Wilcoxon signed rank test at a 5% significance level with adjustment for multiple testing according to Holm. Yellow shading indicates that metric values from the algorithm on the x-axis were significantly superior to those from the algorithm on the y-axis, blue color indicates no significant difference.



### Ranking robustness to ranking methods

*Line plots* for visualizing ranking robustness across different ranking methods. Each algorithm is represented by one colored line. For each ranking method encoded on the x-axis, the height of the line represents the corresponding rank. Horizontal lines indicate identical ranks for all methods.



### 3.3 References

Wiesenfarth, M., Reinke, A., Landman, B.A., Eisenmann, M., Aguilera Saiz, L., Cardoso, M.J., Maier-Hein, L. and Kopp-Schneider, A. Methods and open-source toolkit for analyzing and visualizing challenge results. *Sci Rep* **11**, 2369 (2021). <https://doi.org/10.1038/s41598-021-82017-6>

M. J. A. Eugster, T. Hothorn, and F. Leisch, “Exploratory and inferential analysis of benchmark experiments,” Institut fuer Statistik, Ludwig-Maximilians-Universitaet Muenchen, Germany, Technical Report 30, 2008. [Online]. Available: <http://epub.ub.uni-muenchen.de/4134/>.

## 4 Benchmarking report for multiTaskChallengeHD\_global

created by challengeR v1.0.2  
15 September, 2022

This document presents a systematic report on the benchmark study “multiTask-ChallengeHD\_global”. Input data comprises raw metric values for all algorithms and cases. Generated plots are:

- Visualization of assessment data: Dot- and boxplot, podium plot and ranking heatmap
- Visualization of ranking stability: Blob plot, violin plot and significance map, line plot

Details can be found in Wiesenfarth et al. (2021).

### 4.1 Ranking

Algorithms within a task are ranked according to the following ranking scheme:  
*aggregate using function (“mean”) then rank*

The analysis is based on 17 algorithms and 1120 cases. 0 missing cases have been found in the data set.

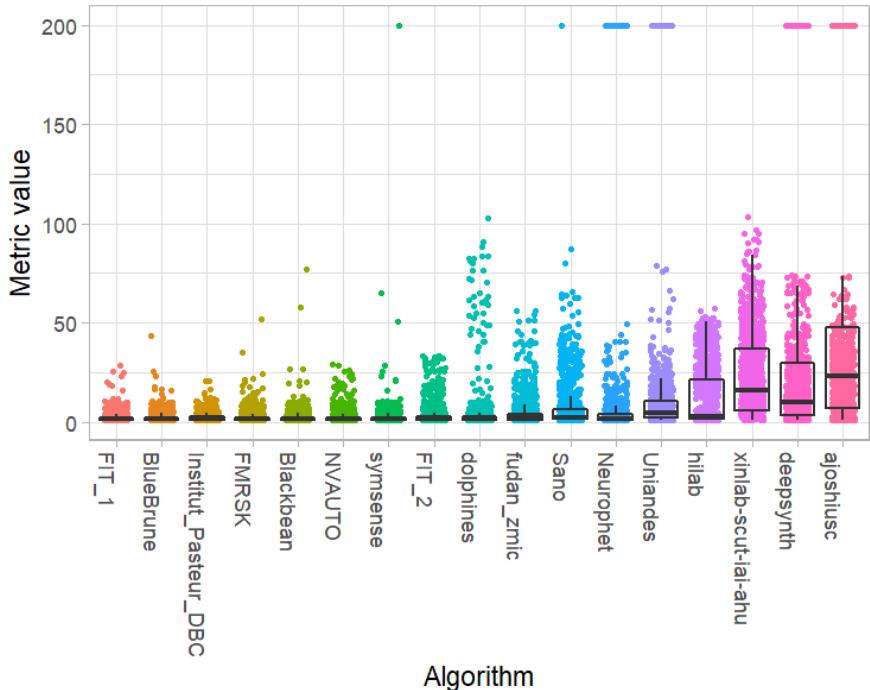
Ranking:

	Hausdorff_mean	rank
FIT_1	2.346567	1
BlueBrune	2.377450	2
Institut_Pasteur_DBC	2.386752	3
FMRSK	2.394580	4
Blackbean	2.506427	5
NVAUTO	2.607866	6
symsense	2.660439	7
FIT_2	3.420916	8
dolphines	4.520666	9
fudan_zmic	4.720102	10
Sano	7.171492	11
Neurophet	10.287913	12
Uniandes	11.366165	13
hilab	13.007935	14
xinlab-scut-iai-ahu	23.149739	15
deepsynth	36.652885	16
ajoshiusc	56.597947	17

## 4.2 Visualization of raw assessment data

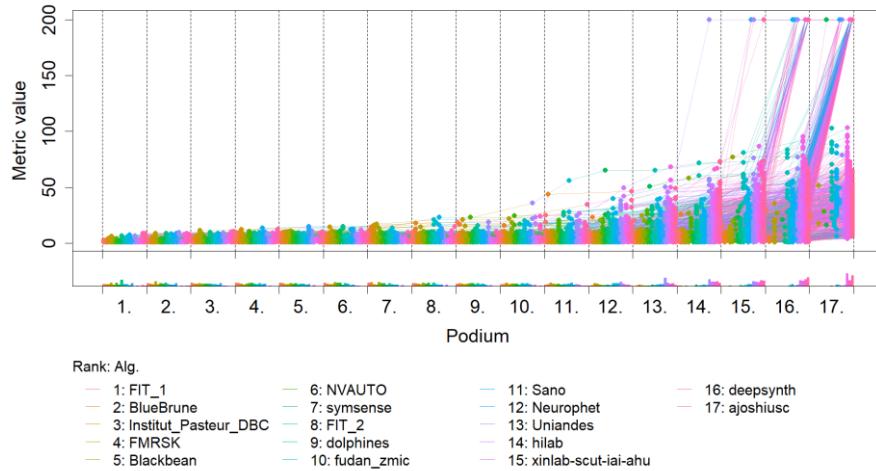
### Dot- and boxplot

*Dot- and boxplots* for visualizing raw assessment data separately for each algorithm. Boxplots representing descriptive statistics over all cases (median, quartiles and outliers) are combined with horizontally jittered dots representing individual cases.



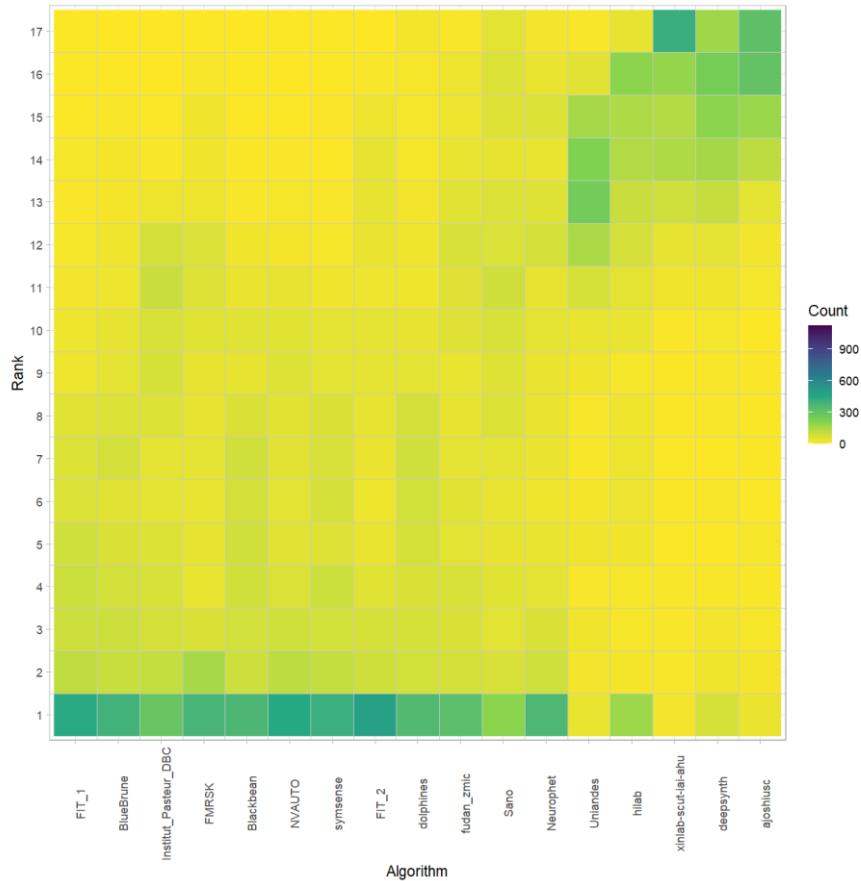
### Podium plot

*Podium plots* (see also Eugster et al., 2008) for visualizing raw assessment data. Upper part (spaghetti plot): Participating algorithms are color-coded, and each colored dot in the plot represents a metric value achieved with the respective algorithm. The actual metric value is encoded by the y-axis. Each podium (here:  $p=17$ ) represents one possible rank, ordered from best (1) to last (here: 17). The assignment of metric values (i.e. colored dots) to one of the podiums is based on the rank that the respective algorithm achieved on the corresponding case. Note that the plot part above each podium place is further subdivided into  $p$  “columns”, where each column represents one participating algorithm (here:  $p = 17$ ). Dots corresponding to identical cases are connected by a line, leading to the shown spaghetti structure. Lower part: Bar charts represent the relative frequency for each algorithm to achieve the rank encoded by the podium place.



## Ranking heatmap

*Ranking heatmaps* for visualizing raw assessment data. Each cell  $(i, A_j)$  shows the absolute frequency of cases in which algorithm  $A_j$  achieved rank  $i$ .

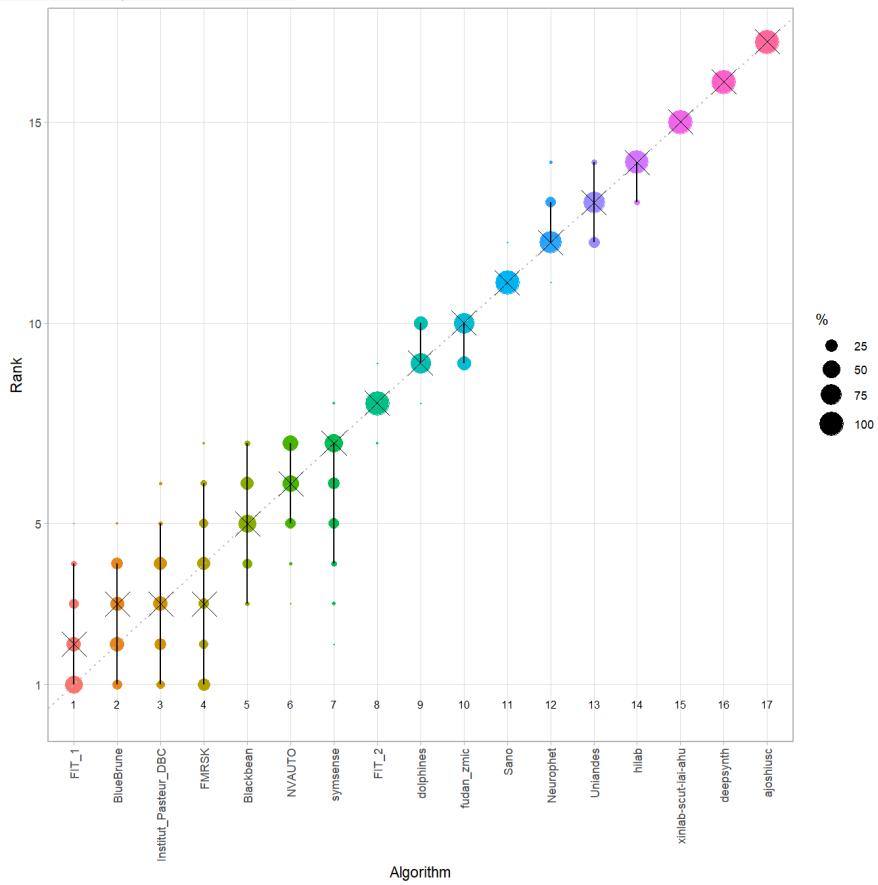


### Visualization of ranking stability

*Blob plot* for visualizing ranking stability based on bootstrap sampling

Algorithms are color-coded, and the area of each blob at position ( $A_i, \text{rank } j$ ) is proportional to the relative frequency  $A_i$  achieved rank  $j$  across  $b = 1000$  bootstrap samples. The median rank for each algorithm is indicated by a black cross. 95% bootstrap intervals across bootstrap samples are indicated by black lines.

```
## Warning: `guides(<scale> = FALSE)` is deprecated.
Please           use           `guides(<scale> =
## "none")` instead.
```

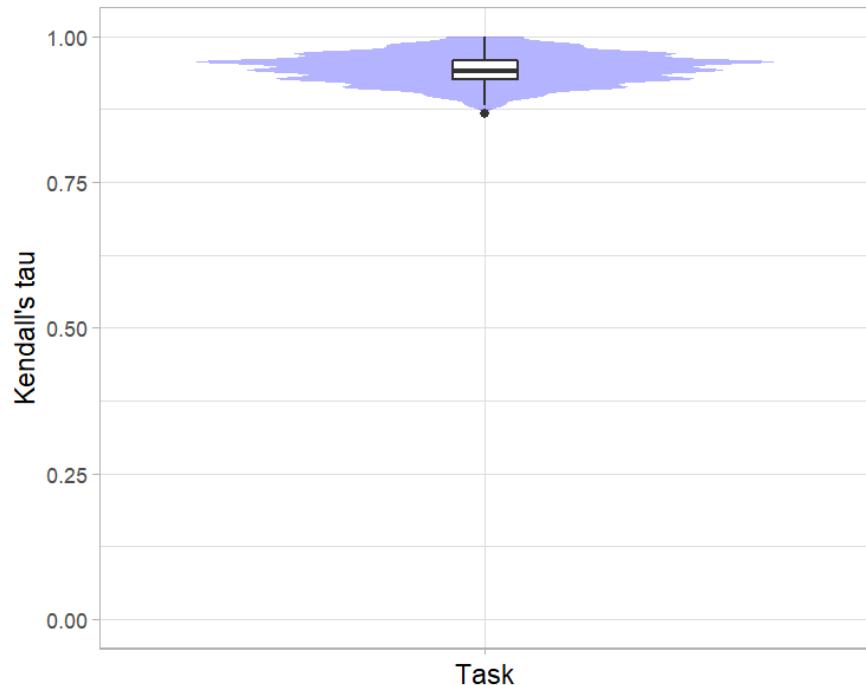


### ***Violin plot for visualizing ranking stability based on bootstrapping***

The ranking list based on the full assessment data is pairwise compared with the ranking lists based on the individual bootstrap samples (here  $b = 1000$  samples). For each pair of rankings, Kendall's  $\tau$  correlation is computed. Kendall's  $\tau$  is a scaled index determining the correlation between the lists. It is computed by evaluating the number of pairwise concordances and discordances between ranking lists and produces values between  $-1$  (for inverted order) and  $1$  (for identical order). A violin plot, which simultaneously depicts a boxplot and a density plot, is generated from the results.

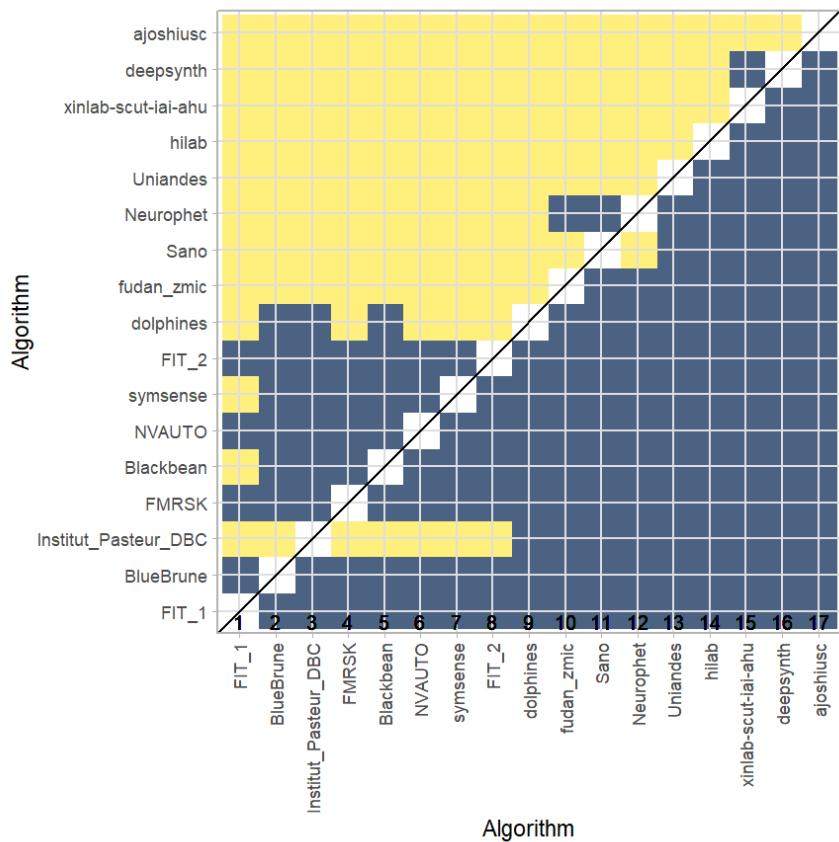
Summary Kendall's tau:

Task	mean	median	q25	q75
dummy-Task	0.9451471	0.9411765	0.9264706	0.9595588



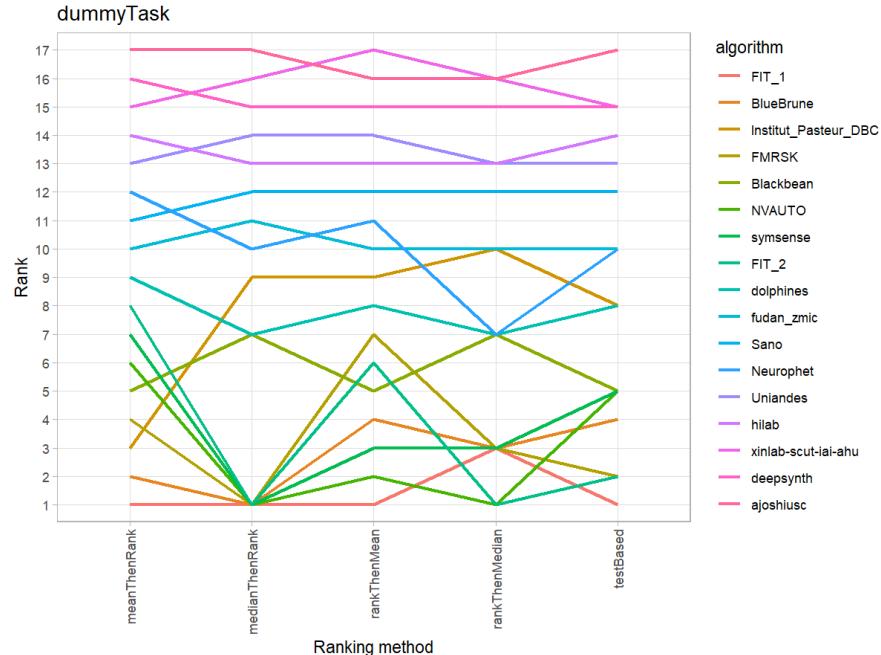
## *Significance maps* for visualizing ranking stability based on statistical significance

*Significance maps* depict incidence matrices of pairwise significant test results for the one-sided Wilcoxon signed rank test at a 5% significance level with adjustment for multiple testing according to Holm. Yellow shading indicates that metric values from the algorithm on the x-axis were significantly superior to those from the algorithm on the y-axis, blue color indicates no significant difference.



### Ranking robustness to ranking methods

*Line plots* for visualizing ranking robustness across different ranking methods. Each algorithm is represented by one colored line. For each ranking method encoded on the x-axis, the height of the line represents the corresponding rank. Horizontal lines indicate identical ranks for all methods.



### 4.3 References

Wiesenfarth, M., Reinke, A., Landman, B.A., Eisenmann, M., Aguilera Saiz, L., Cardoso, M.J., Maier-Hein, L. and Kopp-Schneider, A. Methods and open-source toolkit for analyzing and visualizing challenge results. *Sci Rep* **11**, 2369 (2021). <https://doi.org/10.1038/s41598-021-82017-6>

M. J. A. Eugster, T. Hothorn, and F. Leisch, “Exploratory and inferential analysis of benchmark experiments,” Institut fuer Statistik, Ludwig-Maximilians-Universitaet Muenchen, Germany, Technical Report 30, 2008. [Online]. Available: <http://epub.ub.uni-muenchen.de/4134/>.

## 5 Benchmarking report for multiTaskChallengeVolSim\_global

created by challengeR v1.0.2  
15 September, 2022

This document presents a systematic report on the benchmark study “multiTaskChallengeVolSim\_global”. Input data comprises raw metric values for all algorithms and cases. Generated plots are:

- Visualization of assessment data: Dot- and boxplot, podium plot and ranking heatmap
- Visualization of ranking stability: Blob plot, violin plot and significance map, line plot

Details can be found in Wiesenfarth et al. (2021).

### 5.1 Ranking

Algorithms within a task are ranked according to the following ranking scheme:

*aggregate using function (“mean”) then rank*

The analysis is based on 17 algorithms and 1120 cases. 0 missing cases have been found in the data set.

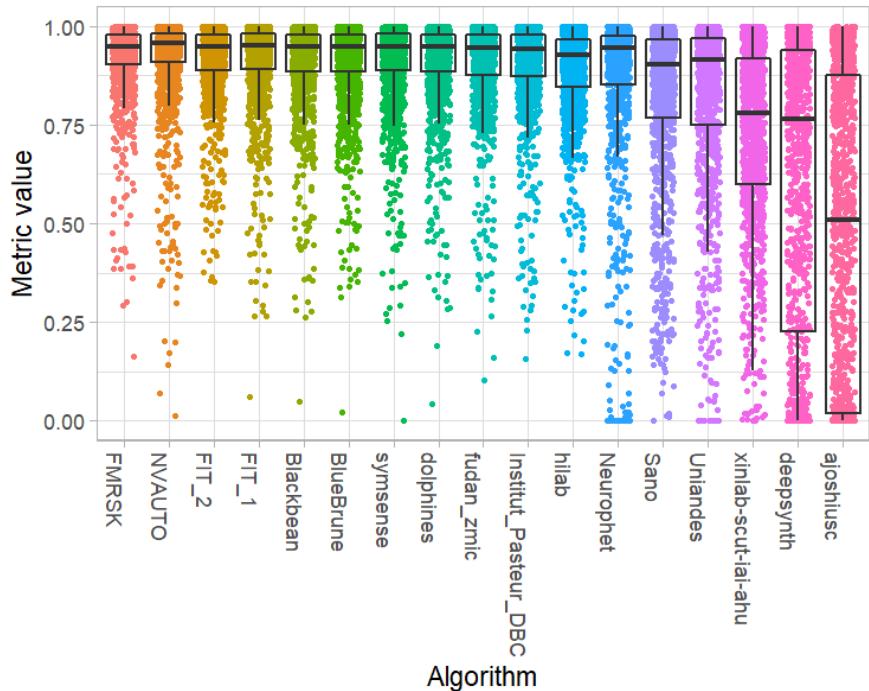
Ranking:

	Volume_Similarity_mean	rank
FMRSK	0.9196037	1
NVAUTO	0.9151729	2
FIT_2	0.9132729	3
FIT_1	0.9100900	4
Blackbean	0.9085861	5
BlueBrune	0.9080167	6
symsense	0.9071442	7
dolphines	0.9051963	8
fudan_zmic	0.9029493	9
Insti-tut_Pasteur_DBC	0.9012806	10
hilab	0.8865124	11
Neurophet	0.8435172	12
Sano	0.8173689	13
Uniandes	0.8137897	14
xinlab-scut-iai-ahu	0.7308521	15
deepsynth	0.6041642	16
ajoshiusc	0.4797518	17

## 5.2 Visualization of raw assessment data

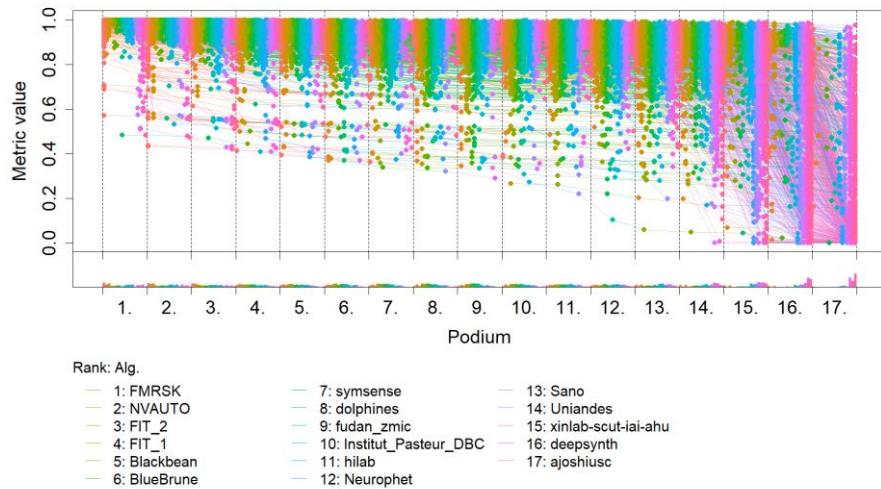
### Dot- and boxplot

*Dot- and boxplots* for visualizing raw assessment data separately for each algorithm. Boxplots representing descriptive statistics over all cases (median, quartiles and outliers) are combined with horizontally jittered dots representing individual cases.



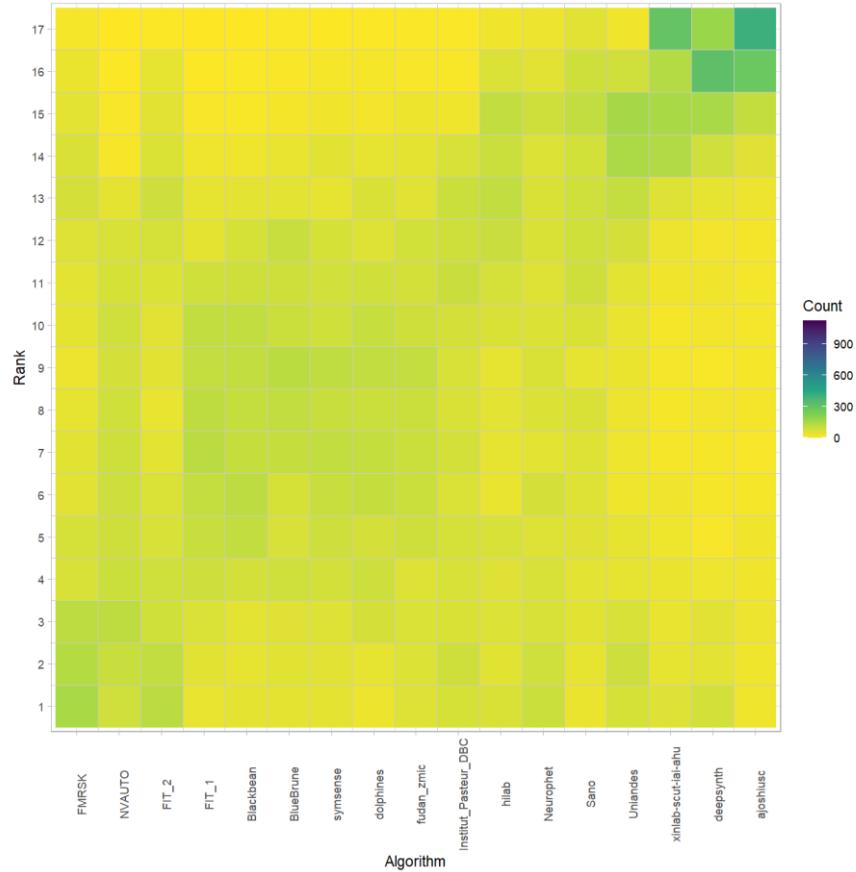
### Podium plot

*Podium plots* (see also Eugster et al., 2008) for visualizing raw assessment data. Upper part (spaghetti plot): Participating algorithms are color-coded, and each colored dot in the plot represents a metric value achieved with the respective algorithm. The actual metric value is encoded by the y-axis. Each podium (here:  $p=17$ ) represents one possible rank, ordered from best (1) to last (here: 17). The assignment of metric values (i.e. colored dots) to one of the podiums is based on the rank that the respective algorithm achieved on the corresponding case. Note that the plot part above each podium place is further subdivided into  $p$  “columns”, where each column represents one participating algorithm (here:  $p = 17$ ). Dots corresponding to identical cases are connected by a line, leading to the shown spaghetti structure. Lower part: Bar charts represent the relative frequency for each algorithm to achieve the rank encoded by the podium place.



### Ranking heatmap

*Ranking heatmaps* for visualizing raw assessment data. Each cell  $(i, A_j)$  shows the absolute frequency of cases in which algorithm  $A_j$  achieved rank  $i$ .

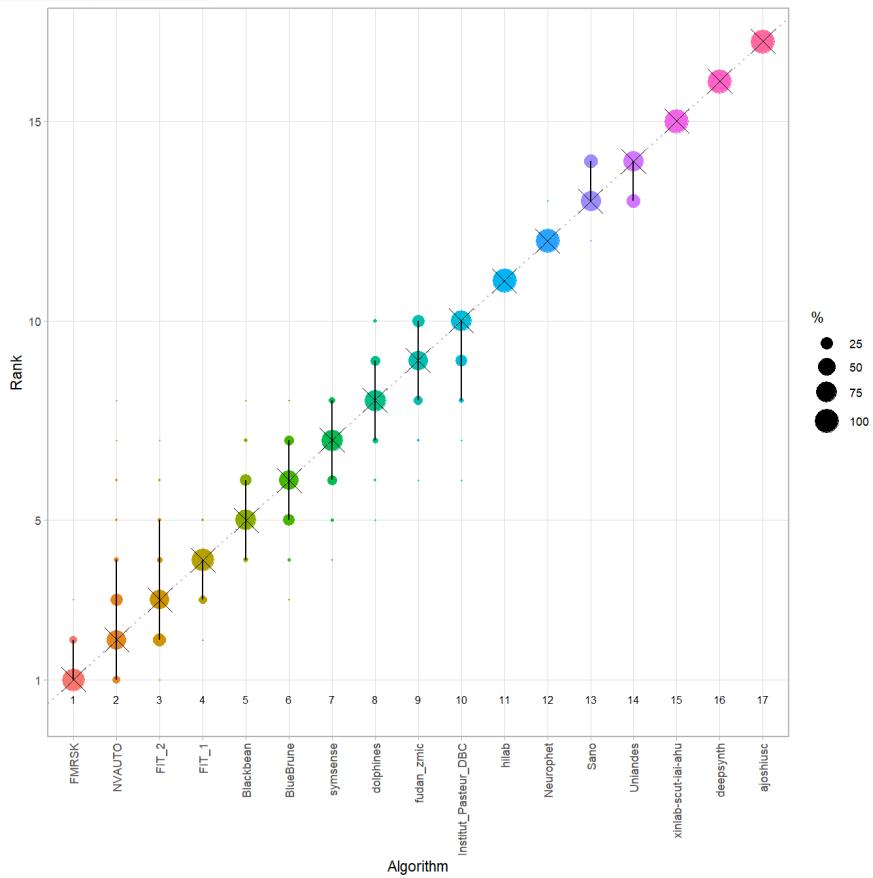


### Visualization of ranking stability

*Blob plot* for visualizing ranking stability based on bootstrap sampling

Algorithms are color-coded, and the area of each blob at position ( $A_i$ , rank  $j$ ) is proportional to the relative frequency  $A_i$  achieved rank  $j$  across  $b = 1000$  bootstrap samples. The median rank for each algorithm is indicated by a black cross. 95% bootstrap intervals across bootstrap samples are indicated by black lines.

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use
`guides(<scale> =
## "none")` instead.
```

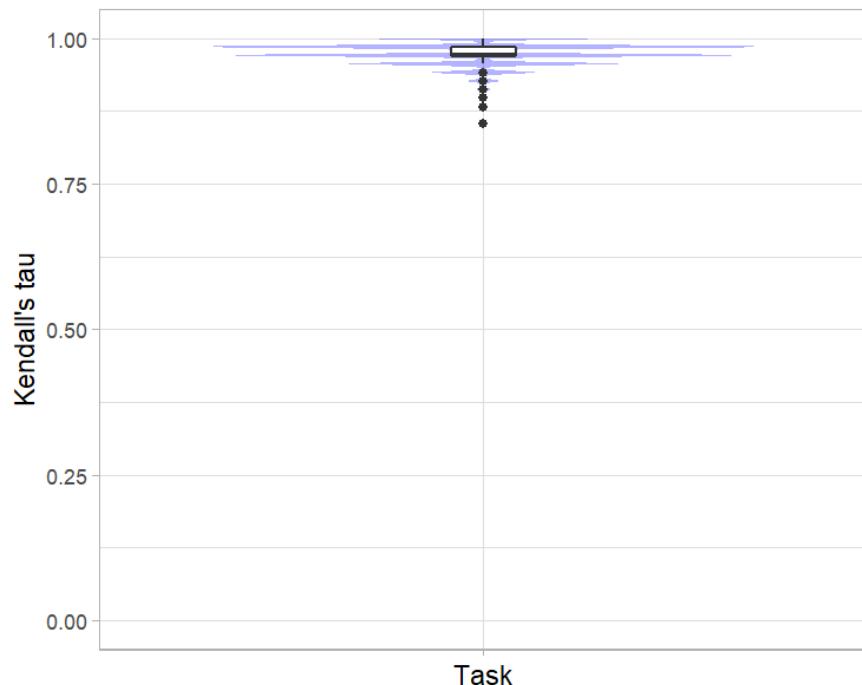


### ***Violin plot for visualizing ranking stability based on bootstrapping***

The ranking list based on the full assessment data is pairwise compared with the ranking lists based on the individual bootstrap samples (here  $b = 1000$  samples). For each pair of rankings, Kendall's  $\tau$  correlation is computed. Kendall's  $\tau$  is a scaled index determining the correlation between the lists. It is computed by evaluating the number of pairwise concordances and discordances between ranking lists and produces values between  $-1$  (for inverted order) and  $1$  (for identical order). A violin plot, which simultaneously depicts a boxplot and a density plot, is generated from the results.

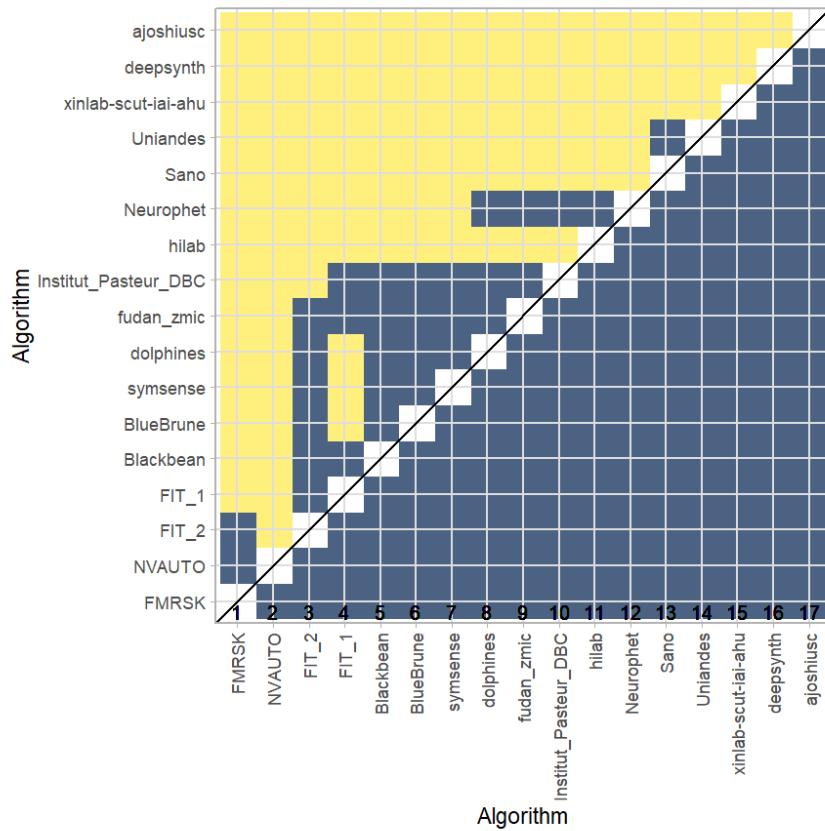
Summary Kendall's tau:

Task	mean	median	q25	q75
dummy-Task	0.9739853	0.9705882	0.9705882	0.9852941



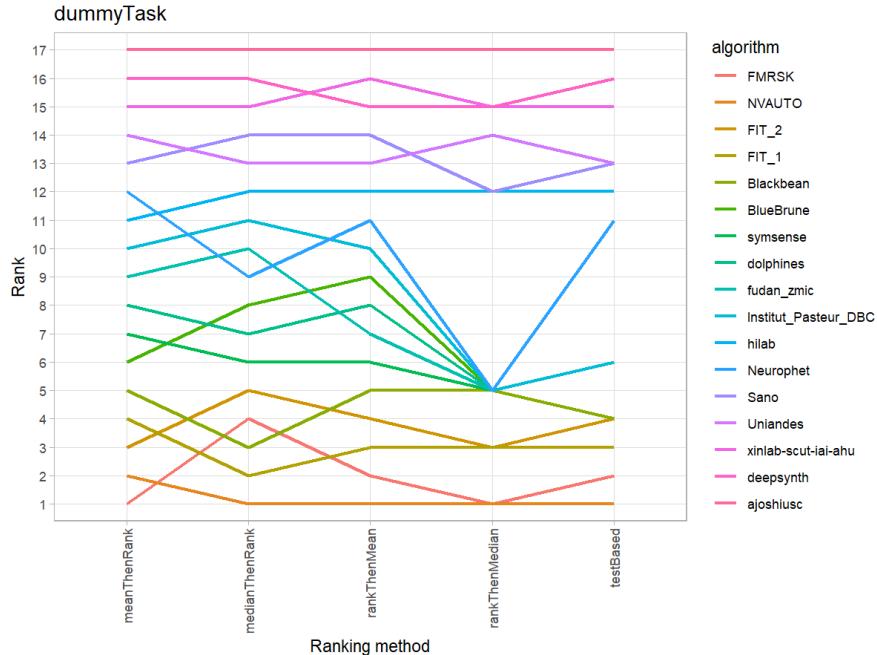
## *Significance maps* for visualizing ranking stability based on statistical significance

*Significance maps* depict incidence matrices of pairwise significant test results for the one-sided Wilcoxon signed rank test at a 5% significance level with adjustment for multiple testing according to Holm. Yellow shading indicates that metric values from the algorithm on the x-axis were significantly superior to those from the algorithm on the y-axis, blue color indicates no significant difference.



### Ranking robustness to ranking methods

*Line plots* for visualizing ranking robustness across different ranking methods. Each algorithm is represented by one colored line. For each ranking method encoded on the x-axis, the height of the line represents the corresponding rank. Horizontal lines indicate identical ranks for all methods.



### 5.3 References

Wiesenfarth, M., Reinke, A., Landman, B.A., Eisenmann, M., Aguilera Saiz, L., Cardoso, M.J., Maier-Hein, L. and Kopp-Schneider, A. Methods and open-source toolkit for analyzing and visualizing challenge results. *Sci Rep* **11**, 2369 (2021). <https://doi.org/10.1038/s41598-021-82017-6>

M. J. A. Eugster, T. Hothorn, and F. Leisch, “Exploratory and inferential analysis of benchmark experiments,” Institut fuer Statistik, Ludwig-Maximilians-Universitaet Muenchen, Germany, Technical Report 30, 2008. [Online]. Available: <http://epub.ub.uni-muenchen.de/4134/>.

## 6 Benchmarking report for Dice Metrics – In Domain

created by challengeR v1.0.2  
07 July, 2023

This document presents a systematic report on the benchmark study “Dice Metrics – In Domain”. Input data comprises raw metric values for all algorithms and cases. Generated plots are:

- Visualization of assessment data: Dot- and boxplot, podium plot and ranking heatmap
- Visualization of ranking stability: Blob plot, violin plot and significance map, line plot

Details can be found in Wiesenfarth et al. (2021).

Ranking

Algorithms within a task are ranked according to the following ranking scheme:

*aggregate using function (“mean”) then rank*

The analysis is based on 17 algorithms and 560 cases. 0 missing cases have been found in the data set.

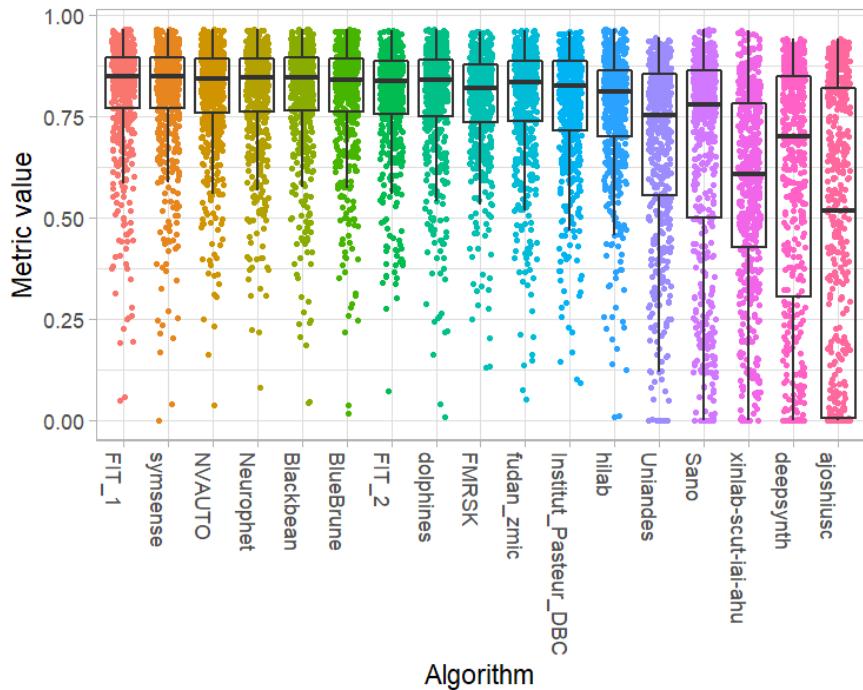
### 6.1 Ranking:

	Dice_mean	rank
FIT_1	0.8052332	1
symsense	0.8046716	2
NVAUTO	0.8041635	3
Neurophet	0.8035026	4
Blackbean	0.8029271	5
BlueBrune	0.8015985	6
FIT_2	0.7965051	7
dolphins	0.7960548	8
FMRSK	0.7888039	9
fudan_zmic	0.7879124	10
Insti-tut_Pasteur_DBC	0.7798630	11
hilab	0.7655387	12
Uniandes	0.6835894	13
Sano	0.6578728	14
xinlab-scut-iai-ahu	0.5781840	15
deepsynth	0.5768365	16
ajoshiusc	0.4553186	17

## 6.2 Visualization of raw assessment data

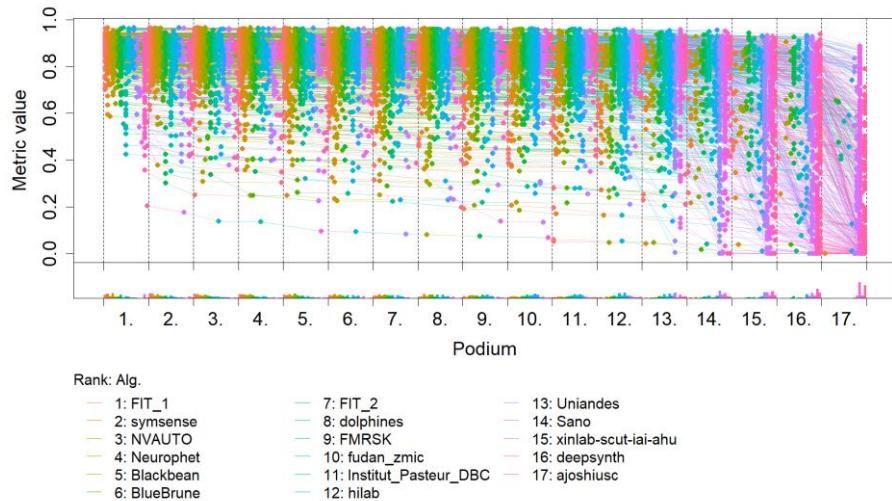
### Dot- and boxplot

*Dot- and boxplots* for visualizing raw assessment data separately for each algorithm. Boxplots representing descriptive statistics over all cases (median, quartiles and outliers) are combined with horizontally jittered dots representing individual cases.



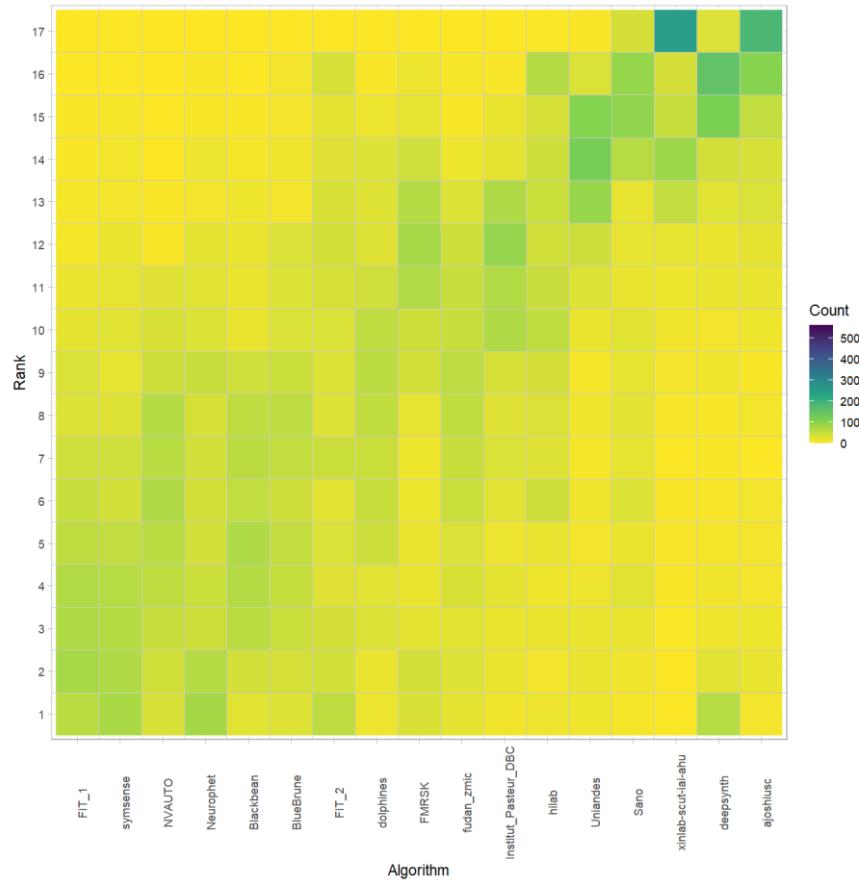
### Podium plot

*Podium plots* (see also Eugster et al., 2008) for visualizing raw assessment data. Upper part (spaghetti plot): Participating algorithms are color-coded, and each colored dot in the plot represents a metric value achieved with the respective algorithm. The actual metric value is encoded by the y-axis. Each podium (here:  $p=17$ ) represents one possible rank, ordered from best (1) to last (here: 17). The assignment of metric values (i.e. colored dots) to one of the podiums is based on the rank that the respective algorithm achieved on the corresponding case. Note that the plot part above each podium place is further subdivided into  $p$  “columns”, where each column represents one participating algorithm (here:  $p = 17$ ). Dots corresponding to identical cases are connected by a line, leading to the shown spaghetti structure. Lower part: Bar charts represent the relative frequency for each algorithm to achieve the rank encoded by the podium place.



## Ranking heatmap

*Ranking heatmaps* for visualizing raw assessment data. Each cell  $(i, A_j)$  shows the absolute frequency of cases in which algorithm  $A_j$  achieved rank  $i$ .

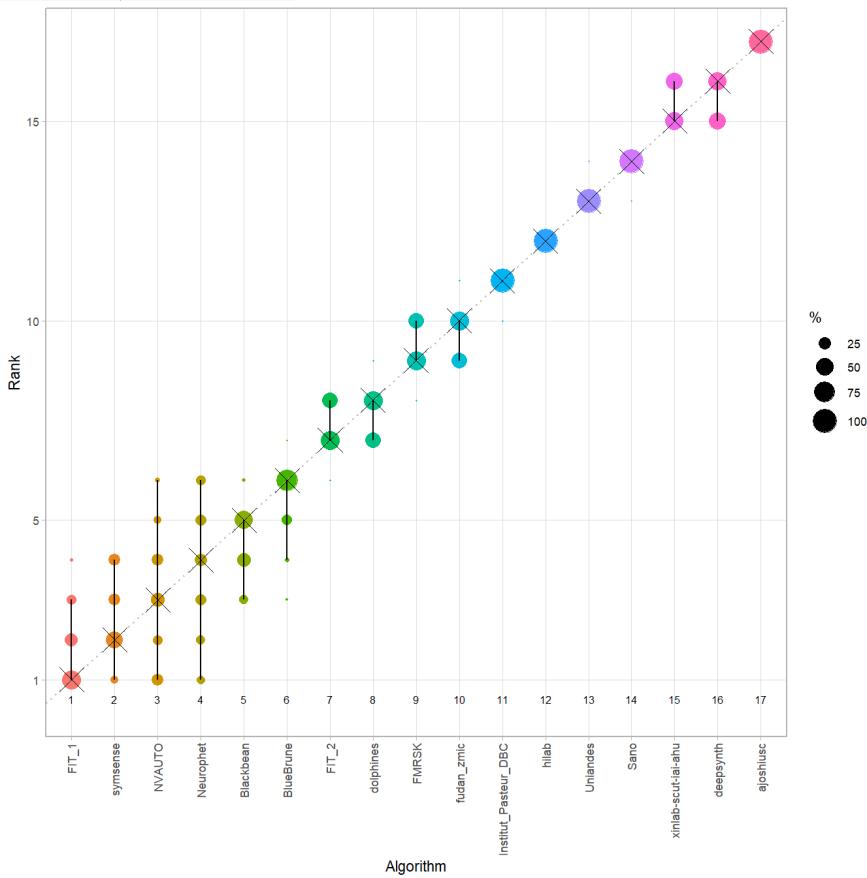


### Visualization of ranking stability

*Blob plot* for visualizing ranking stability based on bootstrap sampling

Algorithms are color-coded, and the area of each blob at position ( $A_i, \text{rank } j$ ) is proportional to the relative frequency  $A_i$  achieved rank  $j$  across  $b = 1000$  bootstrap samples. The median rank for each algorithm is indicated by a black cross. 95% bootstrap intervals across bootstrap samples are indicated by black lines.

```
## Warning: `guides(<scale> = FALSE)` is deprecated.
Please           use           `guides(<scale> =
## "none")` instead.
```

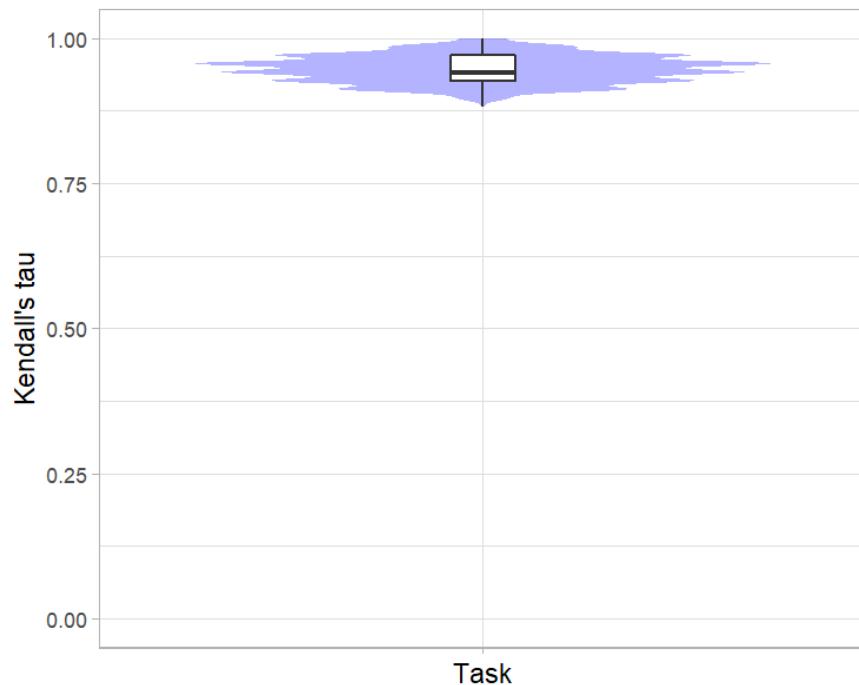


### ***Violin plot for visualizing ranking stability based on bootstrapping***

The ranking list based on the full assessment data is pairwise compared with the ranking lists based on the individual bootstrap samples (here  $b = 1000$  samples). For each pair of rankings, Kendall's  $\tau$  correlation is computed. Kendall's  $\tau$  is a scaled index determining the correlation between the lists. It is computed by evaluating the number of pairwise concordances and discordances between ranking lists and produces values between  $-1$  (for inverted order) and  $1$  (for identical order). A violin plot, which simultaneously depicts a boxplot and a density plot, is generated from the results.

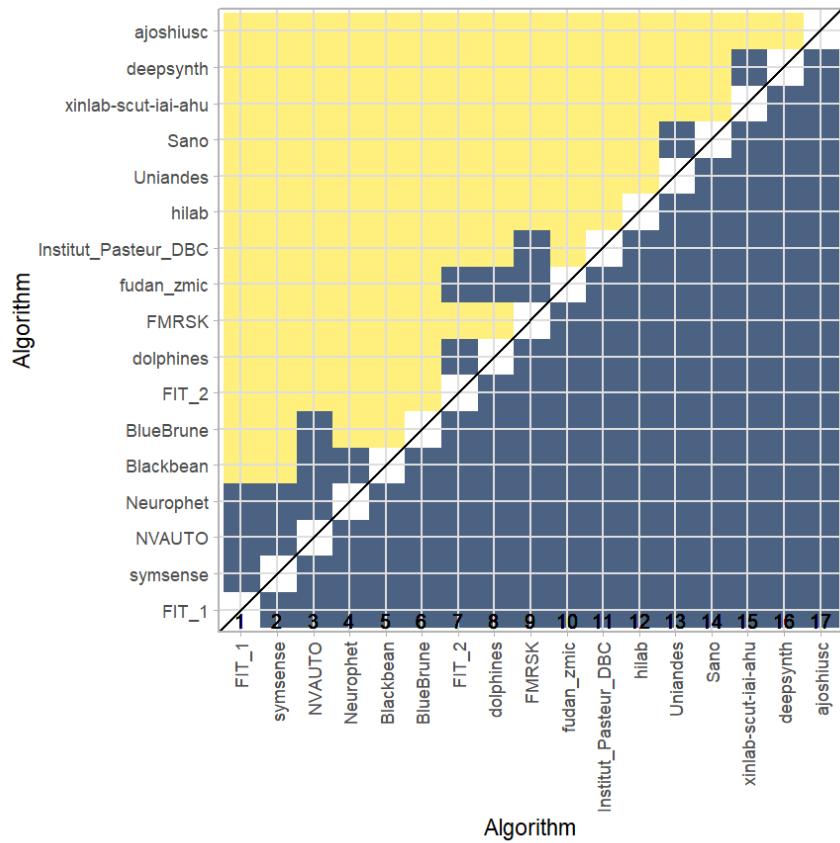
Summary Kendall's tau:

Task	mean	median	q25	q75
dummy-Task	0.9468088	0.9411765	0.9264706	0.9705882



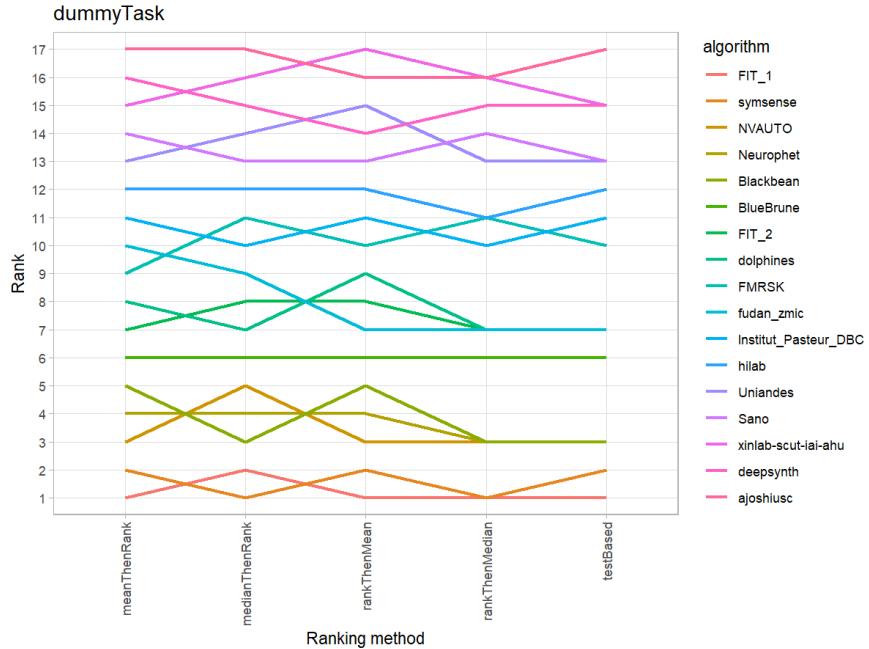
## *Significance maps* for visualizing ranking stability based on statistical significance

*Significance maps* depict incidence matrices of pairwise significant test results for the one-sided Wilcoxon signed rank test at a 5% significance level with adjustment for multiple testing according to Holm. Yellow shading indicates that metric values from the algorithm on the x-axis were significantly superior to those from the algorithm on the y-axis, blue color indicates no significant difference.



### Ranking robustness to ranking methods

*Line plots* for visualizing ranking robustness across different ranking methods. Each algorithm is represented by one colored line. For each ranking method encoded on the x-axis, the height of the line represents the corresponding rank. Horizontal lines indicate identical ranks for all methods.



### 6.3 References

Wiesenfarth, M., Reinke, A., Landman, B.A., Eisenmann, M., Aguilera Saiz, L., Cardoso, M.J., Maier-Hein, L. and Kopp-Schneider, A. Methods and open-source toolkit for analyzing and visualizing challenge results. *Sci Rep* **11**, 2369 (2021). <https://doi.org/10.1038/s41598-021-82017-6>

M. J. A. Eugster, T. Hothorn, and F. Leisch, “Exploratory and inferential analysis of benchmark experiments,” Institut fuer Statistik, Ludwig-Maximilians-Universitaet Muenchen, Germany, Technical Report 30, 2008. [Online]. Available: <http://epub.ub.uni-muenchen.de/4134/>.

## 7 Benchmarking report for Hausdorff Metrics – In Domain

created by challengeR v1.0.2  
07 July, 2023

This document presents a systematic report on the benchmark study “Hausdorff Metrics – In Domain”. Input data comprises raw metric values for all algorithms and cases. Generated plots are:

- Visualization of assessment data: Dot- and boxplot, podium plot and ranking heatmap
- Visualization of ranking stability: Blob plot, violin plot and significance map, line plot

Details can be found in Wiesenfarth et al. (2021).

### 7.1 Ranking

Algorithms within a task are ranked according to the following ranking scheme:

*aggregate using function (“mean”) then rank*

The analysis is based on 17 algorithms and 560 cases. 0 missing cases have been found in the data set.

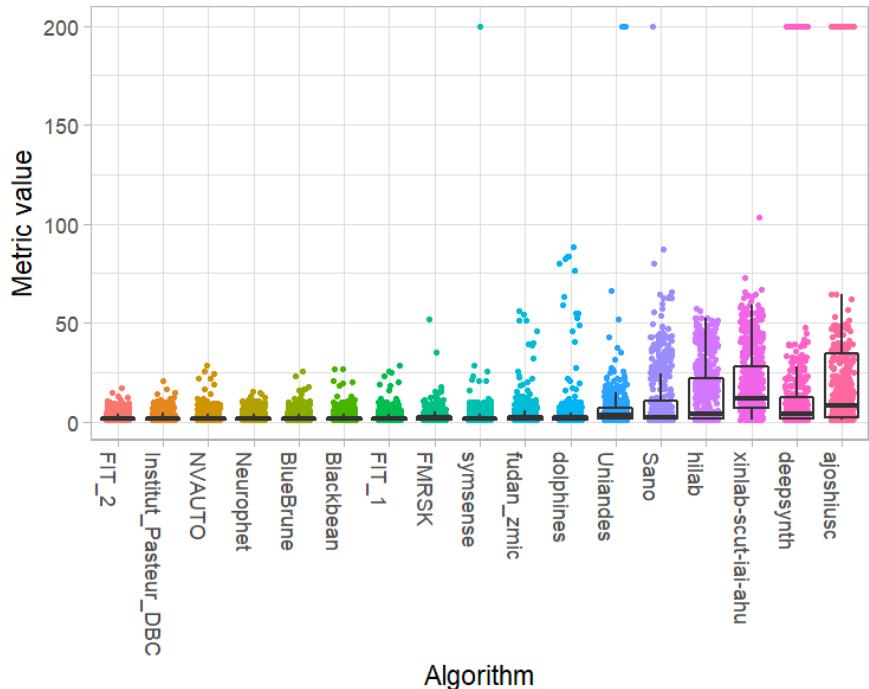
Ranking:

	Hausdorff_mean	rank
FIT_2	2.310282	1
Insti-tut_Pasteur_DBC	2.403996	2
NVAUTO	2.463725	3
Neurophet	2.467510	4
BlueBrune	2.472739	5
Blackbean	2.498323	6
FIT_1	2.520732	7
FMRSK	2.738572	8
symsense	2.828475	9
fudan_zmic	3.520624	10
dolphines	4.091324	11
Uniandes	6.239195	12
Sano	10.399105	13
hilab	13.285462	14
xinlab-scut-iai-ahu	18.562777	15
deepsynth	22.226778	16
ajoshiusc	41.175303	17

## 7.2 Visualization of raw assessment data

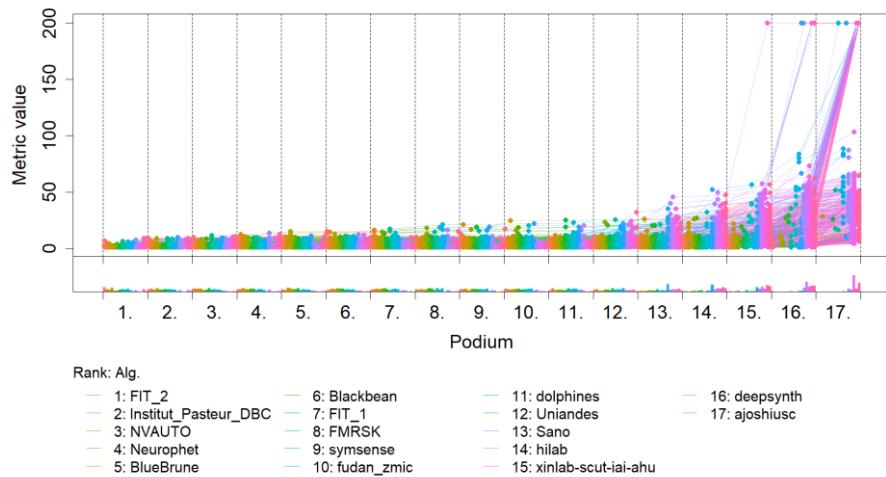
### Dot- and boxplot

*Dot- and boxplots* for visualizing raw assessment data separately for each algorithm. Boxplots representing descriptive statistics over all cases (median, quartiles and outliers) are combined with horizontally jittered dots representing individual cases.



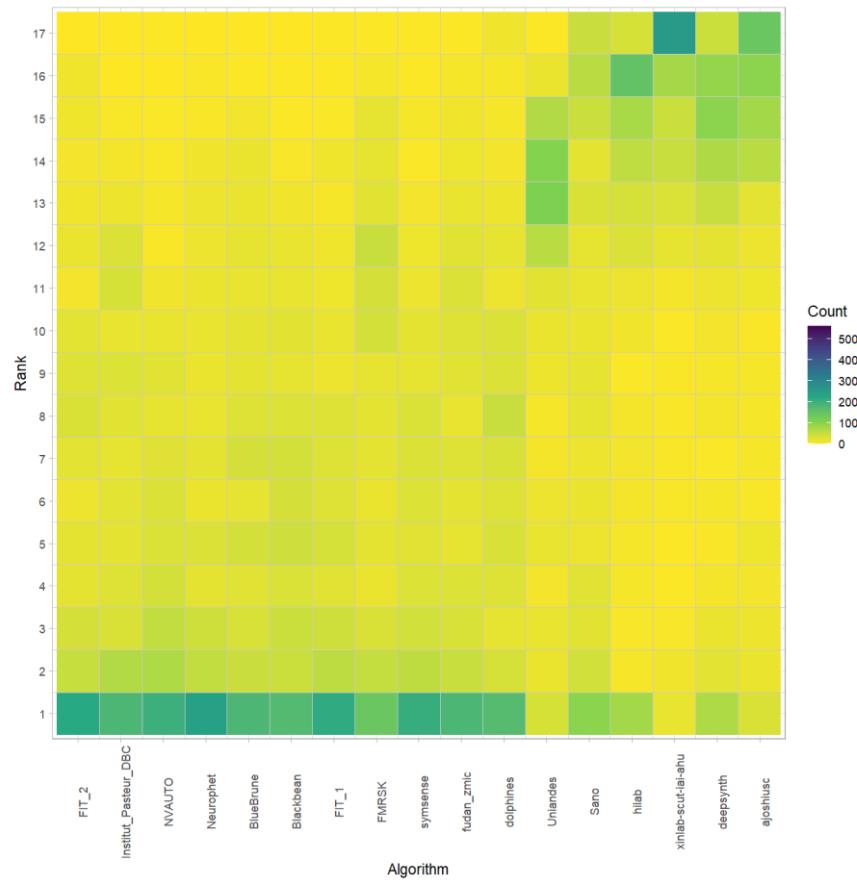
### Podium plot

*Podium plots* (see also Eugster et al., 2008) for visualizing raw assessment data. Upper part (spaghetti plot): Participating algorithms are color-coded, and each colored dot in the plot represents a metric value achieved with the respective algorithm. The actual metric value is encoded by the y-axis. Each podium (here:  $p=17$ ) represents one possible rank, ordered from best (1) to last (here: 17). The assignment of metric values (i.e. colored dots) to one of the podiums is based on the rank that the respective algorithm achieved on the corresponding case. Note that the plot part above each podium place is further subdivided into  $p$  “columns”, where each column represents one participating algorithm (here:  $p = 17$ ). Dots corresponding to identical cases are connected by a line, leading to the shown spaghetti structure. Lower part: Bar charts represent the relative frequency for each algorithm to achieve the rank encoded by the podium place.



## Ranking heatmap

*Ranking heatmaps* for visualizing raw assessment data. Each cell  $(i, A_j)$  shows the absolute frequency of cases in which algorithm  $A_j$  achieved rank  $i$ .

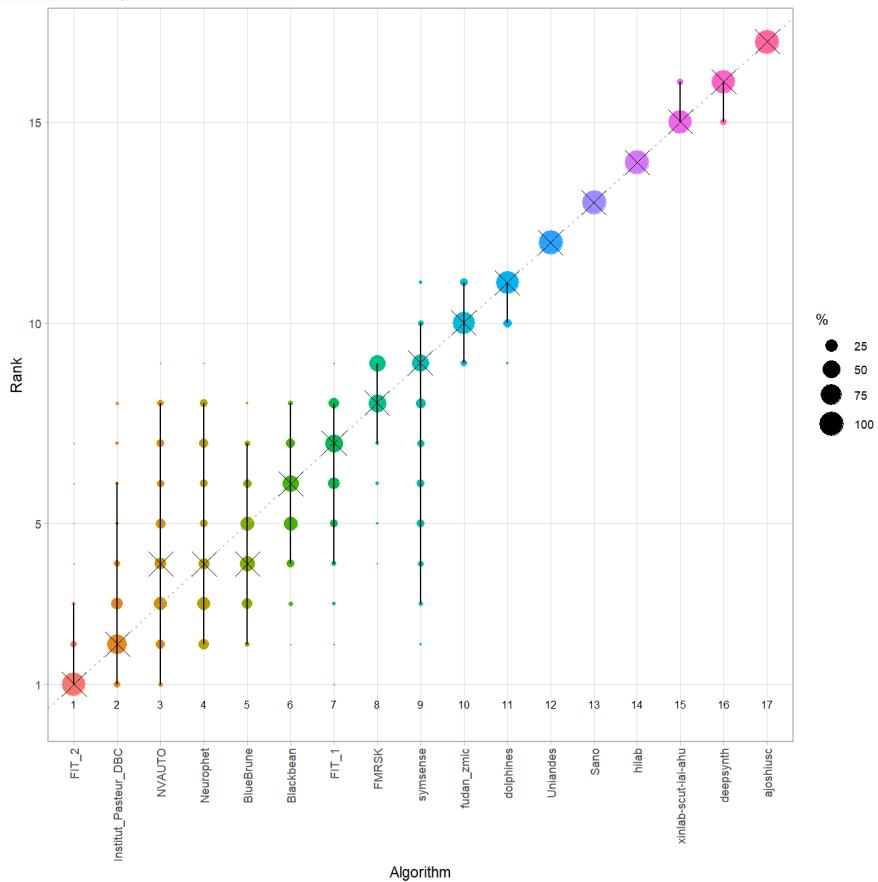


### Visualization of ranking stability

*Blob plot* for visualizing ranking stability based on bootstrap sampling

Algorithms are color-coded, and the area of each blob at position ( $A_i$ , rank  $j$ ) is proportional to the relative frequency  $A_i$  achieved rank  $j$  across  $b = 1000$  bootstrap samples. The median rank for each algorithm is indicated by a black cross. 95% bootstrap intervals across bootstrap samples are indicated by black lines.

```
## Warning: `guides(<scale> = FALSE)` is deprecated.
Please           use           `guides(<scale> =
## "none")` instead.
```

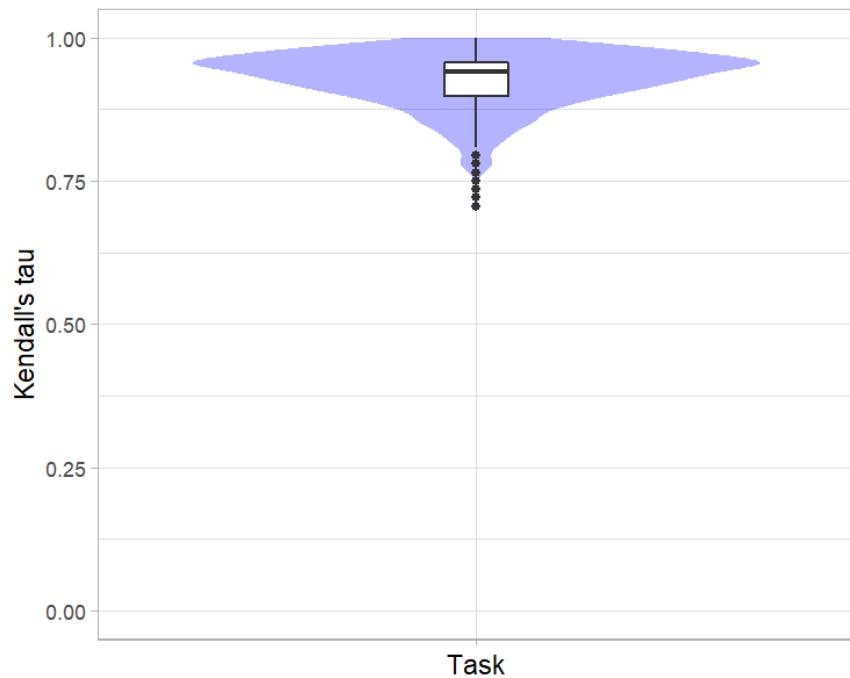


### ***Violin plot for visualizing ranking stability based on bootstrapping***

The ranking list based on the full assessment data is pairwise compared with the ranking lists based on the individual bootstrap samples (here  $b = 1000$  samples). For each pair of rankings, Kendall's  $\tau$  correlation is computed. Kendall's  $\tau$  is a scaled index determining the correlation between the lists. It is computed by evaluating the number of pairwise concordances and discordances between ranking lists and produces values between  $-1$  (for inverted order) and  $1$  (for identical order). A violin plot, which simultaneously depicts a boxplot and a density plot, is generated from the results.

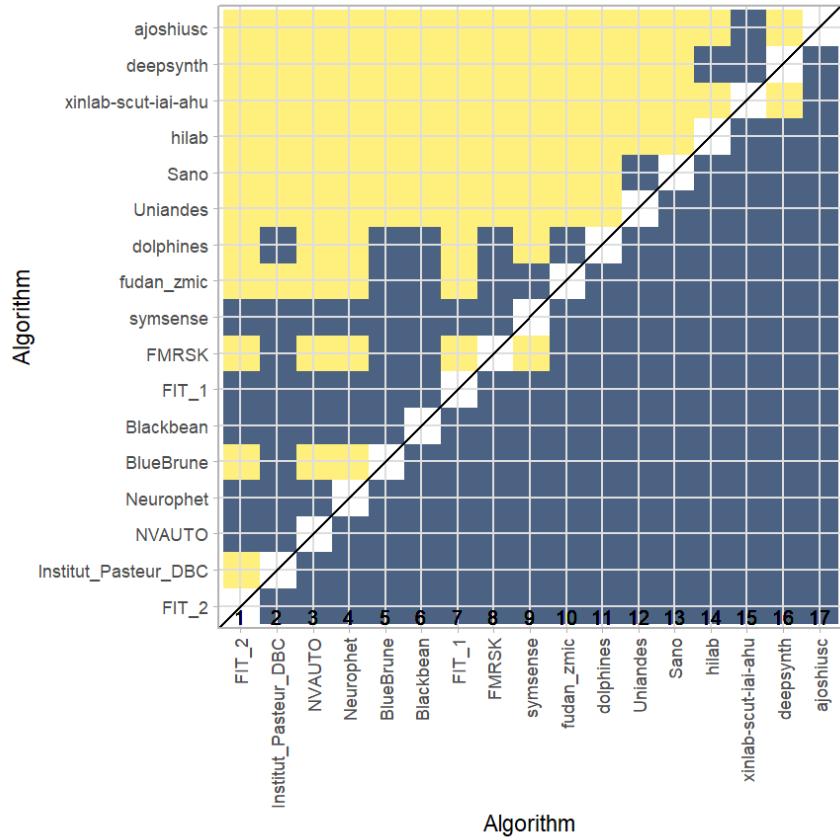
Summary Kendall's tau:

Task	mean	median	q25	q75
dummy-Task	0.9277794	0.9411765	0.8970588	0.9558824



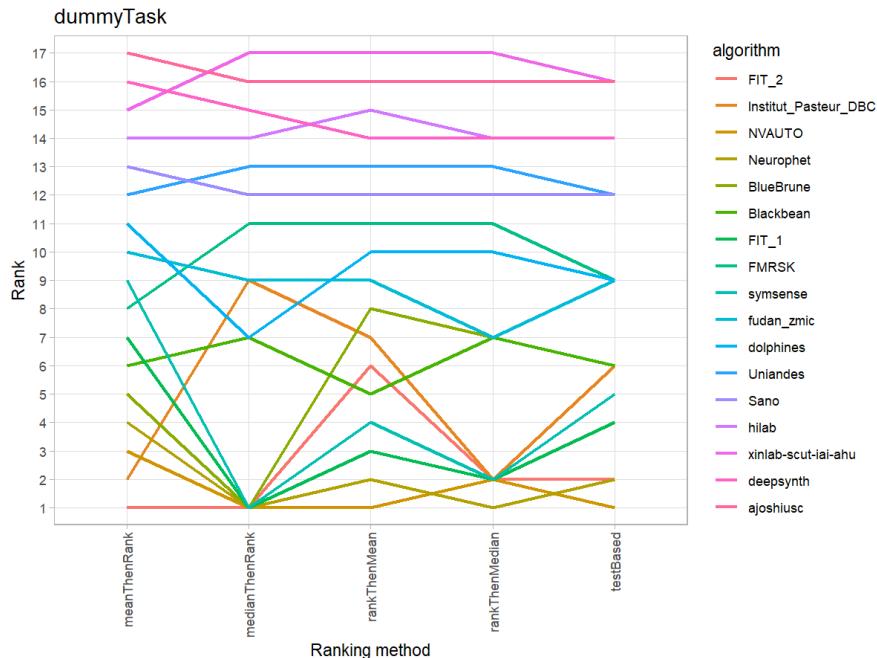
**Significance maps for visualizing ranking stability based on statistical significance**

*Significance maps* depict incidence matrices of pairwise significant test results for the one-sided Wilcoxon signed rank test at a 5% significance level with adjustment for multiple testing according to Holm. Yellow shading indicates that metric values from the algorithm on the x-axis were significantly superior to those from the algorithm on the y-axis, blue color indicates no significant difference.



### Ranking robustness to ranking methods

*Line plots* for visualizing ranking robustness across different ranking methods. Each algorithm is represented by one colored line. For each ranking method encoded on the x-axis, the height of the line represents the corresponding rank. Horizontal lines indicate identical ranks for all methods.



### 7.3 References

Wiesenfarth, M., Reinke, A., Landman, B.A., Eisenmann, M., Aguilera Saiz, L., Cardoso, M.J., Maier-Hein, L. and Kopp-Schneider, A. Methods and open-source toolkit for analyzing and visualizing challenge results. *Sci Rep* **11**, 2369 (2021). <https://doi.org/10.1038/s41598-021-82017-6>

M. J. A. Eugster, T. Hothorn, and F. Leisch, “Exploratory and inferential analysis of benchmark experiments,” Institut fuer Statistik, Ludwig-Maximilians-Universitaet Muenchen, Germany, Technical Report 30, 2008. [Online]. Available: <http://epub.ub.uni-muenchen.de/4134/>.

## 8 Benchmarking report for Volume Similarity Metrics – In Domain

created by challengeR v1.0.2  
07 July, 2023

This document presents a systematic report on the benchmark study “Volume Similarity Metrics – In Domain”. Input data comprises raw metric values for all algorithms and cases. Generated plots are:

- Visualization of assessment data: Dot- and boxplot, podium plot and ranking heatmap
- Visualization of ranking stability: Blob plot, violin plot and significance map, line plot

Details can be found in Wiesenfarth et al. (2021).

### 8.1 Ranking

Algorithms within a task are ranked according to the following ranking scheme:  
*aggregate using function (“mean”) then rank*

The analysis is based on 17 algorithms and 560 cases. 0 missing cases have been found in the data set.

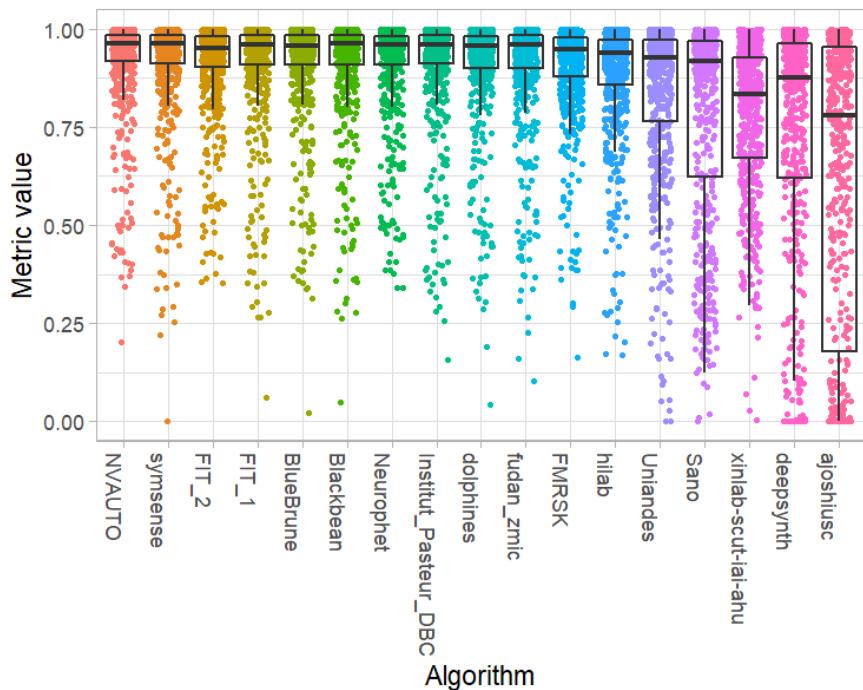
Ranking:

	Volume_Similarity_mean	rank
NVAUTO	0.9141931	1
symsense	0.9099799	2
FIT_2	0.9099546	3
FIT_1	0.9092761	4
BlueBrune	0.9091471	5
Blackbean	0.9090229	6
Neurophet	0.9085816	7
Insti-tut_Pasteur_DBC	0.9079917	8
dolphins	0.9060934	9
fudan_zmic	0.9048224	10
FMRSK	0.9028394	11
hilab	0.8802276	12
Uniandes	0.8361667	13
Sano	0.7797989	14
xinlab-scut-iai-ahu	0.7769642	15
deepsynth	0.7271838	16
ajoshiusc	0.6107316	17

## 8.2 Visualization of raw assessment data

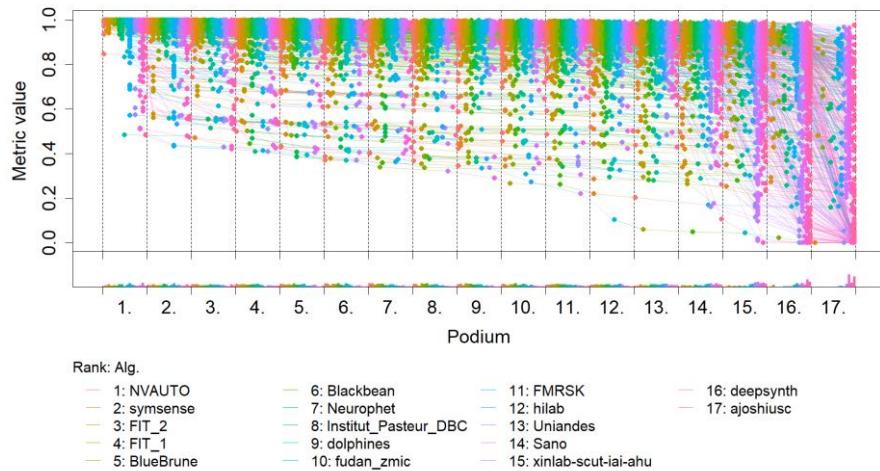
### Dot- and boxplot

*Dot- and boxplots* for visualizing raw assessment data separately for each algorithm. Boxplots representing descriptive statistics over all cases (median, quartiles and outliers) are combined with horizontally jittered dots representing individual cases.



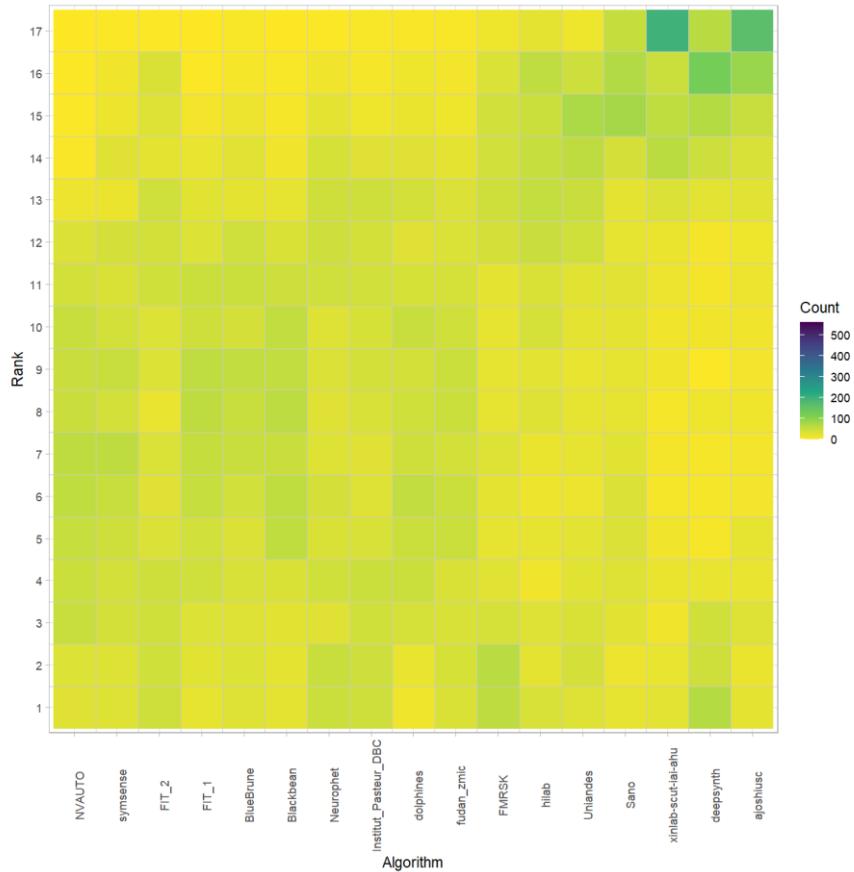
### Podium plot

*Podium plots* (see also Eugster et al., 2008) for visualizing raw assessment data. Upper part (spaghetti plot): Participating algorithms are color-coded, and each colored dot in the plot represents a metric value achieved with the respective algorithm. The actual metric value is encoded by the y-axis. Each podium (here:  $p=17$ ) represents one possible rank, ordered from best (1) to last (here: 17). The assignment of metric values (i.e. colored dots) to one of the podiums is based on the rank that the respective algorithm achieved on the corresponding case. Note that the plot part above each podium place is further subdivided into  $p$  “columns”, where each column represents one participating algorithm (here:  $p = 17$ ). Dots corresponding to identical cases are connected by a line, leading to the shown spaghetti structure. Lower part: Bar charts represent the relative frequency for each algorithm to achieve the rank encoded by the podium place.



## Ranking heatmap

*Ranking heatmaps* for visualizing raw assessment data. Each cell  $(i, A_j)$  shows the absolute frequency of cases in which algorithm  $A_j$  achieved rank  $i$ .

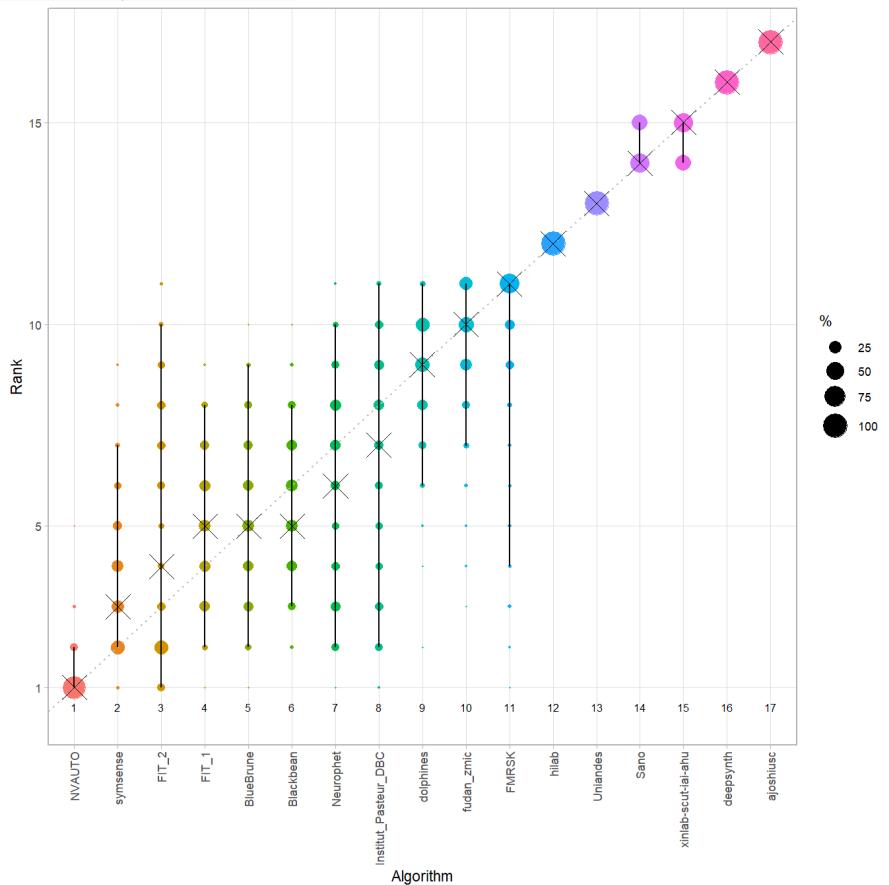


### Visualization of ranking stability

*Blob plot* for visualizing ranking stability based on bootstrap sampling

Algorithms are color-coded, and the area of each blob at position  $(A_i, \text{rank } j)$  is proportional to the relative frequency  $A_i$  achieved rank  $j$  across  $b = 1000$  bootstrap samples. The median rank for each algorithm is indicated by a black cross. 95% bootstrap intervals across bootstrap samples are indicated by black lines.

```
## Warning: `guides(<scale> = FALSE)` is deprecated.
Please           use           `guides(<scale> =
## "none")` instead.
```

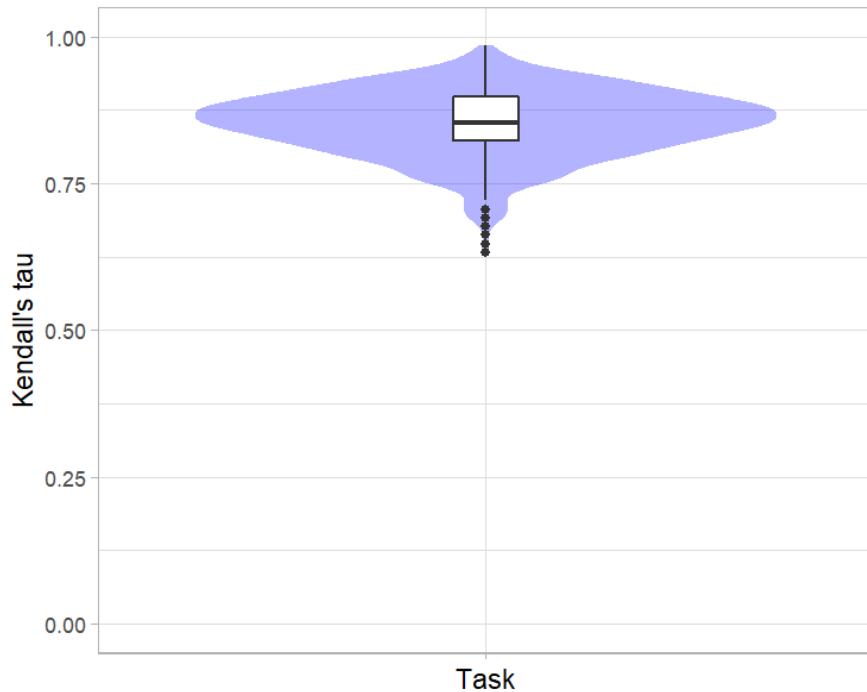


### ***Violin plot for visualizing ranking stability based on bootstrapping***

The ranking list based on the full assessment data is pairwise compared with the ranking lists based on the individual bootstrap samples (here  $b = 1000$  samples). For each pair of rankings, Kendall's  $\tau$  correlation is computed. Kendall's  $\tau$  is a scaled index determining the correlation between the lists. It is computed by evaluating the number of pairwise concordances and discordances between ranking lists and produces values between  $-1$  (for inverted order) and  $1$  (for identical order). A violin plot, which simultaneously depicts a boxplot and a density plot, is generated from the results.

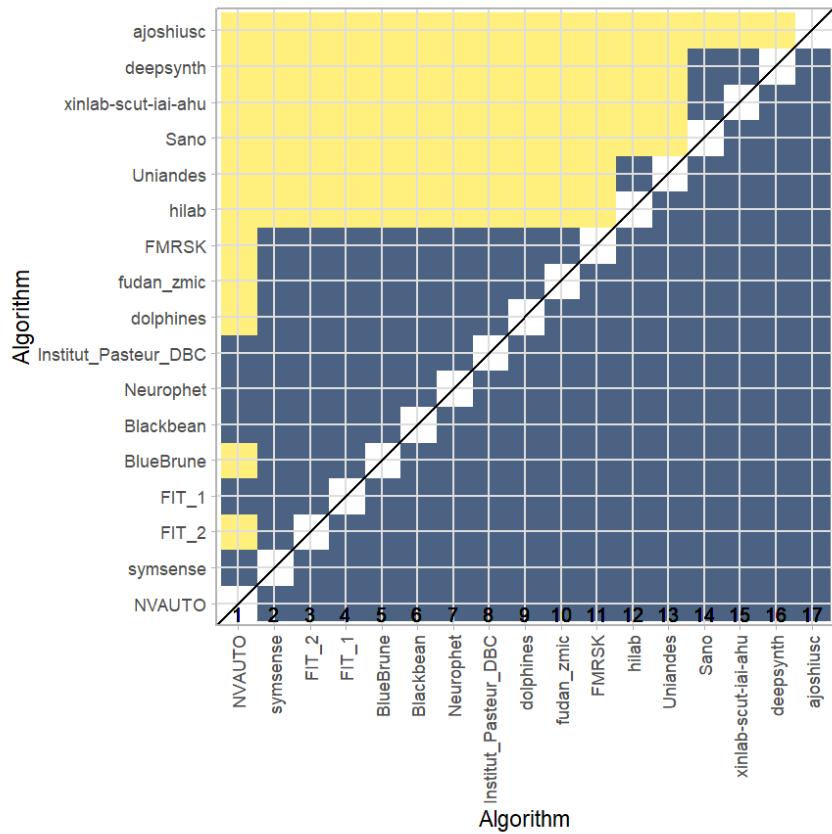
Summary Kendall's tau:

Task	mean	median	q25	q75
dummy-Task	0.8518824	0.8529412	0.8235294	0.8970588



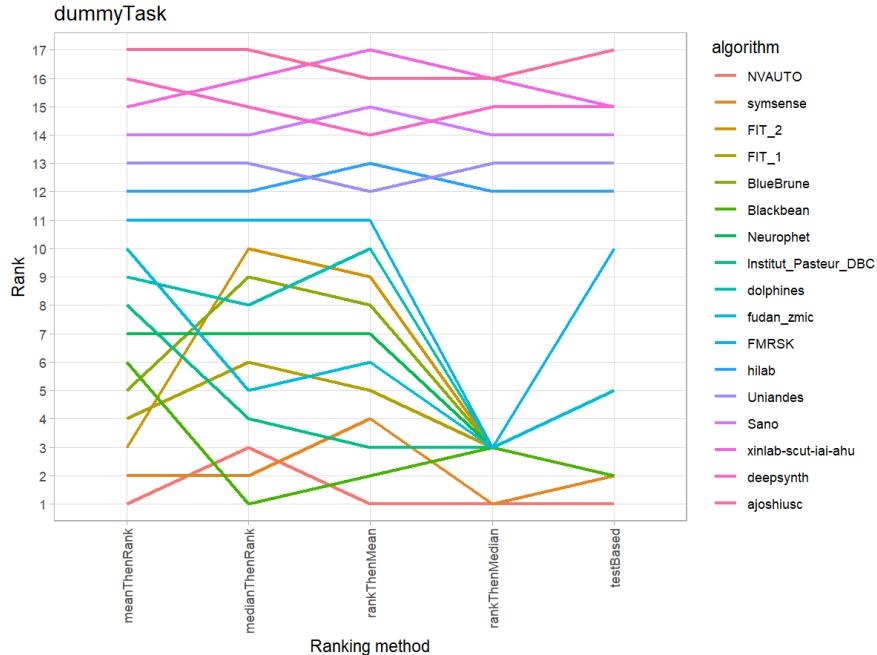
## *Significance maps* for visualizing ranking stability based on statistical significance

*Significance maps* depict incidence matrices of pairwise significant test results for the one-sided Wilcoxon signed rank test at a 5% significance level with adjustment for multiple testing according to Holm. Yellow shading indicates that metric values from the algorithm on the x-axis were significantly superior to those from the algorithm on the y-axis, blue color indicates no significant difference.



### Ranking robustness to ranking methods

*Line plots* for visualizing ranking robustness across different ranking methods. Each algorithm is represented by one colored line. For each ranking method encoded on the x-axis, the height of the line represents the corresponding rank. Horizontal lines indicate identical ranks for all methods.



### 8.3 References

Wiesenfarth, M., Reinke, A., Landman, B.A., Eisenmann, M., Aguilera Saiz, L., Cardoso, M.J., Maier-Hein, L. and Kopp-Schneider, A. Methods and open-source toolkit for analyzing and visualizing challenge results. *Sci Rep* **11**, 2369 (2021). <https://doi.org/10.1038/s41598-021-82017-6>

M. J. A. Eugster, T. Hothorn, and F. Leisch, “Exploratory and inferential analysis of benchmark experiments,” Institut fuer Statistik, Ludwig-Maximilians-Universitaet Muenchen, Germany, Technical Report 30, 2008. [Online]. Available: <http://epub.ub.uni-muenchen.de/4134/>.

## 9 Benchmarking report for Dice Metrics – Out of Domain

created by challengeR v1.0.2  
07 July, 2023

This document presents a systematic report on the benchmark study “Dice Metrics – Out of Domain”. Input data comprises raw metric values for all algorithms and cases. Generated plots are:

- Visualization of assessment data: Dot- and boxplot, podium plot and ranking heatmap
- Visualization of ranking stability: Blob plot, violin plot and significance map, line plot

Details can be found in Wiesenfarth et al. (2021).

### 9.1 Ranking

Algorithms within a task are ranked according to the following ranking scheme:

*aggregate using function (“mean”) then rank*

The analysis is based on 17 algorithms and 560 cases. 0 missing cases have been found in the data set.

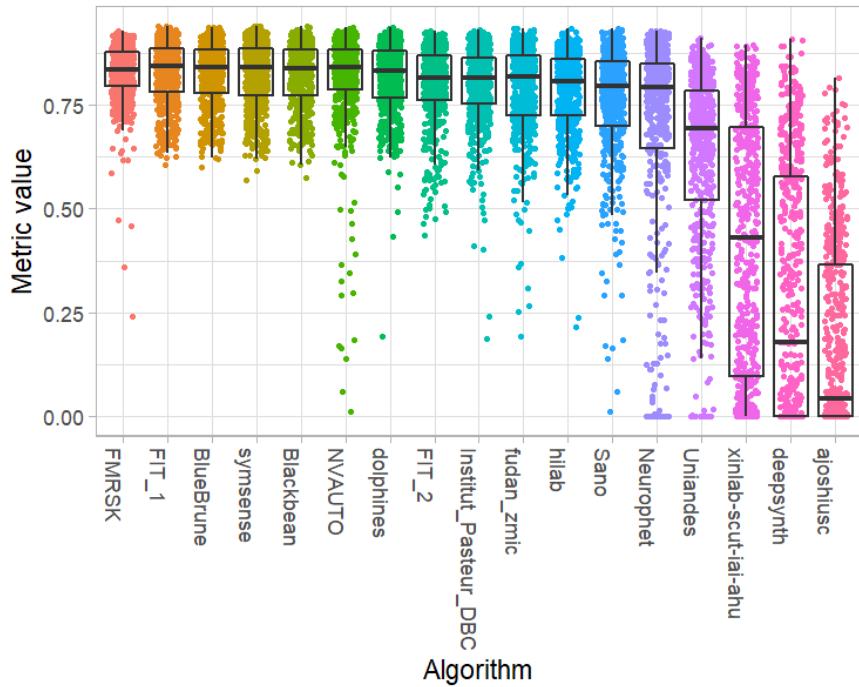
Ranking:

	Dice_mean	rank
FMRSK	0.8276876	1
FIT_1	0.8268333	2
BlueBrune	0.8224908	3
symsense	0.8219764	4
Blackbean	0.8211430	5
NVAUTO	0.8161359	6
dolphines	0.8159020	7
FIT_2	0.7996354	8
Insti-tut_Pasteur_DBC	0.7974522	9
fudan_zmic	0.7884858	10
hilab	0.7815745	11
Sano	0.7609486	12
Neurophet	0.6752764	13
Uniandes	0.6206139	14
xinlab-scut-iai-ahu	0.4099756	15
deepsynth	0.2901309	16
ajoshiusc	0.1816931	17

## 9.2 Visualization of raw assessment data

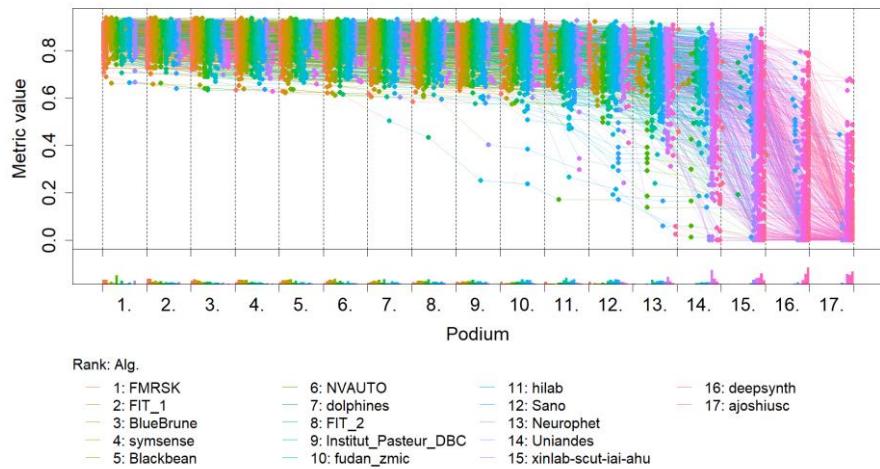
### Dot- and boxplot

*Dot- and boxplots* for visualizing raw assessment data separately for each algorithm. Boxplots representing descriptive statistics over all cases (median, quartiles and outliers) are combined with horizontally jittered dots representing individual cases.



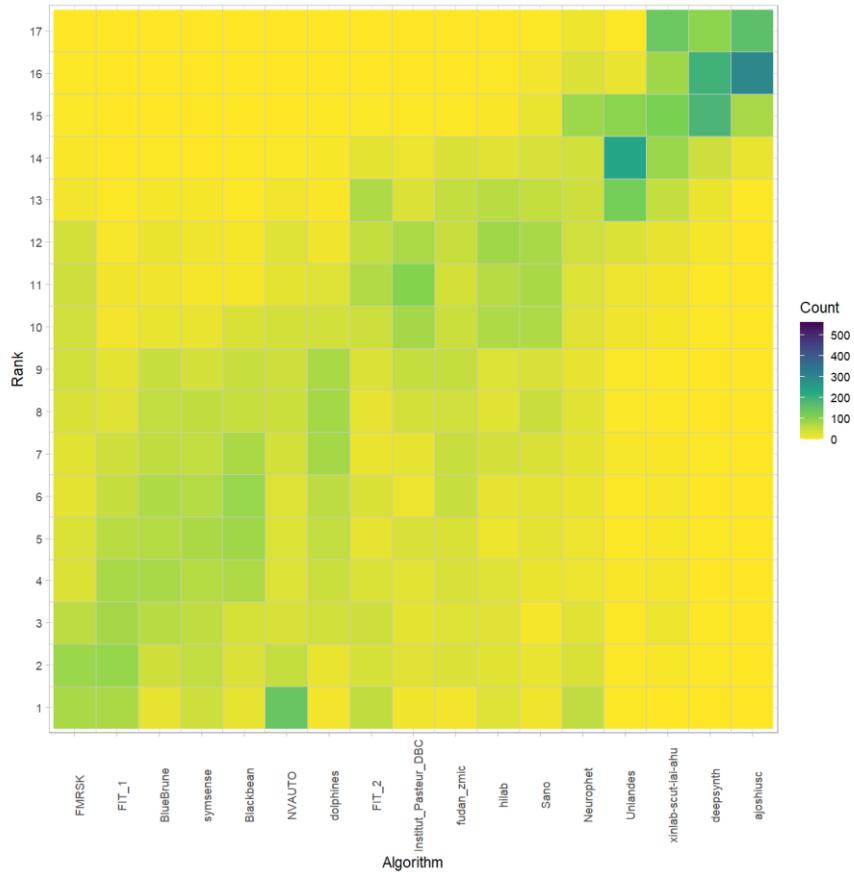
### Podium plot

*Podium plots* (see also Eugster et al., 2008) for visualizing raw assessment data. Upper part (spaghetti plot): Participating algorithms are color-coded, and each colored dot in the plot represents a metric value achieved with the respective algorithm. The actual metric value is encoded by the y-axis. Each podium (here:  $p=17$ ) represents one possible rank, ordered from best (1) to last (here: 17). The assignment of metric values (i.e. colored dots) to one of the podiums is based on the rank that the respective algorithm achieved on the corresponding case. Note that the plot part above each podium place is further subdivided into  $p$  “columns”, where each column represents one participating algorithm (here:  $p = 17$ ). Dots corresponding to identical cases are connected by a line, leading to the shown spaghetti structure. Lower part: Bar charts represent the relative frequency for each algorithm to achieve the rank encoded by the podium place.



## Ranking heatmap

*Ranking heatmaps* for visualizing raw assessment data. Each cell  $(i, A_j)$  shows the absolute frequency of cases in which algorithm  $A_j$  achieved rank  $i$ .

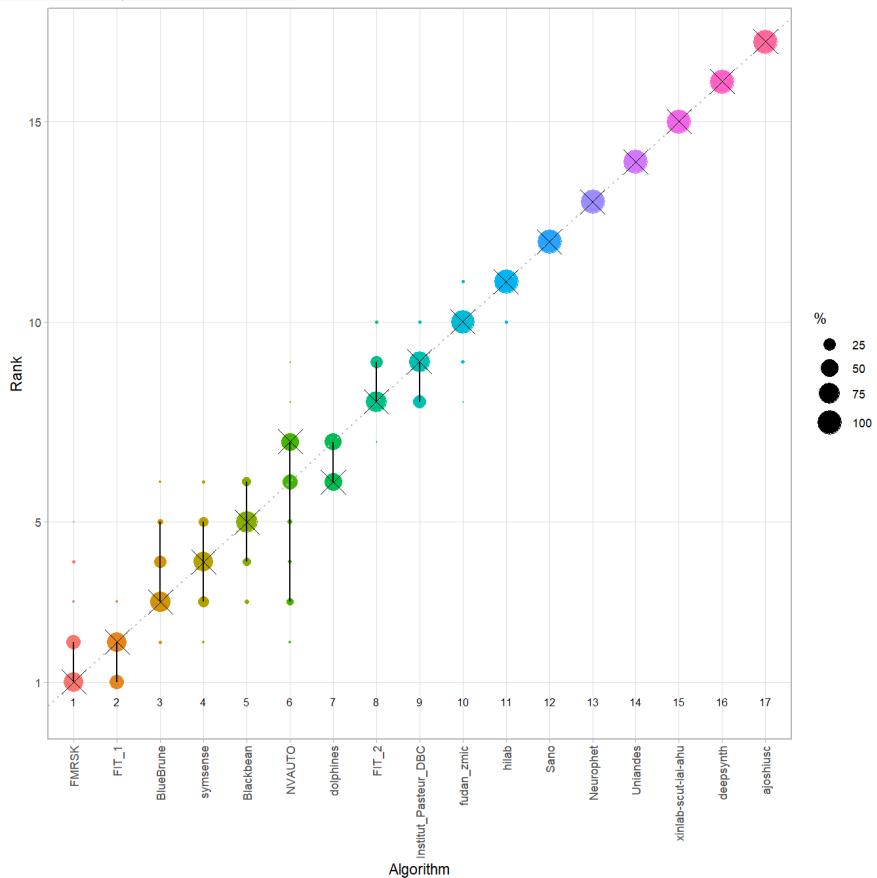


### Visualization of ranking stability

*Blob plot* for visualizing ranking stability based on bootstrap sampling

Algorithms are color-coded, and the area of each blob at position ( $A_i$ , rank  $j$ ) is proportional to the relative frequency  $A_i$  achieved rank  $j$  across  $b = 1000$  bootstrap samples. The median rank for each algorithm is indicated by a black cross. 95% bootstrap intervals across bootstrap samples are indicated by black lines.

```
## Warning: `guides(<scale> = FALSE)` is deprecated.
Please           use           `guides(<scale> =
## "none")` instead.
```

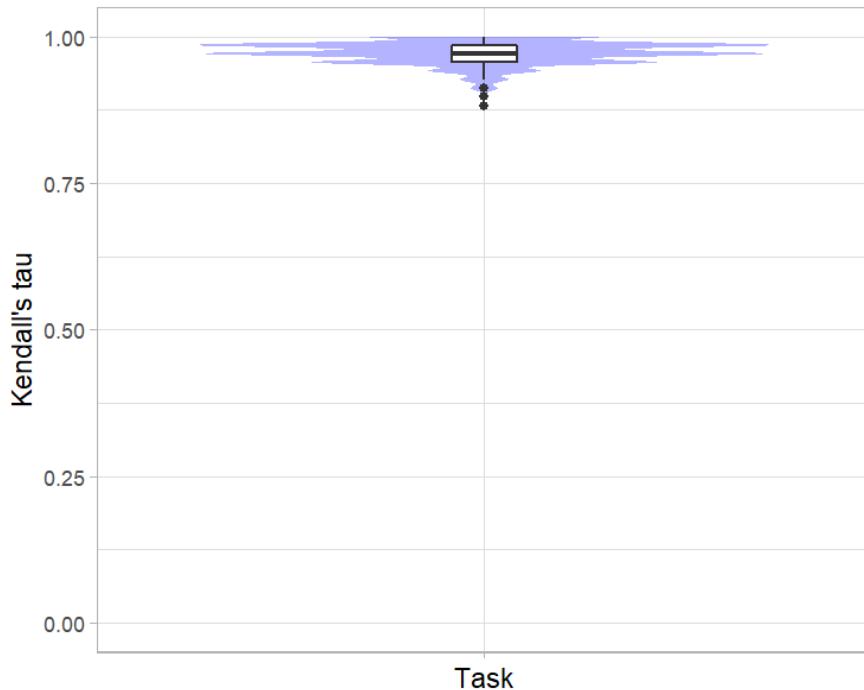


### ***Violin plot for visualizing ranking stability based on bootstrapping***

The ranking list based on the full assessment data is pairwise compared with the ranking lists based on the individual bootstrap samples (here  $b = 1000$  samples). For each pair of rankings, Kendall's  $\tau$  correlation is computed. Kendall's  $\tau$  is a scaled index determining the correlation between the lists. It is computed by evaluating the number of pairwise concordances and discordances between ranking lists and produces values between  $-1$  (for inverted order) and  $1$  (for identical order). A violin plot, which simultaneously depicts a boxplot and a density plot, is generated from the results.

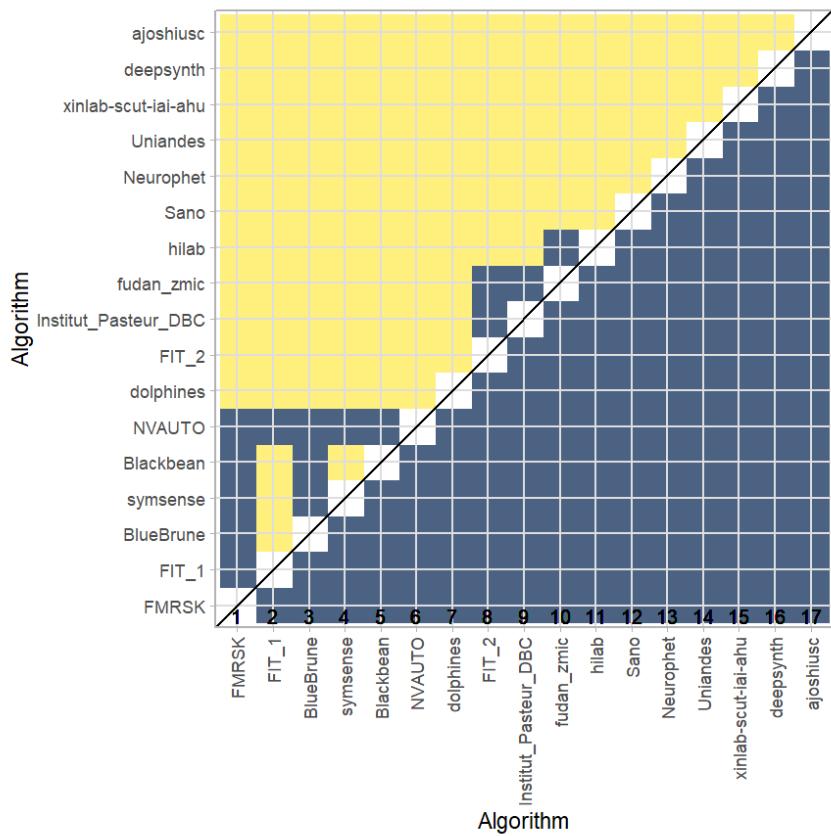
Summary Kendall's tau:

Task	mean	median	q25	q75
dummy- Task	0.9722353	0.9705882	0.9558824	0.9852941



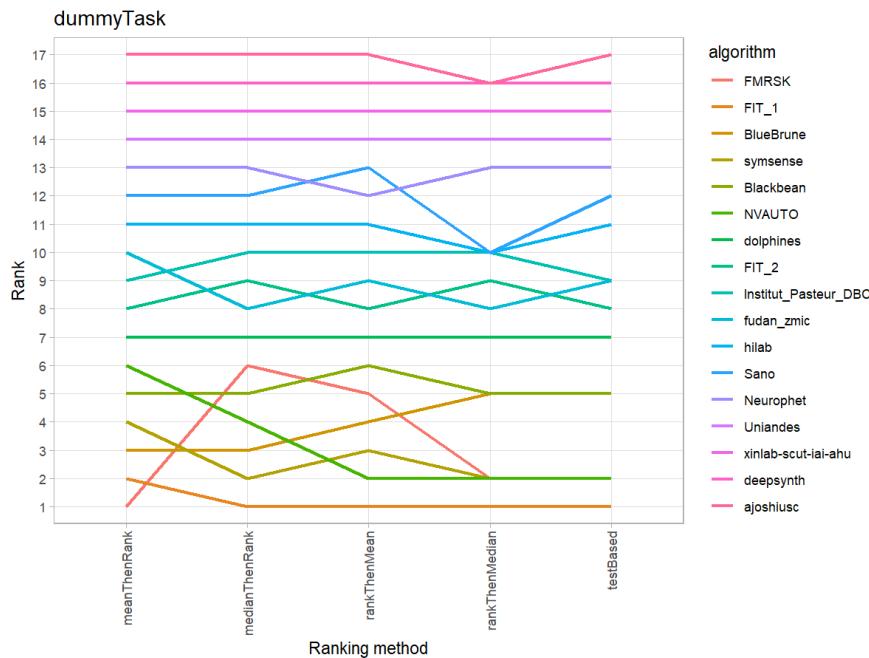
## *Significance maps* for visualizing ranking stability based on statistical significance

*Significance maps* depict incidence matrices of pairwise significant test results for the one-sided Wilcoxon signed rank test at a 5% significance level with adjustment for multiple testing according to Holm. Yellow shading indicates that metric values from the algorithm on the x-axis were significantly superior to those from the algorithm on the y-axis, blue color indicates no significant difference.



### Ranking robustness to ranking methods

*Line plots* for visualizing ranking robustness across different ranking methods. Each algorithm is represented by one colored line. For each ranking method encoded on the x-axis, the height of the line represents the corresponding rank. Horizontal lines indicate identical ranks for all methods.



### 9.3 References

Wiesenfarth, M., Reinke, A., Landman, B.A., Eisenmann, M., Aguilera Saiz, L., Cardoso, M.J., Maier-Hein, L. and Kopp-Schneider, A. Methods and open-source toolkit for analyzing and visualizing challenge results. *Sci Rep* **11**, 2369 (2021). <https://doi.org/10.1038/s41598-021-82017-6>

M. J. A. Eugster, T. Hothorn, and F. Leisch, “Exploratory and inferential analysis of benchmark experiments,” Institut fuer Statistik, Ludwig-Maximilians-Universitaet Muenchen, Germany, Technical Report 30, 2008. [Online]. Available: <http://epub.ub.uni-muenchen.de/4134/>.

## 10 Benchmarking report for Hausdorff Metrics – Out of Domain

created by challengeR v1.0.2  
07 July, 2023

This document presents a systematic report on the benchmark study “Hausdorff Metrics – Out of Domain”. Input data comprises raw metric values for all algorithms and cases. Generated plots are:

- Visualization of assessment data: Dot- and boxplot, podium plot and ranking heatmap
- Visualization of ranking stability: Blob plot, violin plot and significance map, line plot

Details can be found in Wiesenfarth et al. (2021).

### 10.1 Ranking

Algorithms within a task are ranked according to the following ranking scheme:  
*aggregate using function (“mean”) then rank*

The analysis is based on 17 algorithms and 560 cases. 0 missing cases have been found in the data set.

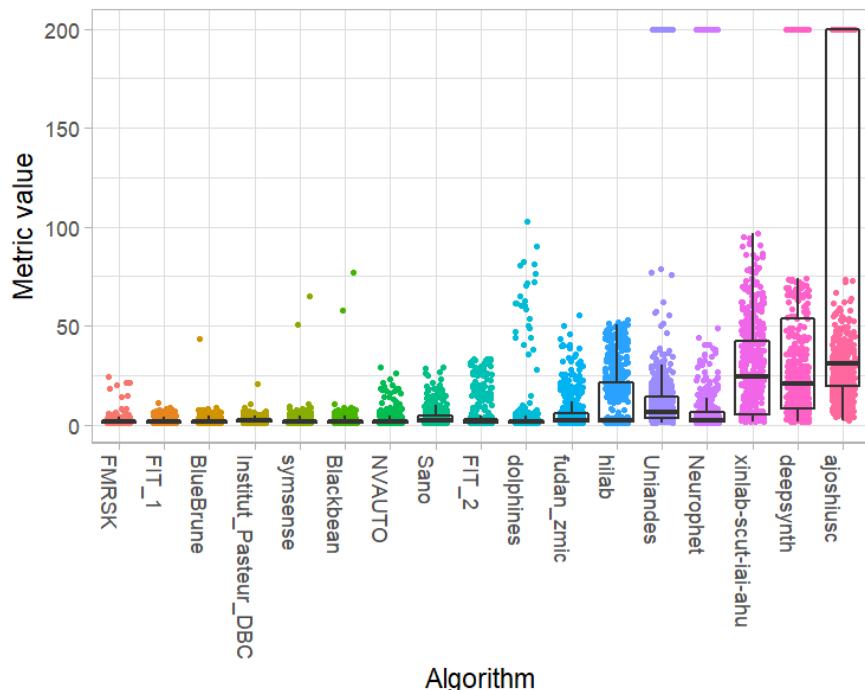
Ranking:

	Hausdorff_mean	rank
FMRSK	2.050589	1
FIT_1	2.172403	2
BlueBrune	2.282161	3
Insti-tut_Pasteur_DBC	2.369508	4
symsense	2.492404	5
Blackbean	2.514531	6
NVAUTO	2.752006	7
Sano	3.943880	8
FIT_2	4.531551	9
dolphins	4.950009	10
fudan_zmic	5.919579	11
hilab	12.730408	12
Uniandes	16.493135	13
Neurophet	18.108315	14
xinlab-scut-iai-ahu	27.736702	15
deepsynth	51.078992	16
ajoshiusc	72.020591	17

## 10.2 Visualization of raw assessment data

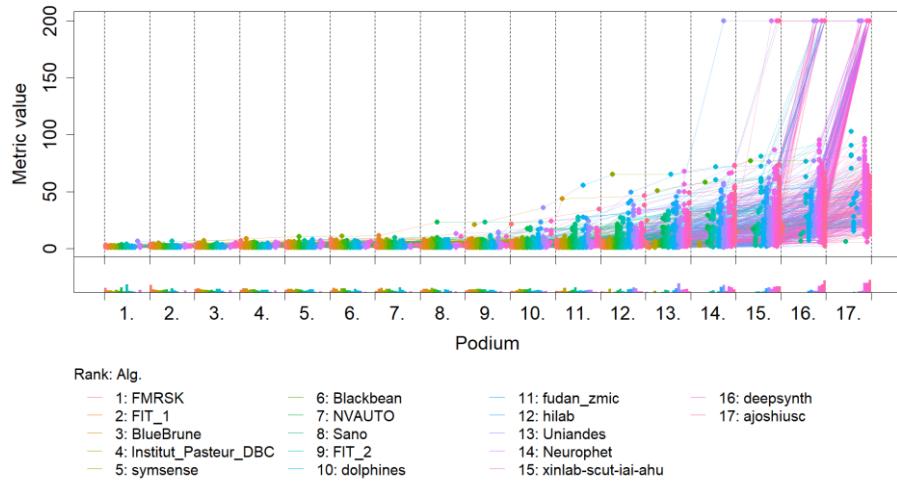
### Dot- and boxplot

*Dot- and boxplots* for visualizing raw assessment data separately for each algorithm. Boxplots representing descriptive statistics over all cases (median, quartiles and outliers) are combined with horizontally jittered dots representing individual cases.



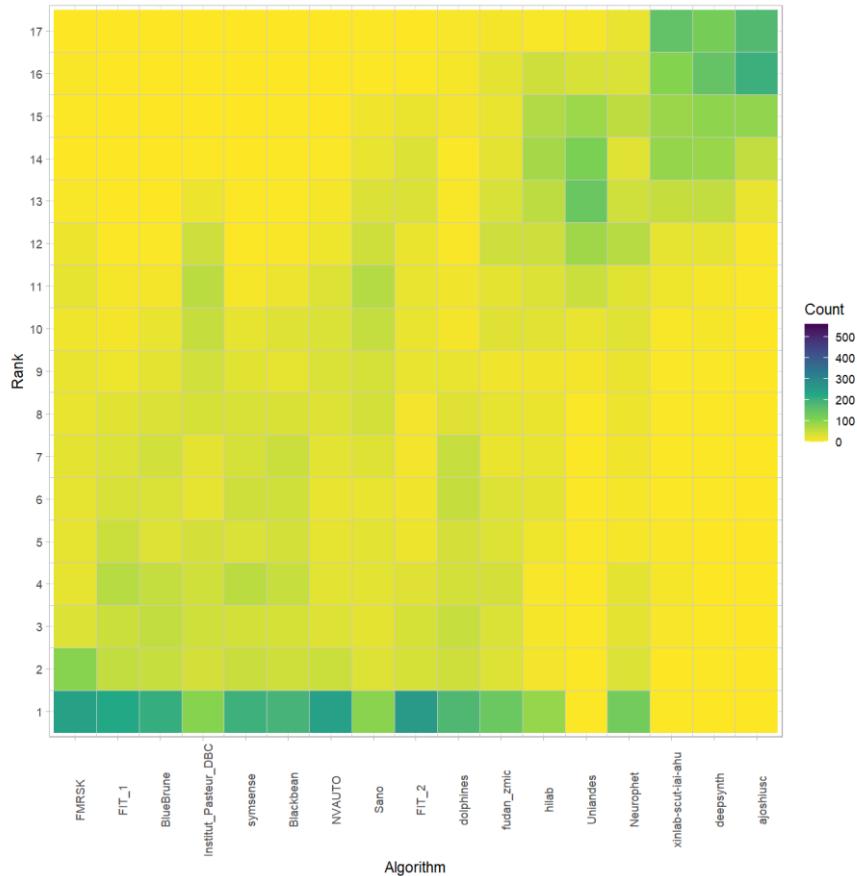
### Podium plot

*Podium plots* (see also Eugster et al., 2008) for visualizing raw assessment data. Upper part (spaghetti plot): Participating algorithms are color-coded, and each colored dot in the plot represents a metric value achieved with the respective algorithm. The actual metric value is encoded by the y-axis. Each podium (here:  $p=17$ ) represents one possible rank, ordered from best (1) to last (here: 17). The assignment of metric values (i.e. colored dots) to one of the podiums is based on the rank that the respective algorithm achieved on the corresponding case. Note that the plot part above each podium place is further subdivided into  $p$  “columns”, where each column represents one participating algorithm (here:  $p = 17$ ). Dots corresponding to identical cases are connected by a line, leading to the shown spaghetti structure. Lower part: Bar charts represent the relative frequency for each algorithm to achieve the rank encoded by the podium place.



## Ranking heatmap

*Ranking heatmaps* for visualizing raw assessment data. Each cell  $(i, A_j)$  shows the absolute frequency of cases in which algorithm  $A_j$  achieved rank  $i$ .

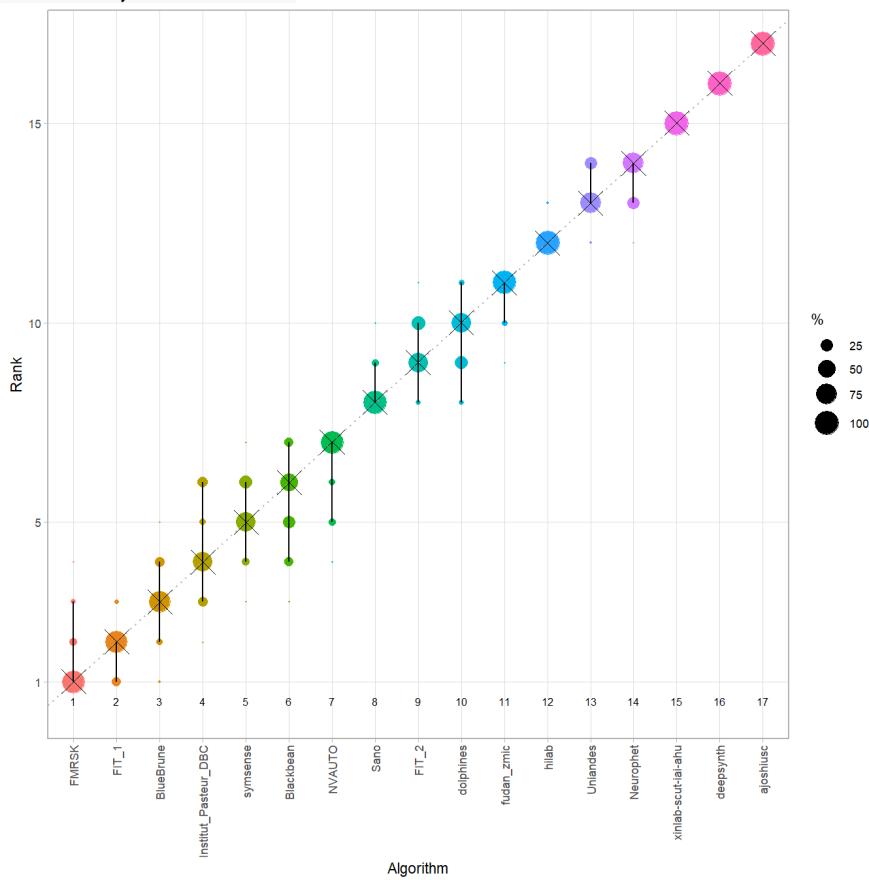


### Visualization of ranking stability

*Blob plot* for visualizing ranking stability based on bootstrap sampling

Algorithms are color-coded, and the area of each blob at position ( $A_i$ , rank  $j$ ) is proportional to the relative frequency  $A_i$  achieved rank  $j$  across  $b = 1000$  bootstrap samples. The median rank for each algorithm is indicated by a black cross. 95% bootstrap intervals across bootstrap samples are indicated by black lines.

```
## Warning: `guides(<scale> = FALSE)` is deprecated.
Please           use           `guides(<scale> =
## "none")` instead.
```

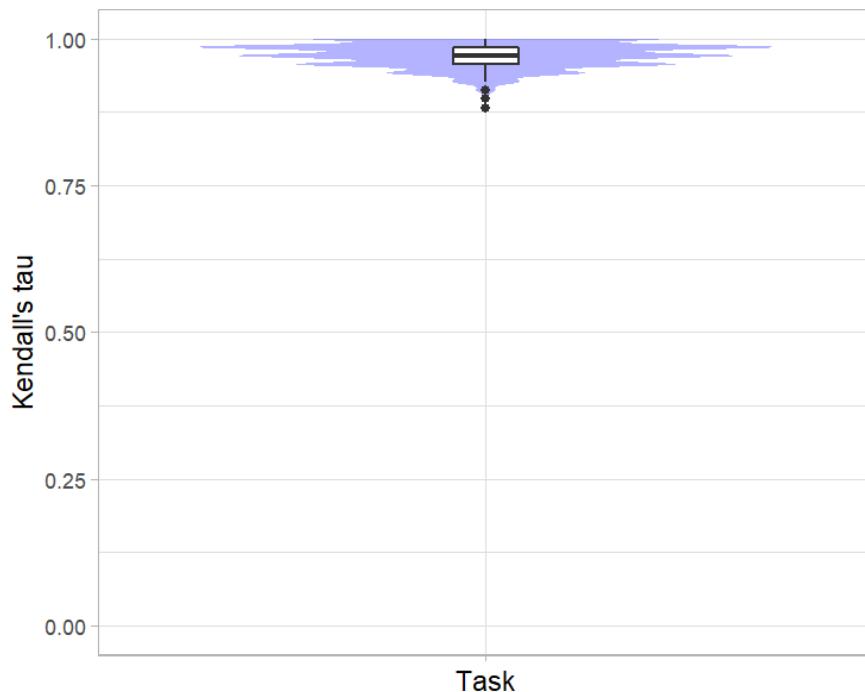


### ***Violin plot for visualizing ranking stability based on bootstrapping***

The ranking list based on the full assessment data is pairwise compared with the ranking lists based on the individual bootstrap samples (here  $b = 1000$  samples). For each pair of rankings, Kendall's  $\tau$  correlation is computed. Kendall's  $\tau$  is a scaled index determining the correlation between the lists. It is computed by evaluating the number of pairwise concordances and discordances between ranking lists and produces values between  $-1$  (for inverted order) and  $1$  (for identical order). A violin plot, which simultaneously depicts a boxplot and a density plot, is generated from the results.

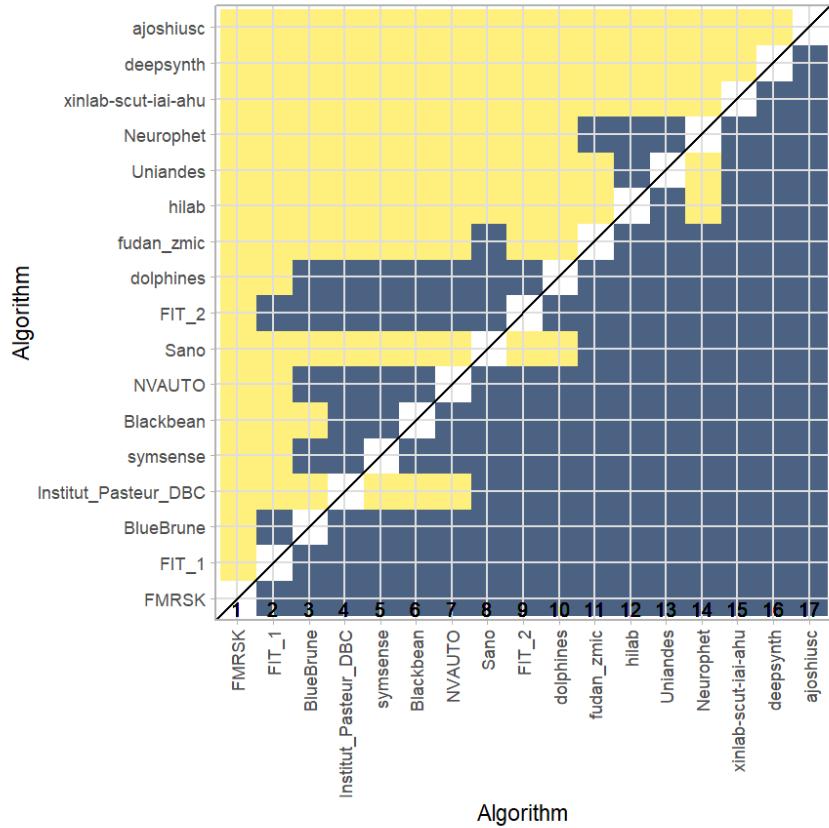
Summary Kendall's tau:

Task	mean	median	q25	q75
dummy-Task	0.9721029	0.9705882	0.9558824	0.9852941



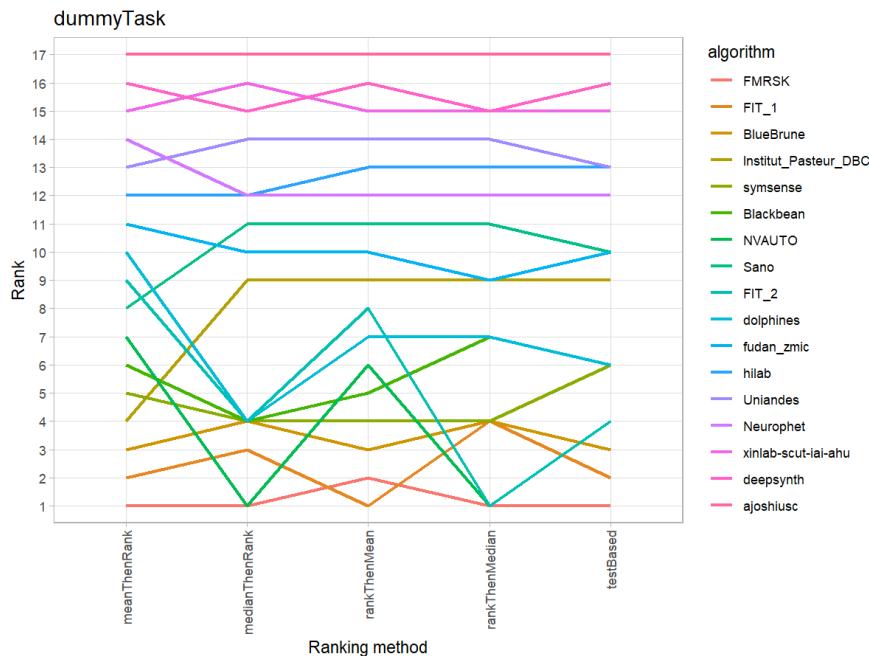
**Significance maps for visualizing ranking stability based on statistical significance**

*Significance maps* depict incidence matrices of pairwise significant test results for the one-sided Wilcoxon signed rank test at a 5% significance level with adjustment for multiple testing according to Holm. Yellow shading indicates that metric values from the algorithm on the x-axis were significantly superior to those from the algorithm on the y-axis, blue color indicates no significant difference.



### Ranking robustness to ranking methods

*Line plots* for visualizing ranking robustness across different ranking methods. Each algorithm is represented by one colored line. For each ranking method encoded on the x-axis, the height of the line represents the corresponding rank. Horizontal lines indicate identical ranks for all methods.



### 10.3 References

Wiesenfarth, M., Reinke, A., Landman, B.A., Eisenmann, M., Aguilera Saiz, L., Cardoso, M.J., Maier-Hein, L. and Kopp-Schneider, A. Methods and open-source toolkit for analyzing and visualizing challenge results. *Sci Rep* **11**, 2369 (2021). <https://doi.org/10.1038/s41598-021-82017-6>

M. J. A. Eugster, T. Hothorn, and F. Leisch, “Exploratory and inferential analysis of benchmark experiments,” Institut fuer Statistik, Ludwig-Maximilians-Universitaet Muenchen, Germany, Technical Report 30, 2008. [Online]. Available: <http://epub.ub.uni-muenchen.de/4134/>.

## 11 Benchmarking report for Volume Similarity Metrics – Out of Domain

created by challengeR v1.0.2  
07 July, 2023

This document presents a systematic report on the benchmark study “Volume Similarity Metrics – Out of Domain”. Input data comprises raw metric values for all algorithms and cases. Generated plots are:

- Visualization of assessment data: Dot- and boxplot, podium plot and ranking heatmap
- Visualization of ranking stability: Blob plot, violin plot and significance map, line plot

Details can be found in Wiesenfarth et al. (2021).

### 11.1 Ranking

Algorithms within a task are ranked according to the following ranking scheme:  
*aggregate using function (“mean”) then rank*

The analysis is based on 17 algorithms and 560 cases. 0 missing cases have been found in the data set.

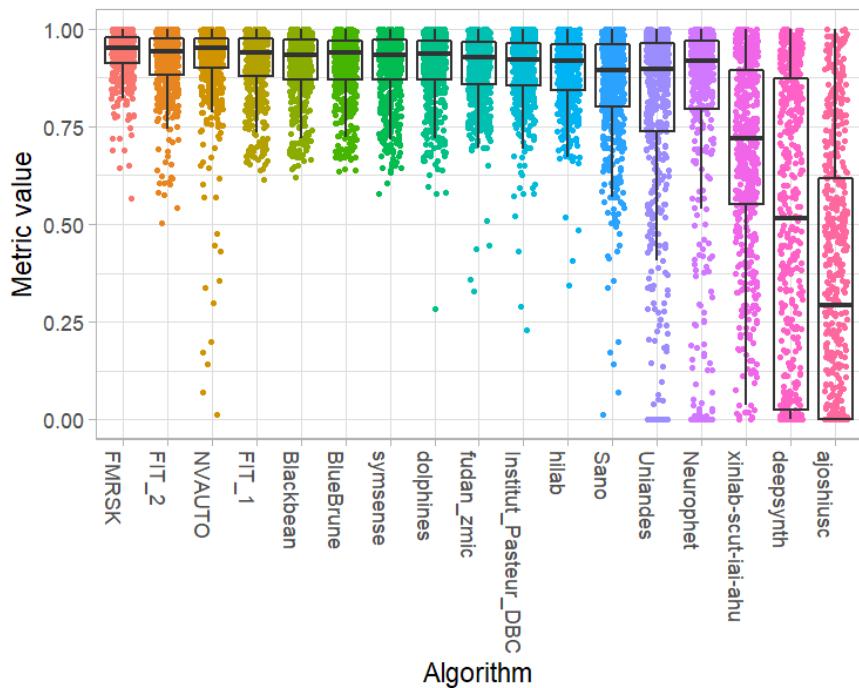
Ranking:

	Volume_Similarity_mean	rank
FMRSK	0.9363680	1
FIT_2	0.9165913	2
NVAUTO	0.9161527	3
FIT_1	0.9109040	4
Blackbean	0.9081494	5
BlueBrune	0.9068862	6
symsense	0.9043084	7
dolphines	0.9042991	8
fudan_zmic	0.9010763	9
Insti-tut_Pasteur_DBC	0.8945695	10
hilab	0.8927971	11
Sano	0.8549389	12
Uniandes	0.7914127	13
Neurophet	0.7784528	14
xinlab-scut-iai-ahu	0.6847400	15
deepsynth	0.4811447	16
ajoshiusc	0.3487719	17

## 11.2 Visualization of raw assessment data

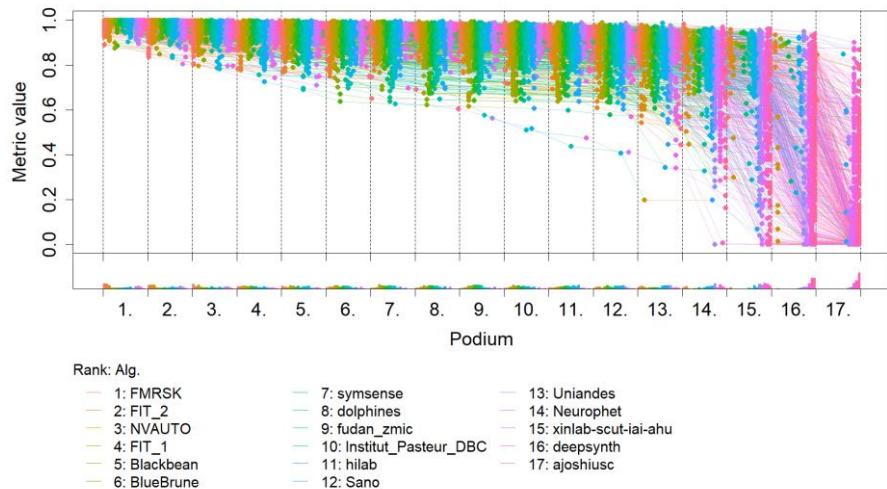
### Dot- and boxplot

*Dot- and boxplots* for visualizing raw assessment data separately for each algorithm. Boxplots representing descriptive statistics over all cases (median, quartiles and outliers) are combined with horizontally jittered dots representing individual cases.



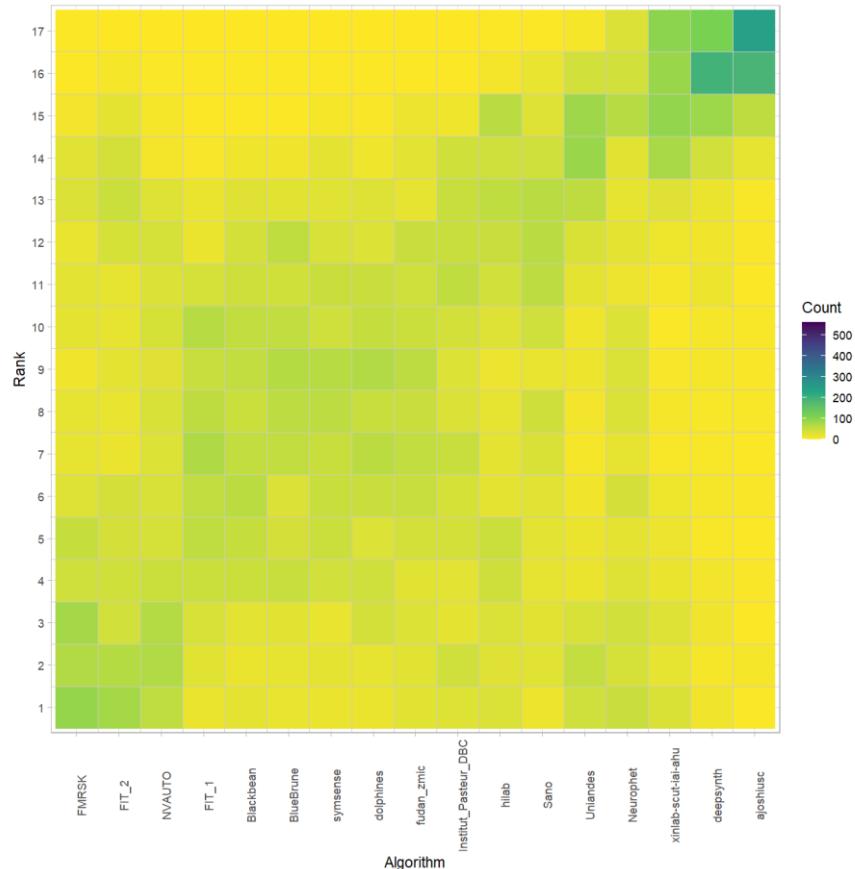
### Podium plot

*Podium plots* (see also Eugster et al., 2008) for visualizing raw assessment data. Upper part (spaghetti plot): Participating algorithms are color-coded, and each colored dot in the plot represents a metric value achieved with the respective algorithm. The actual metric value is encoded by the y-axis. Each podium (here:  $p=17$ ) represents one possible rank, ordered from best (1) to last (here: 17). The assignment of metric values (i.e. colored dots) to one of the podiums is based on the rank that the respective algorithm achieved on the corresponding case. Note that the plot part above each podium place is further subdivided into  $p$  “columns”, where each column represents one participating algorithm (here:  $p = 17$ ). Dots corresponding to identical cases are connected by a line, leading to the shown spaghetti structure. Lower part: Bar charts represent the relative frequency for each algorithm to achieve the rank encoded by the podium place.



## Ranking heatmap

*Ranking heatmaps* for visualizing raw assessment data. Each cell  $(i, A_j)$  shows the absolute frequency of cases in which algorithm  $A_j$  achieved rank  $i$ .

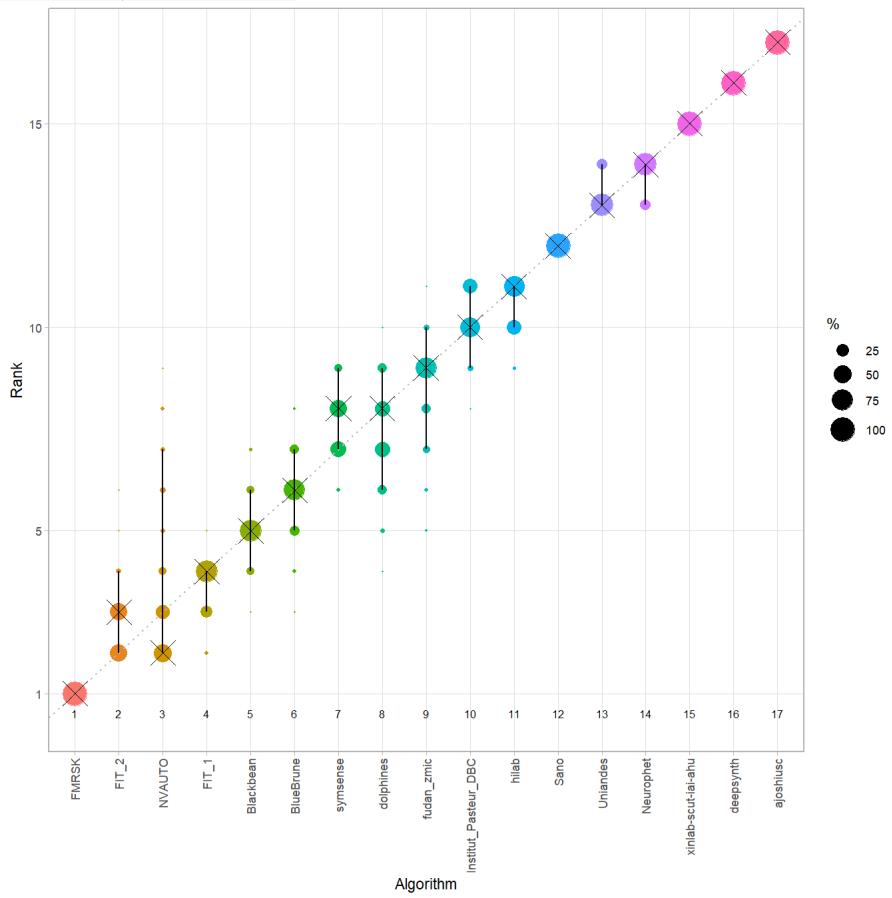


### Visualization of ranking stability

*Blob plot* for visualizing ranking stability based on bootstrap sampling

Algorithms are color-coded, and the area of each blob at position  $(A_i, \text{rank } j)$  is proportional to the relative frequency  $A_i$  achieved rank  $j$  across  $b = 1000$  bootstrap samples. The median rank for each algorithm is indicated by a black cross. 95% bootstrap intervals across bootstrap samples are indicated by black lines.

```
## Warning: `guides(<scale> = FALSE)` is deprecated.
Please           use           `guides(<scale> =
## "none")` instead.
```

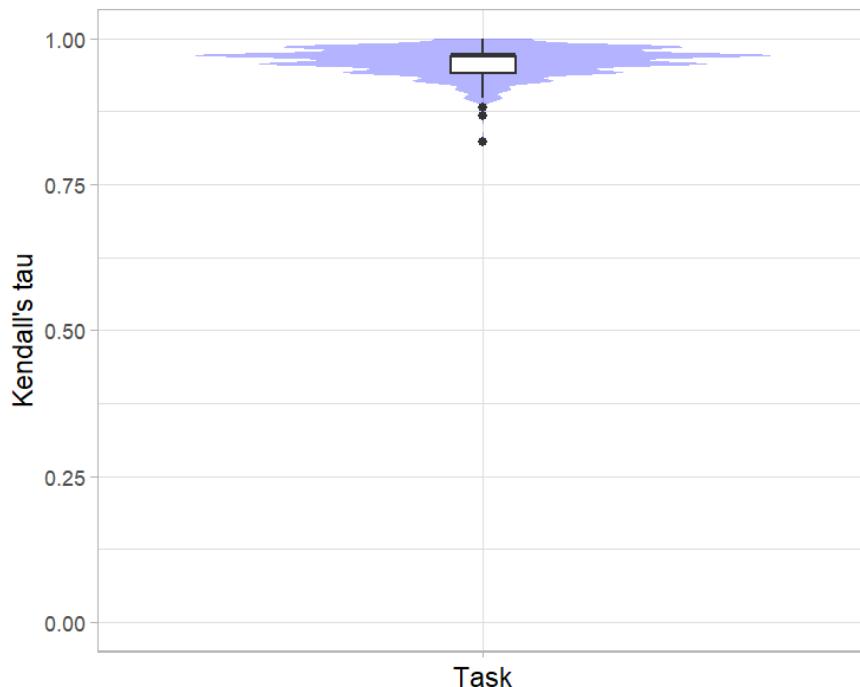


### ***Violin plot for visualizing ranking stability based on bootstrapping***

The ranking list based on the full assessment data is pairwise compared with the ranking lists based on the individual bootstrap samples (here  $b = 1000$  samples). For each pair of rankings, Kendall's  $\tau$  correlation is computed. Kendall's  $\tau$  is a scaled index determining the correlation between the lists. It is computed by evaluating the number of pairwise concordances and discordances between ranking lists and produces values between  $-1$  (for inverted order) and  $1$  (for identical order). A violin plot, which simultaneously depicts a boxplot and a density plot, is generated from the results.

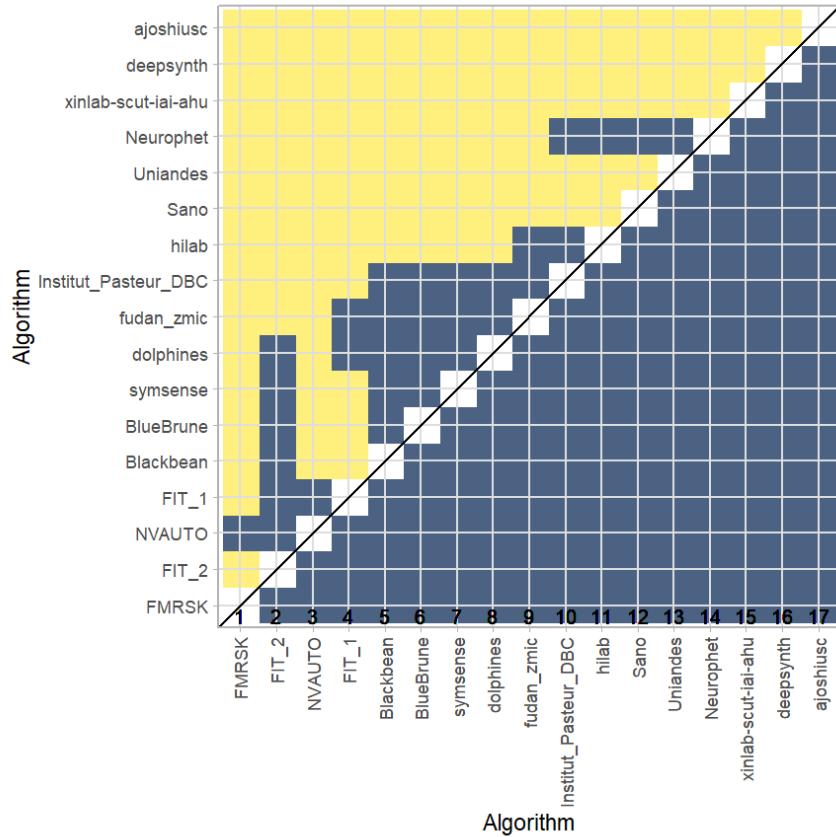
Summary Kendall's tau:

Task	mean	median	q25	q75
dummy-Task	0.9610882	0.9705882	0.9411765	0.9705882



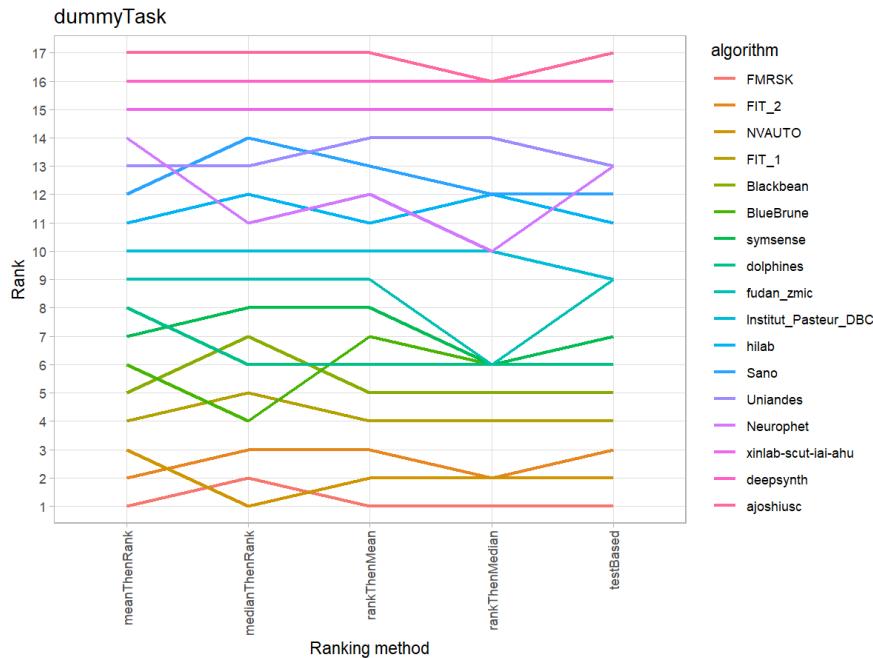
**Significance maps for visualizing ranking stability based on statistical significance**

*Significance maps* depict incidence matrices of pairwise significant test results for the one-sided Wilcoxon signed rank test at a 5% significance level with adjustment for multiple testing according to Holm. Yellow shading indicates that metric values from the algorithm on the x-axis were significantly superior to those from the algorithm on the y-axis, blue color indicates no significant difference.



### Ranking robustness to ranking methods

*Line plots* for visualizing ranking robustness across different ranking methods. Each algorithm is represented by one colored line. For each ranking method encoded on the x-axis, the height of the line represents the corresponding rank. Horizontal lines indicate identical ranks for all methods.



### 11.3 References

Wiesenfarth, M., Reinke, A., Landman, B.A., Eisenmann, M., Aguilera Saiz, L., Cardoso, M.J., Maier-Hein, L. and Kopp-Schneider, A. Methods and open-source toolkit for analyzing and visualizing challenge results. *Sci Rep* **11**, 2369 (2021). <https://doi.org/10.1038/s41598-021-82017-6>

M. J. A. Eugster, T. Hothorn, and F. Leisch, “Exploratory and inferential analysis of benchmark experiments,” Institut fuer Statistik, Ludwig-Maximilians-Universitaet Muenchen, Germany, Technical Report 30, 2008. [Online]. Available: <http://epub.ub.uni-muenchen.de/4134/>.

## 12 Evaluation Metrics per Label

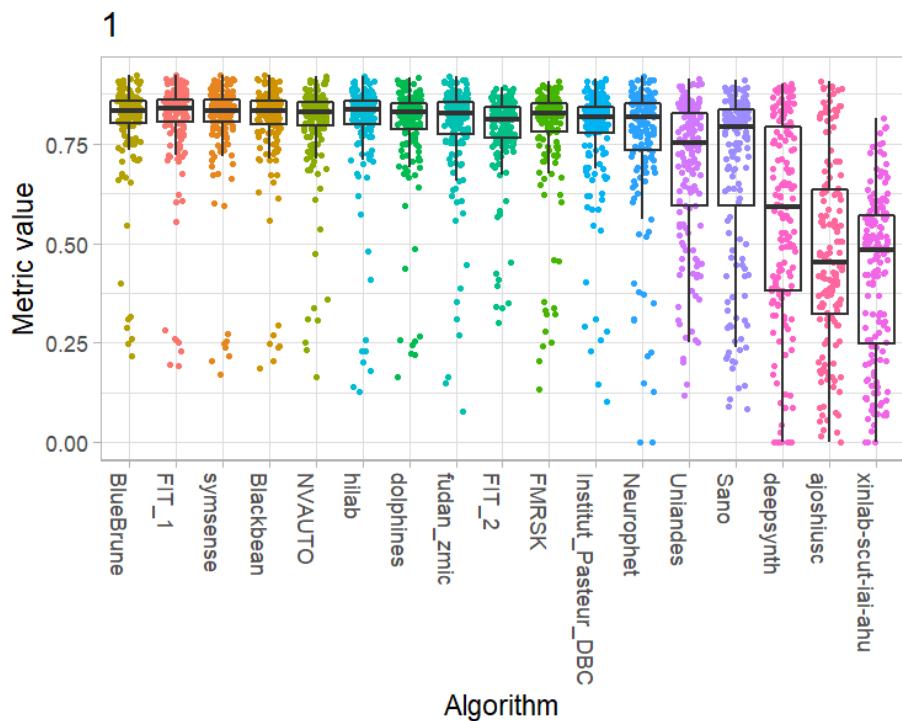
The following figures were created by the ChallengeR Tool, but were edited for brevity. To obtain the full ChallengeR reports by label, contact the authors.

## 12.1 Global Evaluation Metrics per Label

### Dice Similarity Coefficient

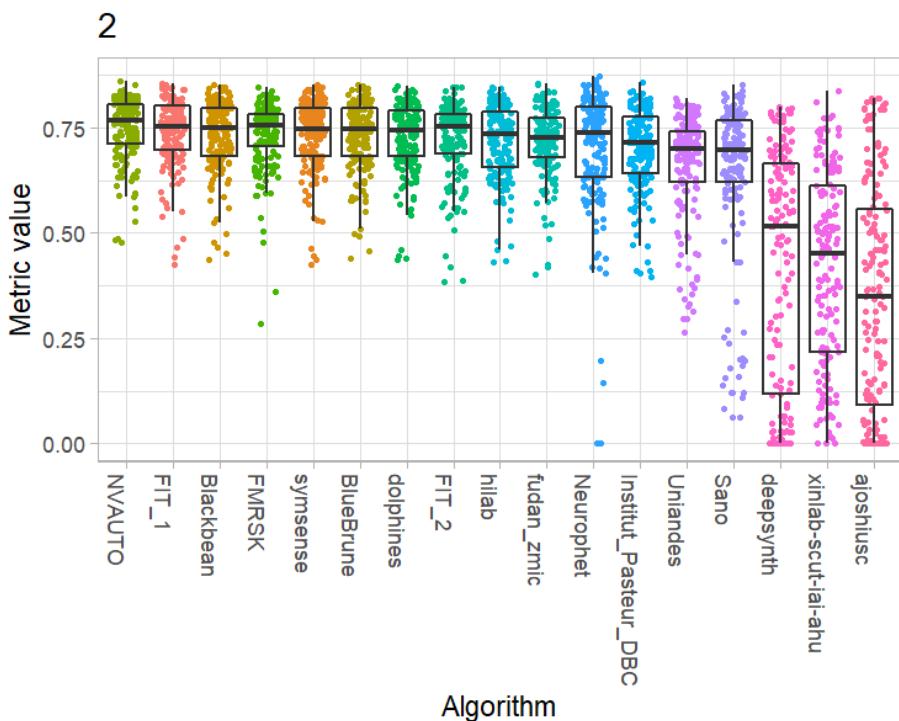
*External Cerebrospinal Fluid*

	Dice_mean	rank
BlueBrune	0.8002536	1
FIT_1	0.8002142	2
symsense	0.7996112	3
Blackbean	0.7975261	4
NVAUTO	0.7942006	5
hilab	0.7940044	6
dolphines	0.7883823	7
fudan_zmic	0.7853473	8
FIT_2	0.7806497	9
FMRSK	0.7754985	10
Institut_Pasteur_DBC	0.7738600	11
Neurophet	0.7552054	12
Uniandes	0.6863355	13
Sano	0.6808355	14
deepsynth	0.5570726	15
ajoshiusc	0.4753010	16
xinlab-scut-iai-ahu	0.4186531	17



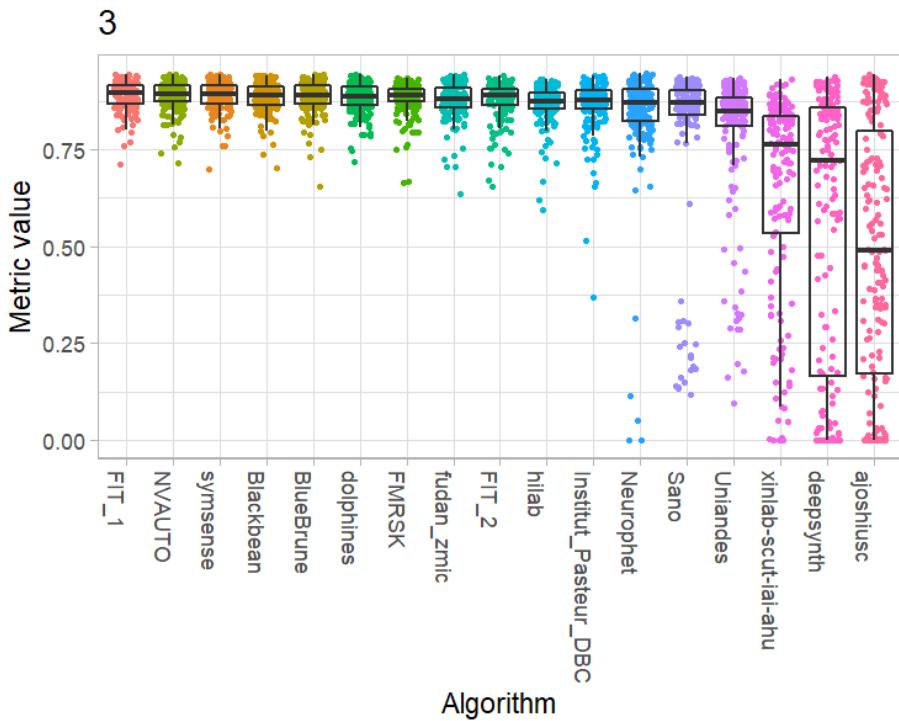
*Grey Matter*

	Dice_mean	rank
NVAUTO	0.7465990	1
FIT_1	0.7390611	2
Blackbean	0.7322551	3
FMRSK	0.7322542	4
symsense	0.7316994	5
BlueBrune	0.7311050	6
dolphines	0.7260992	7
FIT_2	0.7213431	8
hilab	0.7168561	9
fudan_zmic	0.7167084	10
Neurophet	0.6982392	11
Institut_Pasteur_DBC	0.6978033	12
Uniandes	0.6600978	13
Sano	0.6377675	14
deepsynth	0.4188469	15
xinlab-scut-iai-ahu	0.4143335	16
ajoshiusc	0.3460518	17



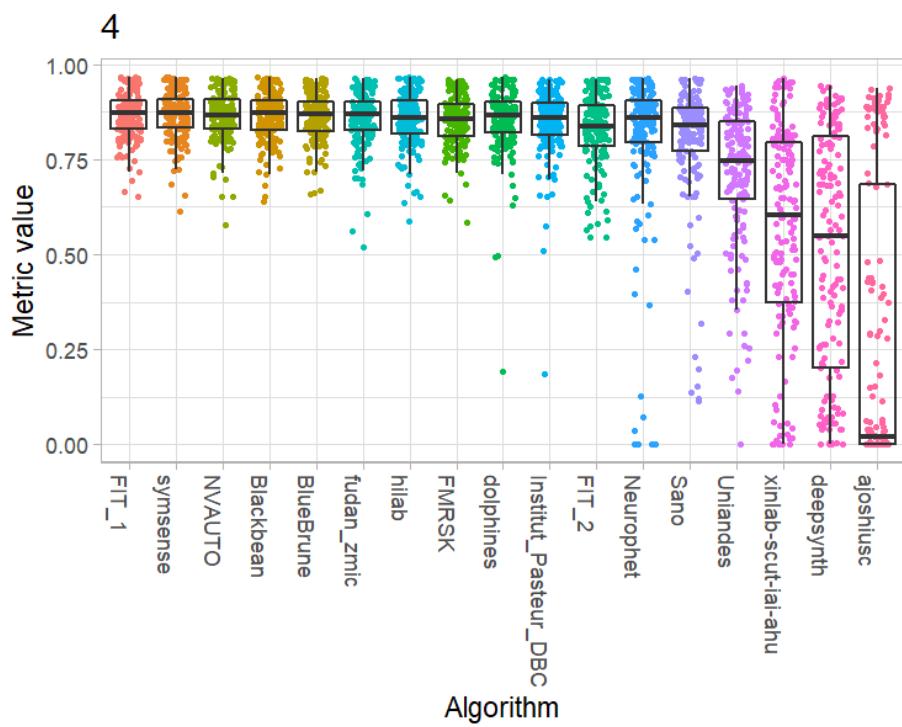
*White Matter*

	Dice_mean	rank
FIT_1	0.8886578	1
NVAUTO	0.8874888	2
symsense	0.8871385	3
Blackbean	0.8861978	4
BlueBrune	0.8851404	5
dolphines	0.8820585	6
FMRSK	0.8813707	7
fudan_zmic	0.8767395	8
FIT_2	0.8756532	9
hilab	0.8665794	10
Institut_Pasteur_DBC	0.8645161	11
Neurophet	0.8396351	12
Sano	0.7976742	13
Uniandes	0.7865351	14
xinlab-scut-iai-ahu	0.6476109	15
deepsynth	0.5587658	16
ajoshiusc	0.4817301	17



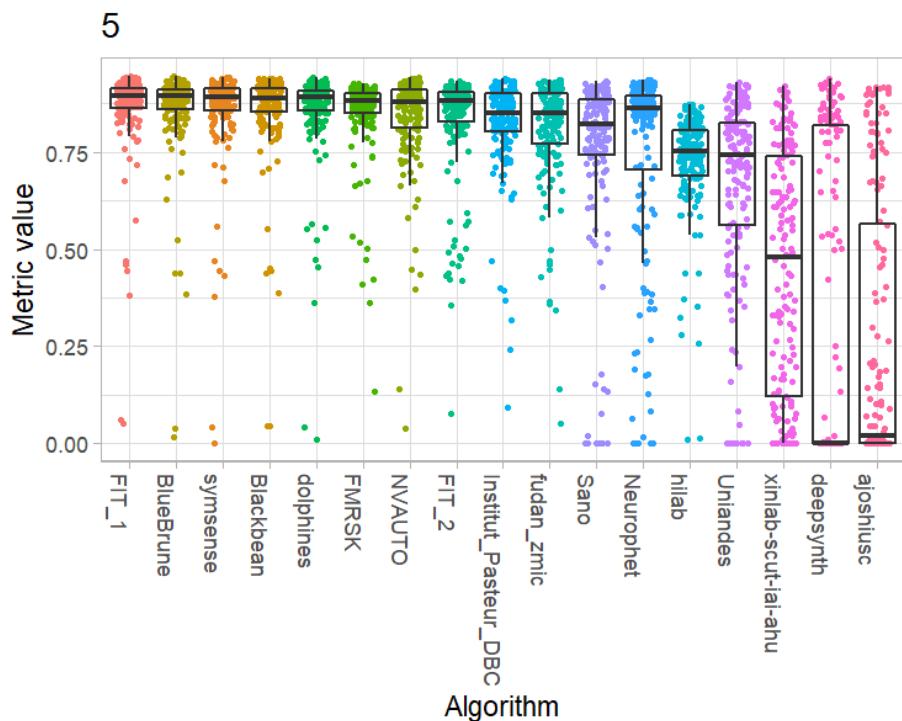
*Ventricles*

	Dice_mean	rank
FIT_1	0.8669385	1
symsense	0.8668682	2
NVAUTO	0.8646941	3
Blackbean	0.8636322	4
BlueBrune	0.8623484	5
fudan_zmic	0.8569926	6
hilab	0.8554609	7
FMRSK	0.8526819	8
dolphines	0.8516589	9
Institut_Pasteur_DBC	0.8479540	10
FIT_2	0.8249273	11
Neurophet	0.8001492	12
Sano	0.7968522	13
Uniandes	0.7078457	14
xinlab-scut-iai-ahu	0.5577158	15
deepsynth	0.5027161	16
ajoshiusc	0.2791750	17



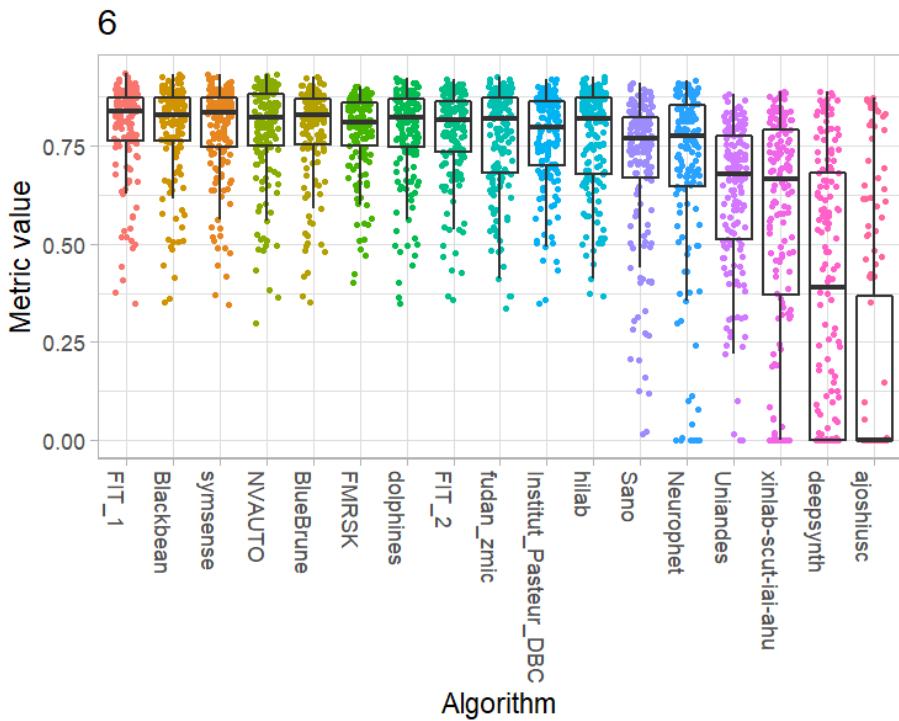
*Cerebellum*

	Dice_mean	rank
FIT_1	0.8616184	1
BlueBrune	0.8596306	2
symsense	0.8580273	3
Blackbean	0.8567526	4
dolphines	0.8547840	5
FMRSK	0.8495308	6
NVAUTO	0.8379790	7
FIT_2	0.8282533	8
Institut_Pasteur_DBC	0.8219894	9
fudan_zmic	0.8066828	10
Sano	0.7431538	11
Neurophet	0.7424658	12
hilab	0.7242887	13
Uniandes	0.6628640	14
xinlab-scut-iai-ahu	0.4366765	15
deepsynth	0.3205607	16
ajoshiusc	0.2580985	17



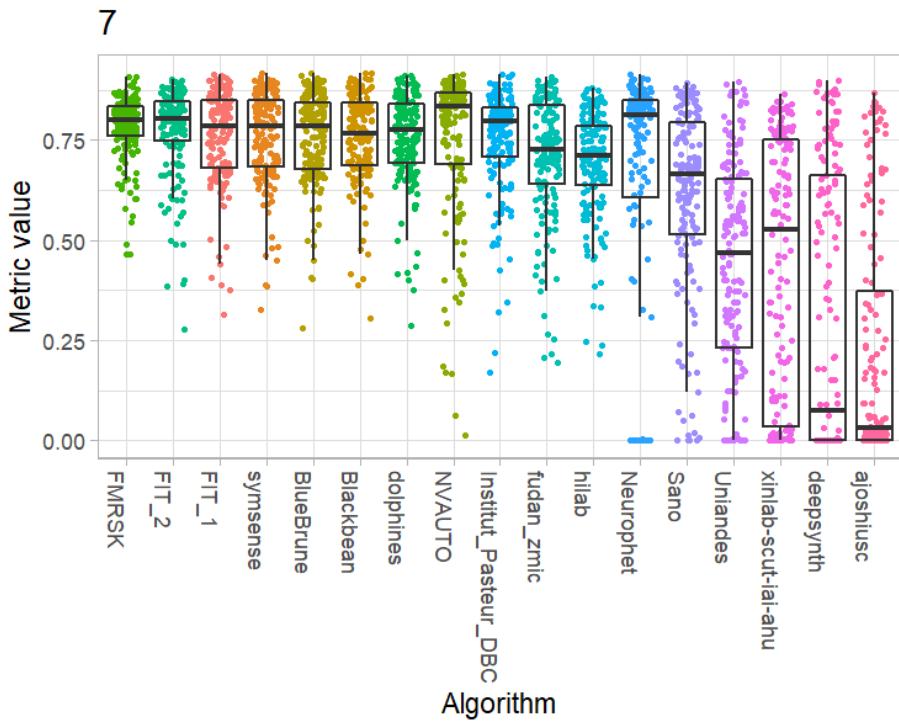
*Deep Grey Matter*

	Dice_mean	rank
FIT_1	0.7970376	1
Blackbean	0.7921152	2
symsense	0.7919593	3
NVAUTO	0.7899541	4
BlueBrune	0.7890816	5
FMRSK	0.7858984	6
dolphines	0.7840985	7
FIT_2	0.7823788	8
fudan_zmic	0.7682912	9
Institut_Pasteur_DBC	0.7655669	10
hilab	0.7654387	11
Sano	0.6987481	12
Neurophet	0.6901062	13
Uniandes	0.6270504	14
xinlab-scut-iai-ahu	0.5547655	15
deepsynth	0.3713239	16
ajoshiusc	0.1719423	17



*Brainstem*

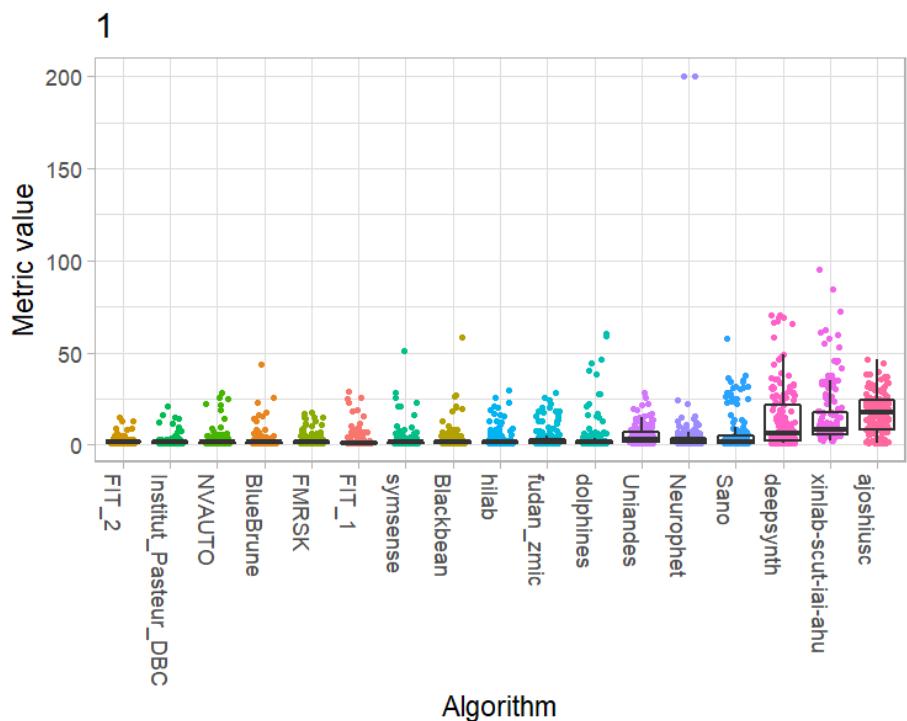
	Dice_mean	rank
FMRSK	0.7804857	1
FIT_2	0.7732864	2
FIT_1	0.7587051	3
symsense	0.7579639	4
BlueBrune	0.7567531	5
Blackbean	0.7557665	6
dolphines	0.7547676	7
NVAUTO	0.7501320	8
Institut_Pasteur_DBC	0.7489135	9
fudan_zmic	0.7066320	10
hilab	0.6922677	11
Neurophet	0.6499254	12
Sano	0.6108436	13
Uniandes	0.4339832	14
xinlab-scut-iai-ahu	0.4288033	15
deepsynth	0.3050998	16
ajoshiusc	0.2172425	17



### 95th percentile Hausdorff Distance

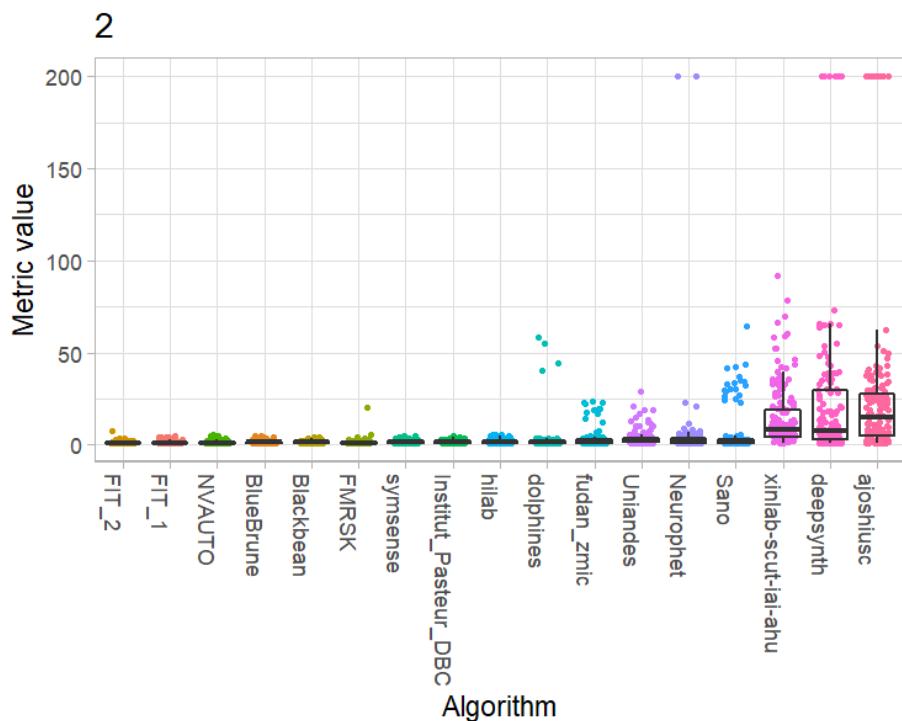
*External Cerebrospinal Fluid*

	Hausdorff_mean	rank
FIT_2	2.007029	1
Institut_Pasteur_DBC	2.310762	2
NVAUTO	2.683908	3
BlueBrune	2.716403	4
FMRSK	2.733759	5
FIT_1	2.840592	6
symsense	3.042505	7
Blackbean	3.128235	8
hilab	3.260556	9
fudan_zmic	3.705524	10
dolphines	4.587188	11
Uniandes	5.265143	12
Neurophet	5.684216	13
Sano	6.540178	14
deepsynth	14.186533	15
xinlab-scut-iai-ahu	15.411957	16
ajoshiusc	16.757040	17



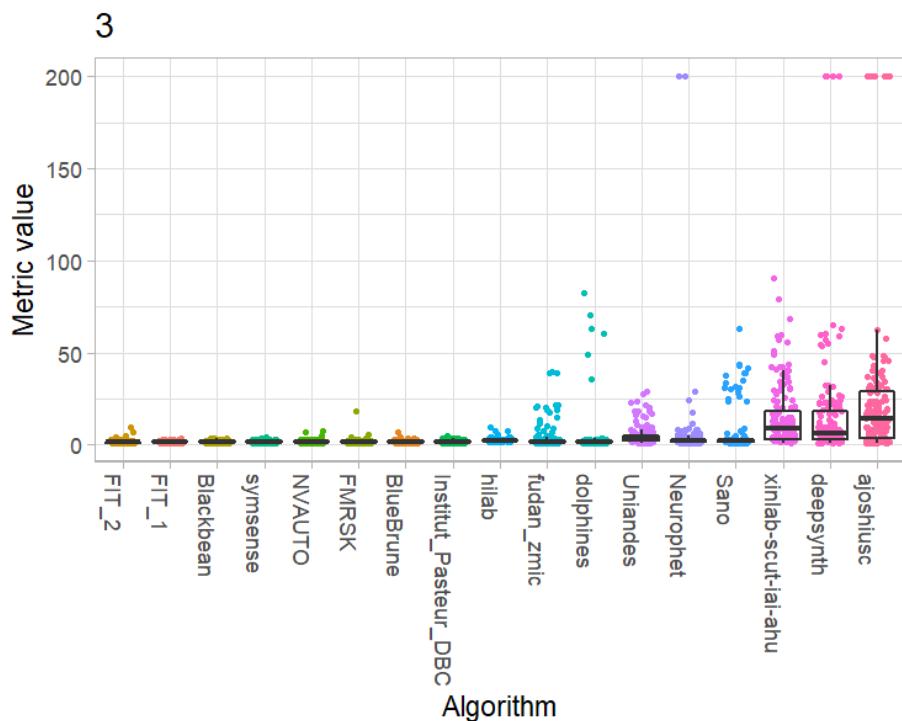
*Grey Matter*

	Hausdorff_mean	rank
FIT_2	1.324447	1
FIT_1	1.466889	2
NVAUTO	1.471739	3
BlueBrune	1.509236	4
Blackbean	1.518263	5
FMRSK	1.522801	6
symsense	1.530787	7
Institut_Pasteur_DBC	1.705699	8
hilab	1.835026	9
dolphines	2.692176	10
fudan_zmic	3.025552	11
Uniandes	3.563289	12
Neurophet	4.982318	13
Sano	5.539338	14
xinlab-scut-iai-ahu	14.806309	15
deepsynth	26.390935	16
ajoshiusc	27.821770	17



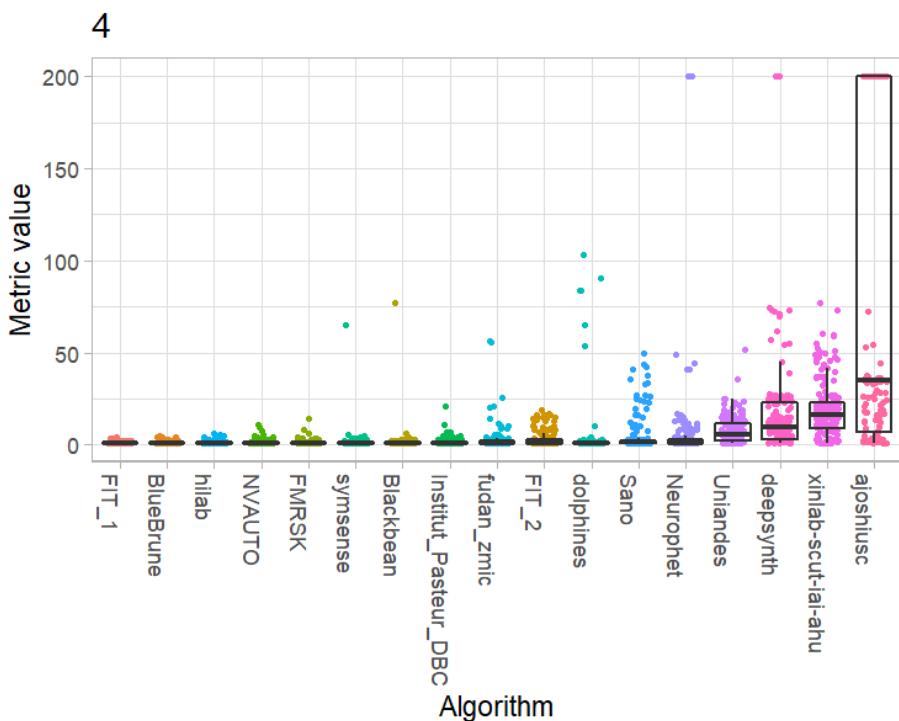
*White Matter*

	Hausdorff_mean	rank
FIT_2	1.663723	1
FIT_1	1.731898	2
Blackbean	1.787248	3
symsense	1.793332	4
NVAUTO	1.826043	5
FMRSK	1.826570	6
BlueBrune	1.833102	7
Institut_Pasteur_DBC	1.936338	8
hilab	2.580569	9
fudan_zmic	3.900536	10
dolphines	3.951597	11
Uniandes	4.757142	12
Neurophet	5.459811	13
Sano	5.925676	14
xinlab-scut-iai-ahu	14.380733	15
deepsynth	17.517663	16
ajoshiusc	29.377441	17



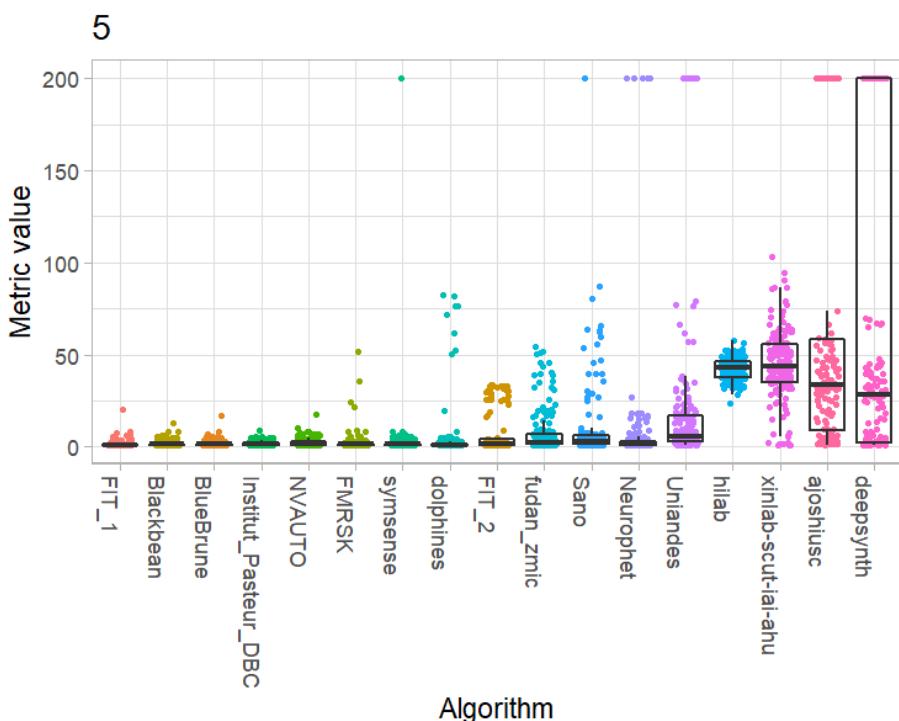
*Ventricles*

	Hausdorff_mean	rank
FIT_1	1.278630	1
BlueBrune	1.355978	2
hilab	1.531157	3
NVAUTO	1.564612	4
FMRSK	1.571398	5
symsense	1.719535	6
Blackbean	1.843204	7
Institut_Pasteur_DBC	1.862322	8
fudan_zmic	3.031148	9
FIT_2	3.423573	10
dolphines	4.324883	11
Sano	5.289352	12
Neurophet	7.347408	13
Uniandes	7.702073	14
deepsynth	16.979665	15
xinlab-scut-iai-ahu	19.254743	16
ajoshiusc	94.309559	17



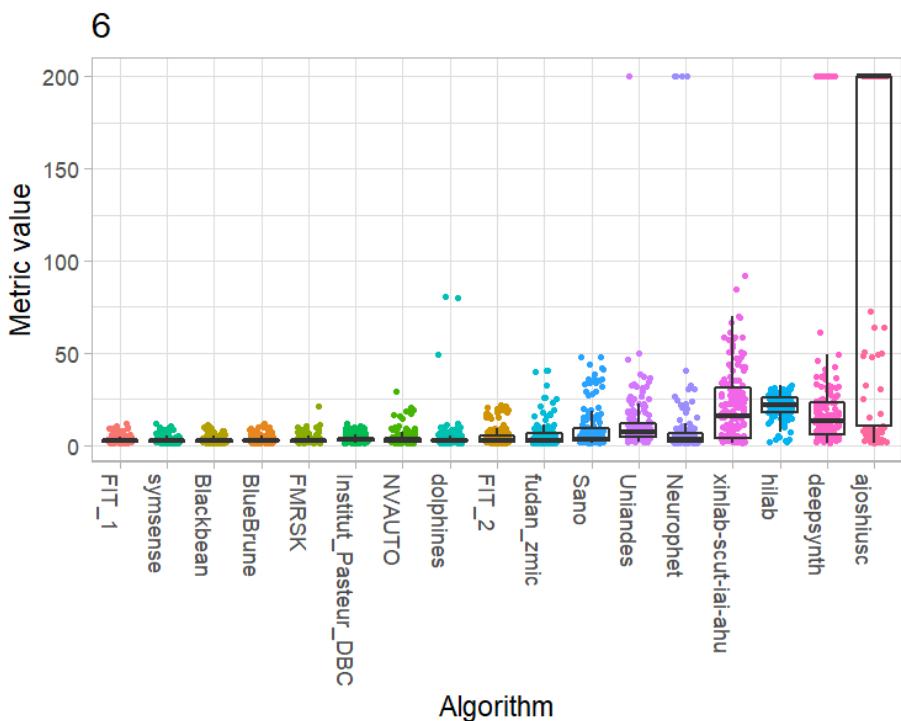
*Cerebellum*

	Hausdorff_mean	rank
FIT_1	1.756305	1
Blackbean	1.828592	2
BlueBrune	1.840775	3
Institut_Pasteur_DBC	1.994657	4
NVAUTO	2.397171	5
FMRSK	2.497374	6
symsense	3.108596	7
dolphines	5.069591	8
FIT_2	7.422129	9
fudan_zmic	7.906044	10
Sano	10.087596	11
Neurophet	10.673282	12
Uniandes	18.745689	13
hilab	42.248560	14
xinlab-scut-iai-ahu	44.732221	15
ajoshiusc	64.650575	16
deepsynth	70.043637	17



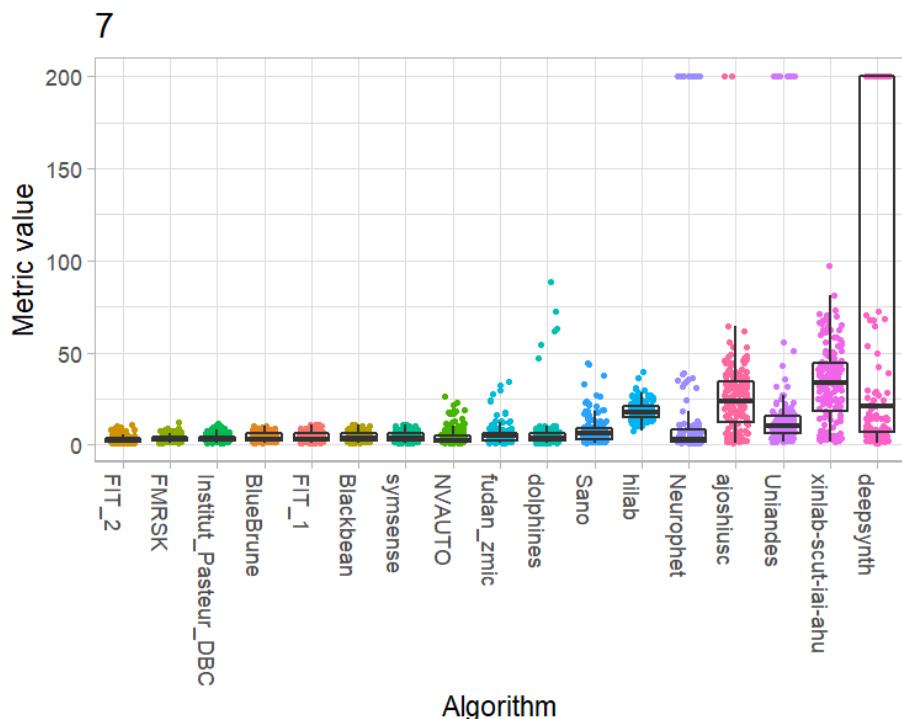
*Deep Grey Matter*

	Hausdorff_mean	rank
FIT_1	3.141916	1
symsense	3.172609	2
Blackbean	3.210812	3
BlueBrune	3.277840	4
FMRSK	3.290733	5
Institut_Pasteur_DBC	3.428397	6
NVAUTO	3.889540	7
dolphines	4.609755	8
FIT_2	5.017189	9
fudan_zmic	5.760197	10
Sano	8.962917	11
Uniandes	11.831873	12
Neurophet	12.841321	13
xinlab-scut-iai-ahu	20.594101	14
hilab	21.215602	15
deepsynth	26.387147	16
ajoshiusc	137.363660	17



*Brainstem*

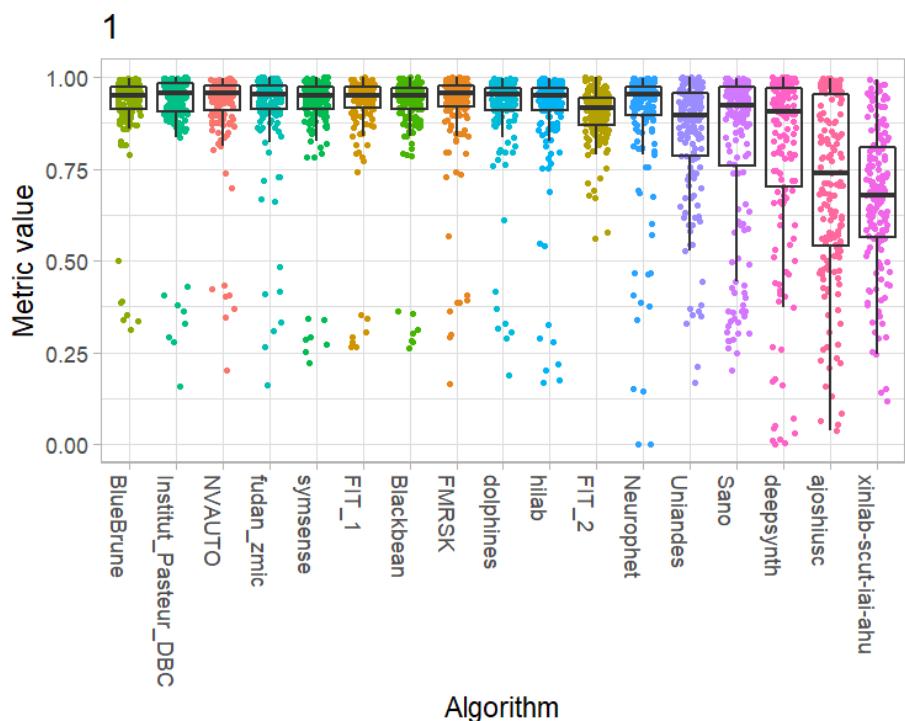
	Hausdorff_mean	rank
FIT_2	3.088323	1
FMRSK	3.319428	2
Institut_Pasteur_DBC	3.469085	3
BlueBrune	4.108818	4
FIT_1	4.209740	5
Blackbean	4.228637	6
symsense	4.255712	7
NVAUTO	4.422046	8
fudan_zmic	5.711710	9
dolphines	6.409475	10
Sano	7.855387	11
hilab	18.384073	12
Neurophet	25.027032	13
ajoshiusc	25.905584	14
Uniandes	27.697947	15
xinlab-scut-iai-ahu	32.868111	16
deepsynth	85.064616	17



### Volume Similarity

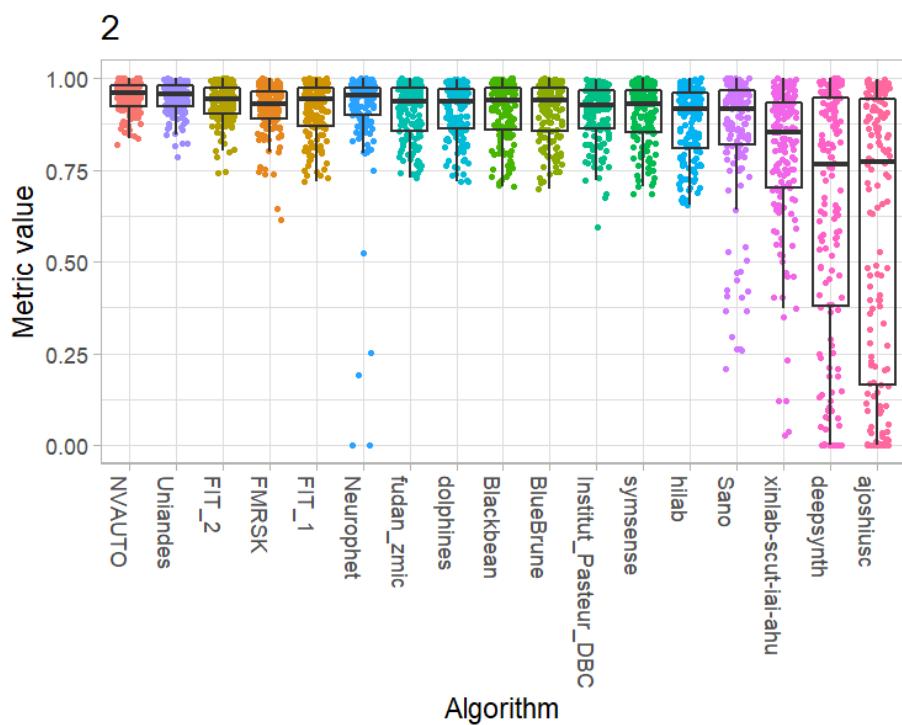
#### *External Cerebrospinal Fluid*

	Volume_Similarity_mean	rank
BlueBrune	0.9181480	1
Institut_Pasteur_DBC	0.9172483	2
NVAUTO	0.9164660	3
fudan_zmic	0.9139644	4
symsense	0.9133753	5
FIT_1	0.9129024	6
Blackbean	0.9123308	7
FMRSK	0.9105632	8
dolphines	0.9101116	9
hilab	0.9001898	10
FIT_2	0.8992279	11
Neurophet	0.8862415	12
Uniandes	0.8406884	13
Sano	0.8112722	14
deepsynth	0.7908622	15
ajoshiusc	0.7049072	16
xinlab-scut-iai-ahu	0.6709563	17



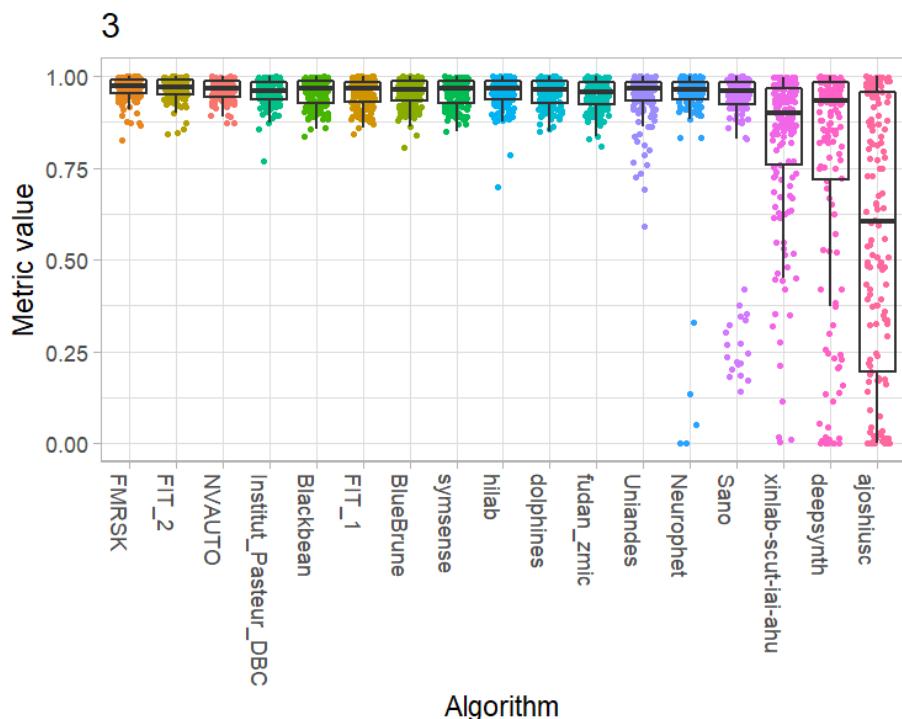
*Grey Matter*

	Volume_Similarity_mean	rank
NVAUTO	0.9493876	1
Uniandes	0.9463811	2
FIT_2	0.9325960	3
FMRSK	0.9185032	4
FIT_1	0.9154207	5
Neurophet	0.9136929	6
fudan_zmic	0.9119098	7
dolphines	0.9114815	8
Blackbean	0.9110063	9
BlueBrune	0.9093676	10
Institut_Pasteur_DBC	0.9041263	11
symsense	0.9035745	12
hilab	0.8811402	13
Sano	0.8521231	14
xinlab-scut-iai-ahu	0.7920461	15
deepsynth	0.6377564	16
ajoshiusc	0.5880742	17



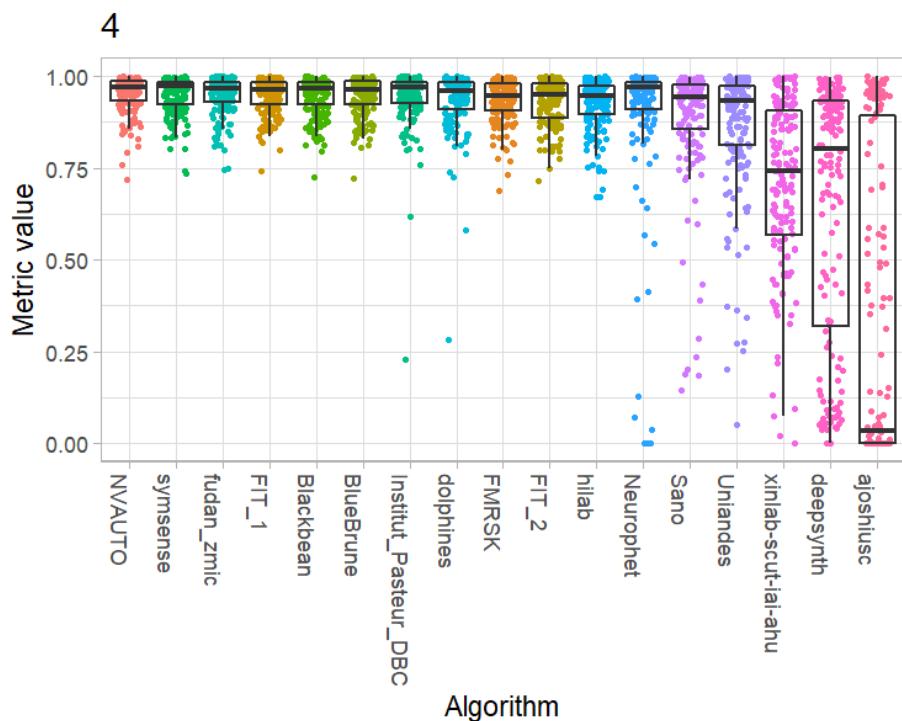
*White Matter*

	Volume_Similarity_mean	rank
FMR SK	0.9677088	1
FIT_2	0.9670448	2
NVAUTO	0.9637872	3
Institut_Pasteur_DBC	0.9571928	4
Blackbean	0.9554196	5
FIT_1	0.9554103	6
BlueBrune	0.9552884	7
symsense	0.9550764	8
hilab	0.9548579	9
dolphines	0.9546843	10
fudan_zmic	0.9484207	11
Uniandes	0.9469805	12
Neurophet	0.9332439	13
Sano	0.8761791	14
xinlab-scut-iai-ahu	0.8232146	15
deepsynth	0.7630108	16
ajoshiusc	0.5630531	17



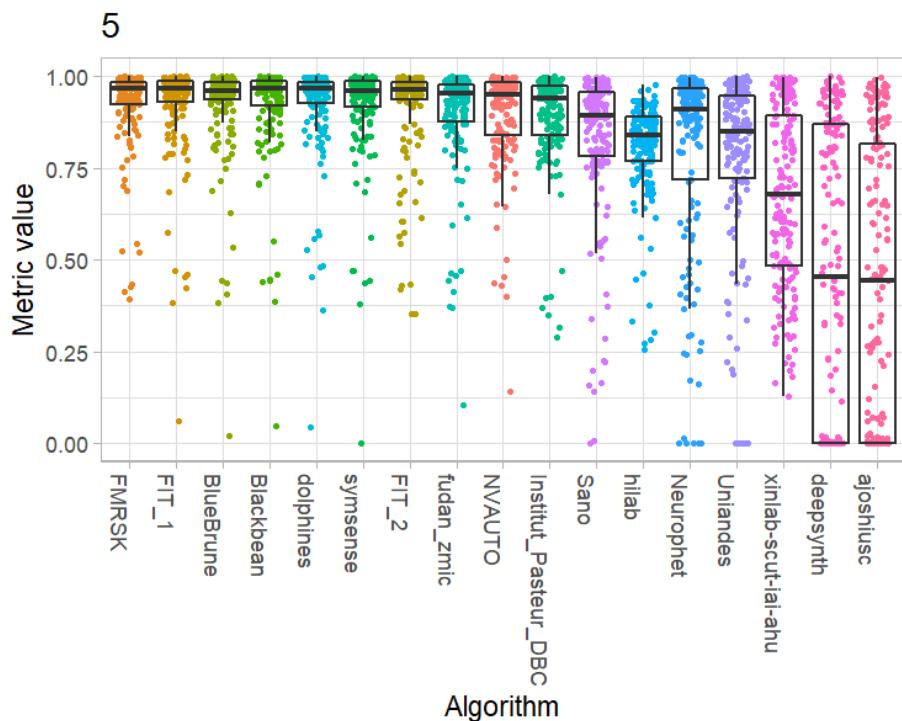
*Ventricles*

	Volume_Similarity_mean	rank
NVAUTO	0.9531792	1
symsense	0.9489486	2
fudan_zmic	0.9485503	3
FIT_1	0.9477089	4
Blackbean	0.9466208	5
BlueBrune	0.9463388	6
Institut_Pasteur_DBC	0.9439894	7
dolphines	0.9360979	8
FMRSK	0.9349780	9
FIT_2	0.9315541	10
hilab	0.9256832	11
Neurophet	0.8885335	12
Sano	0.8813197	13
Uniandes	0.8609946	14
xinlab-scut-iai-ahu	0.7104980	15
deepsynth	0.6432755	16
ajoshiusc	0.3401671	17



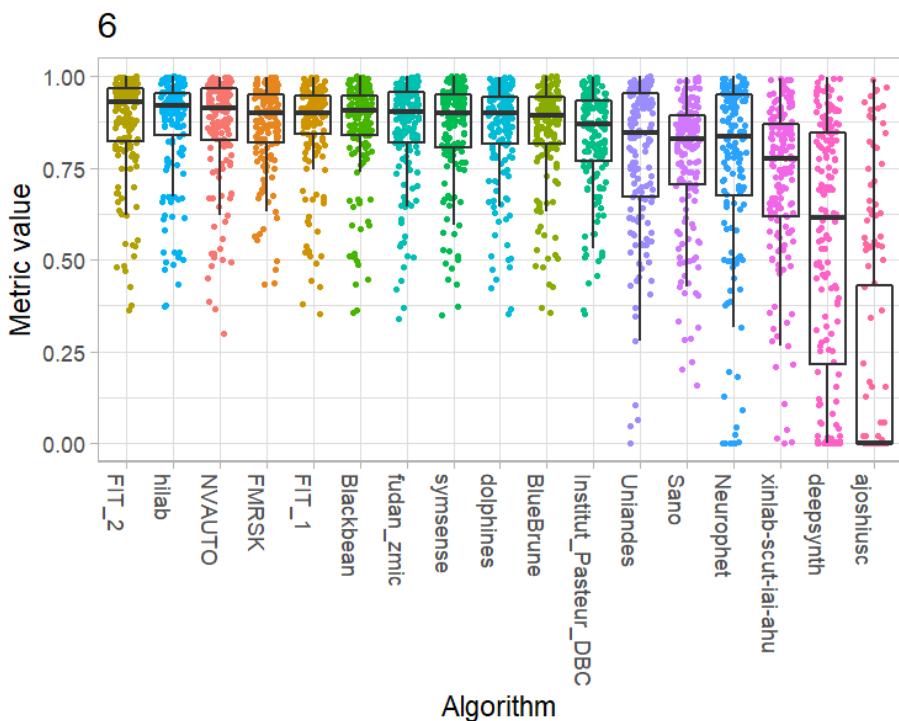
*Cerebellum*

	Volume_Similarity_mean	rank
FMR SK	0.9236694	1
FIT_1	0.9236319	2
BlueBrune	0.9229350	3
Blackbean	0.9219461	4
dolphines	0.9208316	5
symsense	0.9199382	6
FIT_2	0.9129599	7
fudan_zmic	0.8992646	8
NVAUTO	0.8971479	9
Institut_Pasteur_DBC	0.8911547	10
Sano	0.8248674	11
hilab	0.8045768	12
Neurophet	0.7903950	13
Uniandes	0.7812908	14
xinlab-scut-iai-ahu	0.6697599	15
deepsynth	0.4530265	16
ajoshiusc	0.4287538	17



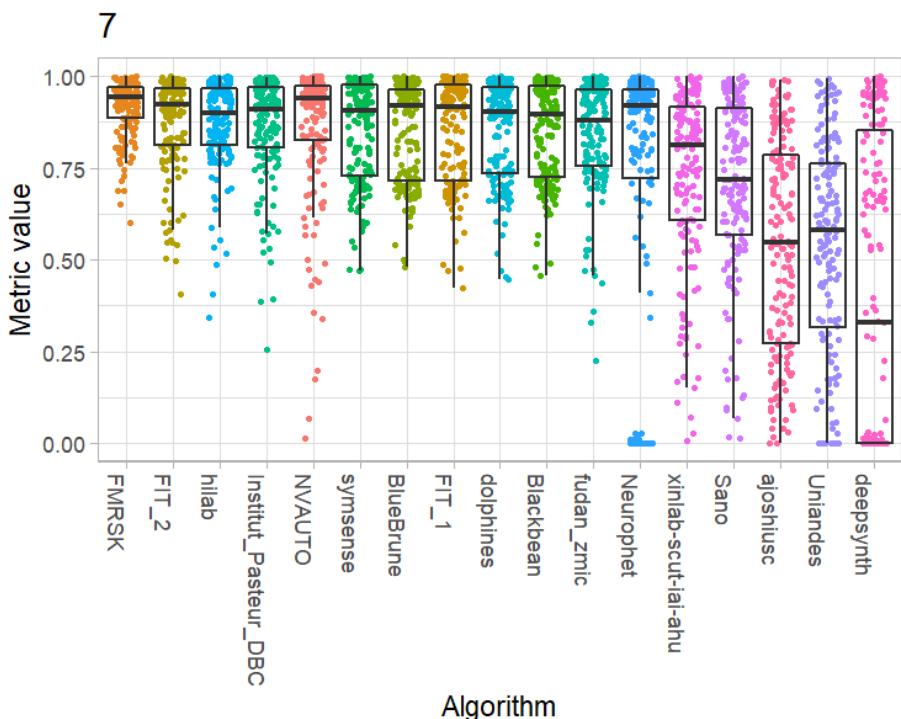
*Deep Grey Matter*

	Volume_Similarity_mean	rank
FIT_2	0.8731193	1
hilab	0.8656961	2
NVAUTO	0.8643086	3
FMRSK	0.8634168	4
FIT_1	0.8627724	5
Blackbean	0.8611927	6
fudan_zmic	0.8563904	7
symsense	0.8556097	8
dolphines	0.8512317	9
BlueBrune	0.8509714	10
Institut_Pasteur_DBC	0.8321411	11
Uniandes	0.7850371	12
Sano	0.7690705	13
Neurophet	0.7530849	14
xinlab-scut-iai-ahu	0.7188615	15
deepsynth	0.5323868	16
ajoshiusc	0.1948204	17



*Brainstem*

	Volume_Similarity_mean	rank
FMR SK	0.9183866	1
FIT_2	0.8764088	2
hilab	0.8734426	3
Institut_Pasteur_DBC	0.8631117	4
NVAUTO	0.8619337	5
symsense	0.8534867	6
BlueBrune	0.8530674	7
FIT_1	0.8527837	8
dolphines	0.8519354	9
Blackbean	0.8515866	10
fudan_zmic	0.8421453	11
Neurophet	0.7394286	12
xinlab-scut-iai-ahu	0.7306282	13
Sano	0.7067502	14
ajoshiusc	0.5384863	15
Uniandes	0.5351554	16
deepsynth	0.4088315	17

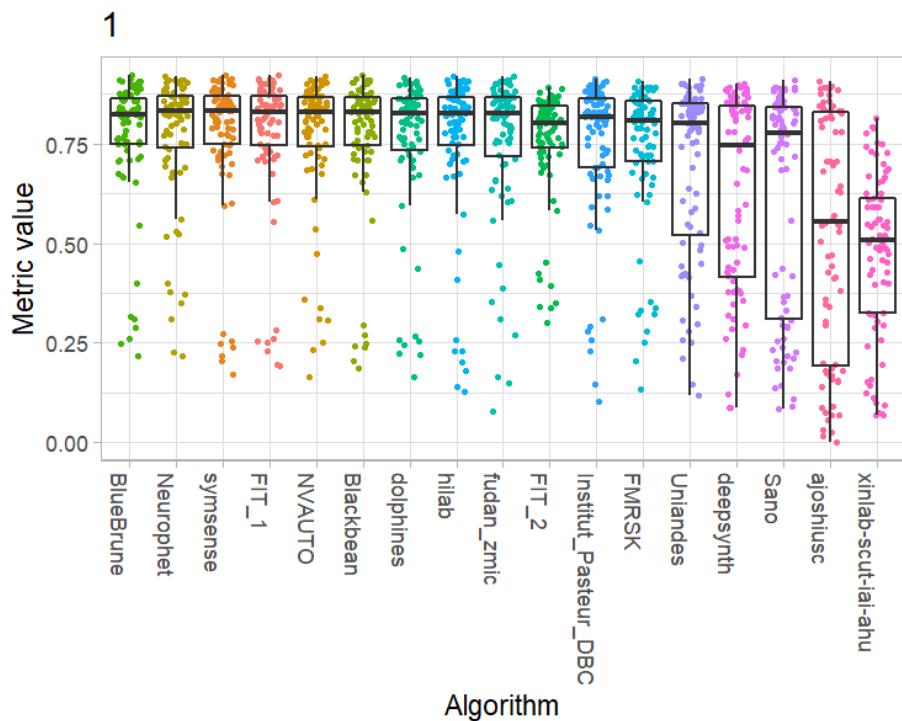


## 12.2 In-Domain Evaluation Metrics per Label

### Dice Similarity Coefficient

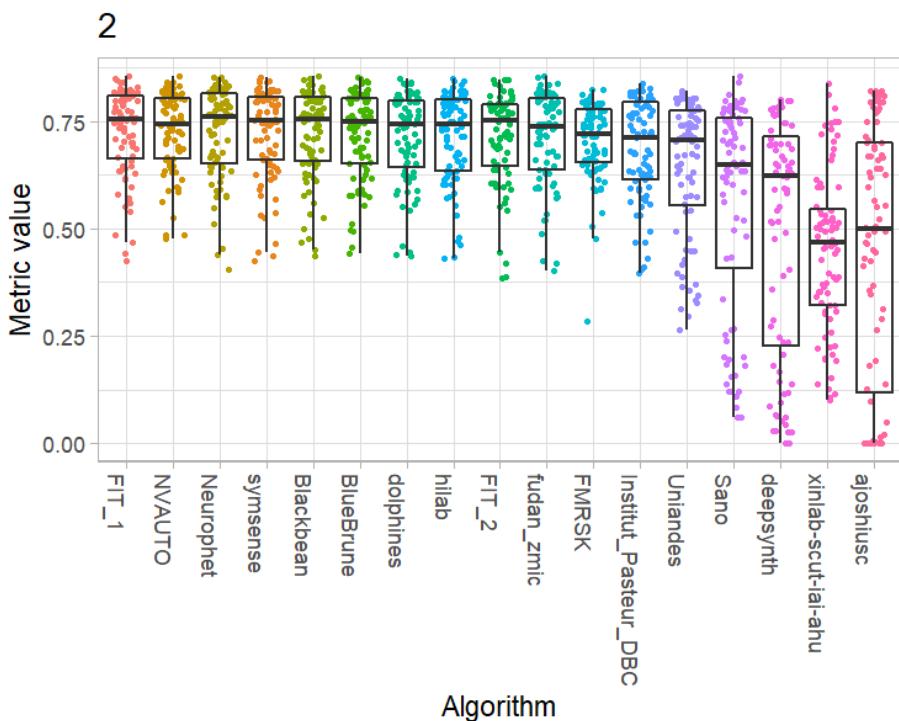
*External Cerebrospinal Fluid*

	Dice_mean	rank
BlueBrune	0.7686780	1
Neurophet	0.7685598	2
symsense	0.7672718	3
FIT_1	0.7666795	4
NVAUTO	0.7659438	5
Blackbean	0.7655063	6
dolphines	0.7562437	7
hilab	0.7548995	8
fudan_zmic	0.7548893	9
FIT_2	0.7528382	10
Institut_Pasteur_DBC	0.7475852	11
FMRSK	0.7357067	12
Uniandes	0.6881726	13
deepsynth	0.6297376	14
Sano	0.6199991	15
ajoshiusc	0.5269903	16
xinlab-scut-iai-ahu	0.4745736	17



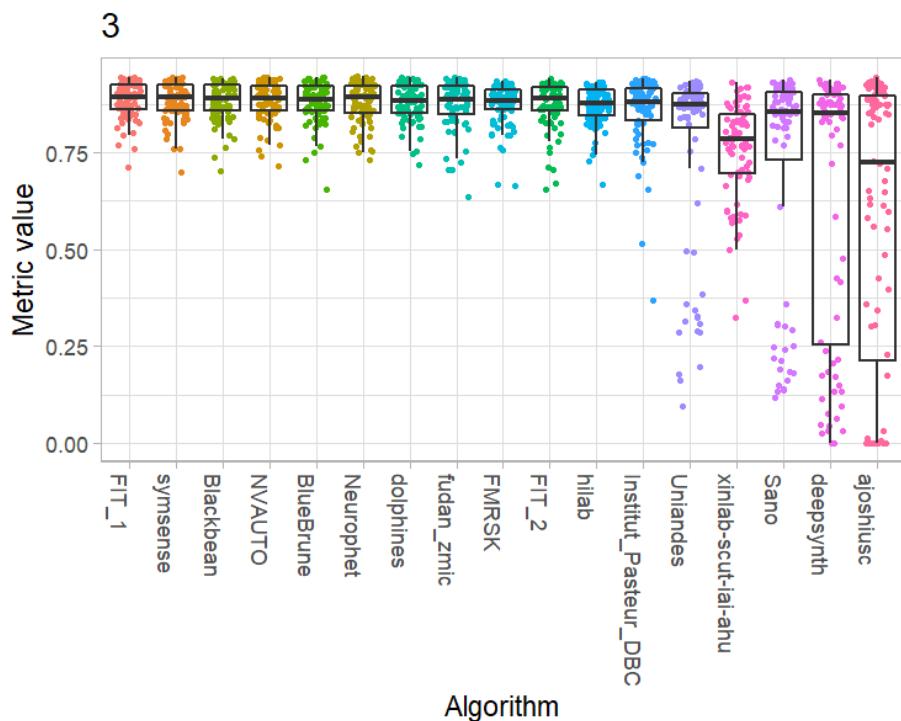
*Grey Matter*

	Dice_mean	rank
FIT_1	0.7263488	1
NVAUTO	0.7249598	2
Neurophet	0.7237017	3
symsense	0.7217628	4
Blackbean	0.7215964	5
BlueBrune	0.7194684	6
dolphines	0.7133646	7
hilab	0.7126315	8
FIT_2	0.7122103	9
fudan_zmic	0.7115305	10
FMRSK	0.7052036	11
Institut_Pasteur_DBC	0.6860686	12
Uniandes	0.6420362	13
Sano	0.5633247	14
deepsynth	0.4930486	15
xinlab-scut-iai-ahu	0.4518581	16
ajoshiusc	0.4414023	17



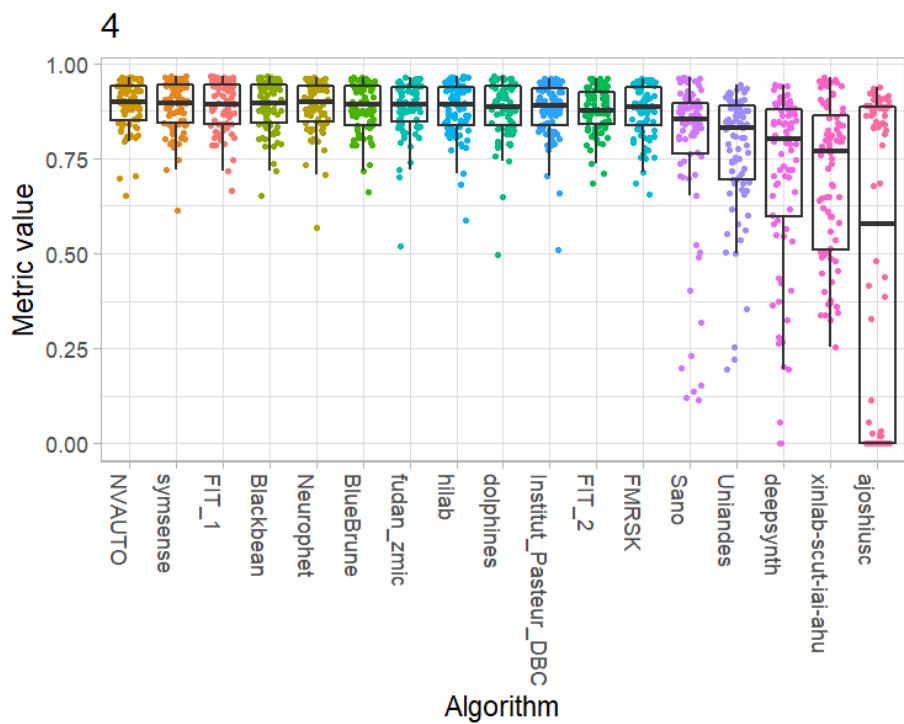
*White Matter*

	Dice_mean	rank
FIT_1	0.8845329	1
symsense	0.8830143	2
Blackbean	0.8821324	3
NVAUTO	0.8818510	4
BlueBrune	0.8799348	5
Neurophet	0.8783639	6
dolphines	0.8783167	7
fudan_zmic	0.8740668	8
FMRSK	0.8734803	9
FIT_2	0.8734289	10
hilab	0.8695814	11
Institut_Pasteur_DBC	0.8572237	12
Uniandes	0.7630115	13
xinlab-scut-iai-ahu	0.7581536	14
Sano	0.7177692	15
deepsynth	0.6507174	16
ajoshiusc	0.5778724	17



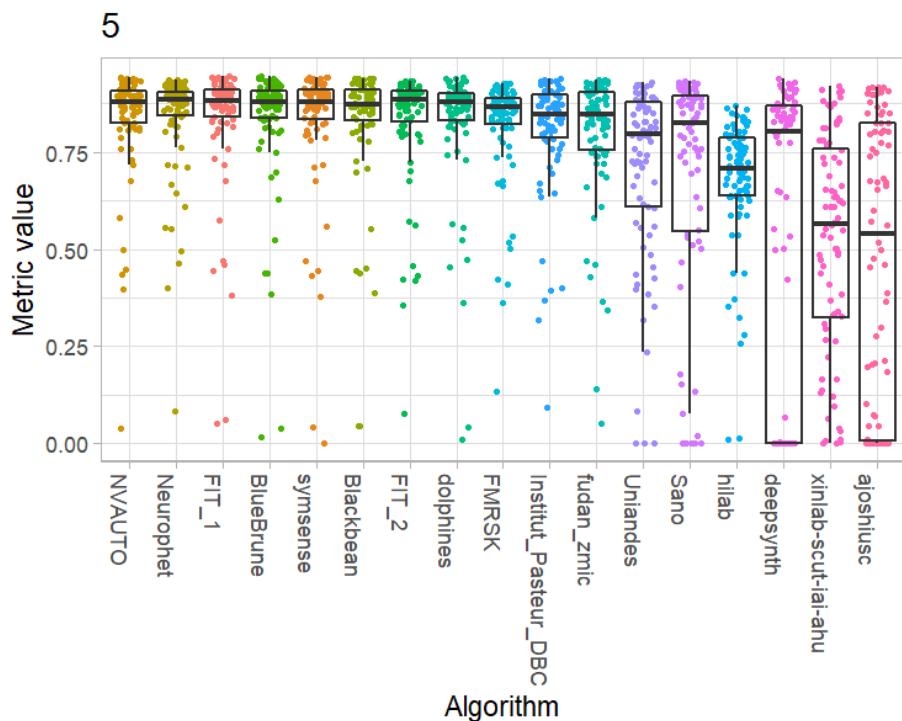
*Ventricles*

	Dice_mean	rank
NVAUTO	0.8892620	1
symsense	0.8890340	2
FIT_1	0.8879099	3
Blackbean	0.8874693	4
Neurophet	0.8869914	5
BlueBrune	0.8850628	6
fudan_zmic	0.8840325	7
hilab	0.8810722	8
dolphines	0.8801130	9
Institut_Pasteur_DBC	0.8779373	10
FIT_2	0.8767312	11
FMRSK	0.8756877	12
Sano	0.7802861	13
Uniandes	0.7702485	14
deepsynth	0.7048879	15
xinlab-scut-iai-ahu	0.7005004	16
ajoshiusc	0.4640130	17



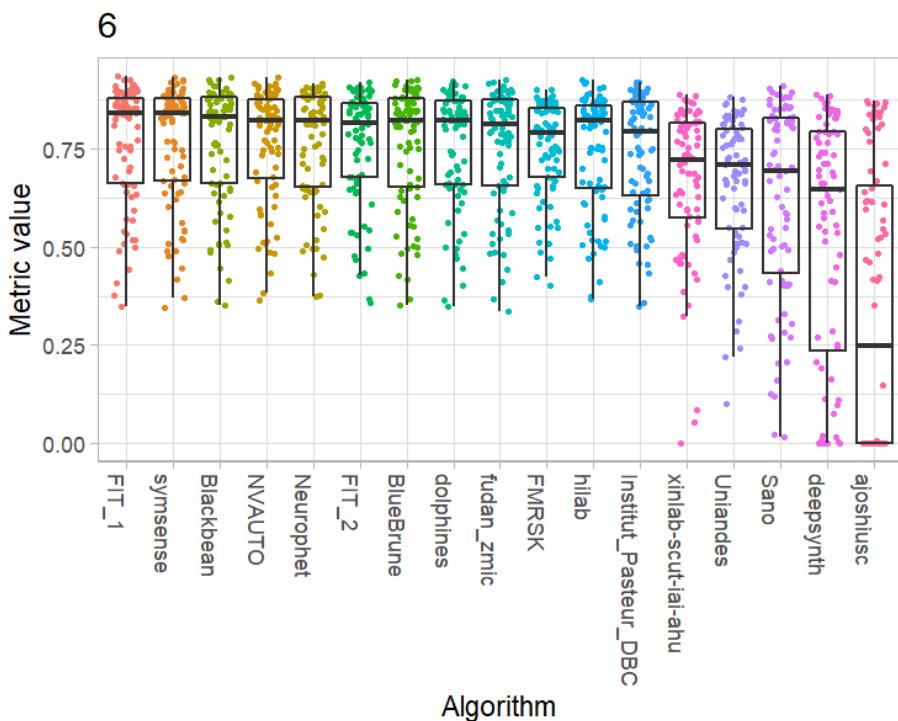
*Cerebellum*

	Dice_mean	rank
NVAUTO	0.8354476	1
Neurophet	0.8347703	2
FIT_1	0.8315482	3
BlueBrune	0.8291744	4
symsense	0.8289905	5
Blackbean	0.8272752	6
FIT_2	0.8270983	7
dolphines	0.8261006	8
FMRSK	0.8179780	9
Institut_Pasteur_DBC	0.8069207	10
fudan_zmic	0.7959490	11
Uniandes	0.7080635	12
Sano	0.6809247	13
hilab	0.6729231	14
deepsynth	0.5425873	15
xinlab-scut-iai-ahu	0.5172309	16
ajoshiusc	0.4643438	17



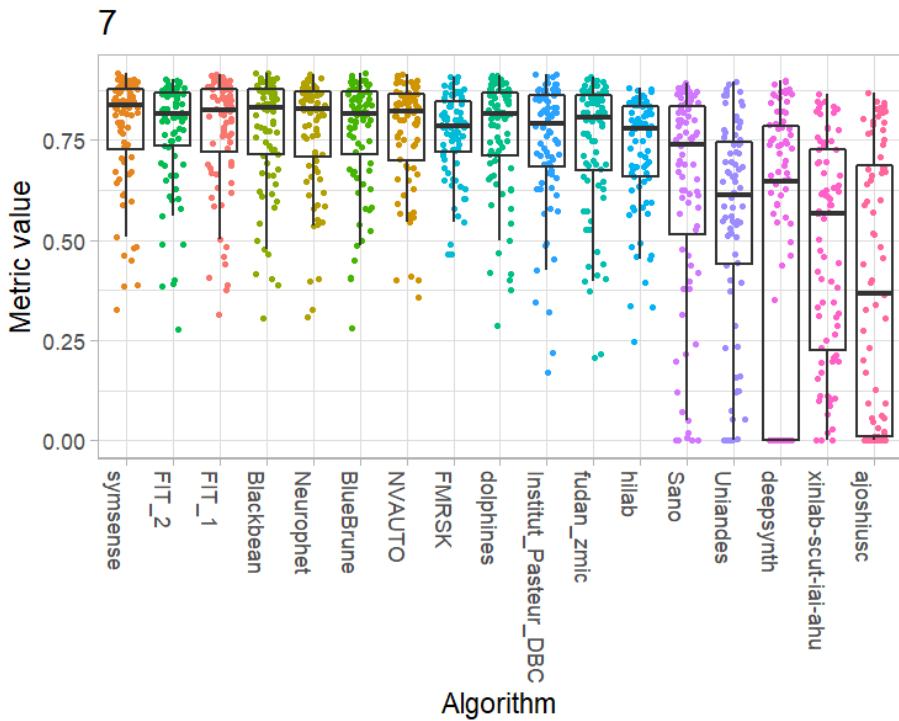
*Deep Grey Matter*

	Dice_mean	rank
FIT_1	0.7672281	1
symsense	0.7653231	2
Blackbean	0.7647831	3
NVAUTO	0.7626383	4
Neurophet	0.7610775	5
FIT_2	0.7605443	6
BlueBrune	0.7591993	7
dolphines	0.7555571	8
fudan_zmic	0.7553751	9
FMRSK	0.7502340	10
hilab	0.7484796	11
Institut_Pasteur_DBC	0.7413080	12
xinlab-scut-iai-ahu	0.6673670	13
Uniandes	0.6620491	14
Sano	0.6197778	15
deepsynth	0.5283210	16
ajoshiusc	0.3363208	17



*Brainstem*

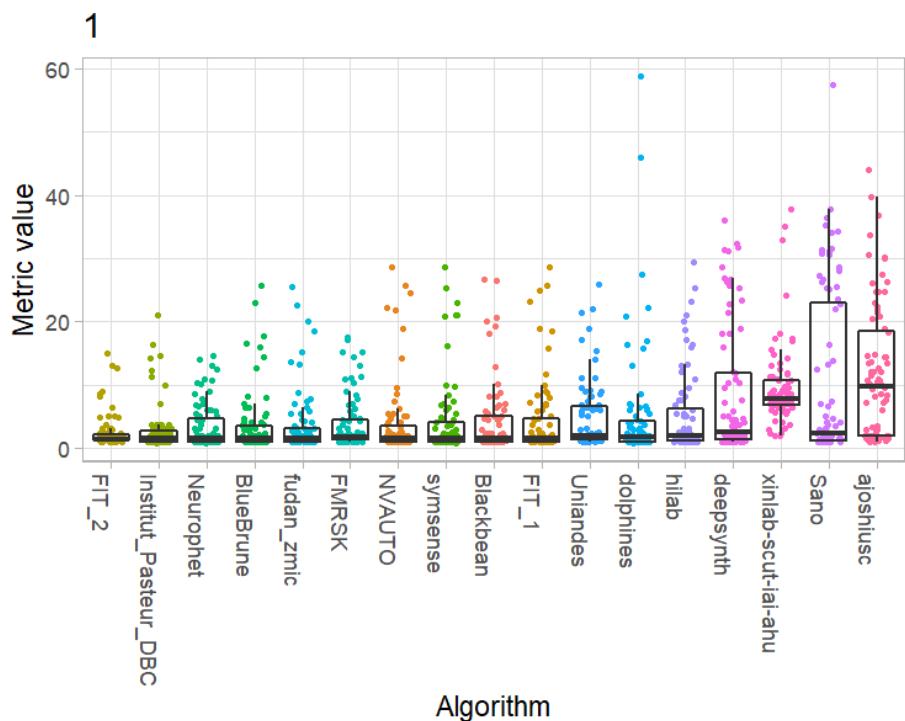
	Dice_mean	rank
symsense	0.7773051	1
FIT_2	0.7726848	2
FIT_1	0.7723850	3
Blackbean	0.7717269	4
Neurophet	0.7710532	5
BlueBrune	0.7696722	6
NVAUTO	0.7690416	7
FMR SK	0.7633371	8
dolphines	0.7626881	9
Institut_Pasteur_DBC	0.7419974	10
fudan_zmic	0.7395436	11
hilab	0.7191834	12
Sano	0.6230278	13
Uniandes	0.5515445	14
deepsynth	0.4885560	15
xinlab-scut-iai-ahu	0.4776047	16
ajoshiusc	0.3762880	17



### 95th percentile Hausdorff Distance

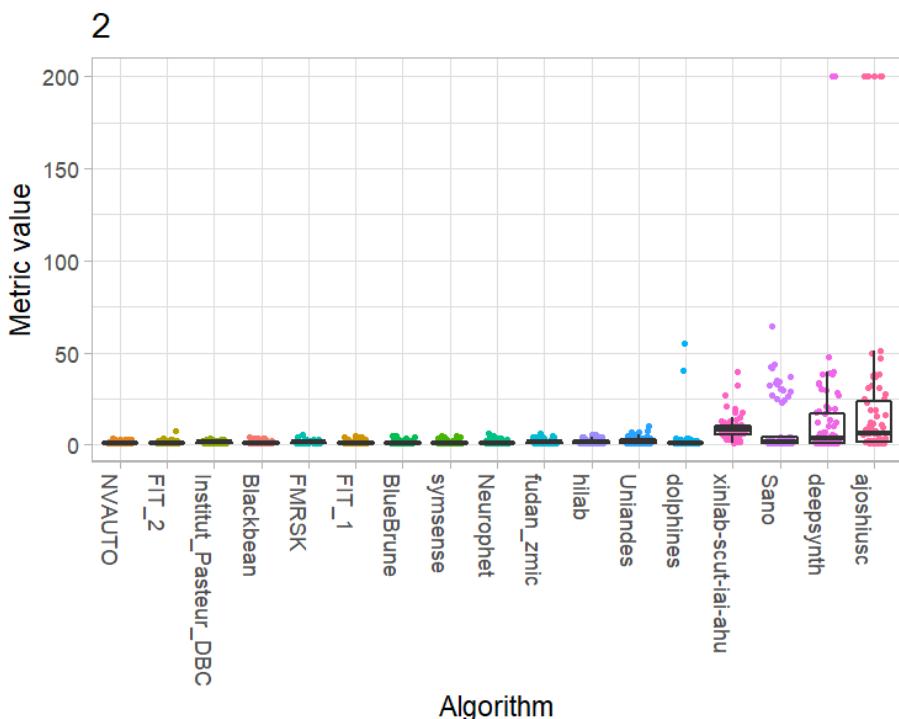
*External Cerebrospinal Fluid*

	Hausdorff_mean	rank
FIT_2	2.550676	1
Institut_Pasteur_DBC	2.897667	2
Neurophet	3.337199	3
BlueBrune	3.597053	4
fudan_zmic	3.642080	5
FMRSK	3.877877	6
NVAUTO	3.955686	7
symsense	4.144267	8
Blackbean	4.175199	9
FIT_1	4.366167	10
Uniandes	4.826249	11
dolphines	5.069797	12
hilab	5.079362	13
deepsynth	8.299171	14
xinlab-scut-iai-ahu	9.569984	15
Sano	10.509291	16
ajoshiusc	11.549493	17



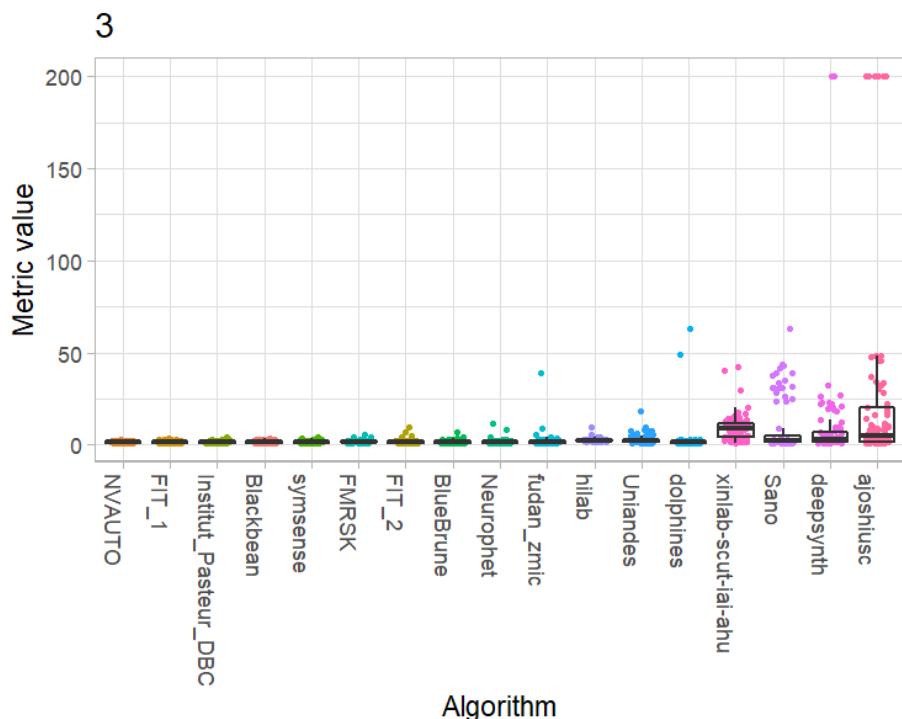
*Grey Matter*

	Hausdorff_mean	rank
NVAUTO	1.363050	1
FIT_2	1.412666	2
Institut_Pasteur_DBC	1.458434	3
Blackbean	1.463699	4
FMRSK	1.464376	5
FIT_1	1.466031	6
BlueBrune	1.514074	7
symsense	1.514167	8
Neurophet	1.562734	9
fudan_zmic	1.663988	10
hilab	1.811408	11
Uniandes	2.363587	12
dolphines	2.566244	13
xinlab-scut-iai-ahu	9.023032	14
Sano	9.271346	15
deepsynth	14.372962	16
ajoshiusc	24.888738	17



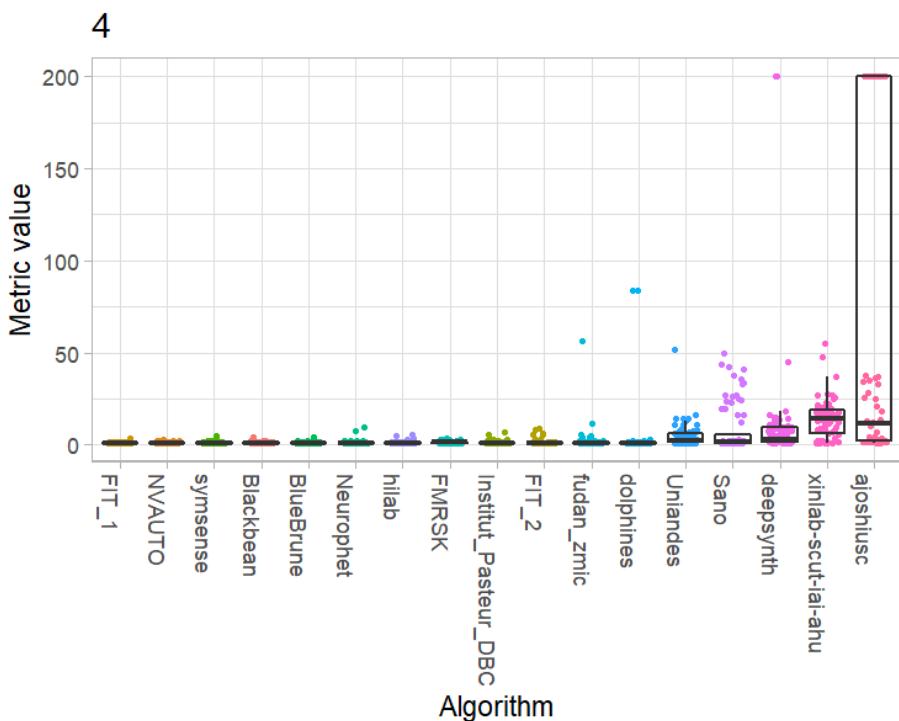
*White Matter*

	Hausdorff_mean	rank
NVAUTO	1.598410	1
FIT_1	1.637384	2
Institut_Pasteur_DBC	1.677778	3
Blackbean	1.703644	4
symsense	1.735831	5
FMRSK	1.759075	6
FIT_2	1.763586	7
BlueBrune	1.835771	8
Neurophet	1.893040	9
fudan_zmic	2.300537	10
hilab	2.609709	11
Uniandes	3.052682	12
dolphines	3.052907	13
xinlab-scut-iai-ahu	9.003149	14
Sano	9.609853	15
deepsynth	11.123079	16
ajoshiusc	29.318986	17



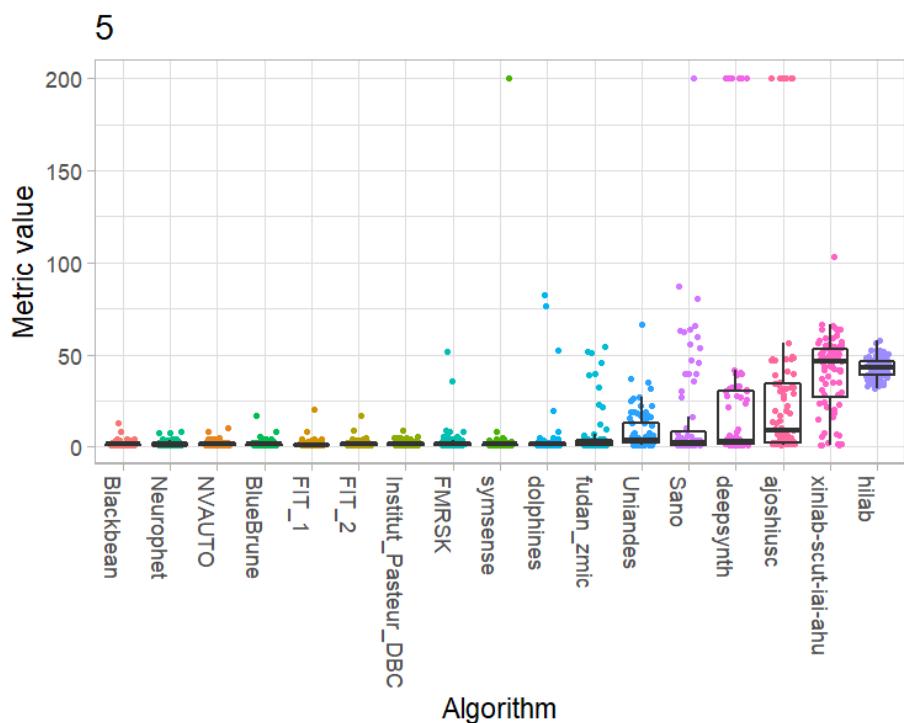
*Ventricles*

	Hausdorff_mean	rank
FIT_1	1.160982	1
NVAUTO	1.161848	2
symsense	1.182994	3
Blackbean	1.197736	4
BlueBrune	1.203656	5
Neurophet	1.287276	6
hilab	1.338571	7
FMRSK	1.347198	8
Institut_Pasteur_DBC	1.361136	9
FIT_2	1.516389	10
fudan_zmic	2.171975	11
dolphines	3.237673	12
Uniandes	4.589975	13
Sano	8.203408	14
deepsynth	10.833369	15
xinlab-scut-iai-ahu	13.851894	16
ajoshiusc	70.784358	17



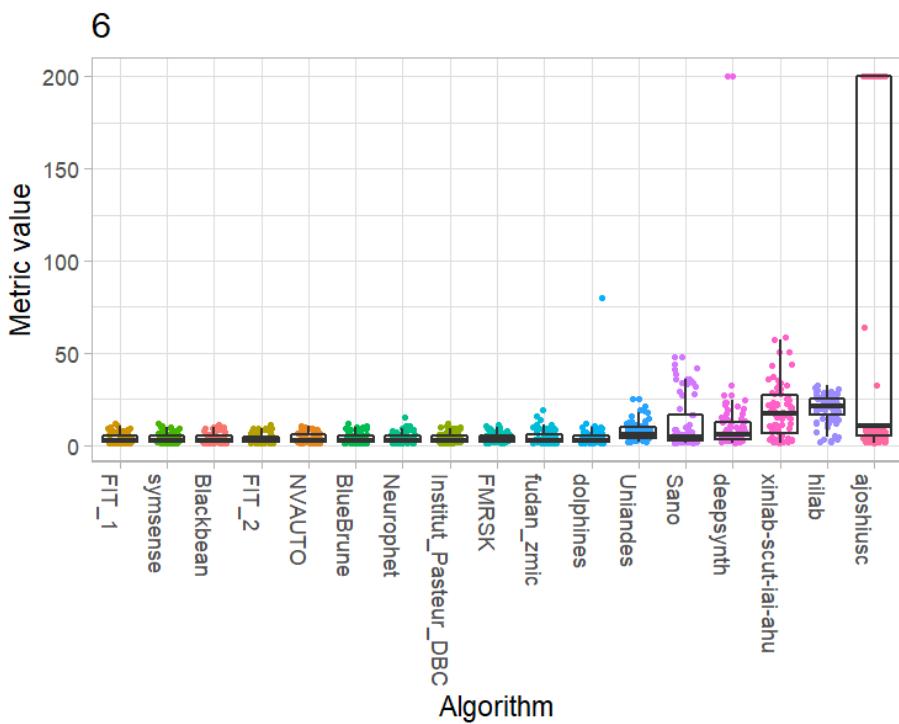
*Cerebellum*

	Hausdorff_mean	rank
Blackbean	1.744056	1
Neurophet	1.747662	2
NVAUTO	1.773582	3
BlueBrune	1.786421	4
FIT_1	1.786947	5
FIT_2	1.864141	6
Institut_Pasteur_DBC	1.952694	7
FMRSK	3.102060	8
symsense	4.092692	9
dolphines	4.452882	10
fudan_zmic	6.458242	11
Uniandes	8.678721	12
Sano	15.465972	13
deepsynth	31.286949	14
ajoshiusc	34.073063	15
xinlab-scut-iai-ahu	39.960761	16
hilab	43.174712	17



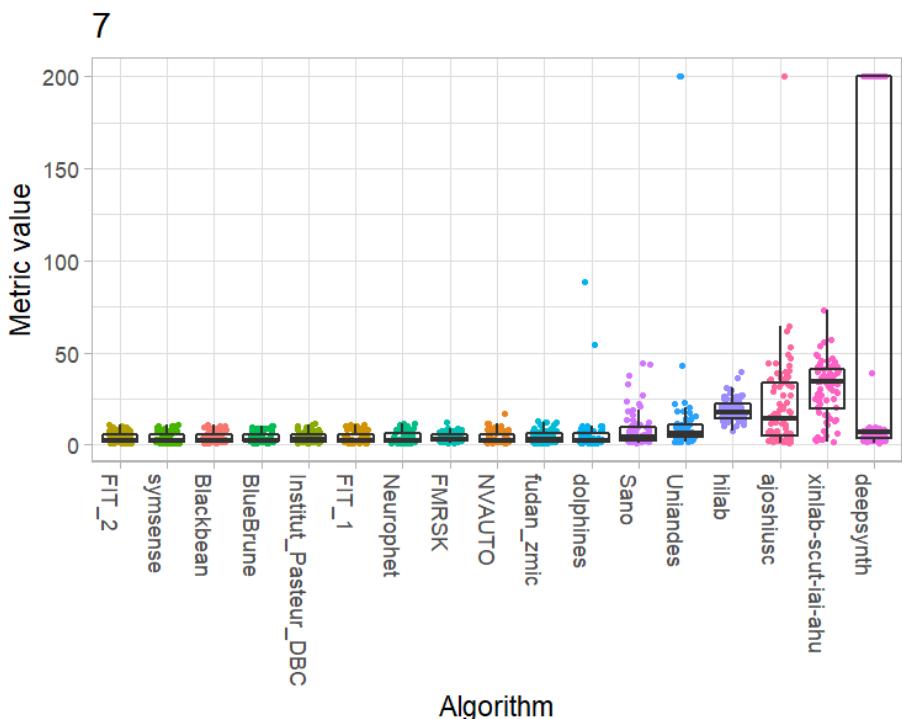
*Deep Grey Matter*

	Hausdorff_mean	rank
FIT_1	3.527885	1
symsense	3.533101	2
Blackbean	3.545642	3
FIT_2	3.567668	4
NVAUTO	3.610600	5
BlueBrune	3.686621	6
Neurophet	3.707675	7
Institut_Pasteur_DBC	3.786003	8
FMRSK	3.870988	9
fudan_zmic	4.373367	10
dolphines	4.717370	11
Uniandes	7.442582	12
Sano	11.782041	13
deepsynth	13.463711	14
xinlab-scut-iai-ahu	18.490138	15
hilab	20.111739	16
ajoshiusc	96.340616	17



*Brainstem*

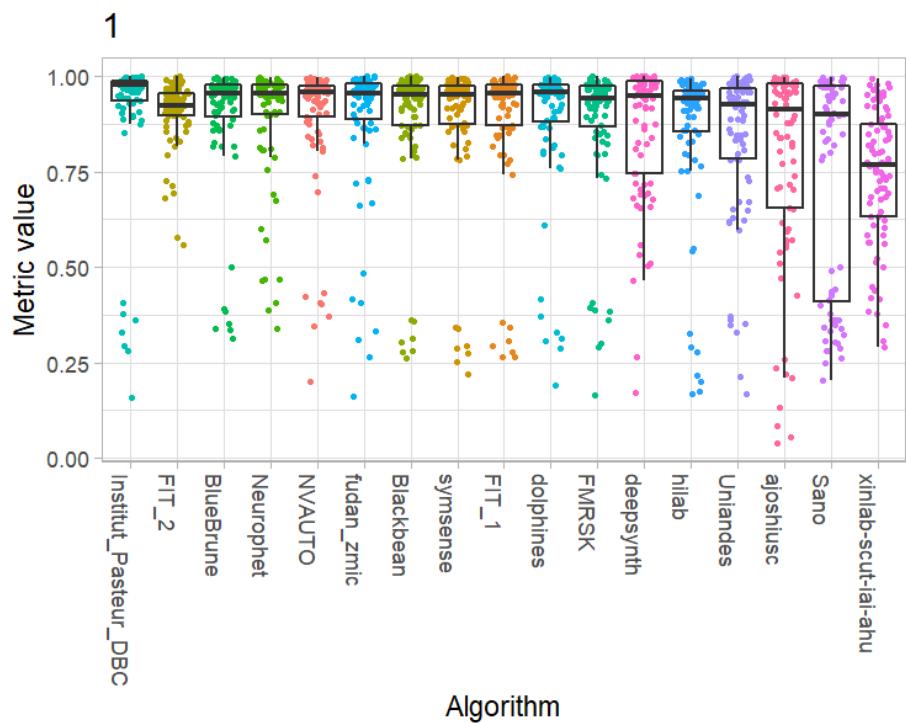
	Hausdorff_mean	rank
FIT_2	3.496845	1
symsense	3.596271	2
Blackbean	3.658285	3
BlueBrune	3.685579	4
Institut_Pasteur_DBC	3.694259	5
FIT_1	3.699724	6
Neurophet	3.736987	7
FMRSK	3.748431	8
NVAUTO	3.782899	9
fudan_zmic	4.034180	10
dolphines	5.542392	11
Sano	7.951822	12
Uniandes	12.720568	13
hilab	18.872729	14
ajoshiusc	21.271864	15
xinlab-scut-iai-ahu	30.040479	16
deepsynth	66.208206	17



### Volume Similarity

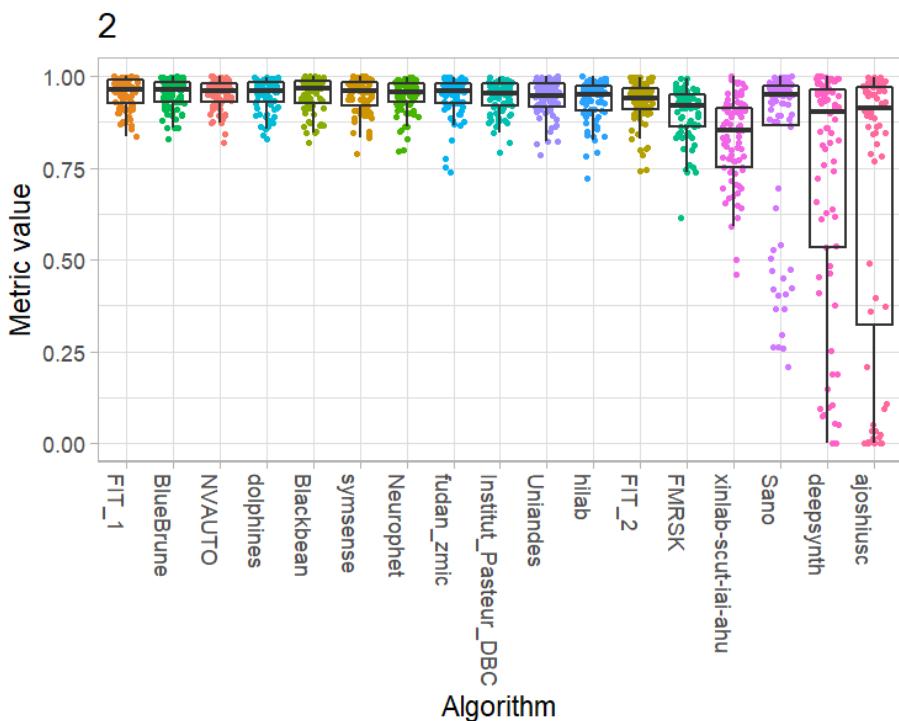
#### *External Cerebrospinal Fluid*

	Volume_Similarity_mean	rank
Institut_Pasteur_DBC	0.9085632	1
FIT_2	0.9065663	2
BlueBrune	0.8921795	3
Neurophet	0.8893738	4
NVAUTO	0.8877628	5
fudan_zmic	0.8828154	6
Blackbean	0.8822232	7
symsense	0.8810547	8
FIT_1	0.8807766	9
dolphines	0.8803072	10
FMRSK	0.8704519	11
deepsynth	0.8622139	12
hilab	0.8538392	13
Uniandes	0.8353976	14
ajoshiusc	0.7865949	15
Sano	0.7465081	16
xinlab-scut-iai-ahu	0.7392403	17



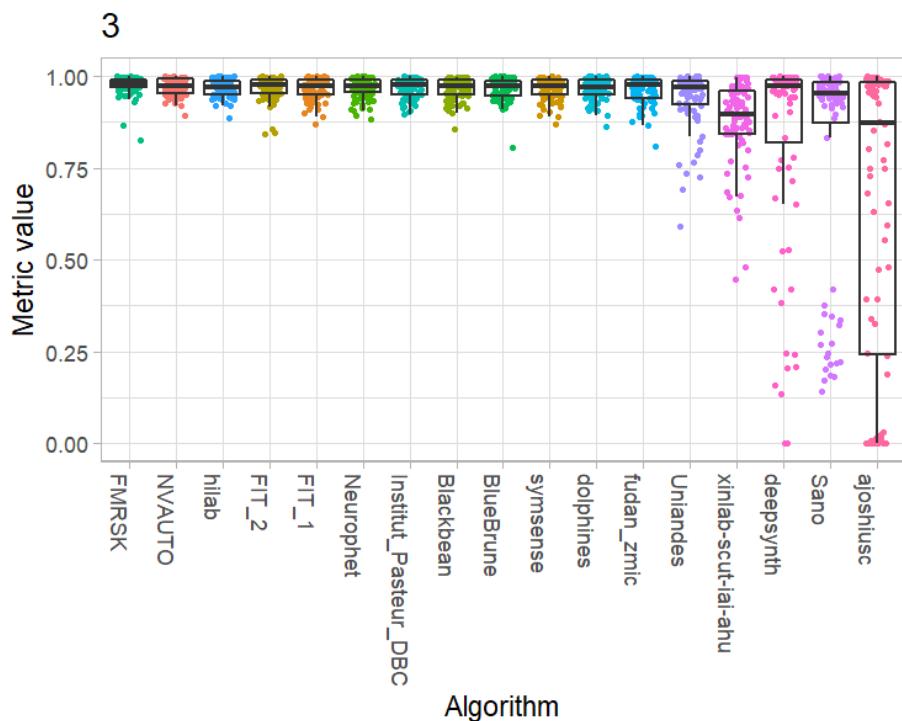
*Grey Matter*

	Volume_Similarity_mean	rank
FIT_1	0.9530493	1
BlueBrune	0.9527876	2
NVAUTO	0.9514285	3
dolphines	0.9508445	4
Blackbean	0.9502379	5
symsense	0.9467269	6
Neurophet	0.9467147	7
fudan_zmic	0.9463077	8
Institut_Pasteur_DBC	0.9433058	9
Uniandes	0.9408114	10
hilab	0.9333762	11
FIT_2	0.9292147	12
FMRSK	0.9009435	13
xinlab-scut-iai-ahu	0.8273008	14
Sano	0.8254385	15
deepsynth	0.7229728	16
ajoshiusc	0.6822188	17



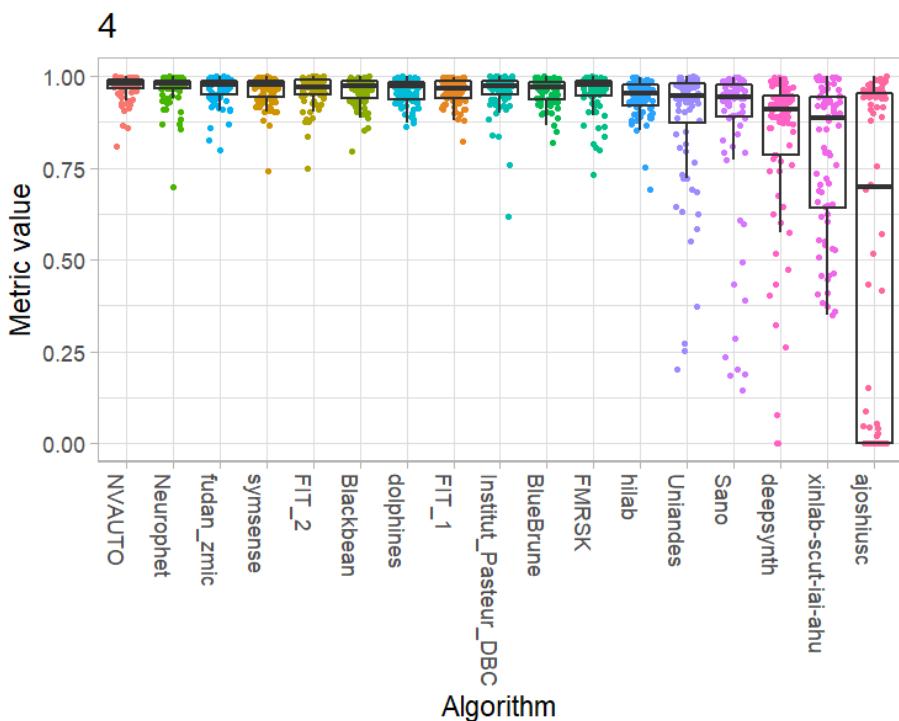
*White Matter*

	Volume_Similarity_mean	rank
FMRSK	0.9756578	1
NVAUTO	0.9721781	2
hilab	0.9687652	3
FIT_2	0.9679916	4
FIT_1	0.9674281	5
Neurophet	0.9672782	6
Institut_Pasteur_DBC	0.9670964	7
Blackbean	0.9667710	8
BlueBrune	0.9666252	9
symsense	0.9659575	10
dolphines	0.9658533	11
fudan_zmic	0.9621530	12
Uniandes	0.9380243	13
xinlab-scut-iai-ahu	0.8769402	14
deepsynth	0.8385679	15
Sano	0.7977872	16
ajoshiusc	0.6463211	17



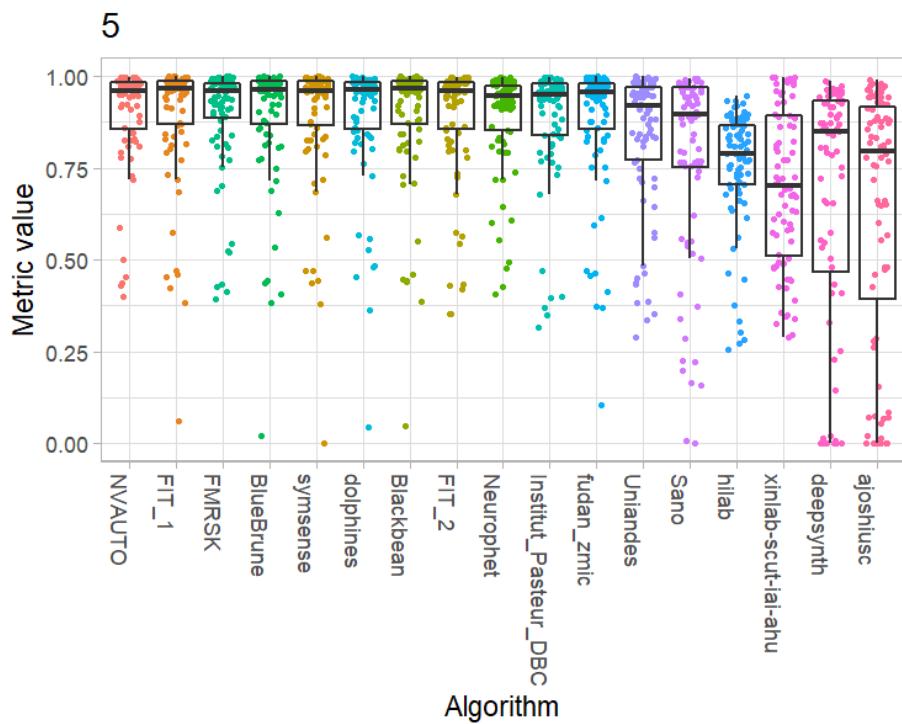
*Ventricles*

	Volume_Similarity_mean	rank
NVAUTO	0.9702868	1
Neurophet	0.9661515	2
fudan_zmic	0.9632481	3
symsense	0.9621225	4
FIT_2	0.9616560	5
Blackbean	0.9602027	6
dolphines	0.9590036	7
FIT_1	0.9588118	8
Institut_Pasteur_DBC	0.9585284	9
BlueBrune	0.9580399	10
FMRSK	0.9550511	11
hilab	0.9420679	12
Uniandes	0.8782691	13
Sano	0.8577305	14
deepsynth	0.8211856	15
xinlab-scut-iai-ahu	0.7863451	16
ajoshiusc	0.5103154	17



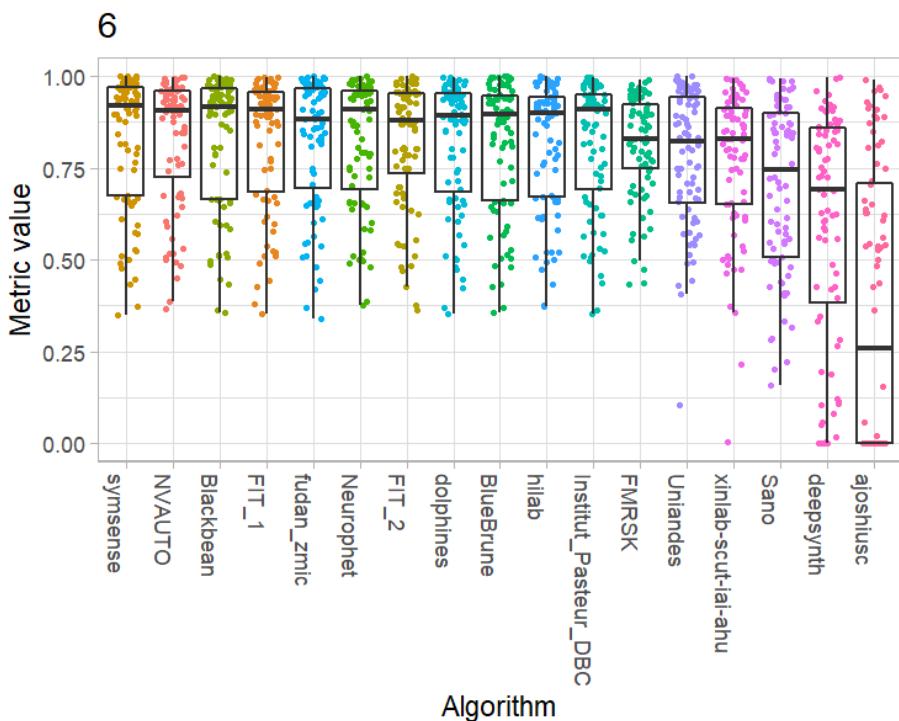
*Cerebellum*

	Volume_Similarity_mean	rank
NVAUTO	0.8997494	1
FIT_1	0.8931584	2
FMRSK	0.8922363	3
BlueBrune	0.8913910	4
symsense	0.8911255	5
dolphines	0.8904698	6
Blackbean	0.8896161	7
FIT_2	0.8882053	8
Neurophet	0.8879441	9
Institut_Pasteur_DBC	0.8832012	10
fudan_zmic	0.8795568	11
Uniandes	0.8335002	12
Sano	0.7809725	13
hilab	0.7520474	14
xinlab-scut-iai-ahu	0.7026824	15
deepsynth	0.6633126	16
ajoshiusc	0.6296582	17



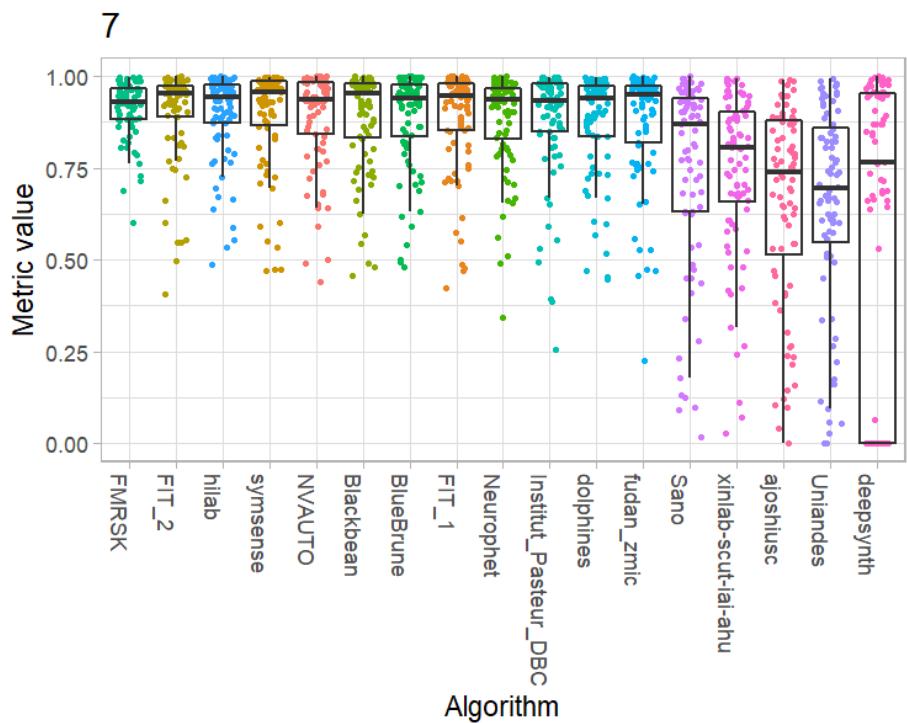
*Deep Grey Matter*

	Volume_Similarity_mean	rank
symsense	0.8298686	1
NVAUTO	0.8296028	2
Blackbean	0.8287232	3
FIT_1	0.8279880	4
fudan_zmic	0.8239713	5
Neurophet	0.8210637	6
FIT_2	0.8191854	7
dolphines	0.8186769	8
BlueBrune	0.8183607	9
hilab	0.8161053	10
Institut_Pasteur_DBC	0.8156715	11
FMRSK	0.8121560	12
Uniandes	0.7777304	13
xinlab-scut-iai-ahu	0.7697305	14
Sano	0.7014797	15
deepsynth	0.5953164	16
ajoshiusc	0.3616672	17



*Brainstem*

	Volume_Similarity_mean	rank
FMR SK	0.9133793	1
FIT_2	0.8968626	2
hilab	0.8953920	3
symsense	0.8930038	4
NVAUTO	0.8883429	5
Blackbean	0.8853860	6
BlueBrune	0.8846456	7
FIT_1	0.8837207	8
Neurophet	0.8815451	9
Institut_Pasteur_DBC	0.8795756	10
dolphines	0.8774987	11
fudan_zmic	0.8757044	12
Sano	0.7486758	13
xinlab-scut-iai-ahu	0.7365100	14
ajoshiusc	0.6583459	15
Uniandes	0.6494336	16
deepsynth	0.5867176	17

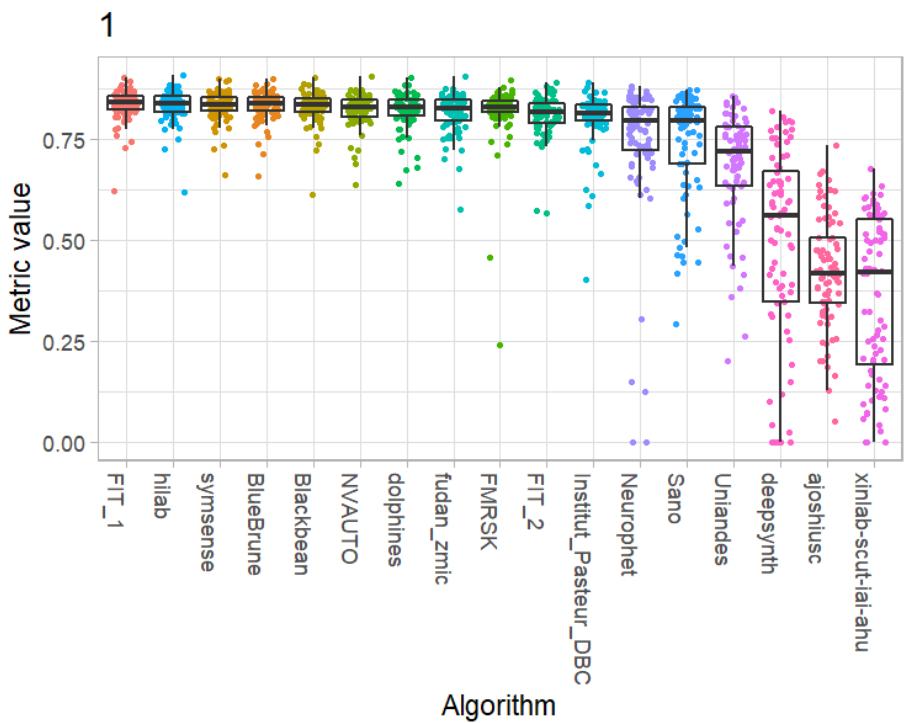


### 12.3 Out-of-Domain Evaluation Metrics per Label

#### Dice Similarity Coefficient

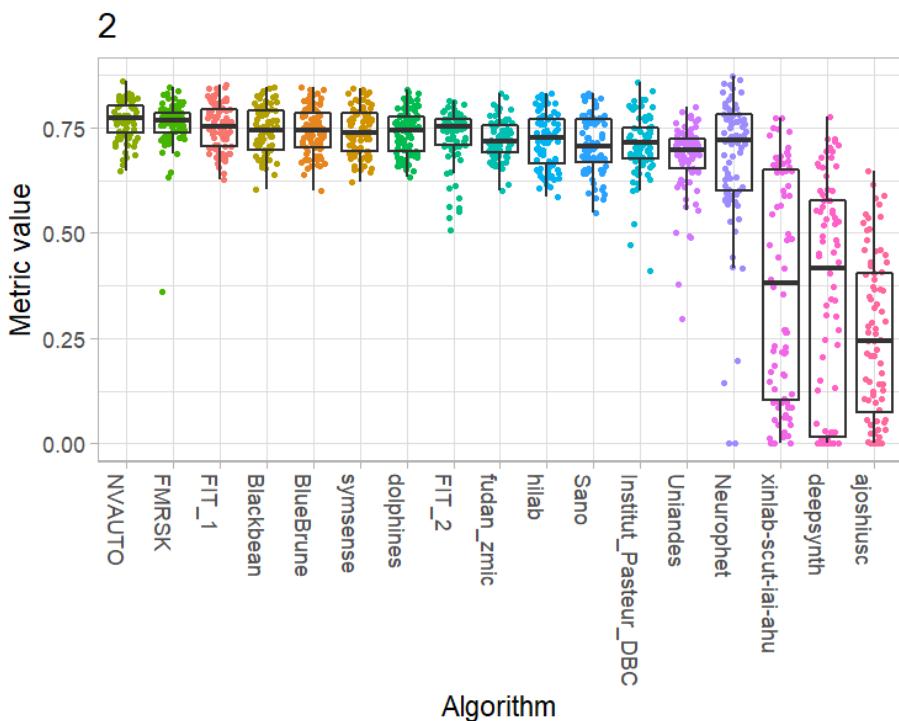
*External Cerebrospinal Fluid*

	Dice_mean	rank
FIT_1	0.8337490	1
hilab	0.8331093	2
symsense	0.8319507	3
BlueBrune	0.8318292	4
Blackbean	0.8295458	5
NVAUTO	0.8224574	6
dolphines	0.8205208	7
fudan_zmic	0.8158054	8
FMRSK	0.8152904	9
FIT_2	0.8084611	10
Institut_Pasteur_DBC	0.8001349	11
Neurophet	0.7418510	12
Sano	0.7416720	13
Uniandes	0.6844984	14
deepsynth	0.4844075	15
ajoshiusc	0.4236117	16
xinlab-scut-iai-ahu	0.3627327	17



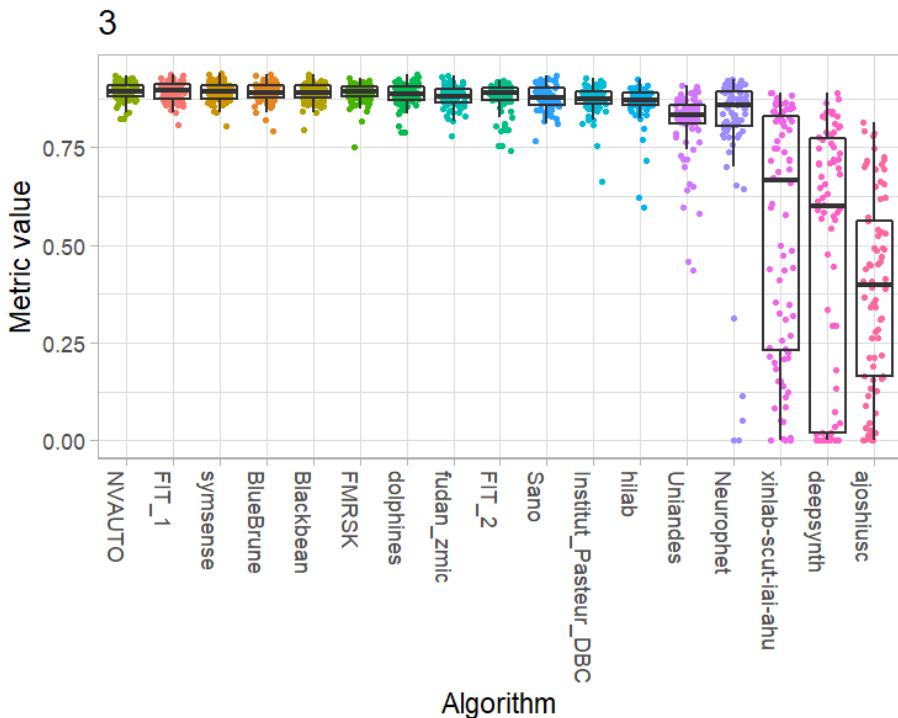
*Grey Matter*

	Dice_mean	rank
NVAUTO	0.7682381	1
FMRSK	0.7593048	2
FIT_1	0.7517735	3
Blackbean	0.7429138	4
BlueBrune	0.7427417	5
symsense	0.7416361	6
dolphines	0.7388338	7
FIT_2	0.7304760	8
fudan_zmic	0.7218863	9
hilab	0.7210807	10
Sano	0.7122103	11
Institut_Pasteur_DBC	0.7095380	12
Uniandes	0.6781594	13
Neurophet	0.6727766	14
xinlab-scut-iai-ahu	0.3768089	15
deepsynth	0.3446453	16
ajoshiusc	0.2507014	17



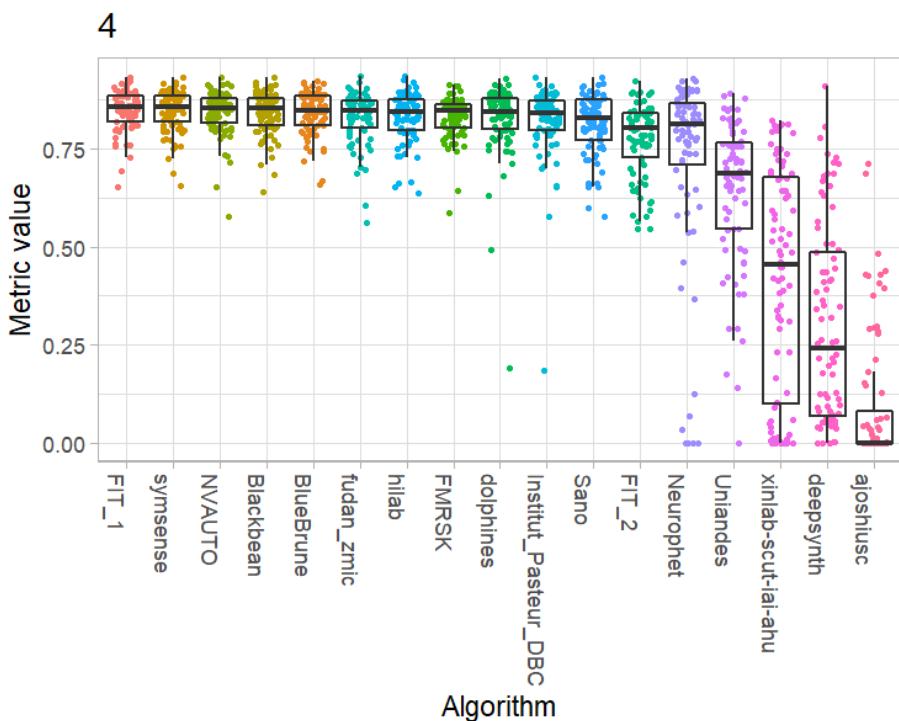
*White Matter*

	Dice_mean	rank
NVAUTO	0.8931265	1
FIT_1	0.8927828	2
symsense	0.8912627	3
BlueBrune	0.8903460	4
Blackbean	0.8902632	5
FMRSK	0.8892611	6
dolphines	0.8858002	7
fudan_zmic	0.8794122	8
FIT_2	0.8778776	9
Sano	0.8775791	10
Institut_Pasteur_DBC	0.8718085	11
hilab	0.8635775	12
Uniandes	0.8100587	13
Neurophet	0.8009063	14
xinlab-scut-iai-ahu	0.5370683	15
deepsynth	0.4668142	16
ajoshiusc	0.3855877	17



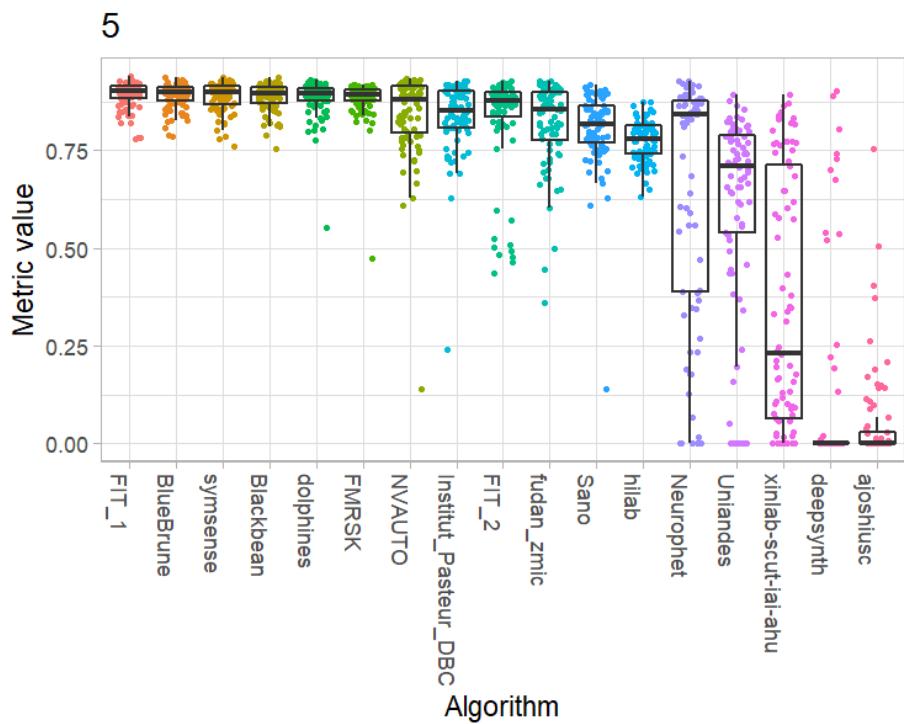
*Ventricles*

	Dice_mean	rank
FIT_1	0.8459670	1
symsense	0.8447024	2
NVAUTO	0.8401262	3
Blackbean	0.8397951	4
BlueBrune	0.8396340	5
fudan_zmic	0.8299527	6
hilab	0.8298495	7
FMRSK	0.8296761	8
dolphines	0.8232049	9
Institut_Pasteur_DBC	0.8179707	10
Sano	0.8134183	11
FIT_2	0.7731233	12
Neurophet	0.7133069	13
Uniandes	0.6454430	14
xinlab-scut-iai-ahu	0.4149313	15
deepsynth	0.3005444	16
ajoshiusc	0.0943369	17



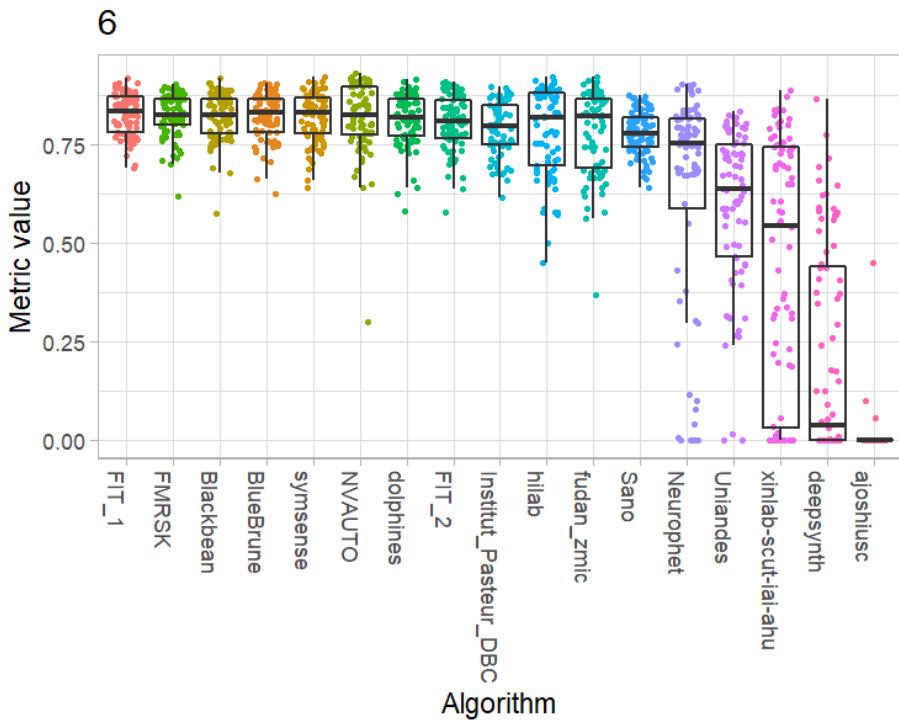
*Cerebellum*

	Dice_mean	rank
FIT_1	0.8916886	1
BlueBrune	0.8900868	2
symsense	0.8870642	3
Blackbean	0.8862300	4
dolphines	0.8834674	5
FMRSK	0.8810837	6
NVAUTO	0.8405104	7
Institut_Pasteur_DBC	0.8370581	8
FIT_2	0.8294083	9
fudan_zmic	0.8174166	10
Sano	0.8053829	11
hilab	0.7756544	12
Neurophet	0.6501613	13
Uniandes	0.6176645	14
xinlab-scut-iai-ahu	0.3561221	15
deepsynth	0.0985342	16
ajoshiusc	0.0518532	17



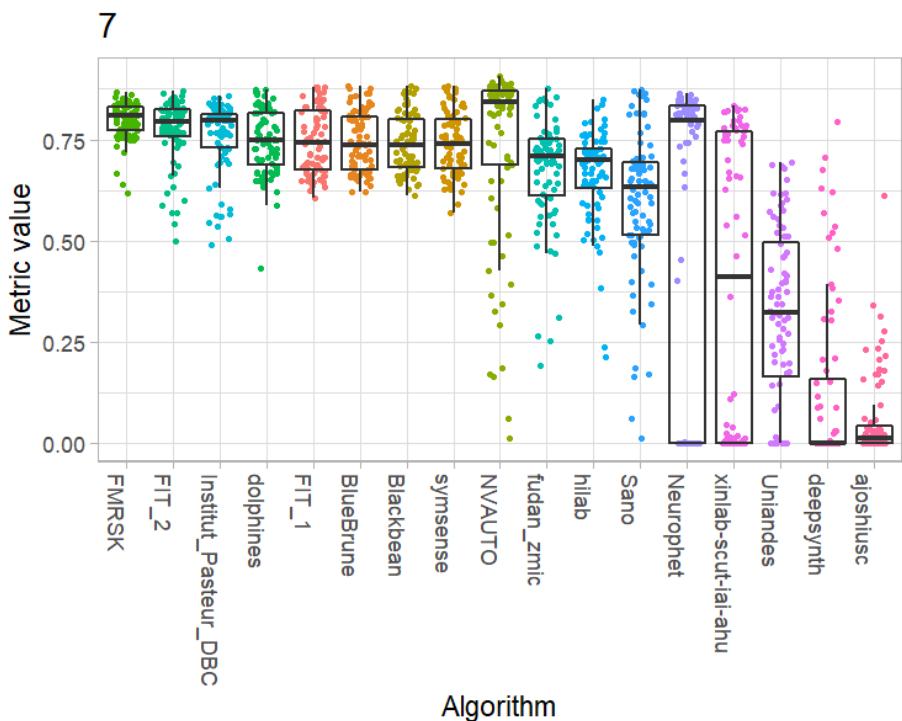
*Deep Grey Matter*

	Dice_mean	rank
FIT_1	0.8268471	1
FMRSK	0.8215627	2
Blackbean	0.8194472	3
BlueBrune	0.8189640	4
symsense	0.8185955	5
NVAUTO	0.8172699	6
dolphines	0.8126398	7
FIT_2	0.8042133	8
Institut_Pasteur_DBC	0.7898259	9
hilab	0.7823978	10
fudan_zmic	0.7812073	11
Sano	0.7777184	12
Neurophet	0.6191348	13
Uniandes	0.5920517	14
xinlab-scut-iai-ahu	0.4421640	15
deepsynth	0.2143268	16
ajoshiusc	0.0075639	17



*Brainstem*

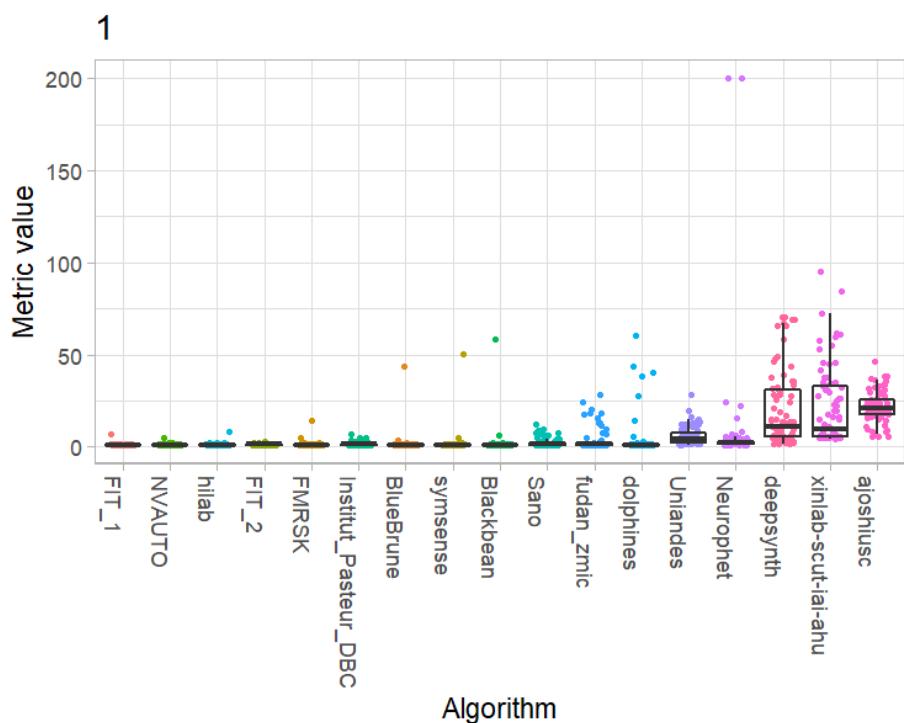
	Dice_mean	rank
FMRSK	0.7976343	1
FIT_2	0.7738880	2
Institut_Pasteur_DBC	0.7558296	3
dolphines	0.7468470	4
FIT_1	0.7450253	5
BlueBrune	0.7438340	6
Blackbean	0.7398061	7
symsense	0.7386228	8
NVAUTO	0.7312224	9
fudan_zmic	0.6737203	10
hilab	0.6653521	11
Sano	0.5986593	12
Neurophet	0.5287977	13
xinlab-scut-iai-ahu	0.3800020	14
Uniandes	0.3164218	15
deepsynth	0.1216436	16
ajoshiusc	0.0581970	17



### 95th percentile Hausdorff Distance

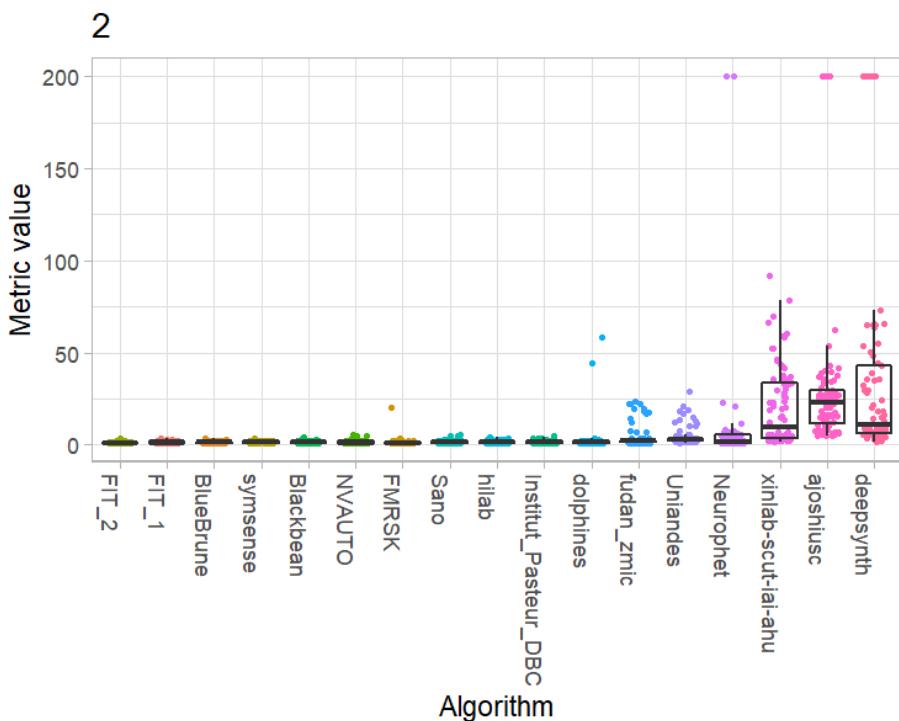
*External Cerebrospinal Fluid*

	Hausdorff_mean	rank
FIT_1	1.315017	1
NVAUTO	1.412130	2
hilab	1.441750	3
FIT_2	1.463382	4
FMRSK	1.589641	5
Institut_Pasteur_DBC	1.723857	6
BlueBrune	1.835754	7
symsense	1.940743	8
Blackbean	2.081272	9
Sano	2.571066	10
fudan_zmic	3.768968	11
dolphines	4.104579	12
Uniandes	5.704037	13
Neurophet	8.031233	14
deepsynth	20.073895	15
xinlab-scut-iai-ahu	21.253929	16
ajoshiusc	21.964587	17



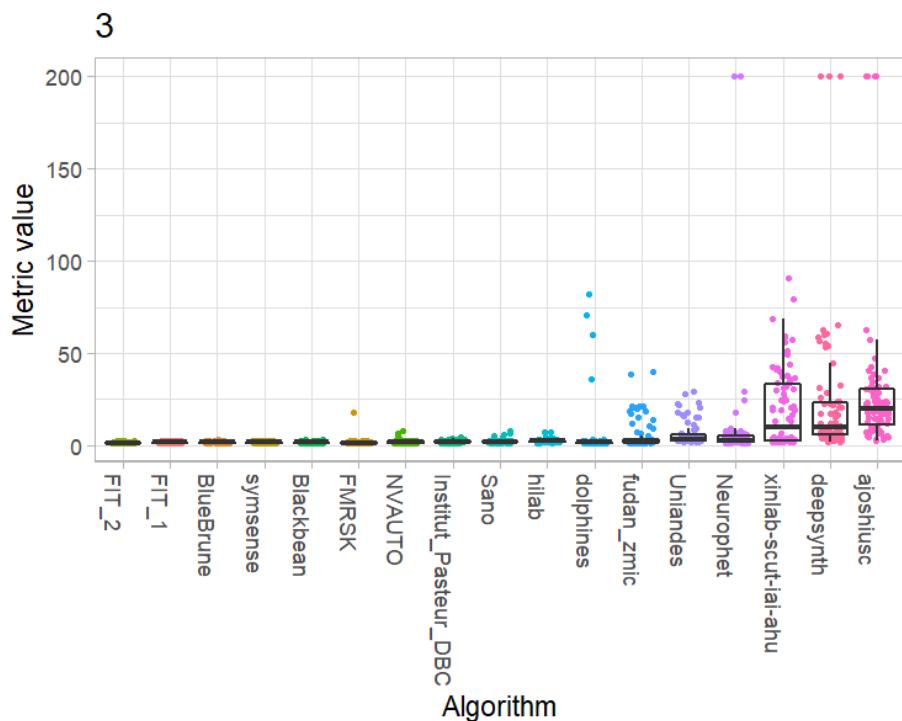
*Grey Matter*

	Hausdorff_mean	rank
FIT_2	1.236228	1
FIT_1	1.467746	2
BlueBrune	1.504399	3
symsense	1.547406	4
Blackbean	1.572826	5
NVAUTO	1.580428	6
FMRSK	1.581225	7
Sano	1.807330	8
hilab	1.858643	9
Institut_Pasteur_DBC	1.952965	10
dolphines	2.818108	11
fudan_zmic	4.387116	12
Uniandes	4.762992	13
Neurophet	8.401903	14
xinlab-scut-iai-ahu	20.589585	15
ajoshiusc	30.754803	16
deepsynth	38.408908	17



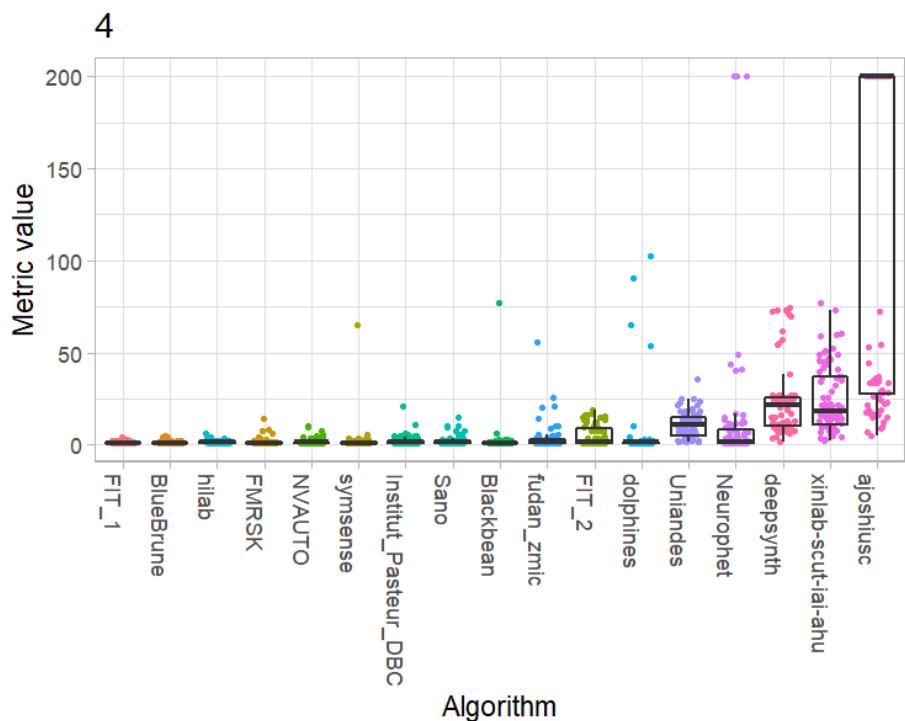
*White Matter*

	Hausdorff_mean	rank
FIT_2	1.563861	1
FIT_1	1.826411	2
BlueBrune	1.830433	3
symsense	1.850833	4
Blackbean	1.870851	5
FMRSK	1.894066	6
NVAUTO	2.053675	7
Institut_Pasteur_DBC	2.194898	8
Sano	2.241500	9
hilab	2.551430	10
dolphines	4.850287	11
fudan_zmic	5.500536	12
Uniandes	6.461602	13
Neurophet	9.026583	14
xinlab-scut-iai-ahu	19.758316	15
deepsynth	23.912247	16
ajoshiusc	29.435896	17



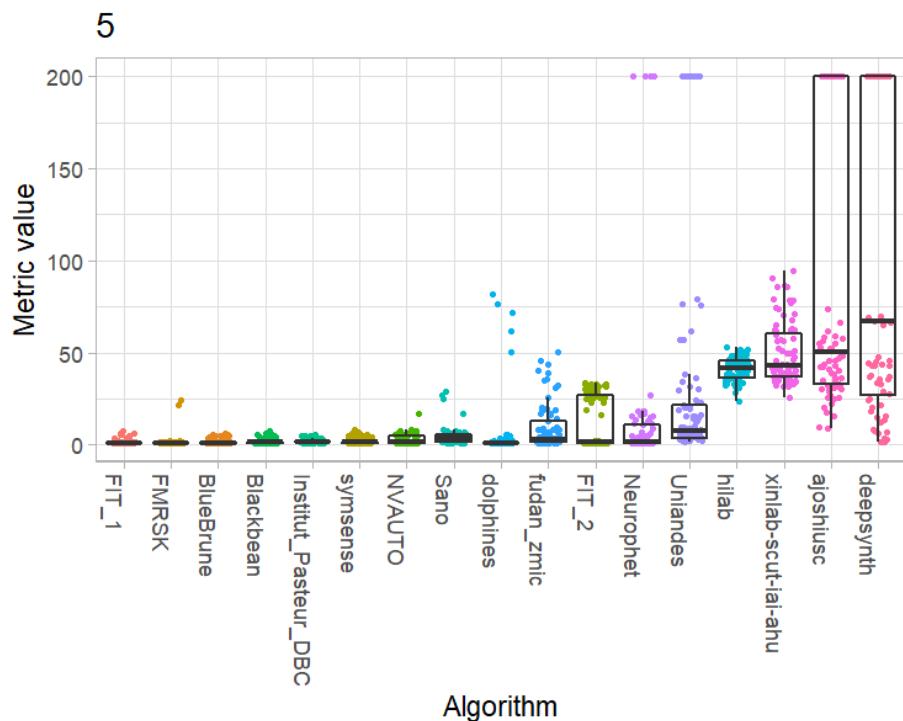
*Ventricles*

	Hausdorff_mean	rank
FIT_1	1.396277	1
BlueBrune	1.508299	2
hilab	1.723743	3
FMRSK	1.795598	4
NVAUTO	1.967377	5
symsense	2.256076	6
Institut_Pasteur_DBC	2.363509	7
Sano	2.375296	8
Blackbean	2.488673	9
fudan_zmic	3.890320	10
FIT_2	5.330758	11
dolphines	5.412092	12
Uniandes	10.814170	13
Neurophet	13.407540	14
deepsynth	23.125961	15
xinlab-scut-iai-ahu	24.657591	16
ajoshiusc	117.834760	17



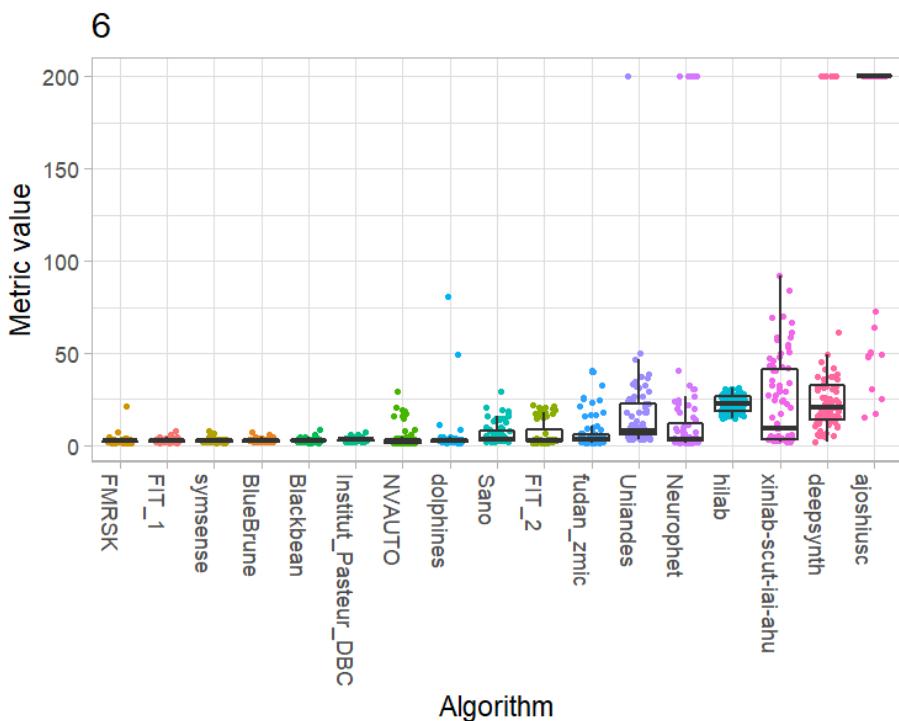
*Cerebellum*

	Hausdorff_mean	rank
FIT_1	1.725663	1
FMRSK	1.892687	2
BlueBrune	1.895130	3
Blackbean	1.913128	4
Institut_Pasteur_DBC	2.036621	5
symsense	2.124500	6
NVAUTO	3.020760	7
Sano	4.709219	8
dolphines	5.686301	9
fudan_zmic	9.353846	10
FIT_2	12.980117	11
Neurophet	19.598903	12
Uniandes	28.812656	13
hilab	41.322407	14
xinlab-scut-iai-ahu	49.503681	15
ajoshiusc	95.228086	16
deepsynth	108.800325	17



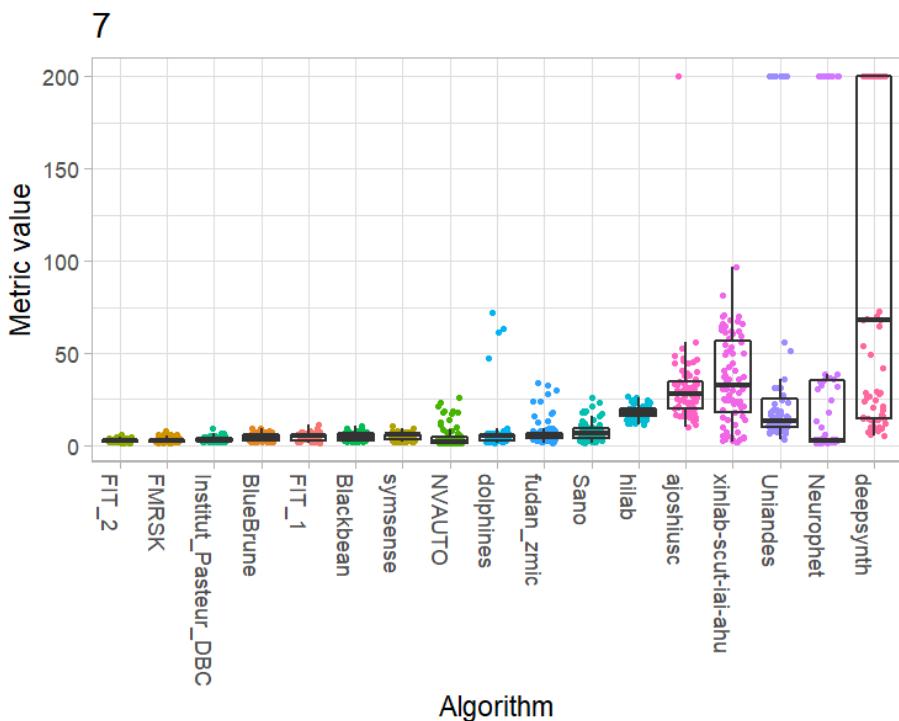
*Deep Grey Matter*

	Hausdorff_mean	rank
FMR SK	2.710479	1
FIT_1	2.755947	2
symsense	2.812116	3
BlueBrune	2.869058	4
Blackbean	2.875981	5
Institut_Pasteur_DBC	3.070791	6
NVAUTO	4.168480	7
dolphines	4.502141	8
Sano	6.143795	9
FIT_2	6.466709	10
fudan_zmic	7.147026	11
Uniandes	16.221163	12
Neurophet	21.974968	13
hilab	22.319464	14
xinlab-scut-iai-ahu	22.698064	15
deepsynth	39.310583	16
ajoshiusc	178.386704	17



*Brainstem*

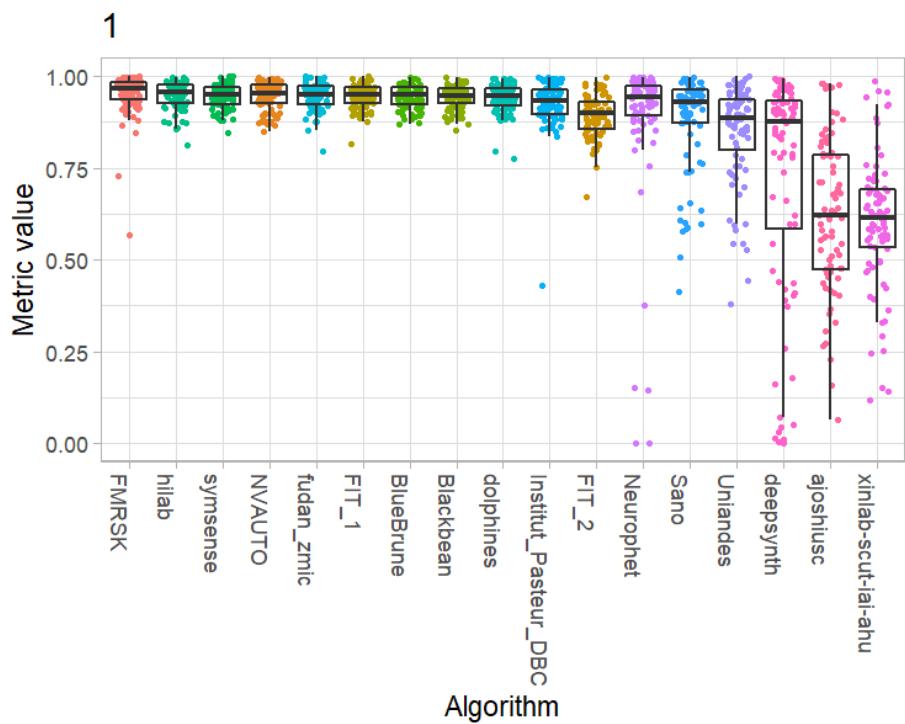
	Hausdorff_mean	rank
FIT_2	2.679801	1
FMRSK	2.890425	2
Institut_Pasteur_DBC	3.243911	3
BlueBrune	4.532057	4
FIT_1	4.719756	5
Blackbean	4.798987	6
symsense	4.915153	7
NVAUTO	5.061193	8
dolphines	7.276558	9
fudan_zmic	7.389241	10
Sano	7.758951	11
hilab	17.895417	12
ajoshiusc	30.539303	13
xinlab-scut-iai-ahu	35.695744	14
Uniandes	42.675325	15
Neurophet	46.317077	16
deepsynth	103.921027	17



### Volume Similarity

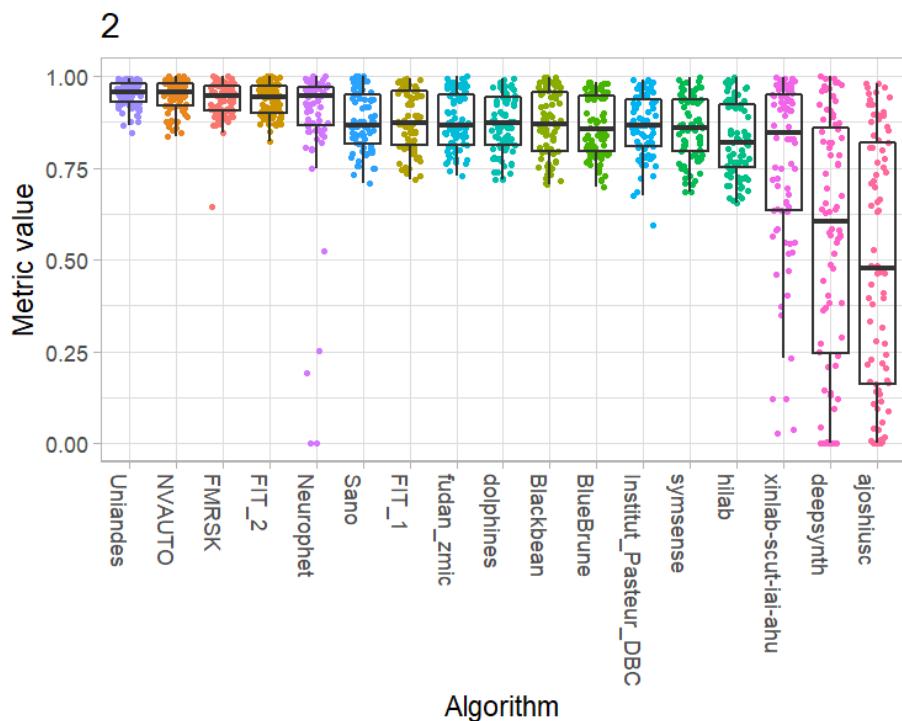
*External Cerebrospinal Fluid*

	Volume_Similarity_mean	rank
FMR SK	0.9506744	1
hilab	0.9465403	2
symsense	0.9456959	3
NVAUTO	0.9451693	4
fudan_zmic	0.9451134	5
FIT_1	0.9450281	6
BlueBrune	0.9441165	7
Blackbean	0.9424384	8
dolphines	0.9399160	9
Institut_Pasteur_DBC	0.9259333	10
FIT_2	0.8918895	11
Neurophet	0.8831092	12
Sano	0.8760364	13
Uniandes	0.8459791	14
deepsynth	0.7195105	15
ajoshiusc	0.6232196	16
xinlab-scut-iai-ahu	0.6026724	17



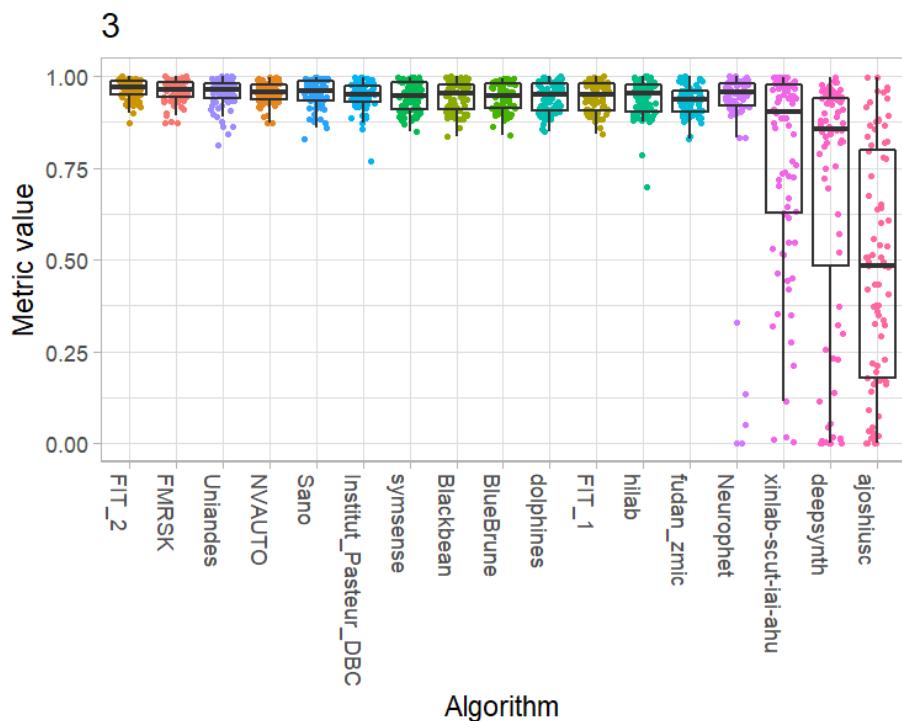
*Grey Matter*

	Volume_Similarity_mean	rank
Uniandes	0.9519508	1
NVAUTO	0.9473467	2
FMRSK	0.9360629	3
FIT_2	0.9359772	4
Neurophet	0.8806710	5
Sano	0.8788077	6
FIT_1	0.8777920	7
fudan_zmic	0.8775119	8
dolphines	0.8721186	9
Blackbean	0.8717747	10
BlueBrune	0.8659476	11
Institut_Pasteur_DBC	0.8649468	12
symsense	0.8604222	13
hilab	0.8289043	14
xinlab-scut-iai-ahu	0.7567914	15
deepsynth	0.5525400	16
ajoshiusc	0.4939296	17



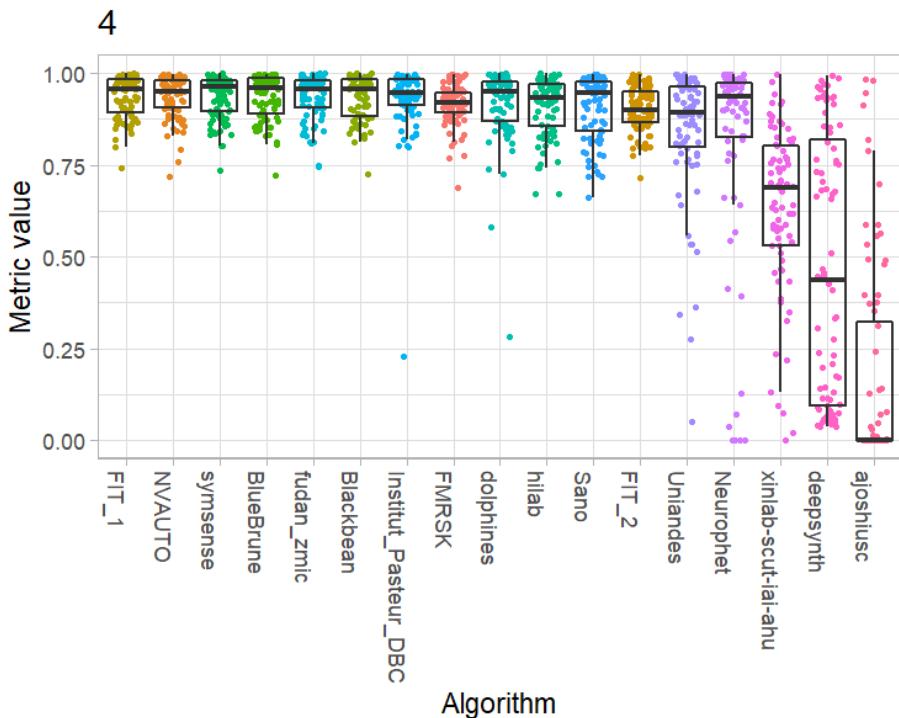
*White Matter*

	Volume_Similarity_mean	rank
FIT_2	0.9660979	1
FMRSK	0.9597598	2
Uniandes	0.9559368	3
NVAUTO	0.9553962	4
Sano	0.9545711	5
Institut_Pasteur_DBC	0.9472892	6
symsense	0.9441952	7
Blackbean	0.9440683	8
BlueBrune	0.9439515	9
dolphines	0.9435153	10
FIT_1	0.9433925	11
hilab	0.9409505	12
fudan_zmic	0.9346883	13
Neurophet	0.8992096	14
xinlab-scut-iai-ahu	0.7694890	15
deepsynth	0.6874537	16
ajoshiusc	0.4797851	17



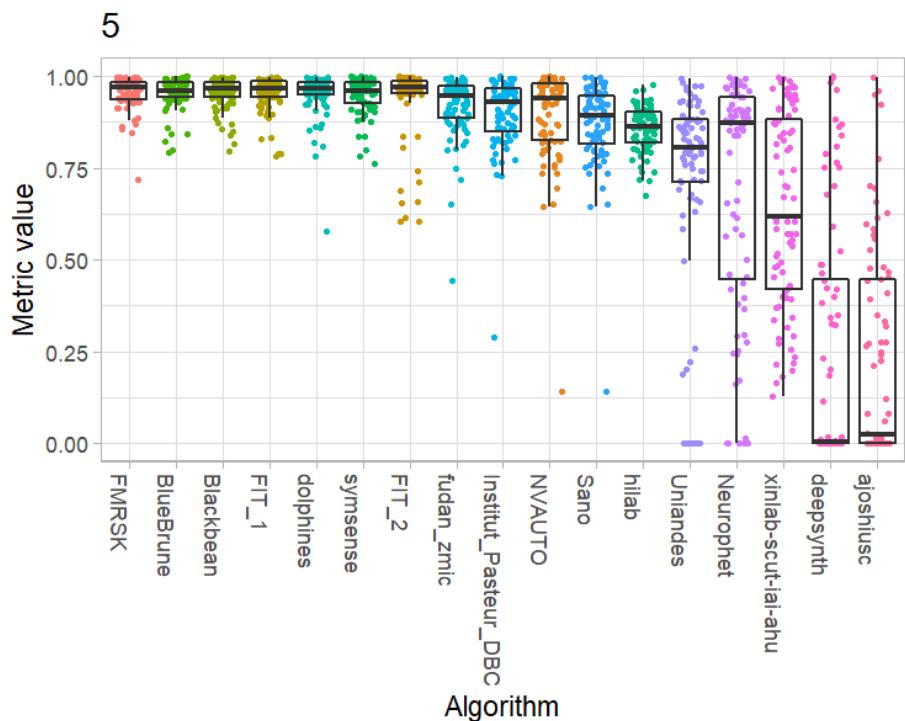
*Ventricles*

	Volume_Similarity_mean	rank
FIT_1	0.9366059	1
NVAUTO	0.9360715	2
symsense	0.9357746	3
BlueBrune	0.9346377	4
fudan_zmic	0.9338525	5
Blackbean	0.9330388	6
Institut_Pasteur_DBC	0.9294503	7
FMRSK	0.9149049	8
dolphines	0.9131922	9
hilab	0.9092984	10
Sano	0.9049090	11
FIT_2	0.9014521	12
Uniandes	0.8437202	13
Neurophet	0.8109155	14
xinlab-scut-iai-ahu	0.6346509	15
deepsynth	0.4653655	16
ajoshiusc	0.1700188	17



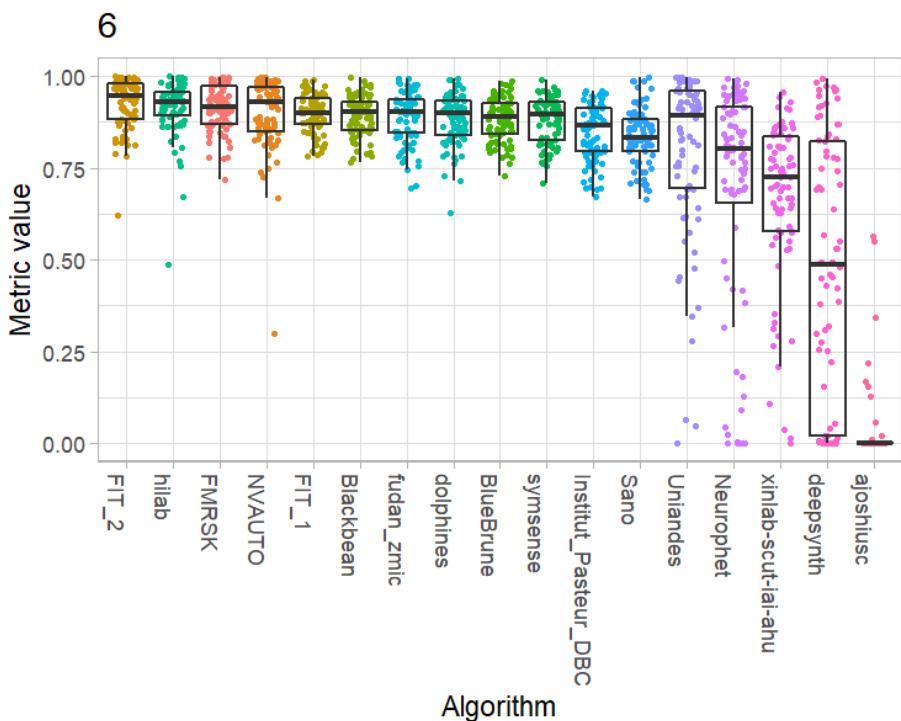
*Cerebellum*

	Volume_Similarity_mean	rank
FMRSK	0.9551024	1
BlueBrune	0.9544790	2
Blackbean	0.9542761	3
FIT_1	0.9541054	4
dolphines	0.9511934	5
symsense	0.9487508	6
FIT_2	0.9377144	7
fudan_zmic	0.9189724	8
Institut_Pasteur_DBC	0.8991082	9
NVAUTO	0.8945464	10
Sano	0.8687622	11
hilab	0.8571062	12
Uniandes	0.7290814	13
Neurophet	0.6928458	14
xinlab-scut-iai-ahu	0.6368375	15
deepsynth	0.2427404	16
ajoshiusc	0.2278496	17



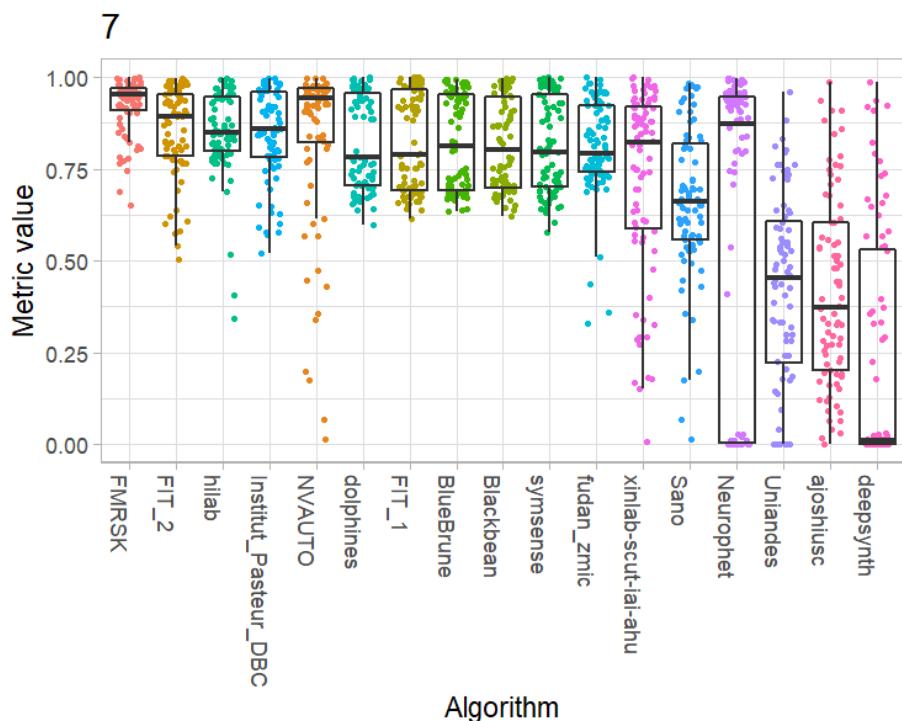
*Deep Grey Matter*

	Volume_Similarity_mean	rank
FIT_2	0.9270531	1
hilab	0.9152869	2
FMRSK	0.9146775	3
NVAUTO	0.8990144	4
FIT_1	0.8975569	5
Blackbean	0.8936621	6
fudan_zmic	0.8888094	7
dolphines	0.8837865	8
BlueBrune	0.8835821	9
symsense	0.8813508	10
Institut_Pasteur_DBC	0.8486107	11
Sano	0.8366612	12
Uniandes	0.7923437	13
Neurophet	0.6851061	14
xinlab-scut-iai-ahu	0.6679924	15
deepsynth	0.4694572	16
ajoshiusc	0.0279737	17



*Brainstem*

	Volume_Similarity_mean	rank
FMR SK	0.9233938	1
FIT_2	0.8559550	2
hilab	0.8514931	3
Institut_Pasteur_DBC	0.8466478	4
NVAUTO	0.8355246	5
dolphines	0.8263721	6
FIT_1	0.8218468	7
BlueBrune	0.8214891	8
Blackbean	0.8177871	9
symsense	0.8139695	10
fudan_zmic	0.8085862	11
xinlab-scut-iai-ahu	0.7247465	12
Sano	0.6648246	13
Neurophet	0.5973122	14
Uniandes	0.4208773	15
ajoshiusc	0.4186267	16
deepsynth	0.2309454	17



## 13 Benchmarking report for Dice Metrics – Excellent Quality Reconstructions

created by challengeR v1.0.2  
29 September, 2023

This document presents a systematic report on the benchmark study “Dice Metrics”. Input data comprises raw metric values for all algorithms and cases. Generated plots are:

- Visualization of assessment data: Dot- and boxplot, podium plot and ranking heatmap
- Visualization of ranking stability: Blob plot, violin plot and significance map, line plot

Details can be found in Wiesenfarth et al. (2021).

### 13.1 Ranking

Algorithms within a task are ranked according to the following ranking scheme:  
*aggregate using function (“mean”) then rank*

The analysis is based on 17 algorithms and 364 cases. 0 missing cases have been found in the data set.

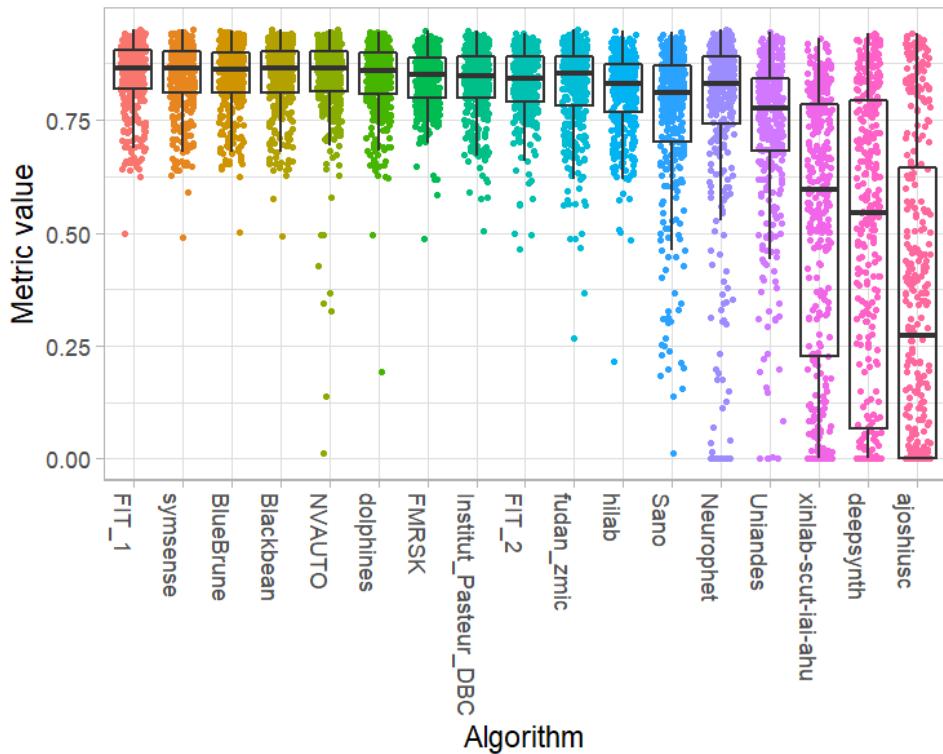
Ranking:

	Dice_mean	rank
FIT_1	0.8475250	1
symsense	0.8448590	2
BlueBrune	0.8444390	3
Blackbean	0.8435391	4
NVAUTO	0.8415211	5
dolphines	0.8407688	6
FMRSK	0.8388783	7
Institut_Pasteur_DBC	0.8335641	8
FIT_2	0.8310134	9
fudan_zmic	0.8235194	10
hilab	0.8112582	11
Sano	0.7583044	12
Neurophet	0.7469227	13
Uniandes	0.7258189	14
xinlab-scut-iai-ahu	0.5143970	15
deepsynth	0.4738044	16
ajoshiusc	0.3421430	17

### 13.2 Visualization of raw assessment data

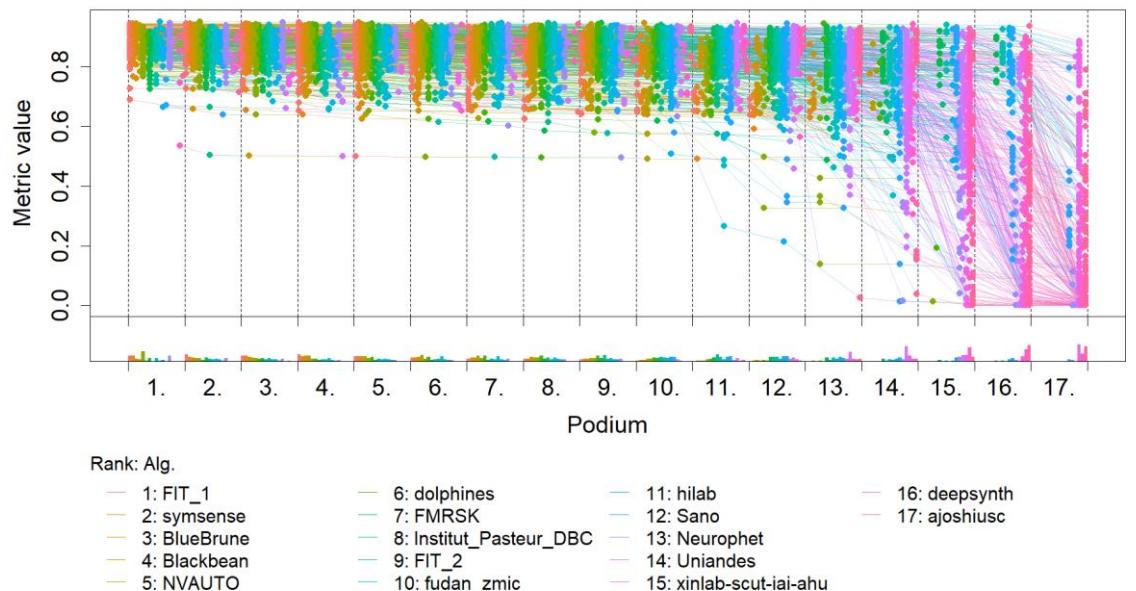
#### Dot- and boxplot

*Dot- and boxplots* for visualizing raw assessment data separately for each algorithm. Boxplots representing descriptive statistics over all cases (median, quartiles and outliers) are combined with horizontally jittered dots representing individual cases.



### Podium plot

*Podium plots* (see also Eugster et al., 2008) for visualizing raw assessment data. Upper part (spaghetti plot): Participating algorithms are color-coded, and each colored dot in the plot represents a metric value achieved with the respective algorithm. The actual metric value is encoded by the y-axis. Each podium (here:  $p=17$ ) represents one possible rank, ordered from best (1) to last (here: 17). The assignment of metric values (i.e. colored dots) to one of the podiums is based on the rank that the respective algorithm achieved on the corresponding case. Note that the plot part above each podium place is further subdivided into  $p$  “columns”, where each column represents one participating algorithm (here:  $p = 17$ ). Dots corresponding to identical cases are connected by a line, leading to the shown spaghetti structure. Lower part: Bar charts represent the relative frequency for each algorithm to achieve the rank encoded by the podium place.



## Ranking heatmap

*Ranking heatmaps* for visualizing raw assessment data. Each cell  $(i, A_j)$  shows the absolute frequency of cases in which algorithm  $A_j$  achieved rank  $i$ .

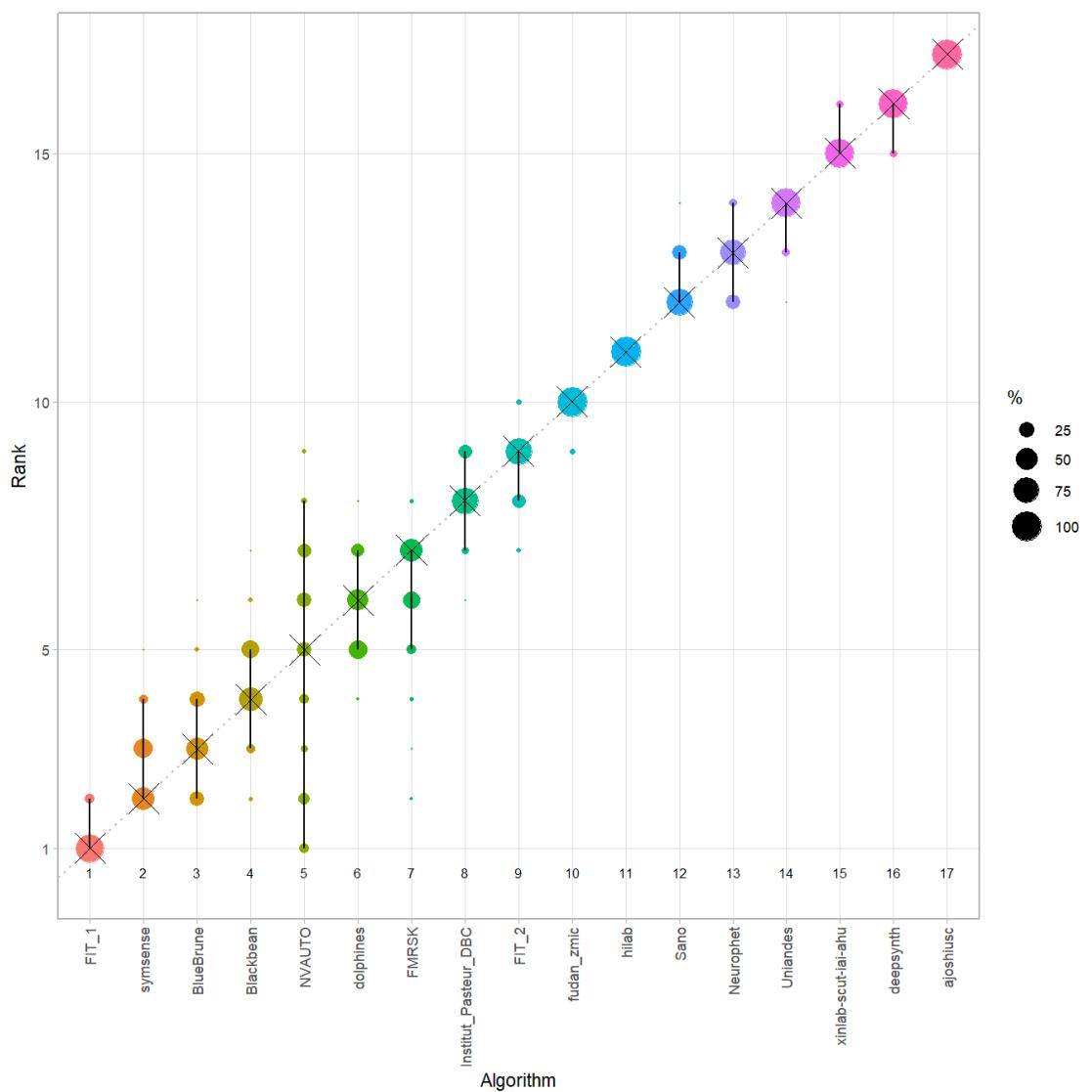


## Visualization of ranking stability

*Blob plot* for visualizing ranking stability based on bootstrap sampling

Algorithms are color-coded, and the area of each blob at position  $(A_i, \text{rank } j)$  is proportional to the relative frequency  $A_i$  achieved rank  $j$  across  $b = 1000$  bootstrap samples. The median rank for each algorithm is indicated by a black cross. 95% bootstrap intervals across bootstrap samples are indicated by black lines.

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> = "none")` instead.
```

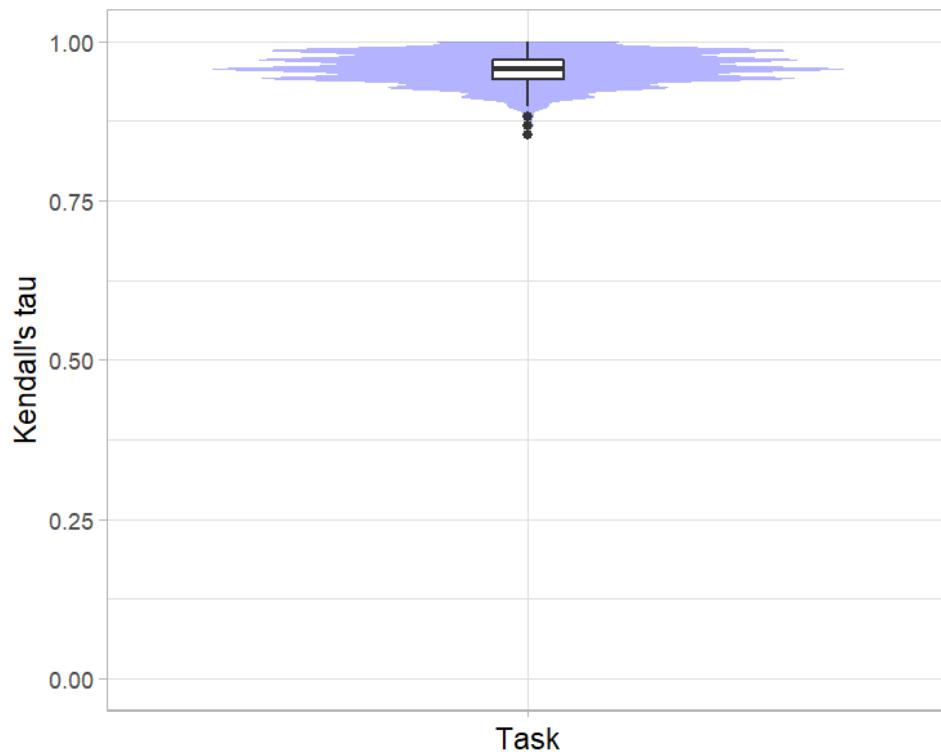


### ***Violin plot for visualizing ranking stability based on bootstrapping***

The ranking list based on the full assessment data is pairwise compared with the ranking lists based on the individual bootstrap samples (here  $b = 1000$  samples). For each pair of rankings, Kendall's  $\tau$  correlation is computed. Kendall's  $\tau$  is a scaled index determining the correlation between the lists. It is computed by evaluating the number of pairwise concordances and discordances between ranking lists and produces values between  $-1$  (for inverted order) and  $1$  (for identical order). A violin plot, which simultaneously depicts a boxplot and a density plot, is generated from the results.

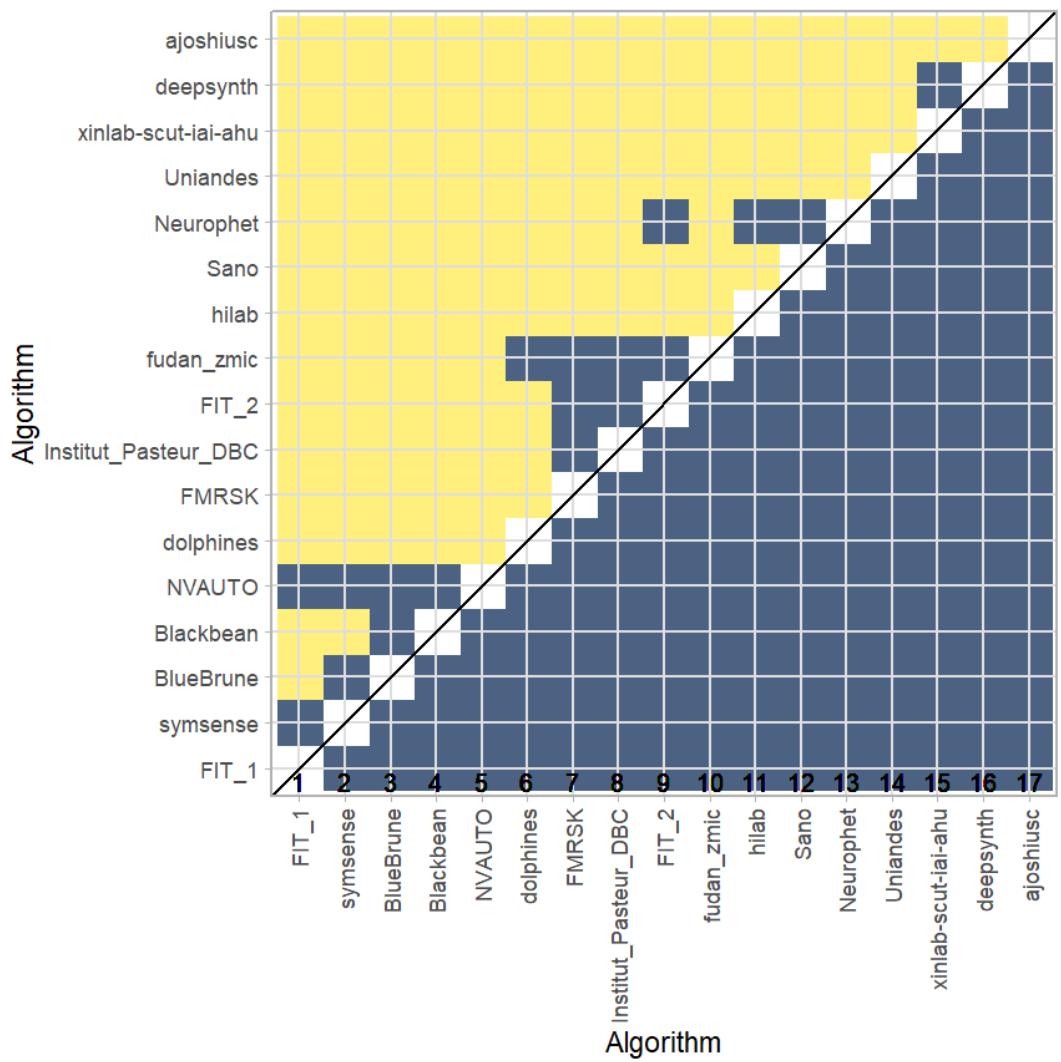
Summary Kendall's tau:

Task	mean	median	q25	q75
dummyTask	0.9575147	0.9558824	0.9411765	0.9705882



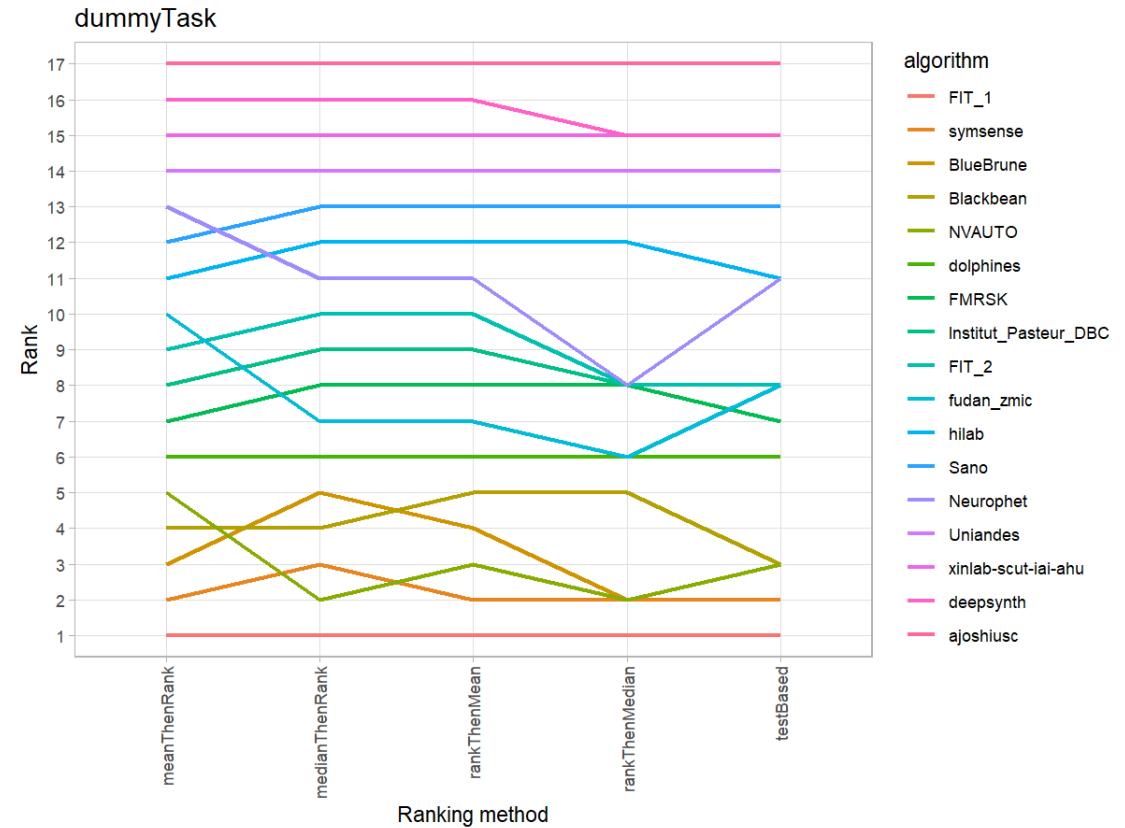
## **Significance maps for visualizing ranking stability based on statistical significance**

*Significance maps* depict incidence matrices of pairwise significant test results for the one-sided Wilcoxon signed rank test at a 5% significance level with adjustment for multiple testing according to Holm. Yellow shading indicates that metric values from the algorithm on the x-axis were significantly superior to those from the algorithm on the y-axis, blue color indicates no significant difference.



### Ranking robustness to ranking methods

*Line plots* for visualizing ranking robustness across different ranking methods. Each algorithm is represented by one colored line. For each ranking method encoded on the x-axis, the height of the line represents the corresponding rank. Horizontal lines indicate identical ranks for all methods.



### 13.3 References

Wiesenfarth, M., Reinke, A., Landman, B.A., Eisenmann, M., Aguilera Saiz, L., Cardoso, M.J., Maier-Hein, L. and Kopp-Schneider, A. Methods and open-source toolkit for analyzing and visualizing challenge results. *Sci Rep* **11**, 2369 (2021). <https://doi.org/10.1038/s41598-021-82017-6>

M. J. A. Eugster, T. Hothorn, and F. Leisch, “Exploratory and inferential analysis of benchmark experiments,” Institut fuer Statistik, Ludwig-Maximilians-Universitaet Muenchen, Germany, Technical Report 30, 2008. [Online]. Available: <http://epub.ub.uni-muenchen.de/4134/>.

## 14 Benchmarking report for Hausdorff Metrics – Excellent Quality Reconstructions

created by challengeR v1.0.2  
29 September, 2023

This document presents a systematic report on the benchmark study “Hausdorff Metrics”. Input data comprises raw metric values for all algorithms and cases. Generated plots are:

- Visualization of assessment data: Dot- and boxplot, podium plot and ranking heatmap
- Visualization of ranking stability: Blob plot, violin plot and significance map, line plot

Details can be found in Wiesenfarth et al. (2021).

### 14.1 Ranking

Algorithms within a task are ranked according to the following ranking scheme:  
*aggregate using function (“mean”) then rank*

The analysis is based on 17 algorithms and 364 cases. 0 missing cases have been found in the data set.

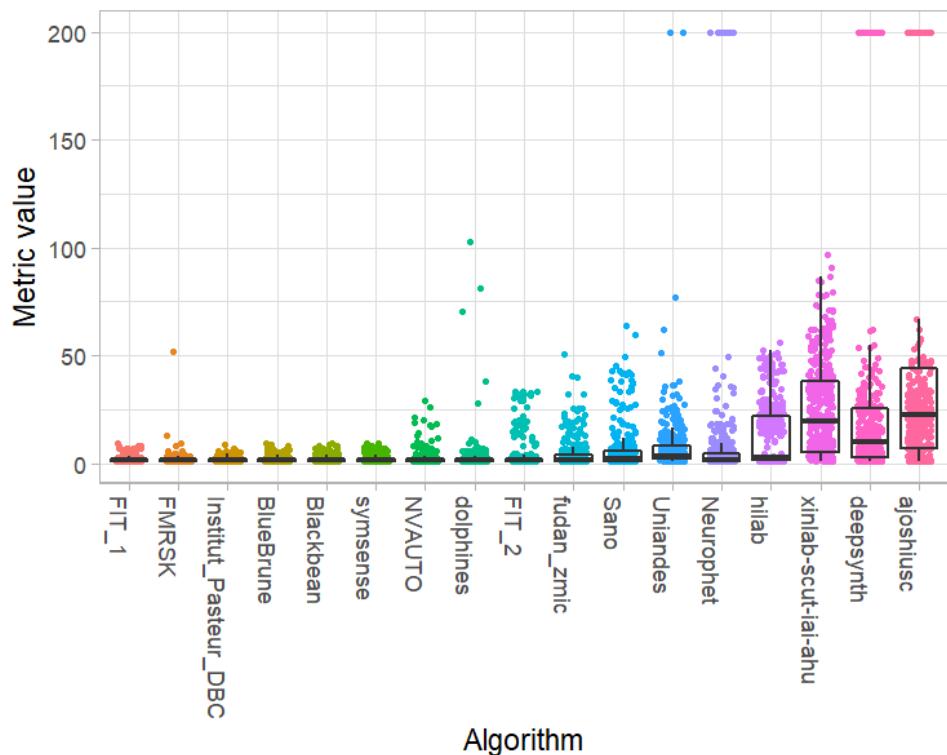
Ranking:

	Hausdorff_mean	rank
FIT_1	1.896977	1
FMRSK	1.906656	2
Insti-	1.920439	3
tut_Pasteur_DBC		
BlueBrune	1.921569	4
Blackbean	1.951724	5
symsense	1.988744	6
NVAUTO	2.397795	7
dolphines	2.755148	8
FIT_2	3.753654	9
fudan_zmic	4.196704	10
Sano	6.168081	11
Uniandes	7.881959	12
Neurophet	9.299844	13
hilab	12.854700	14
xinlab-scut-iai-ahu	24.849282	15
deepsynth	32.525491	16
ajoshiusc	52.881767	17

## 14.2 Visualization of raw assessment data

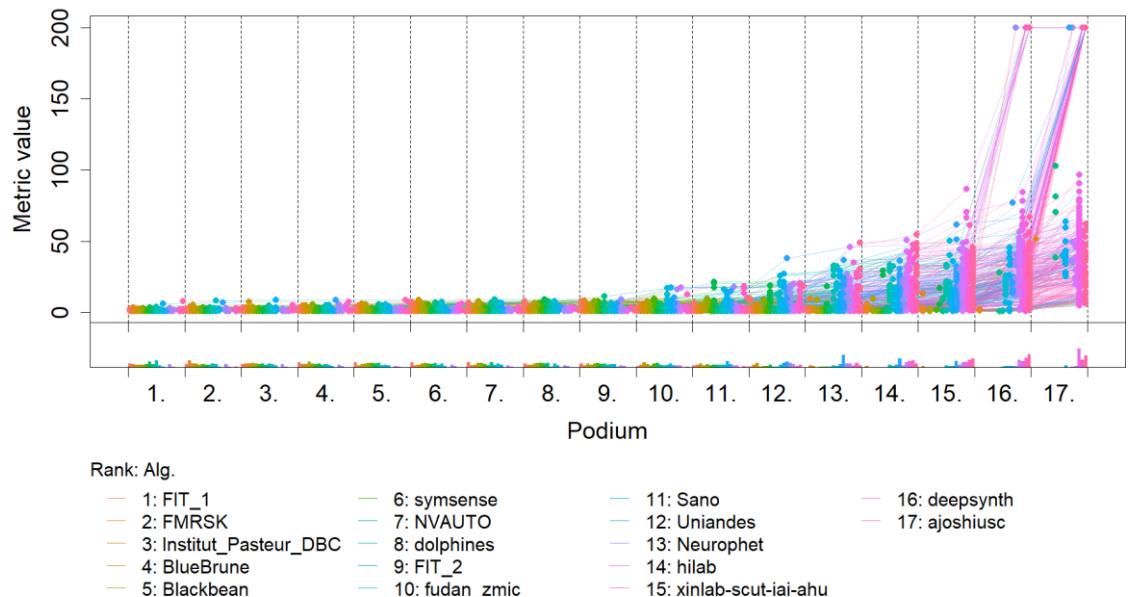
### Dot- and boxplot

*Dot- and boxplots* for visualizing raw assessment data separately for each algorithm. Boxplots representing descriptive statistics over all cases (median, quartiles and outliers) are combined with horizontally jittered dots representing individual cases.



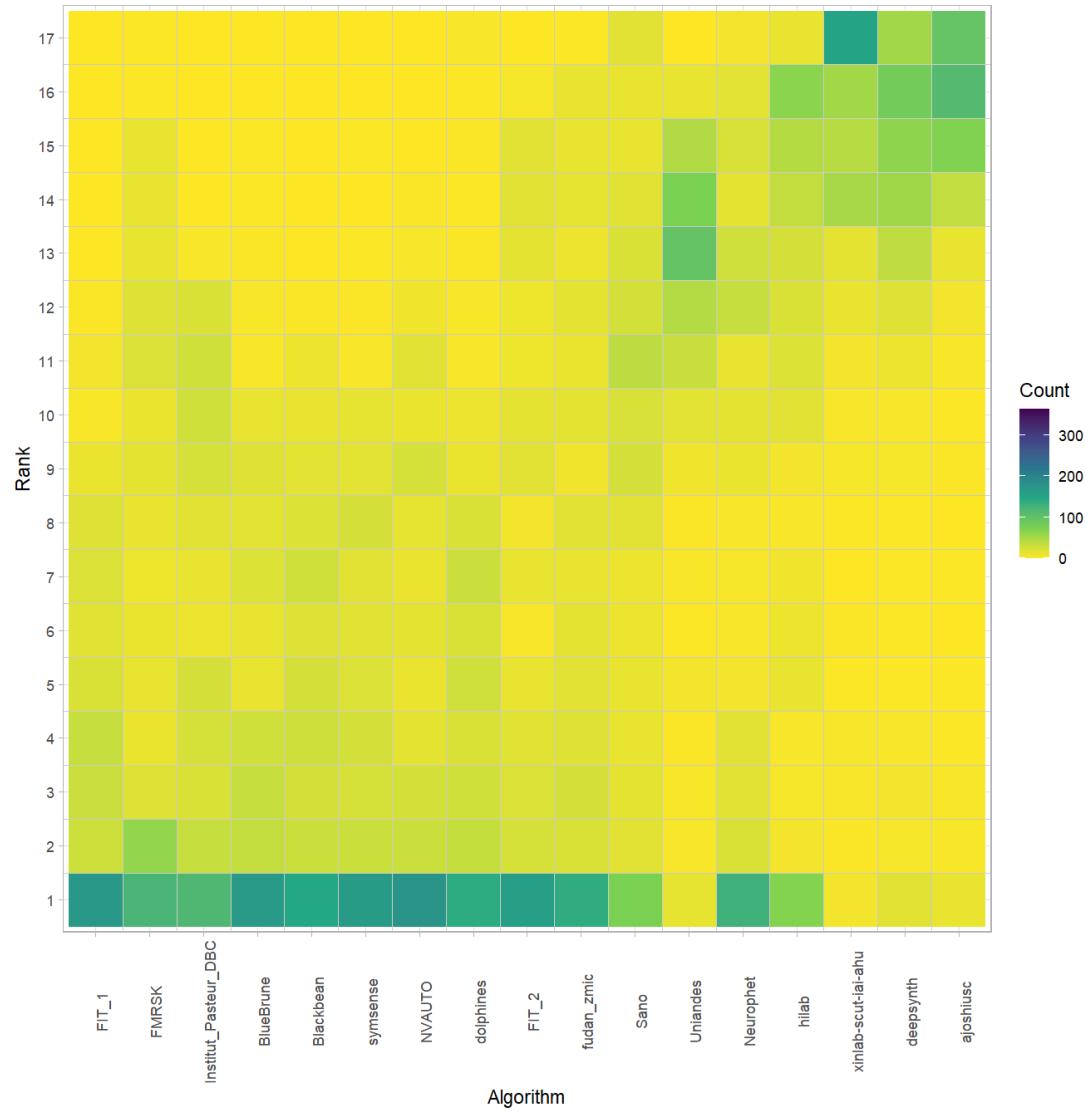
### Podium plot

*Podium plots* (see also Eugster et al., 2008) for visualizing raw assessment data. Upper part (spaghetti plot): Participating algorithms are color-coded, and each colored dot in the plot represents a metric value achieved with the respective algorithm. The actual metric value is encoded by the y-axis. Each podium (here:  $p=17$ ) represents one possible rank, ordered from best (1) to last (here: 17). The assignment of metric values (i.e. colored dots) to one of the podiums is based on the rank that the respective algorithm achieved on the corresponding case. Note that the plot part above each podium place is further subdivided into  $p$  “columns”, where each column represents one participating algorithm (here:  $p = 17$ ). Dots corresponding to identical cases are connected by a line, leading to the shown spaghetti structure. Lower part: Bar charts represent the relative frequency for each algorithm to achieve the rank encoded by the podium place.



## Ranking heatmap

*Ranking heatmaps* for visualizing raw assessment data. Each cell  $(i, A_j)$  shows the absolute frequency of cases in which algorithm  $A_j$  achieved rank  $i$ .

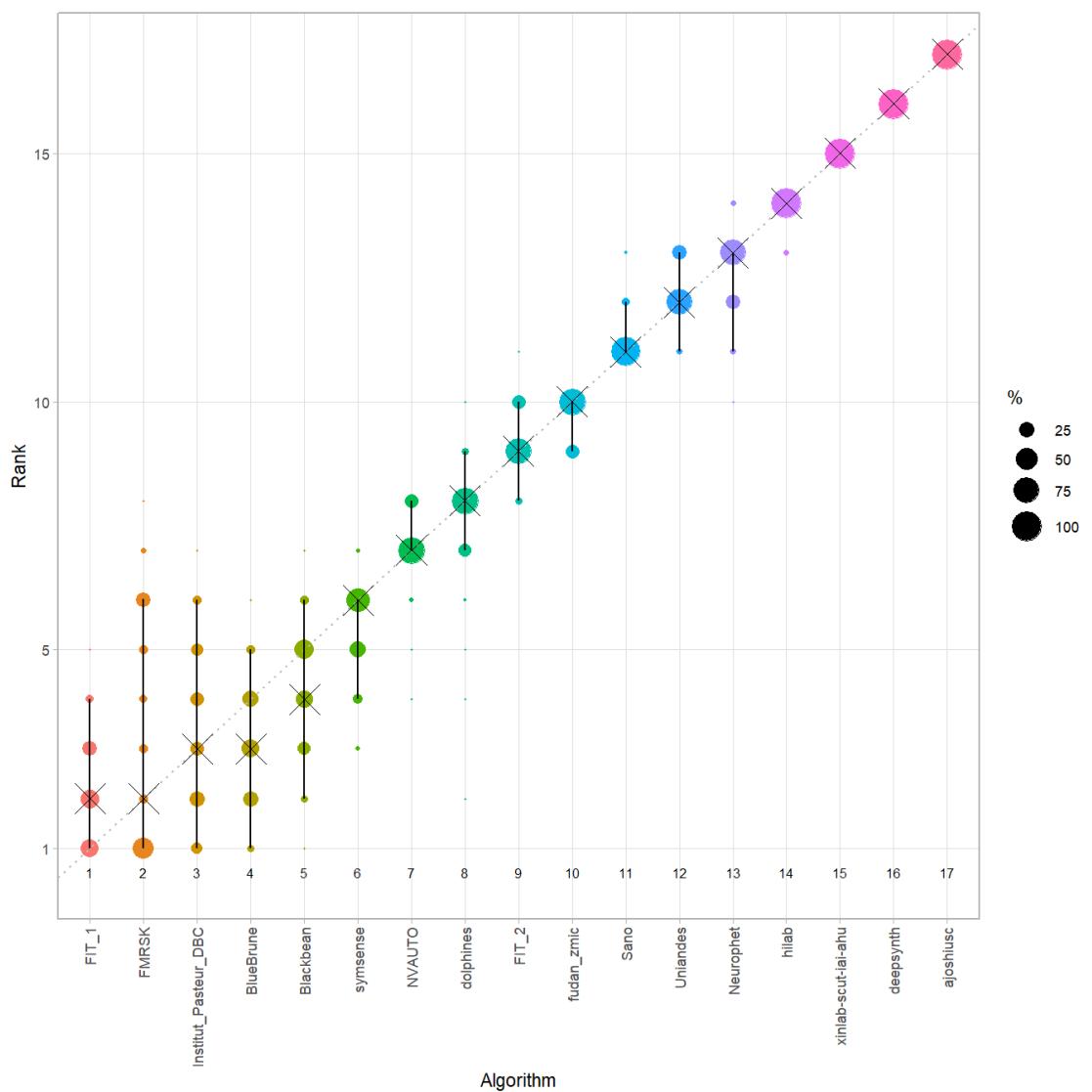


## Visualization of ranking stability

*Blob plot* for visualizing ranking stability based on bootstrap sampling

Algorithms are color-coded, and the area of each blob at position  $(A_i, \text{rank } j)$  is proportional to the relative frequency  $A_i$  achieved rank  $j$  across  $b = 1000$  bootstrap samples. The median rank for each algorithm is indicated by a black cross. 95% bootstrap intervals across bootstrap samples are indicated by black lines.

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> = "none")` instead.
```



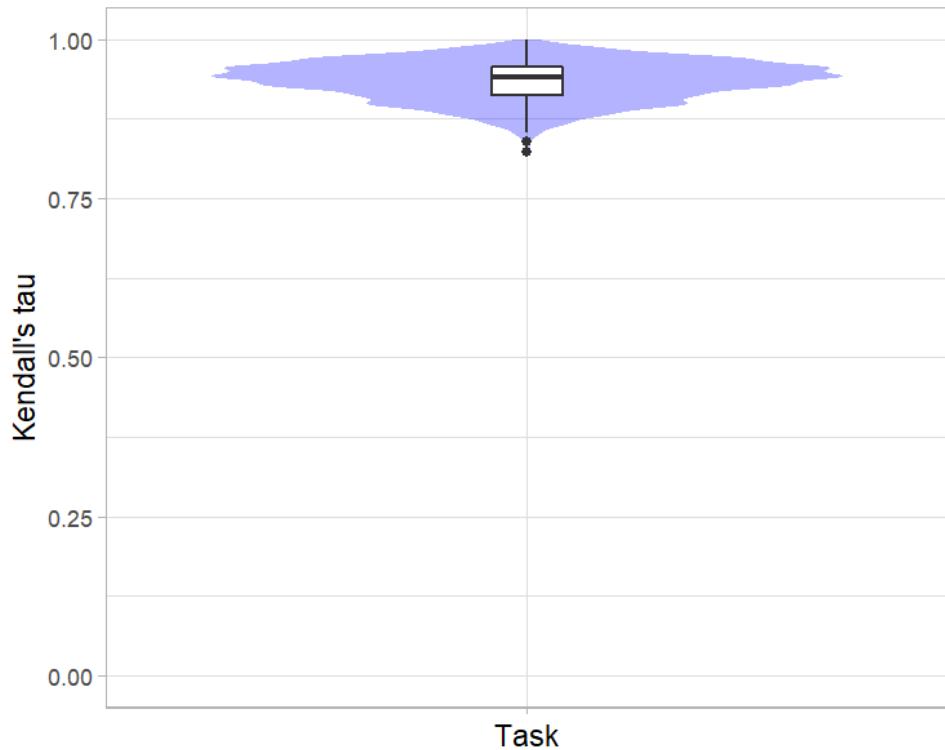
### ***Violin plot for visualizing ranking stability based on bootstrapping***

The ranking list based on the full assessment data is pairwise compared with the ranking lists based on the individual bootstrap samples (here  $b = 1000$  samples). For each pair of rankings, Kendall's  $\tau$  correlation is computed. Kendall's  $\tau$  is a scaled index determining the correlation between the lists. It is computed by evaluating the number of pairwise concordances and discordances between ranking lists and produces values between  $-1$  (for inverted order) and  $1$  (for identical order). A violin plot, which simultaneously depicts a boxplot and a density plot, is generated from the results.

Summary Kendall's tau:

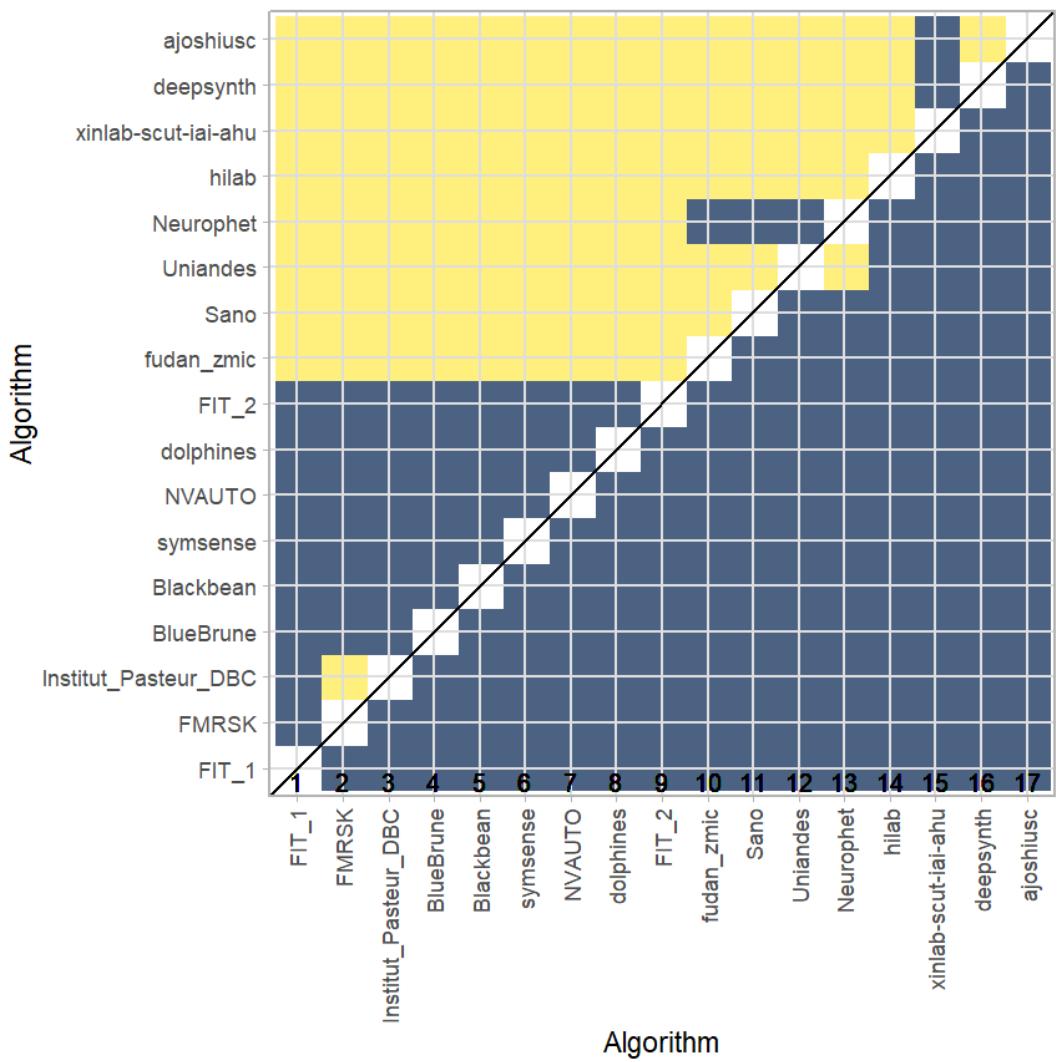
Task	mean	median	q25	q75
dummy-	0.9345588	0.9411765	0.9117647	0.9558824

Task



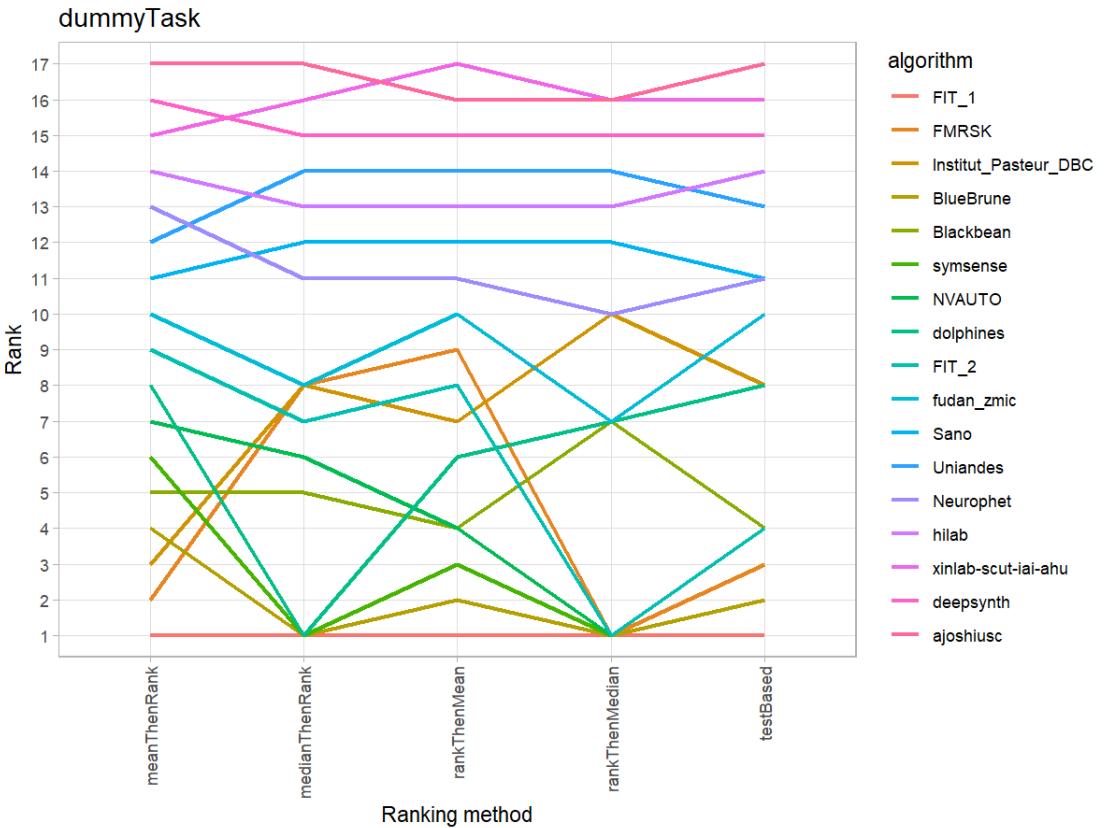
## *Significance maps* for visualizing ranking stability based on statistical significance

*Significance maps* depict incidence matrices of pairwise significant test results for the one-sided Wilcoxon signed rank test at a 5% significance level with adjustment for multiple testing according to Holm. Yellow shading indicates that metric values from the algorithm on the x-axis were significantly superior to those from the algorithm on the y-axis, blue color indicates no significant difference.



### Ranking robustness to ranking methods

*Line plots* for visualizing ranking robustness across different ranking methods. Each algorithm is represented by one colored line. For each ranking method encoded on the x-axis, the height of the line represents the corresponding rank. Horizontal lines indicate identical ranks for all methods.



### 14.3 References

Wiesenfarth, M., Reinke, A., Landman, B.A., Eisenmann, M., Aguilera Saiz, L., Cardoso, M.J., Maier-Hein, L. and Kopp-Schneider, A. Methods and open-source toolkit for analyzing and visualizing challenge results. *Sci Rep* **11**, 2369 (2021). <https://doi.org/10.1038/s41598-021-82017-6>

M. J. A. Eugster, T. Hothorn, and F. Leisch, “Exploratory and inferential analysis of benchmark experiments,” Institut fuer Statistik, Ludwig-Maximilians-Universitaet Muenchen, Germany, Technical Report 30, 2008. [Online]. Available: <http://epub.ub.uni-muenchen.de/4134/>.

## 15 Benchmarking report for Volume Similarity Metrics – Excellent Quality Reconstructions

created by challengeR v1.0.2  
29 September, 2023

This document presents a systematic report on the benchmark study “Volume Similarity Metrics”. Input data comprises raw metric values for all algorithms and cases. Generated plots are:

- Visualization of assessment data: Dot- and boxplot, podium plot and ranking heatmap
- Visualization of ranking stability: Blob plot, violin plot and significance map, line plot

Details can be found in Wiesenfarth et al. (2021).

### 15.1 Ranking

Algorithms within a task are ranked according to the following ranking scheme:  
*aggregate using function (“mean”) then rank*

The analysis is based on 17 algorithms and 364 cases. 0 missing cases have been found in the data set.

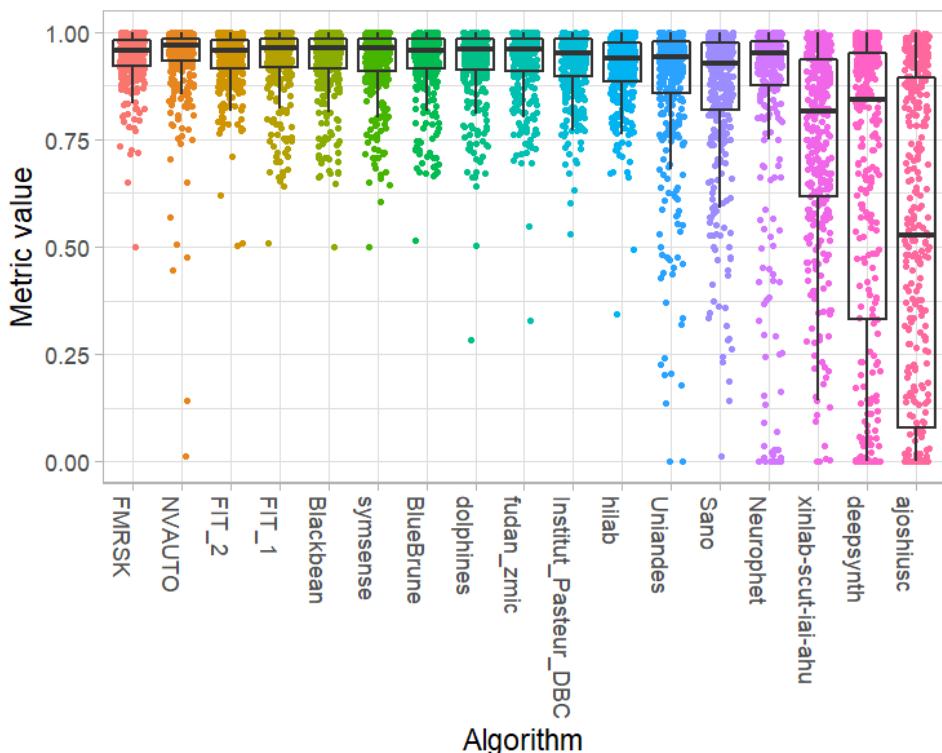
Ranking:

	Vol- ume_Similarity_mean	rank
FMRSK	0.9414758	1
NVAUTO	0.9399236	2
FIT_2	0.9369680	3
FIT_1	0.9343779	4
Blackbean	0.9321776	5
symsense	0.9317690	6
BlueBrune	0.9312333	7
dolphines	0.9306901	8
fudan_zmic	0.9295045	9
Institut_Pasteur_DBC	0.9267254	10
hilab	0.9155821	11
Uniandes	0.8742822	12
Sano	0.8564062	13
Neurophet	0.8447772	14
xinlab-scut-iai-ahu	0.7463051	15
deepsynth	0.6456027	16
ajoshiusc	0.5071440	17

## 15.2 Visualization of raw assessment data

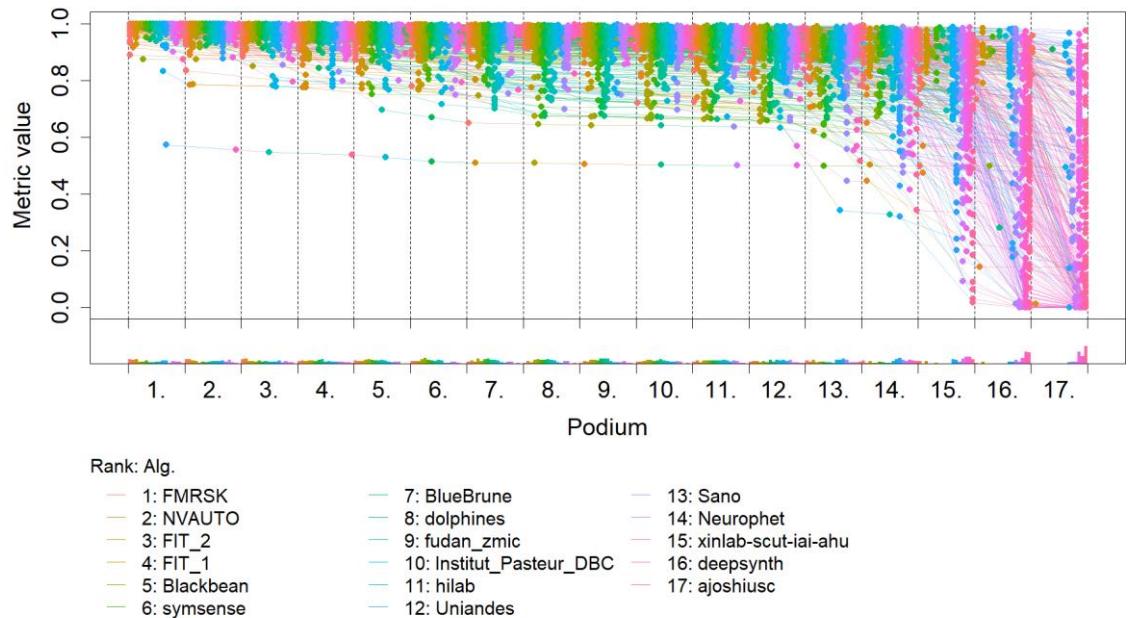
### Dot- and boxplot

*Dot- and boxplots* for visualizing raw assessment data separately for each algorithm. Boxplots representing descriptive statistics over all cases (median, quartiles and outliers) are combined with horizontally jittered dots representing individual cases.



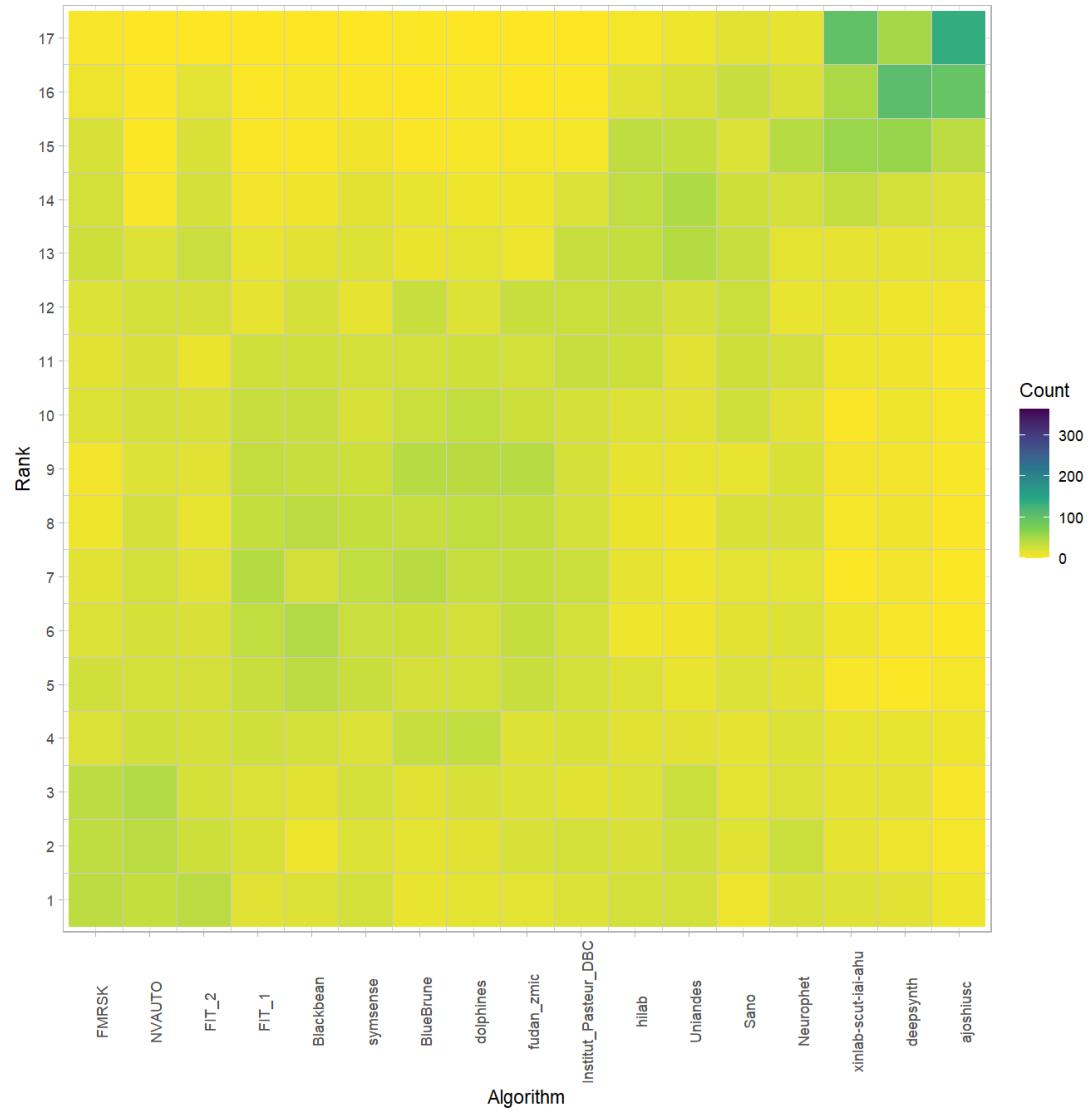
### Podium plot

*Podium plots* (see also Eugster et al., 2008) for visualizing raw assessment data. Upper part (spaghetti plot): Participating algorithms are color-coded, and each colored dot in the plot represents a metric value achieved with the respective algorithm. The actual metric value is encoded by the y-axis. Each podium (here:  $p=17$ ) represents one possible rank, ordered from best (1) to last (here: 17). The assignment of metric values (i.e. colored dots) to one of the podiums is based on the rank that the respective algorithm achieved on the corresponding case. Note that the plot part above each podium place is further subdivided into  $p$  “columns”, where each column represents one participating algorithm (here:  $p = 17$ ). Dots corresponding to identical cases are connected by a line, leading to the shown spaghetti structure. Lower part: Bar charts represent the relative frequency for each algorithm to achieve the rank encoded by the podium place.



## Ranking heatmap

*Ranking heatmaps* for visualizing raw assessment data. Each cell  $(i, A_j)$  shows the absolute frequency of cases in which algorithm  $A_j$  achieved rank  $i$ .

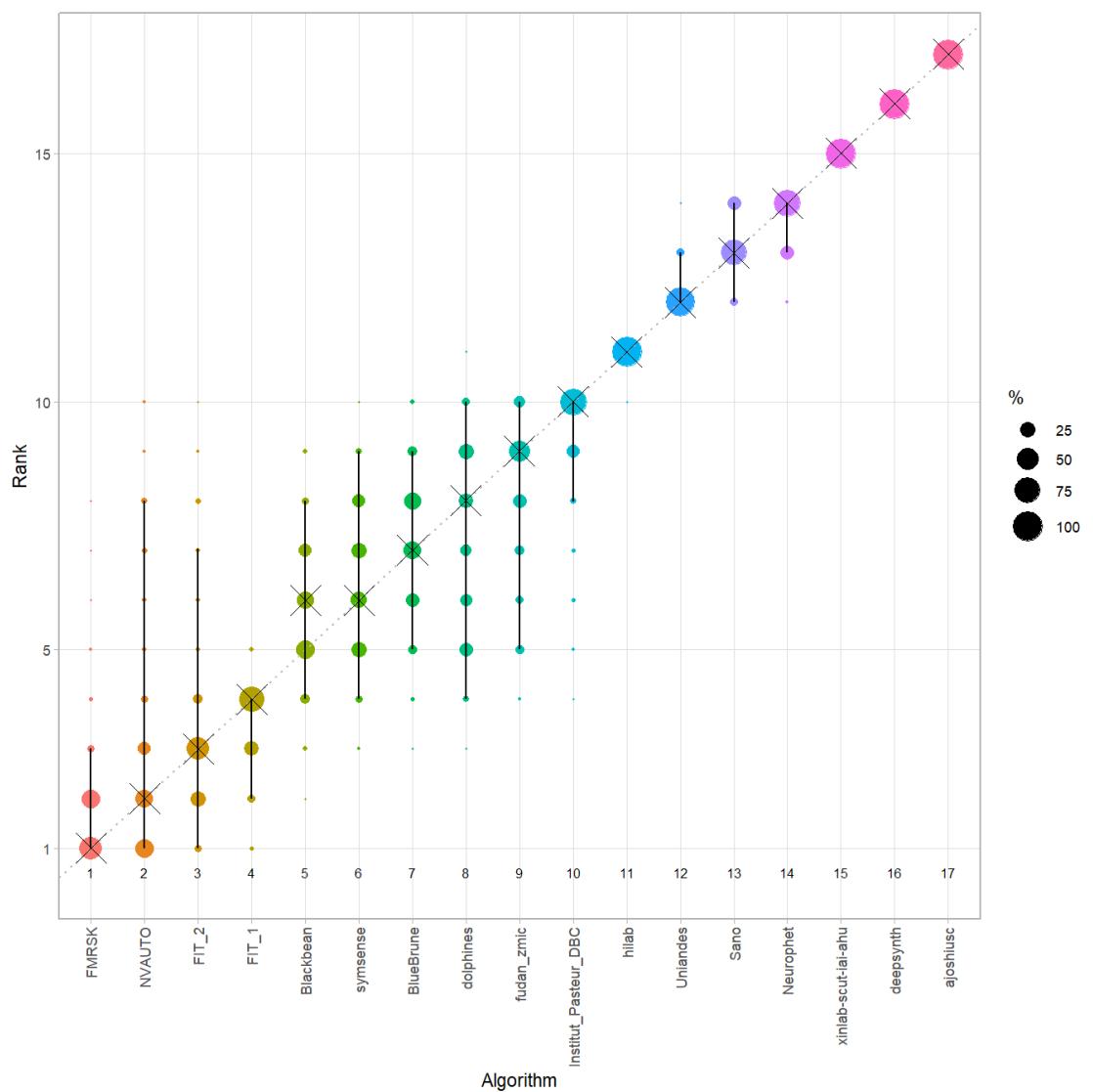


## Visualization of ranking stability

*Blob plot* for visualizing ranking stability based on bootstrap sampling

Algorithms are color-coded, and the area of each blob at position ( $A_i$ ,rank  $j$ ) is proportional to the relative frequency  $A_i$  achieved rank  $j$  across  $b = 1000$  bootstrap samples. The median rank for each algorithm is indicated by a black cross. 95% bootstrap intervals across bootstrap samples are indicated by black lines.

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> = ## "none")` instead.
```

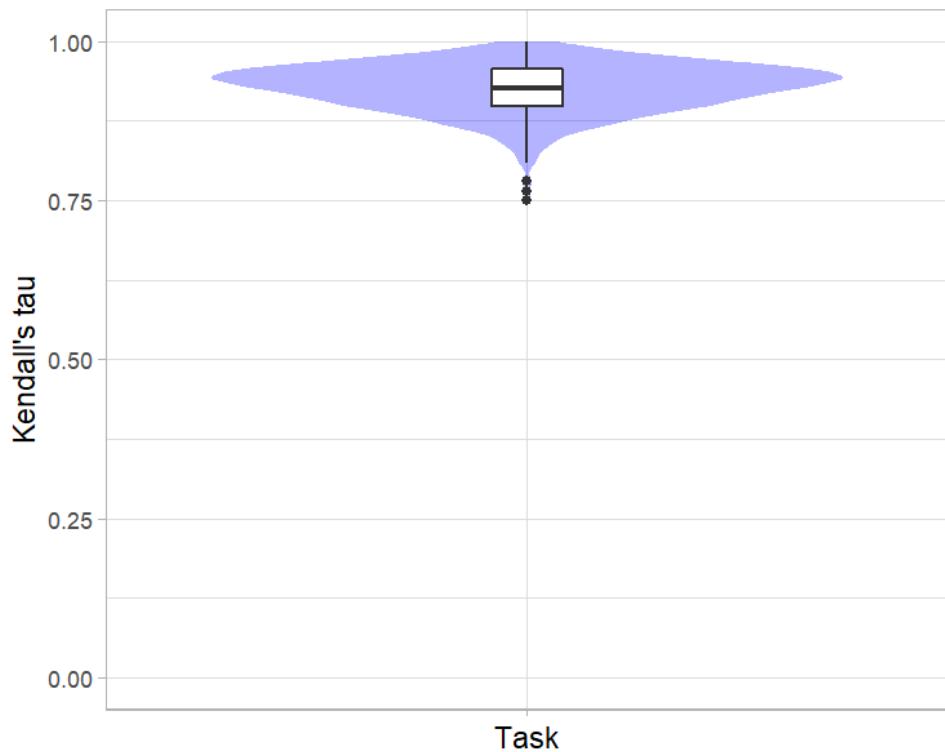


### ***Violin plot for visualizing ranking stability based on bootstrapping***

The ranking list based on the full assessment data is pairwise compared with the ranking lists based on the individual bootstrap samples (here  $b = 1000$  samples). For each pair of rankings, Kendall's  $\tau$  correlation is computed. Kendall's  $\tau$  is a scaled index determining the correlation between the lists. It is computed by evaluating the number of pairwise concordances and discordances between ranking lists and produces values between  $-1$  (for inverted order) and  $1$  (for identical order). A violin plot, which simultaneously depicts a boxplot and a density plot, is generated from the results.

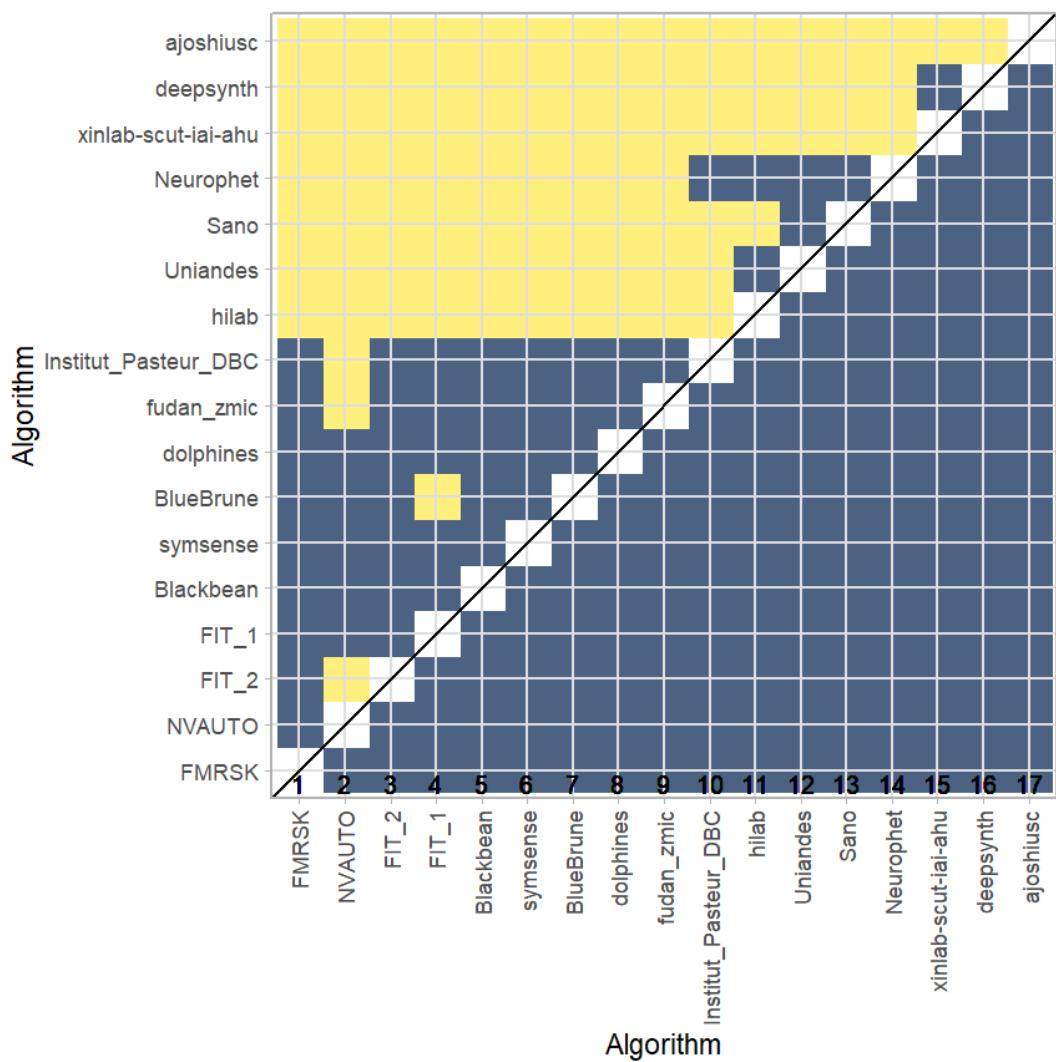
Summary Kendall's tau:

Task	mean	median	q25	q75
dummy-Task	0.9268529	0.9264706	0.8970588	0.9558824



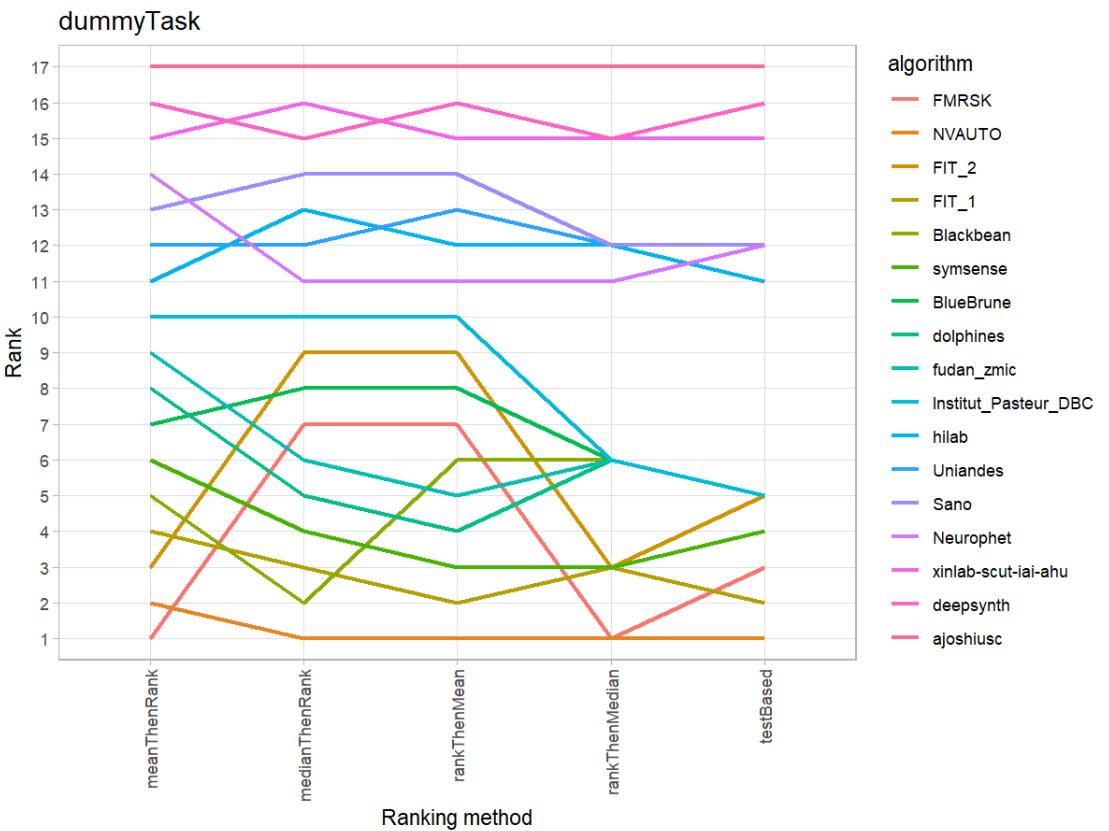
## *Significance maps* for visualizing ranking stability based on statistical significance

*Significance maps* depict incidence matrices of pairwise significant test results for the one-sided Wilcoxon signed rank test at a 5% significance level with adjustment for multiple testing according to Holm. Yellow shading indicates that metric values from the algorithm on the x-axis were significantly superior to those from the algorithm on the y-axis, blue color indicates no significant difference.



### Ranking robustness to ranking methods

*Line plots* for visualizing ranking robustness across different ranking methods. Each algorithm is represented by one colored line. For each ranking method encoded on the x-axis, the height of the line represents the corresponding rank. Horizontal lines indicate identical ranks for all methods.



### 15.3 References

Wiesenfarth, M., Reinke, A., Landman, B.A., Eisenmann, M., Aguilera Saiz, L., Cardoso, M.J., Maier-Hein, L. and Kopp-Schneider, A. Methods and open-source toolkit for analyzing and visualizing challenge results. *Sci Rep* **11**, 2369 (2021). <https://doi.org/10.1038/s41598-021-82017-6>

M. J. A. Eugster, T. Hothorn, and F. Leisch, “Exploratory and inferential analysis of benchmark experiments,” Institut fuer Statistik, Ludwig-Maximilians-Universitaet Muenchen, Germany, Technical Report 30, 2008. [Online]. Available: <http://epub.ub.uni-muenchen.de/4134/>.

## 16 Benchmarking report for Dice Metrics – Good Quality Reconstructions

created by challengeR v1.0.2  
29 September, 2023

This document presents a systematic report on the benchmark study “Dice Metrics”. Input data comprises raw metric values for all algorithms and cases. Generated plots are:

- Visualization of assessment data: Dot- and boxplot, podium plot and ranking heatmap
- Visualization of ranking stability: Blob plot, violin plot and significance map, line plot

Details can be found in Wiesenfarth et al. (2021).

### 16.1 Ranking

Algorithms within a task are ranked according to the following ranking scheme:  
*aggregate using function (“mean”) then rank*

The analysis is based on 17 algorithms and 616 cases. 0 missing cases have been found in the data set.

Ranking:

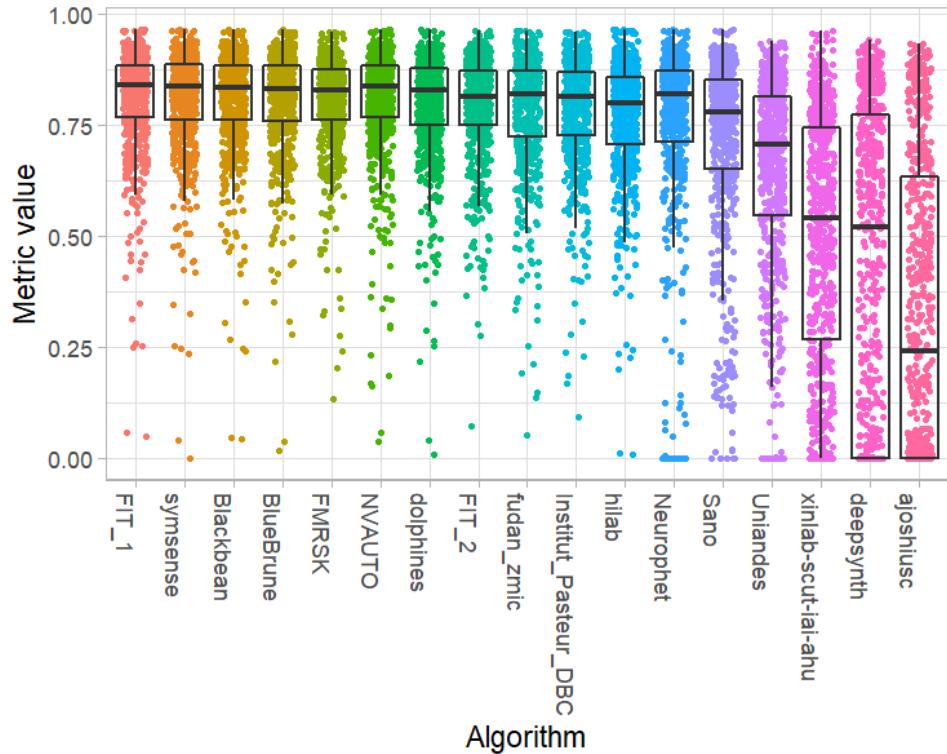
	Dice_mean	rank
FIT_1	0.8102157	1
symsense	0.8075720	2
Blackbean	0.8059347	3
BlueBrune	0.8053248	4
FMRSK	0.8044571	5
NVAUTO	0.8039744	6
dolphines	0.7979522	7
FIT_2	0.7919042	8
fudan_zmic	0.7830144	9

Institut_Pasteur_DBC	0.7826290	10
hilab	0.7675142	11
Neurophet	0.7385599	12
Sano	0.7128628	13
Uniandes	0.6483182	14
xinlab-scut-iai-ahu	0.4965492	15
deepsynth	0.4368088	16
ajoshiusc	0.3279813	17

## 16.2 Visualization of raw assessment data

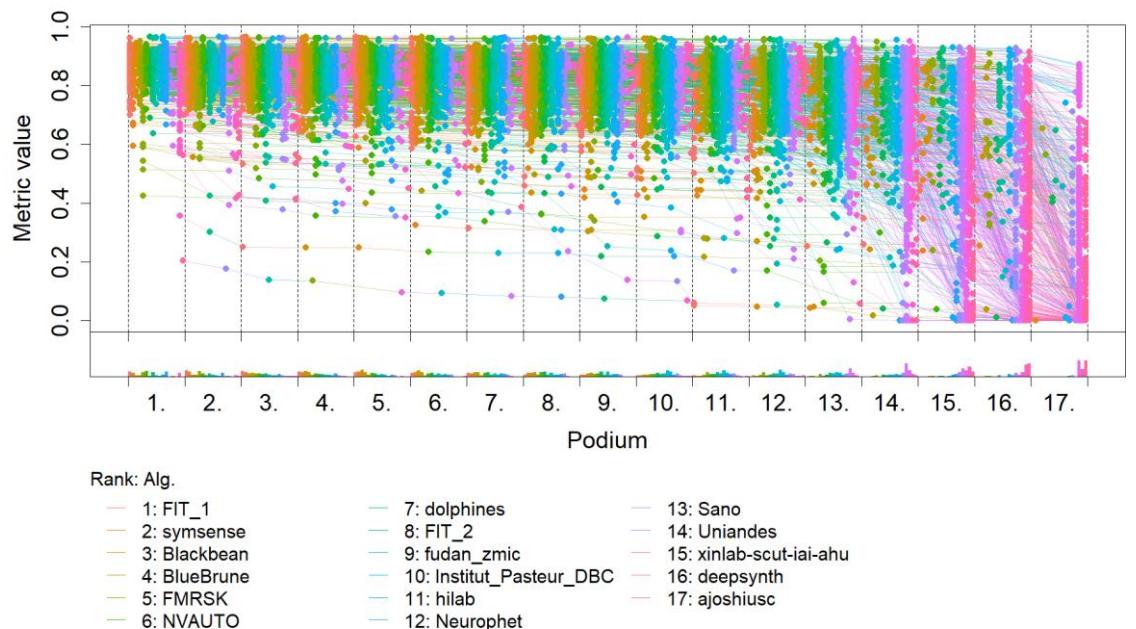
### Dot- and boxplot

*Dot- and boxplots* for visualizing raw assessment data separately for each algorithm. Boxplots representing descriptive statistics over all cases (median, quartiles and outliers) are combined with horizontally jittered dots representing individual cases.



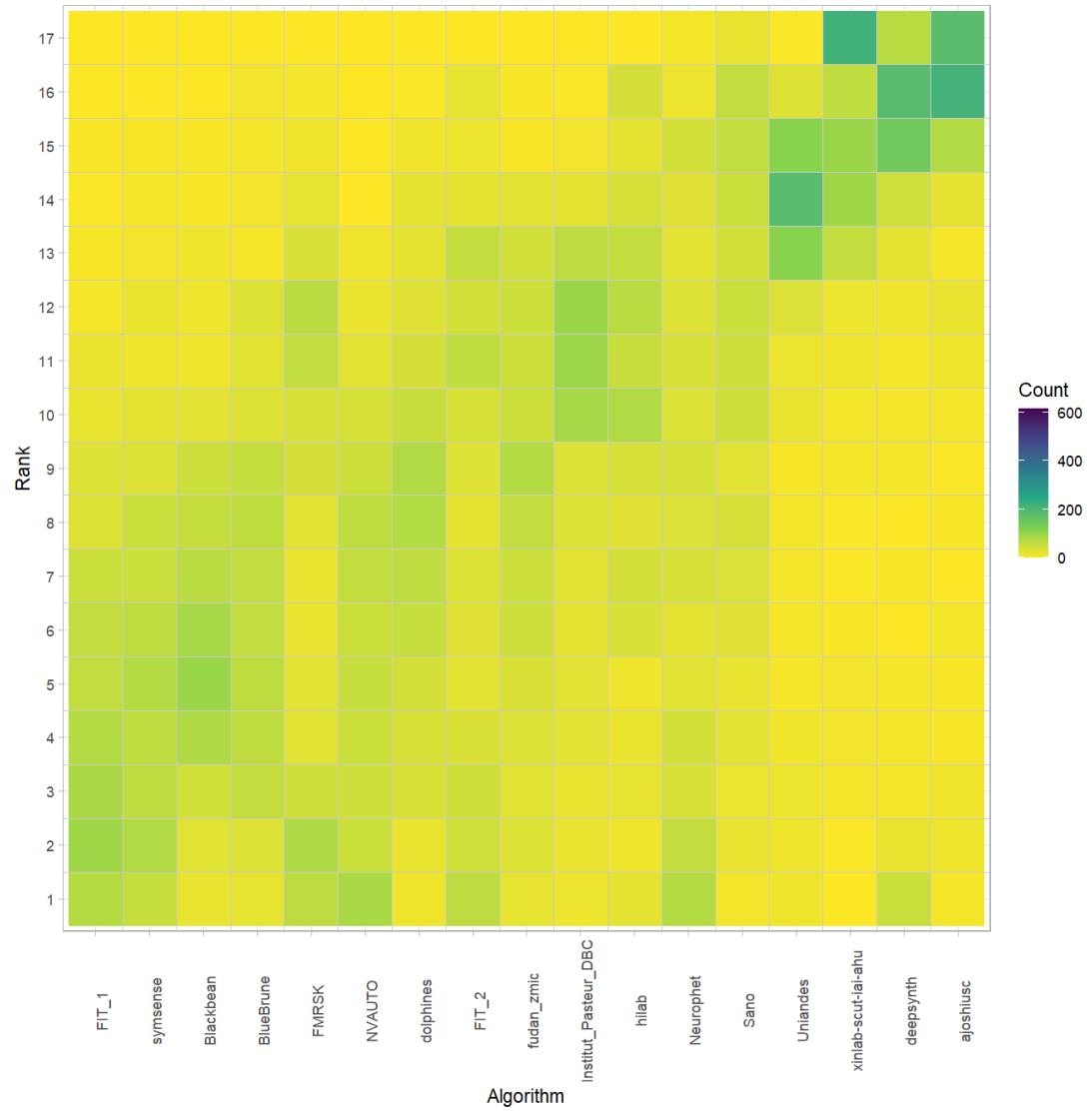
### Podium plot

*Podium plots* (see also Eugster et al., 2008) for visualizing raw assessment data. Upper part (spaghetti plot): Participating algorithms are color-coded, and each colored dot in the plot represents a metric value achieved with the respective algorithm. The actual metric value is encoded by the y-axis. Each podium (here:  $p=17$ ) represents one possible rank, ordered from best (1) to last (here: 17). The assignment of metric values (i.e. colored dots) to one of the podiums is based on the rank that the respective algorithm achieved on the corresponding case. Note that the plot part above each podium place is further subdivided into  $p$  “columns”, where each column represents one participating algorithm (here:  $p = 17$ ). Dots corresponding to identical cases are connected by a line, leading to the shown spaghetti structure. Lower part: Bar charts represent the relative frequency for each algorithm to achieve the rank encoded by the podium place.



## Ranking heatmap

*Ranking heatmaps* for visualizing raw assessment data. Each cell  $(i, A_j)$  shows the absolute frequency of cases in which algorithm  $A_j$  achieved rank  $i$ .

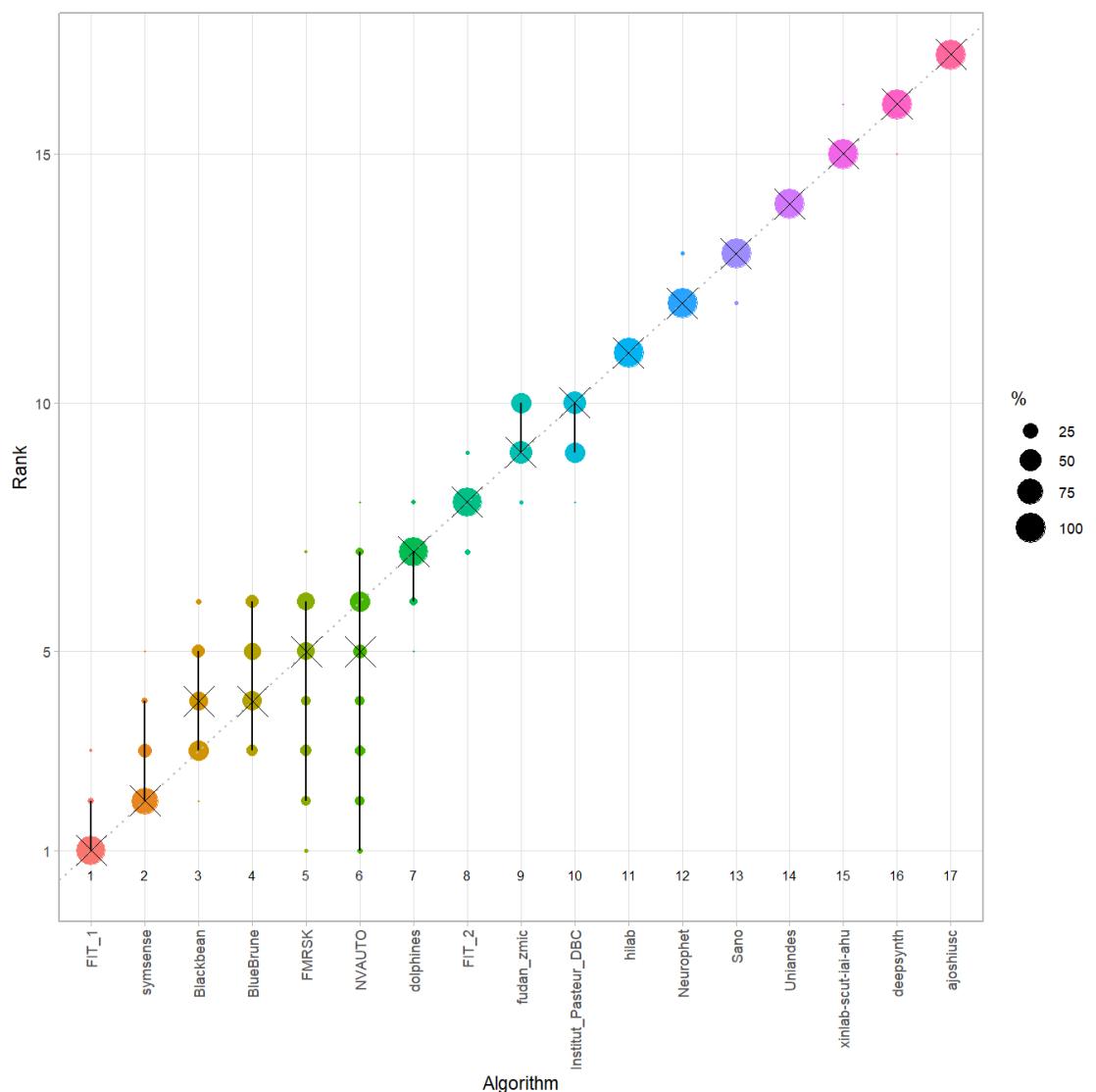


### Visualization of ranking stability

*Blob plot* for visualizing ranking stability based on bootstrap sampling

Algorithms are color-coded, and the area of each blob at position  $(A_i, \text{rank } j)$  is proportional to the relative frequency  $A_i$  achieved rank  $j$  across  $b = 1000$  bootstrap samples. The median rank for each algorithm is indicated by a black cross. 95% bootstrap intervals across bootstrap samples are indicated by black lines.

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.
```



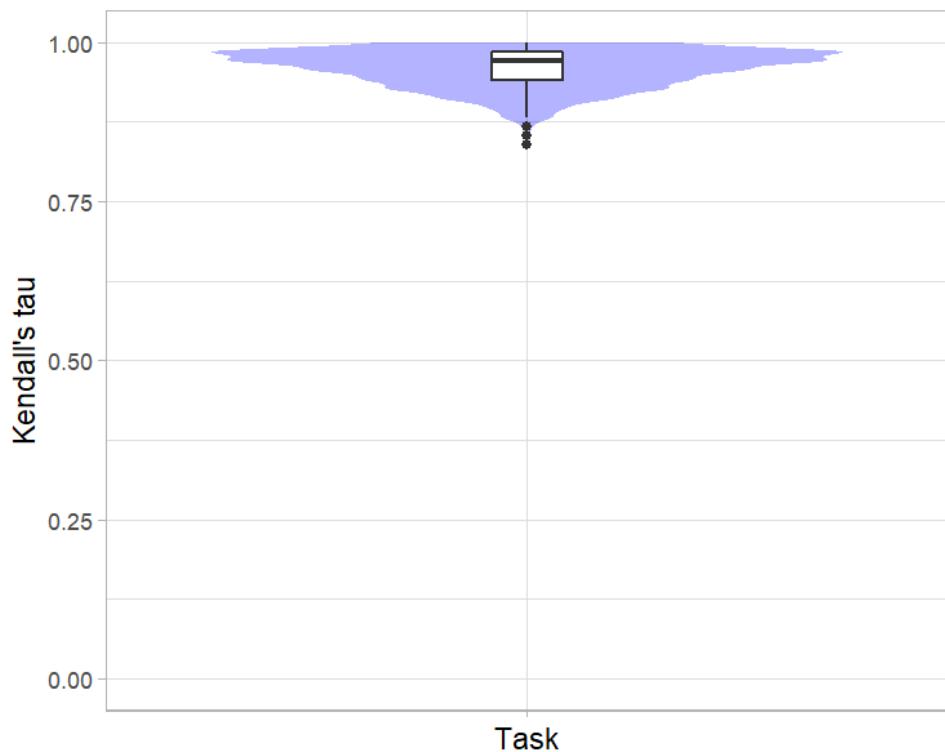
### ***Violin plot for visualizing ranking stability based on bootstrapping***

The ranking list based on the full assessment data is pairwise compared with the ranking lists based on the individual bootstrap samples (here  $b = 1000$  samples). For each pair of rankings, Kendall's  $\tau$  correlation is computed. Kendall's  $\tau$  is a scaled index determining the correlation between the lists. It is computed by evaluating the number of pairwise concordances and discordances between ranking lists and produces values between  $-1$  (for inverted order) and  $1$  (for identical order). A violin plot, which simultaneously depicts a boxplot and a density plot, is generated from the results.

Summary Kendall's tau:

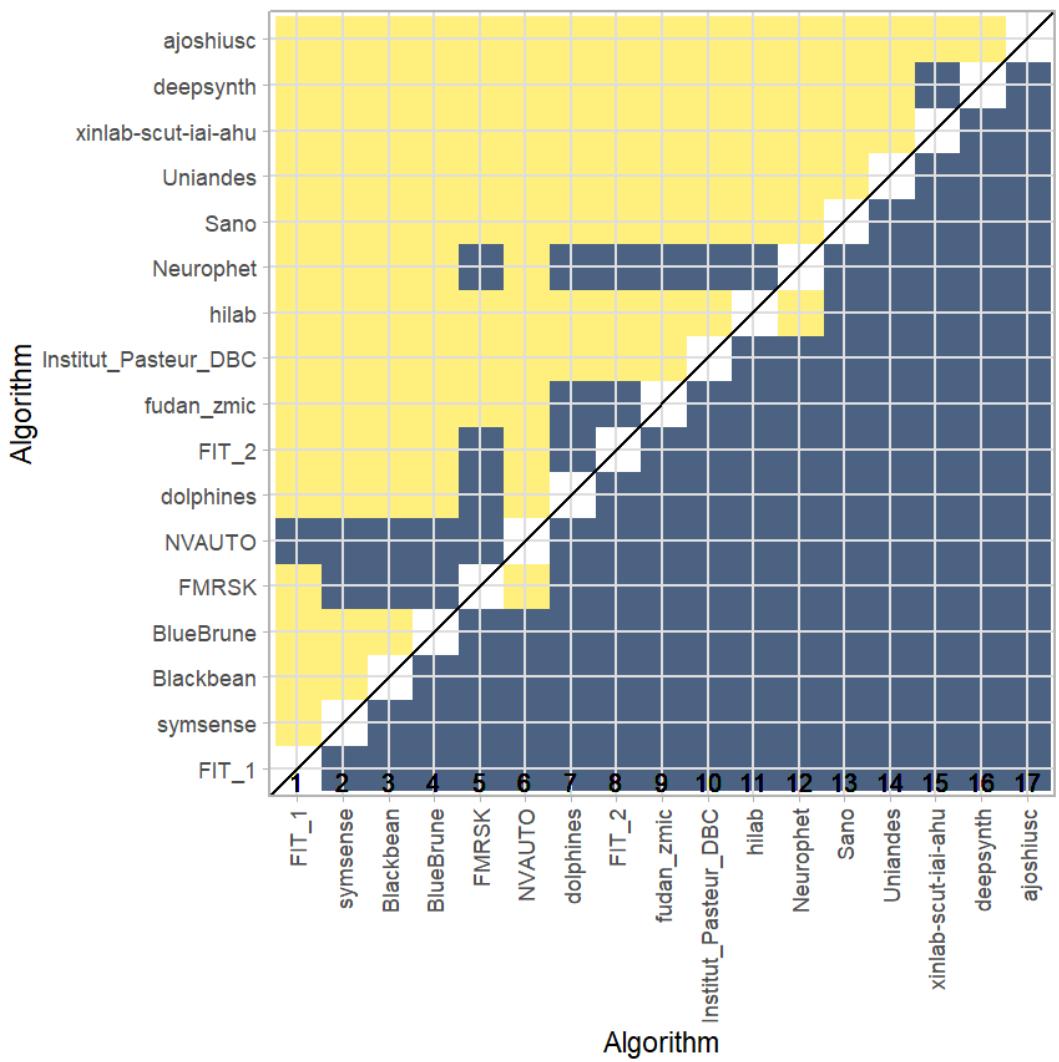
Task	mean	median	q25	q75
dummy-	0.9591912	0.9705882	0.9411765	0.9852941

Task



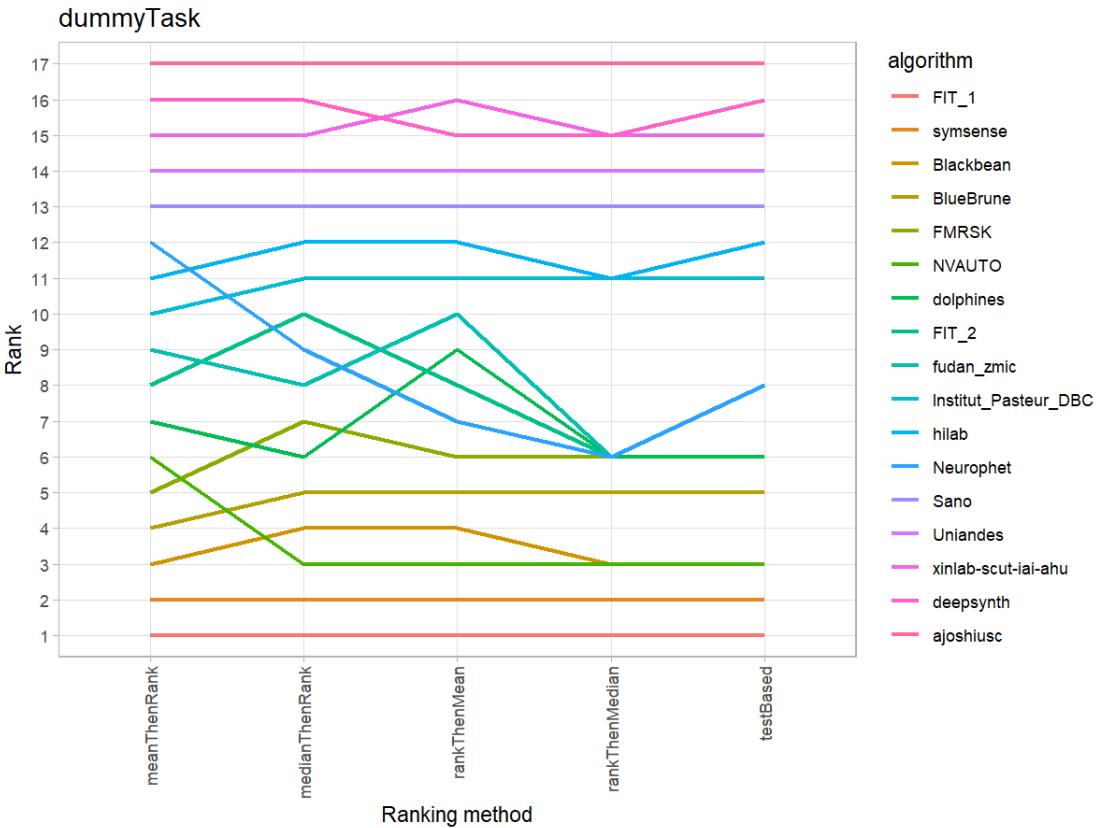
**Significance maps for visualizing ranking stability based on statistical significance**

*Significance maps* depict incidence matrices of pairwise significant test results for the one-sided Wilcoxon signed rank test at a 5% significance level with adjustment for multiple testing according to Holm. Yellow shading indicates that metric values from the algorithm on the x-axis were significantly superior to those from the algorithm on the y-axis, blue color indicates no significant difference.



### Ranking robustness to ranking methods

*Line plots* for visualizing ranking robustness across different ranking methods. Each algorithm is represented by one colored line. For each ranking method encoded on the x-axis, the height of the line represents the corresponding rank. Horizontal lines indicate identical ranks for all methods.



### 16.3 References

Wiesenfarth, M., Reinke, A., Landman, B.A., Eisenmann, M., Aguilera Saiz, L., Cardoso, M.J., Maier-Hein, L. and Kopp-Schneider, A. Methods and open-source toolkit for analyzing and visualizing challenge results. *Sci Rep* **11**, 2369 (2021). <https://doi.org/10.1038/s41598-021-82017-6>

M. J. A. Eugster, T. Hothorn, and F. Leisch, “Exploratory and inferential analysis of benchmark experiments,” Institut fuer Statistik, Ludwig-Maximilians-Universitaet Muenchen, Germany, Technical Report 30, 2008. [Online]. Available: <http://epub.ub.uni-muenchen.de/4134/>.

## 17 Benchmarking report for Hausdorff Metrics – Good Quality Reconstructions

created by challengeR v1.0.2  
29 September, 2023

This document presents a systematic report on the benchmark study “Hausdorff Metrics”. Input data comprises raw metric values for all algorithms and cases. Generated plots are:

- Visualization of assessment data: Dot- and boxplot, podium plot and ranking heatmap
- Visualization of ranking stability: Blob plot, violin plot and significance map, line plot

Details can be found in Wiesenfarth et al. (2021).

### 17.1 Ranking

Algorithms within a task are ranked according to the following ranking scheme:  
*aggregate using function (“mean”) then rank*

The analysis is based on 17 algorithms and 616 cases. 0 missing cases have been found in the data set.

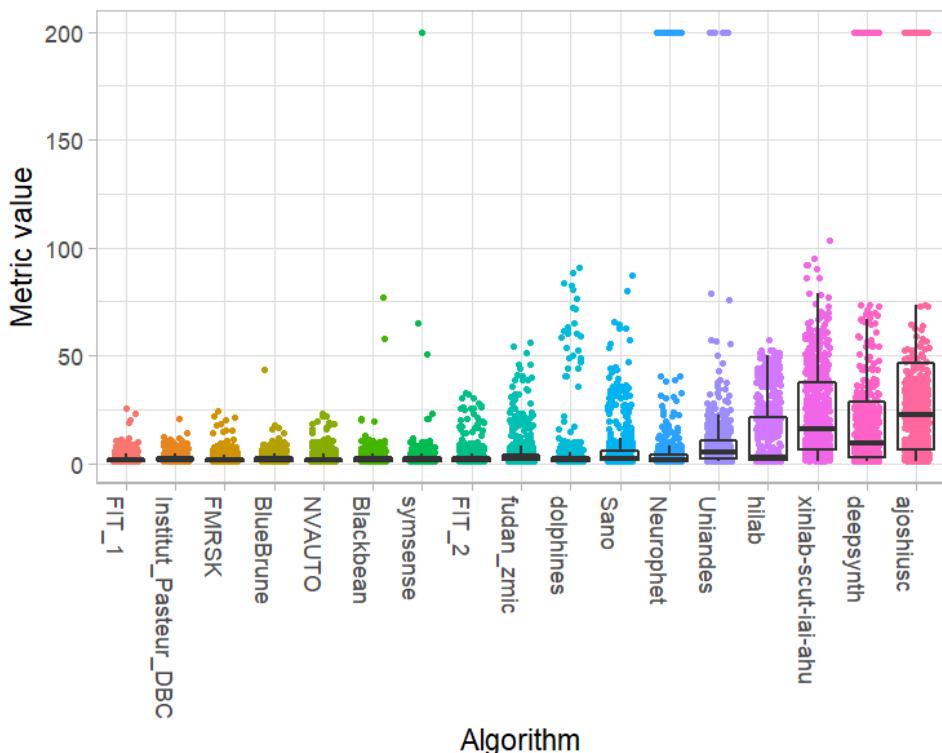
Ranking:

	Hausdorff_mean	rank
FIT_1	2.404649	1
Institut_Pasteur_DBC	2.442461	2
FMRSK	2.461099	3
BlueBrune	2.483286	4
NVAUTO	2.606146	5
Blackbean	2.675704	6
symsense	2.906762	7
FIT_2	3.200410	8
fudan_zmic	4.906101	9
dolphines	5.040367	10
Sano	6.545363	11
Neurophet	11.211864	12
Uniandes	11.850204	13
hilab	12.984241	14
xinlab-scut-iai-ahu	23.097476	15
deepsynth	36.828092	16
ajoshiusc	53.840463	17

## 17.2 Visualization of raw assessment data

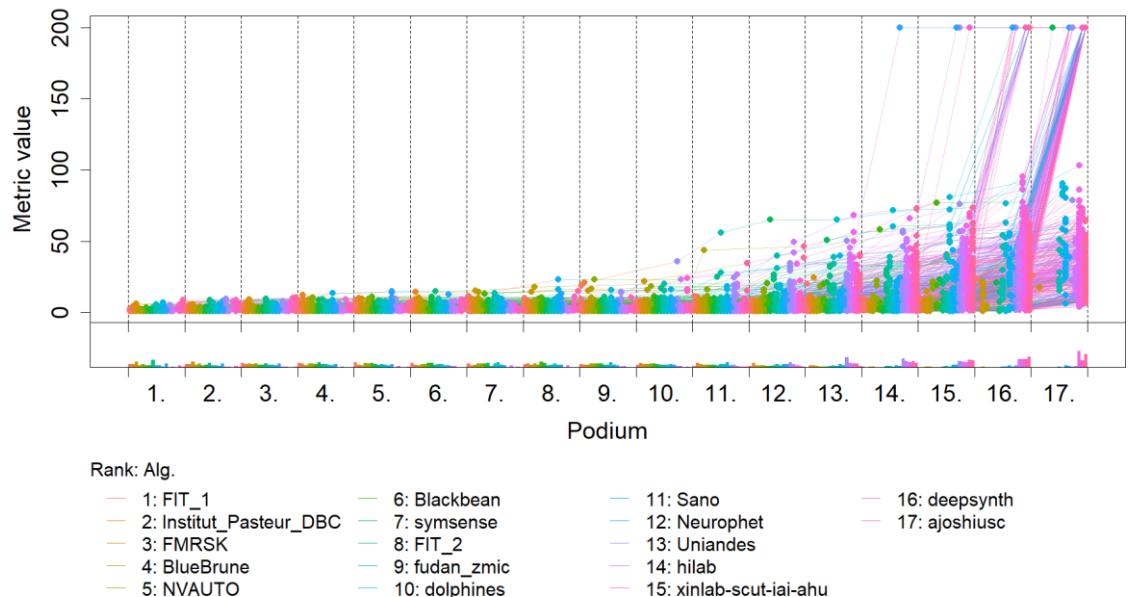
### Dot- and boxplot

*Dot- and boxplots* for visualizing raw assessment data separately for each algorithm. Boxplots representing descriptive statistics over all cases (median, quartiles and outliers) are combined with horizontally jittered dots representing individual cases.



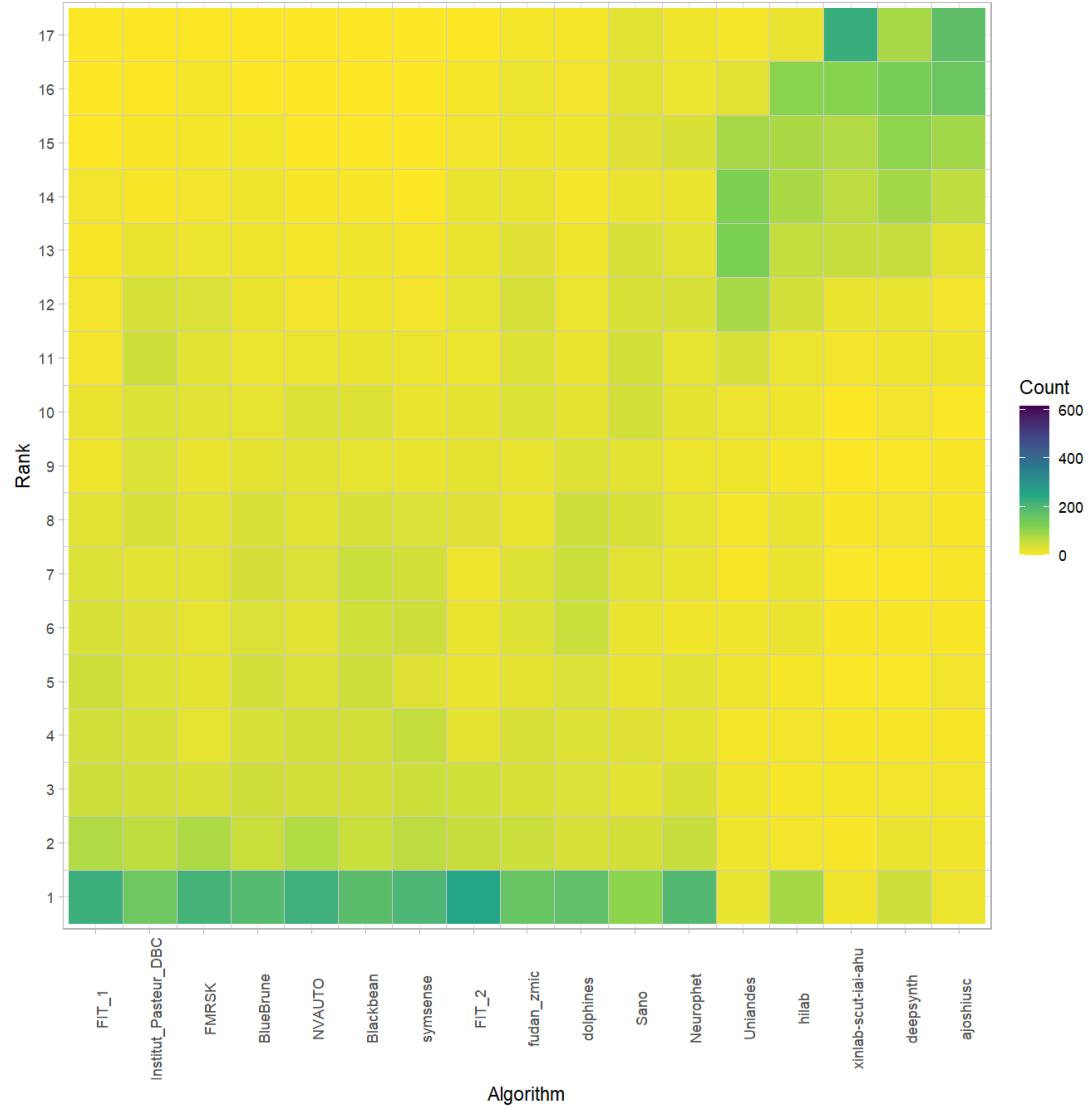
### Podium plot

*Podium plots* (see also Eugster et al., 2008) for visualizing raw assessment data. Upper part (spaghetti plot): Participating algorithms are color-coded, and each colored dot in the plot represents a metric value achieved with the respective algorithm. The actual metric value is encoded by the y-axis. Each podium (here:  $p=17$ ) represents one possible rank, ordered from best (1) to last (here: 17). The assignment of metric values (i.e. colored dots) to one of the podiums is based on the rank that the respective algorithm achieved on the corresponding case. Note that the plot part above each podium place is further subdivided into  $p$  “columns”, where each column represents one participating algorithm (here:  $p = 17$ ). Dots corresponding to identical cases are connected by a line, leading to the shown spaghetti structure. Lower part: Bar charts represent the relative frequency for each algorithm to achieve the rank encoded by the podium place.



## Ranking heatmap

*Ranking heatmaps* for visualizing raw assessment data. Each cell  $(i, A_j)$  shows the absolute frequency of cases in which algorithm  $A_j$  achieved rank  $i$ .

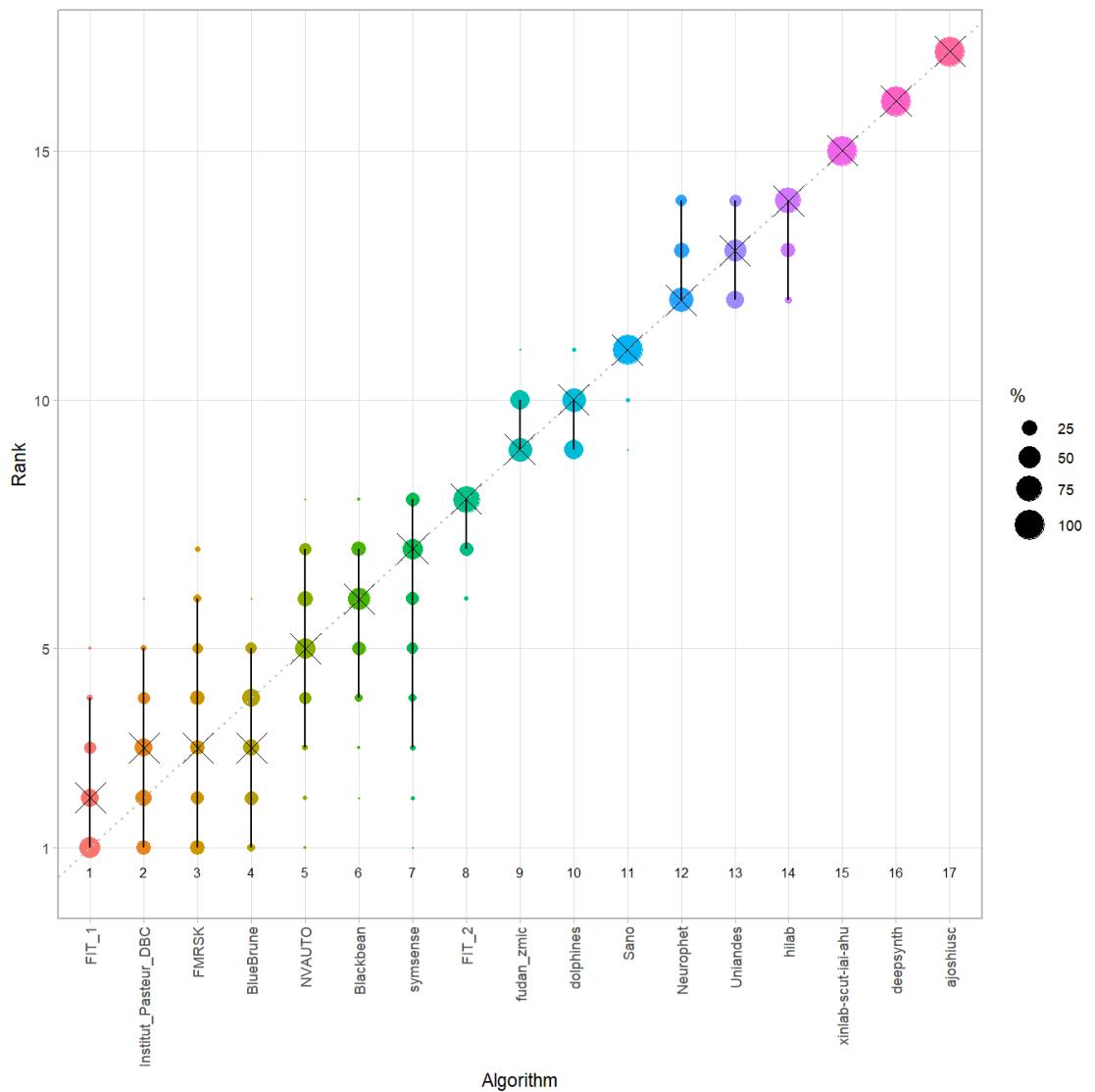


### Visualization of ranking stability

*Blob plot* for visualizing ranking stability based on bootstrap sampling

Algorithms are color-coded, and the area of each blob at position  $(A_i, \text{rank } j)$  is proportional to the relative frequency  $A_i$  achieved rank  $j$  across  $b = 1000$  bootstrap samples. The median rank for each algorithm is indicated by a black cross. 95% bootstrap intervals across bootstrap samples are indicated by black lines.

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.
```



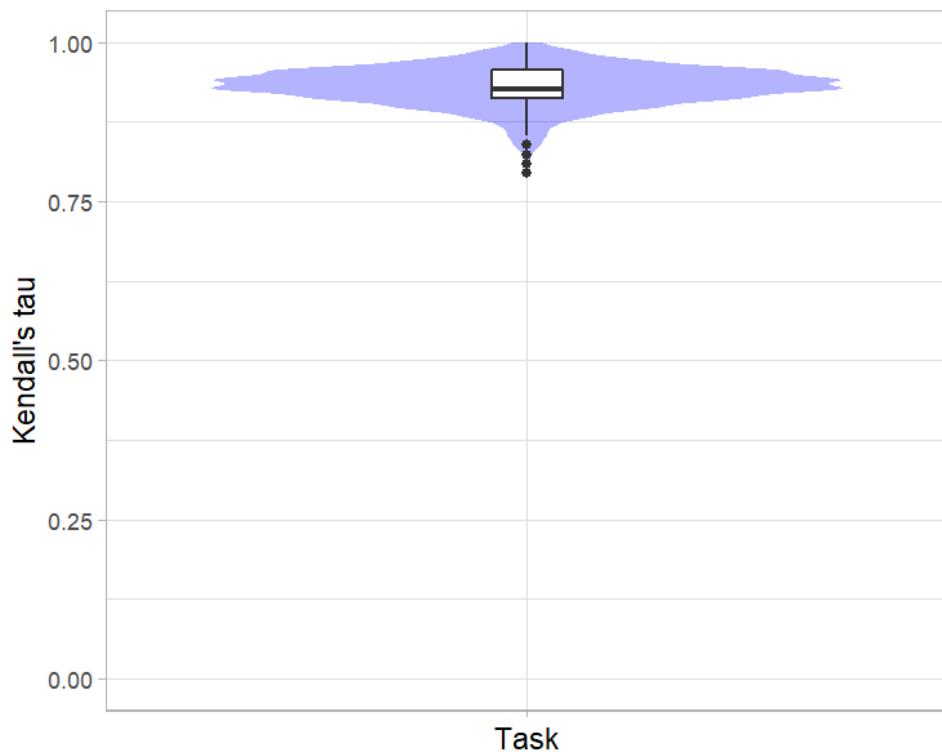
### ***Violin plot for visualizing ranking stability based on bootstrapping***

The ranking list based on the full assessment data is pairwise compared with the ranking lists based on the individual bootstrap samples (here  $b = 1000$  samples). For each pair of rankings, Kendall's  $\tau$  correlation is computed. Kendall's  $\tau$  is a scaled index determining the correlation between the lists. It is computed by evaluating the number of pairwise concordances and discordances between ranking lists and produces values between  $-1$  (for inverted order) and  $1$  (for identical order). A violin plot, which simultaneously depicts a boxplot and a density plot, is generated from the results.

Summary Kendall's tau:

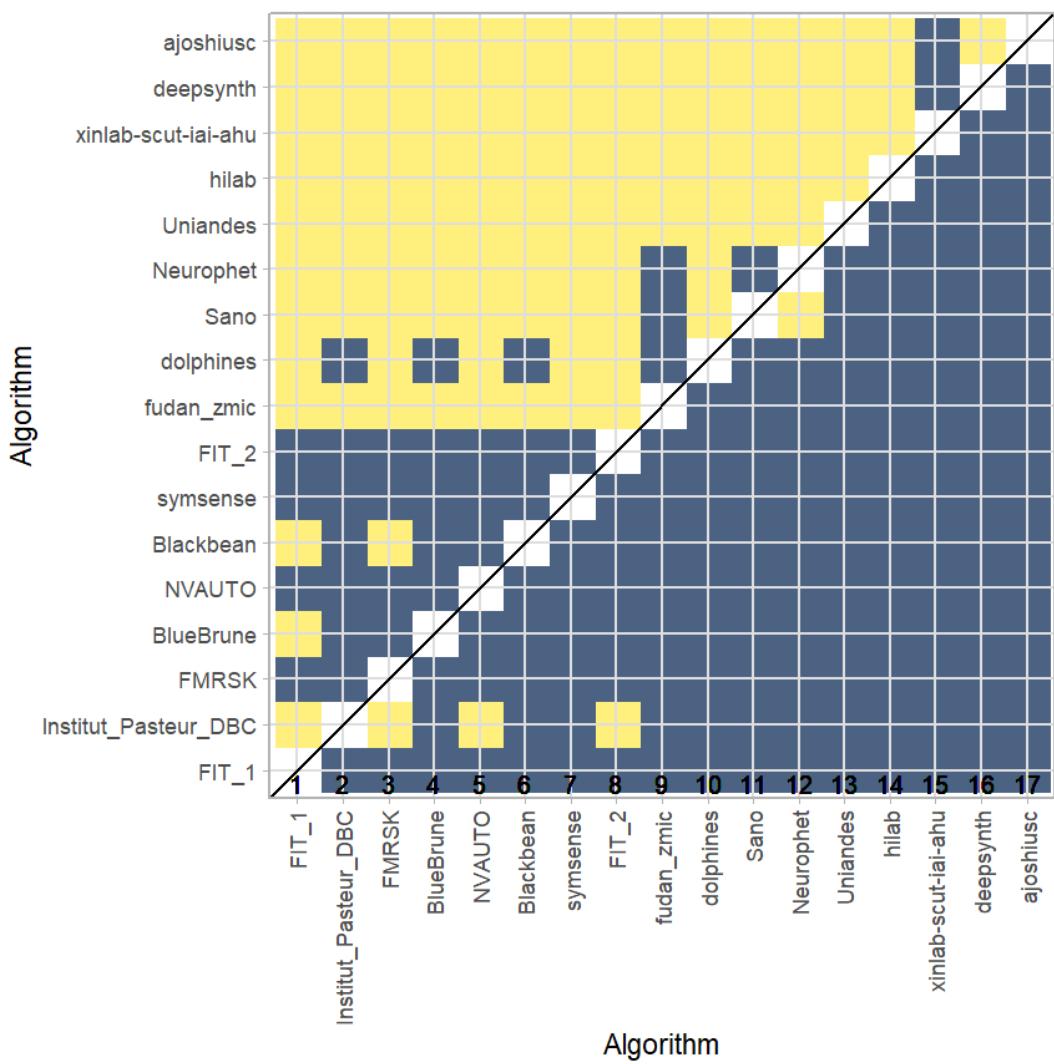
Task	mean	median	q25	q75
dummy-	0.9306471	0.9264706	0.9117647	0.9558824

Task



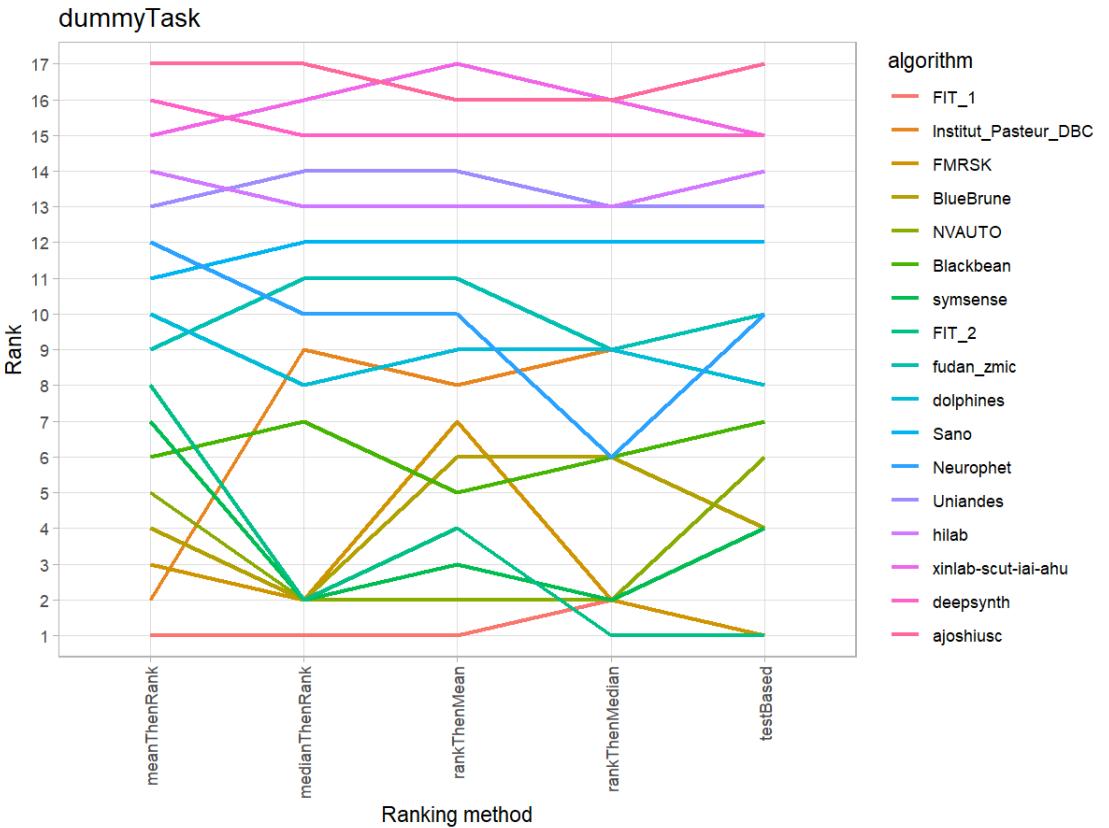
## *Significance maps* for visualizing ranking stability based on statistical significance

*Significance maps* depict incidence matrices of pairwise significant test results for the one-sided Wilcoxon signed rank test at a 5% significance level with adjustment for multiple testing according to Holm. Yellow shading indicates that metric values from the algorithm on the x-axis were significantly superior to those from the algorithm on the y-axis, blue color indicates no significant difference.



### Ranking robustness to ranking methods

*Line plots* for visualizing ranking robustness across different ranking methods. Each algorithm is represented by one colored line. For each ranking method encoded on the x-axis, the height of the line represents the corresponding rank. Horizontal lines indicate identical ranks for all methods.



### 17.3 References

Wiesenfarth, M., Reinke, A., Landman, B.A., Eisenmann, M., Aguilera Saiz, L., Cardoso, M.J., Maier-Hein, L. and Kopp-Schneider, A. Methods and open-source toolkit for analyzing and visualizing challenge results. *Sci Rep* **11**, 2369 (2021). <https://doi.org/10.1038/s41598-021-82017-6>

M. J. A. Eugster, T. Hothorn, and F. Leisch, “Exploratory and inferential analysis of benchmark experiments,” Institut fuer Statistik, Ludwig-Maximilians-Universitaet Muenchen, Germany, Technical Report 30, 2008. [Online]. Available: <http://epub.ub.uni-muenchen.de/4134/>.

## 18 Benchmarking report for Volume Similarity Metrics – Good Quality Reconstructions

created by challengeR v1.0.2

29 September, 2023

This document presents a systematic report on the benchmark study “Volume Similarity Metrics”. Input data comprises raw metric values for all algorithms and cases. Generated plots are:

- Visualization of assessment data: Dot- and boxplot, podium plot and ranking heatmap
- Visualization of ranking stability: Blob plot, violin plot and significance map, line plot

Details can be found in Wiesenfarth et al. (2021).

### 18.1 Ranking

Algorithms within a task are ranked according to the following ranking scheme:

*aggregate using function (“mean”) then rank*

The analysis is based on 17 algorithms and 616 cases. 0 missing cases have been found in the data set.

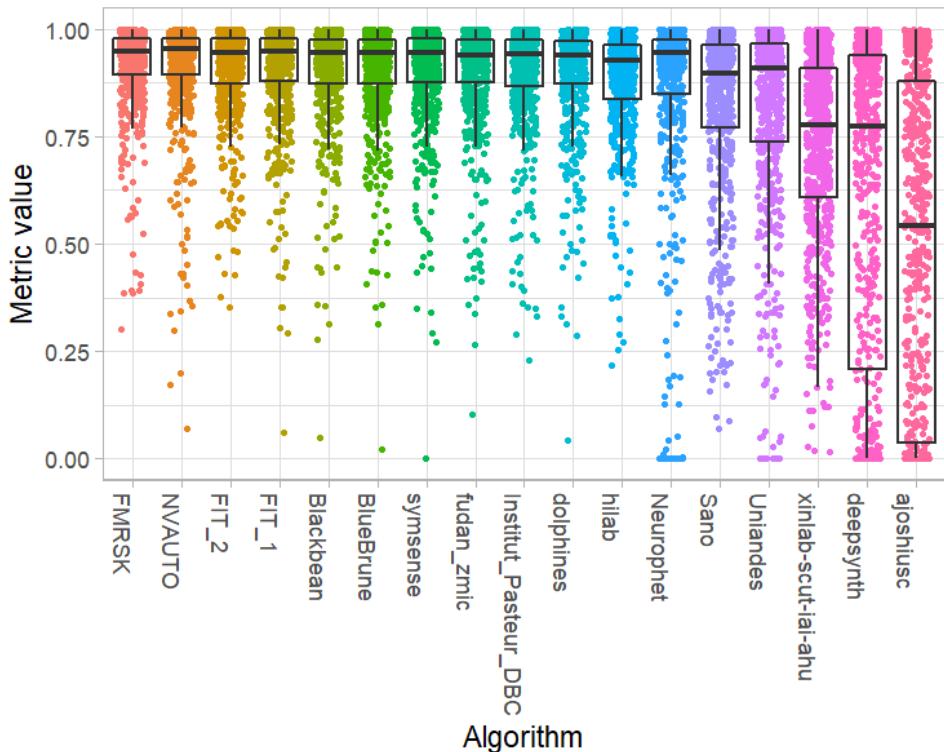
**Ranking:**

	Volume_Similarity_mean	rank
FMRSK	0.9172204	1
NVAUTO	0.9097767	2
FIT_2	0.9075901	3
FIT_1	0.9063503	4
Blackbean	0.9050488	5
BlueBrune	0.9032744	6
symsense	0.9031506	7
fudan_zmic	0.8996209	8
Institut_Pasteur_DBC	0.8989827	9
dolphins	0.8987604	10
hilab	0.8828352	11
Neurophet	0.8429007	12
Sano	0.8234222	13
Uniandes	0.8104665	14
xinlab-scut-iai-ahu	0.7322457	15
deepsynth	0.6032616	16
ajoshiusc	0.4944589	17

## 18.2 Visualization of raw assessment data

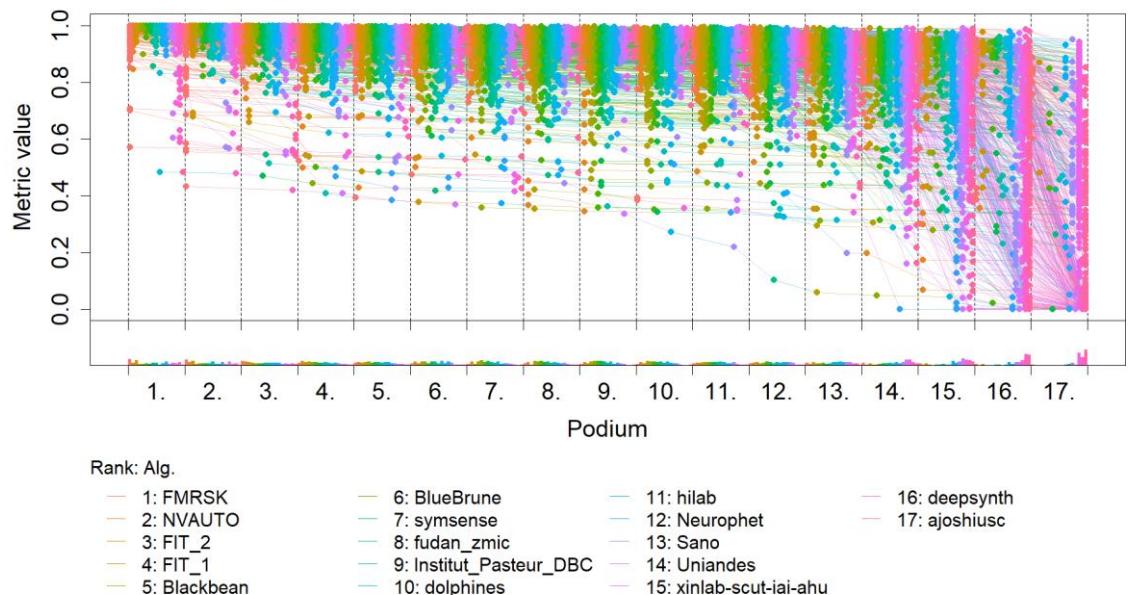
### Dot- and boxplot

*Dot- and boxplots* for visualizing raw assessment data separately for each algorithm. Boxplots representing descriptive statistics over all cases (median, quartiles and outliers) are combined with horizontally jittered dots representing individual cases.



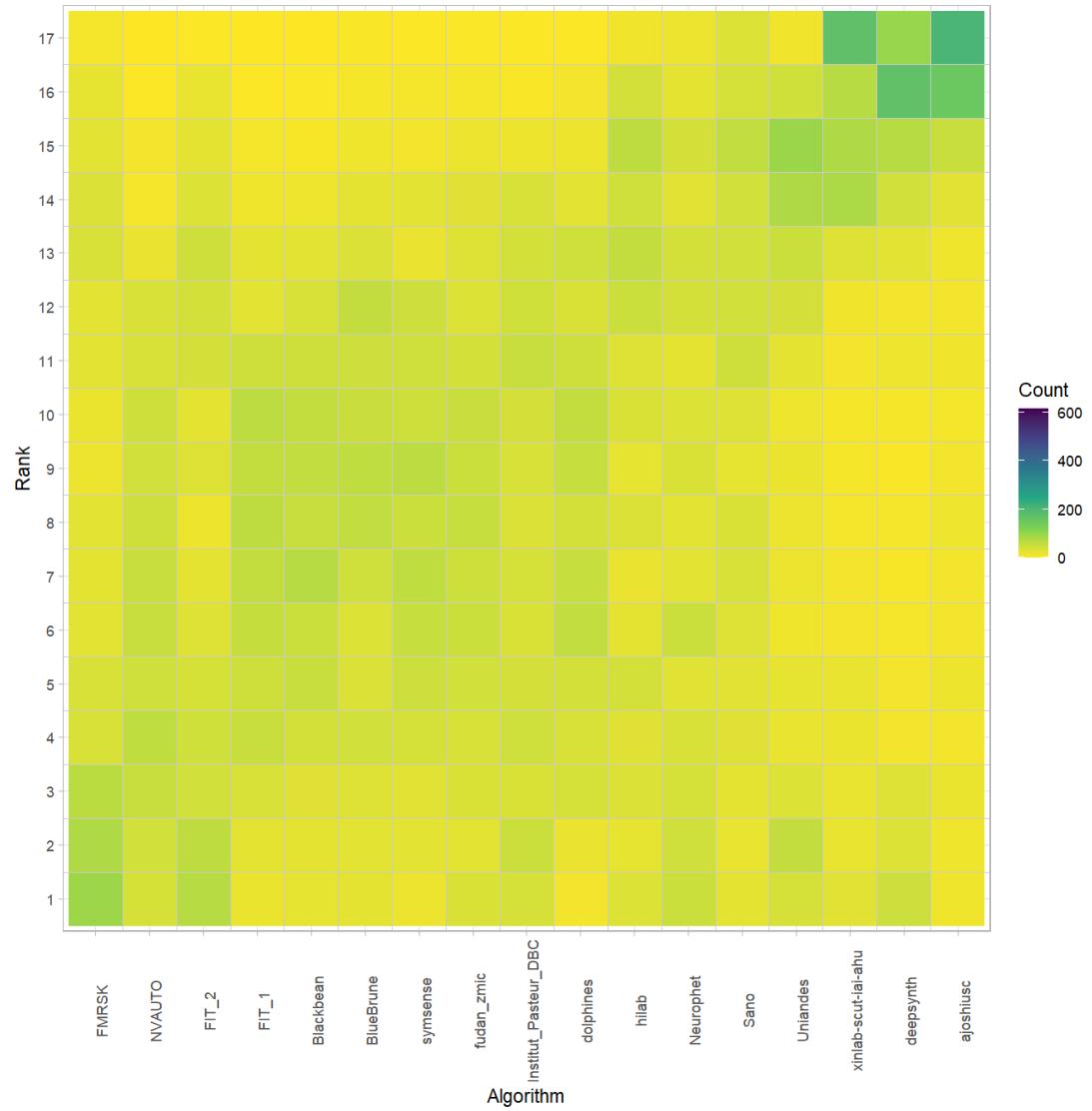
## Podium plot

*Podium plots* (see also Eugster et al., 2008) for visualizing raw assessment data. Upper part (spaghetti plot): Participating algorithms are color-coded, and each colored dot in the plot represents a metric value achieved with the respective algorithm. The actual metric value is encoded by the y-axis. Each podium (here:  $p=17$ ) represents one possible rank, ordered from best (1) to last (here: 17). The assignment of metric values (i.e. colored dots) to one of the podiums is based on the rank that the respective algorithm achieved on the corresponding case. Note that the plot part above each podium place is further subdivided into  $p$  “columns”, where each column represents one participating algorithm (here:  $p = 17$ ). Dots corresponding to identical cases are connected by a line, leading to the shown spaghetti structure. Lower part: Bar charts represent the relative frequency for each algorithm to achieve the rank encoded by the podium place.



### Ranking heatmap

*Ranking heatmaps* for visualizing raw assessment data. Each cell  $(i, A_j)$  shows the absolute frequency of cases in which algorithm  $A_j$  achieved rank  $i$ .

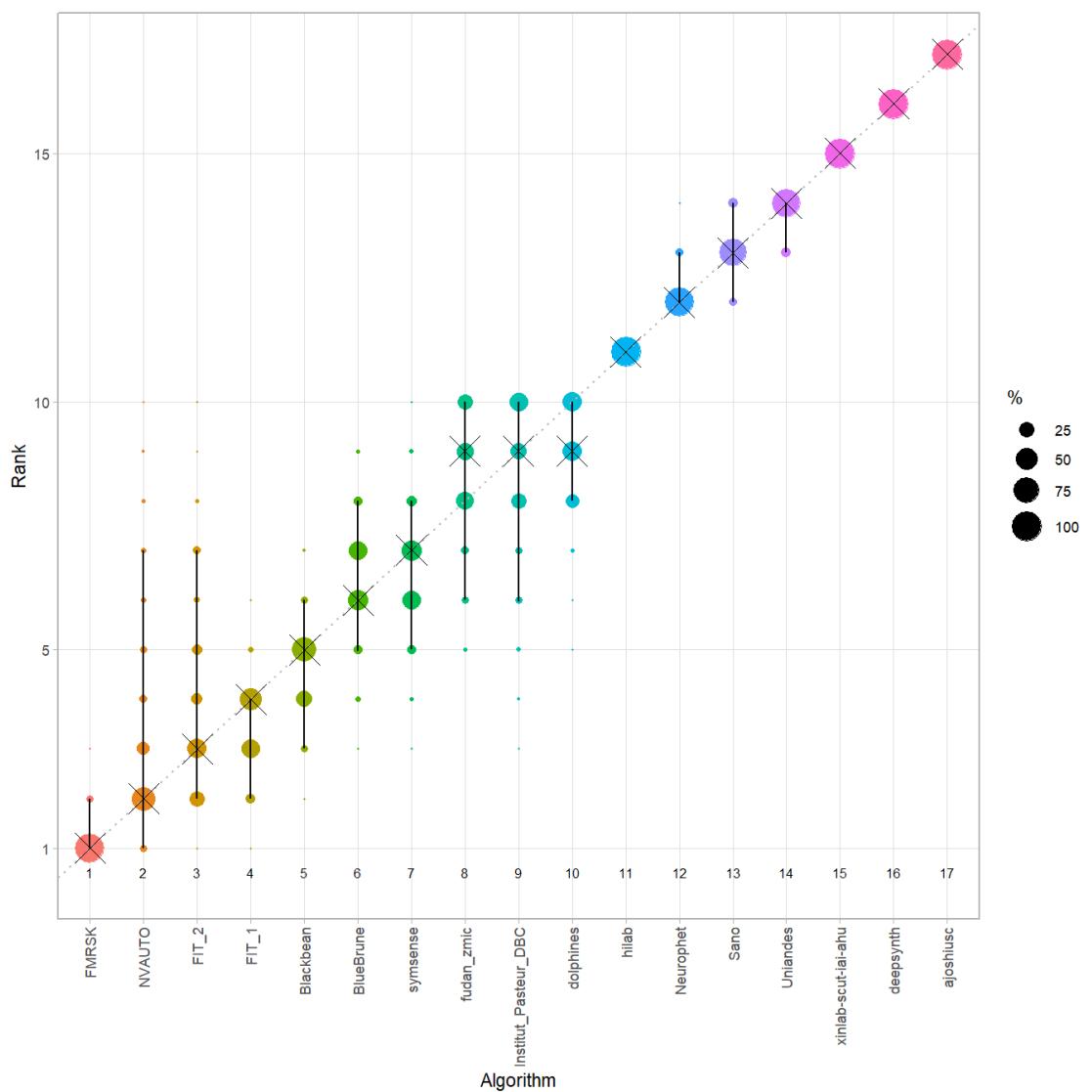


### Visualization of ranking stability

*Blob plot* for visualizing ranking stability based on bootstrap sampling

Algorithms are color-coded, and the area of each blob at position  $(A_i, \text{rank } j)$  is proportional to the relative frequency  $A_i$  achieved rank  $j$  across  $b = 1000$  bootstrap samples. The median rank for each algorithm is indicated by a black cross. 95% bootstrap intervals across bootstrap samples are indicated by black lines.

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.
```

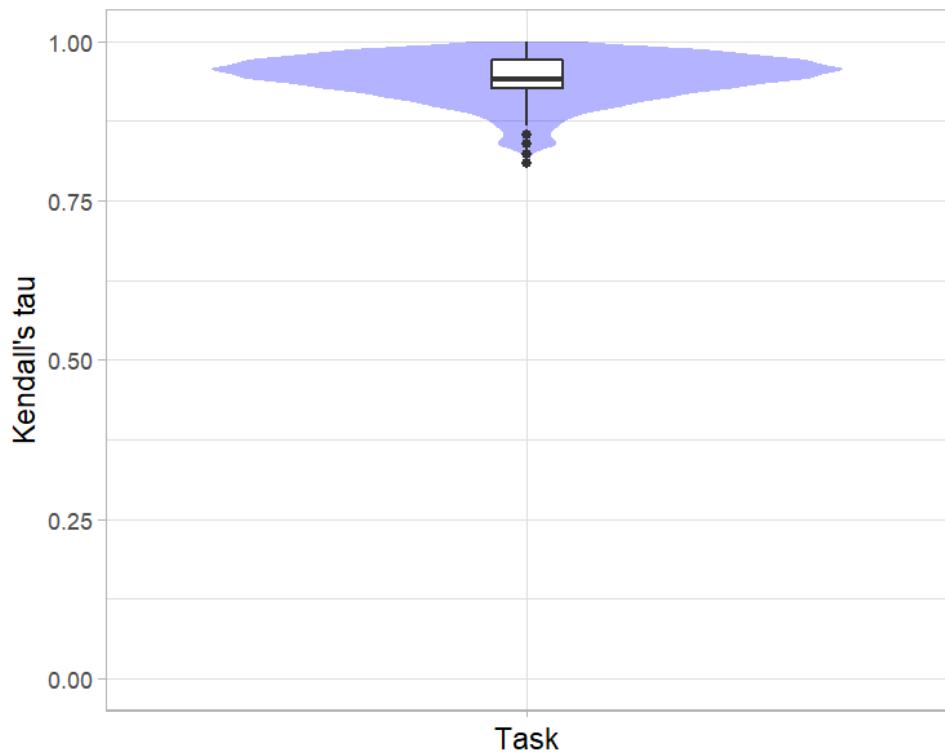


### ***Violin plot for visualizing ranking stability based on bootstrapping***

The ranking list based on the full assessment data is pairwise compared with the ranking lists based on the individual bootstrap samples (here  $b = 1000$  samples). For each pair of rankings, Kendall's  $\tau$  correlation is computed. Kendall's  $\tau$  is a scaled index determining the correlation between the lists. It is computed by evaluating the number of pairwise concordances and discordances between ranking lists and produces values between  $-1$  (for inverted order) and  $1$  (for identical order). A violin plot, which simultaneously depicts a boxplot and a density plot, is generated from the results.

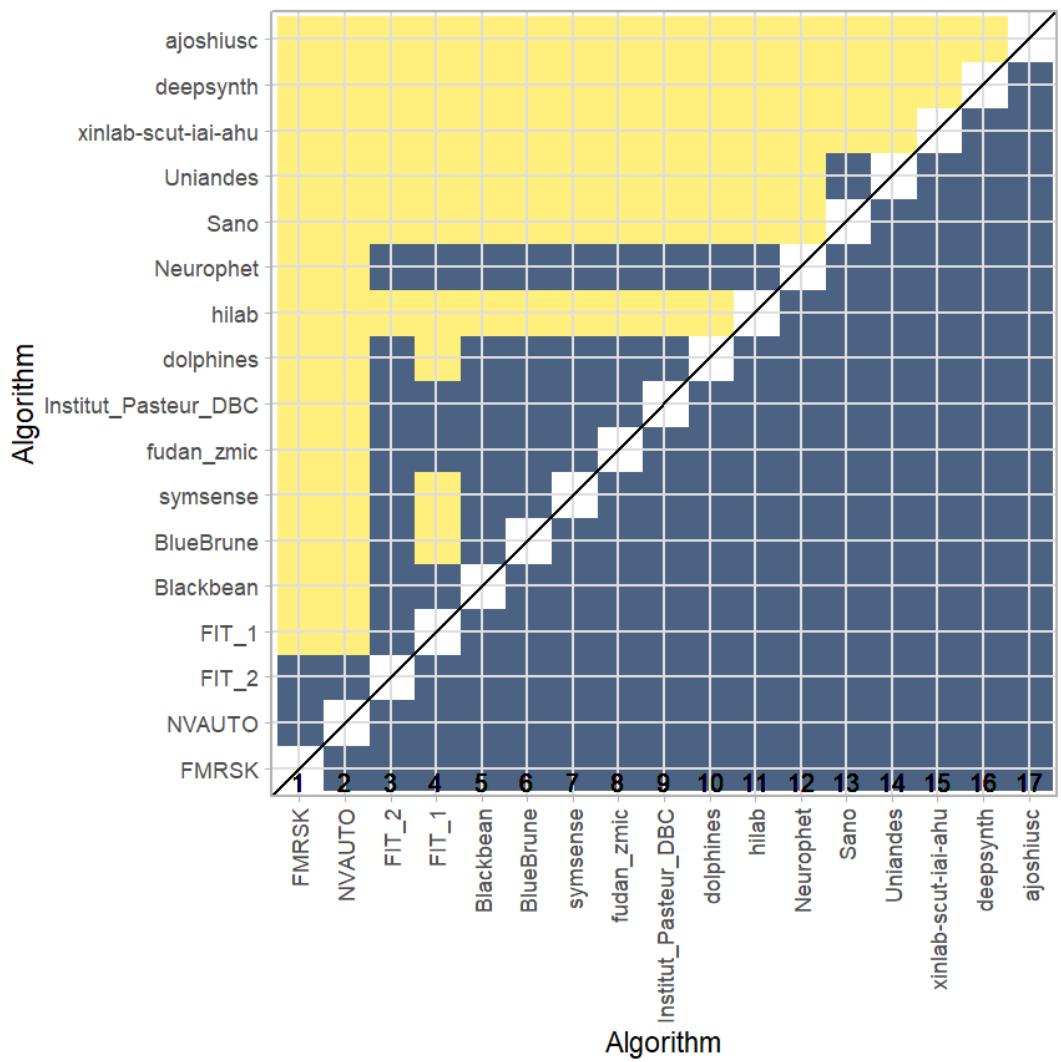
Summary Kendall's tau:

Task	mean	median	q25	q75
dummyTask	0.9421029	0.9411765	0.9264706	0.9705882



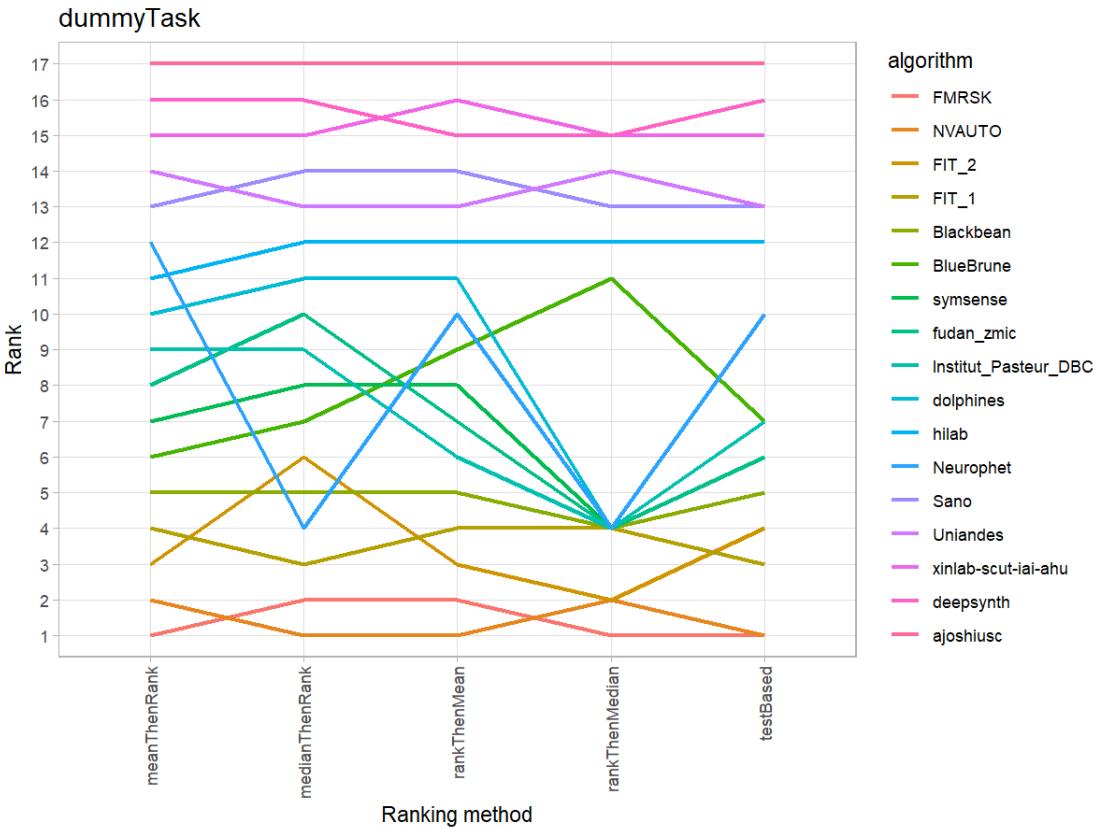
## *Significance maps* for visualizing ranking stability based on statistical significance

*Significance maps* depict incidence matrices of pairwise significant test results for the one-sided Wilcoxon signed rank test at a 5% significance level with adjustment for multiple testing according to Holm. Yellow shading indicates that metric values from the algorithm on the x-axis were significantly superior to those from the algorithm on the y-axis, blue color indicates no significant difference.



### Ranking robustness to ranking methods

*Line plots* for visualizing ranking robustness across different ranking methods. Each algorithm is represented by one colored line. For each ranking method encoded on the x-axis, the height of the line represents the corresponding rank. Horizontal lines indicate identical ranks for all methods.



### 18.3 References

Wiesenfarth, M., Reinke, A., Landman, B.A., Eisenmann, M., Aguilera Saiz, L., Cardoso, M.J., Maier-Hein, L. and Kopp-Schneider, A. Methods and open-source toolkit for analyzing and visualizing challenge results. *Sci Rep* **11**, 2369 (2021). <https://doi.org/10.1038/s41598-021-82017-6>

M. J. A. Eugster, T. Hothorn, and F. Leisch, “Exploratory and inferential analysis of benchmark experiments,” Institut fuer Statistik, Ludwig-Maximilians-Universitaet Muenchen, Germany, Technical Report 30, 2008. [Online]. Available: <http://epub.ub.uni-muenchen.de/4134/>.

## 19 Benchmarking report for Dice Metrics – Poor Quality Reconstructions

created by challengeR v1.0.2  
29 September, 2023

This document presents a systematic report on the benchmark study “Dice Metrics”. Input data comprises raw metric values for all algorithms and cases. Generated plots are:

- Visualization of assessment data: Dot- and boxplot, podium plot and ranking heatmap
- Visualization of ranking stability: Blob plot, violin plot and significance map, line plot

Details can be found in Wiesenfarth et al. (2021).

### 19.1 Ranking

Algorithms within a task are ranked according to the following ranking scheme:  
*aggregate using function (“mean”) then rank*

The analysis is based on 17 algorithms and 140 cases. 0 missing cases have been found in the data set.

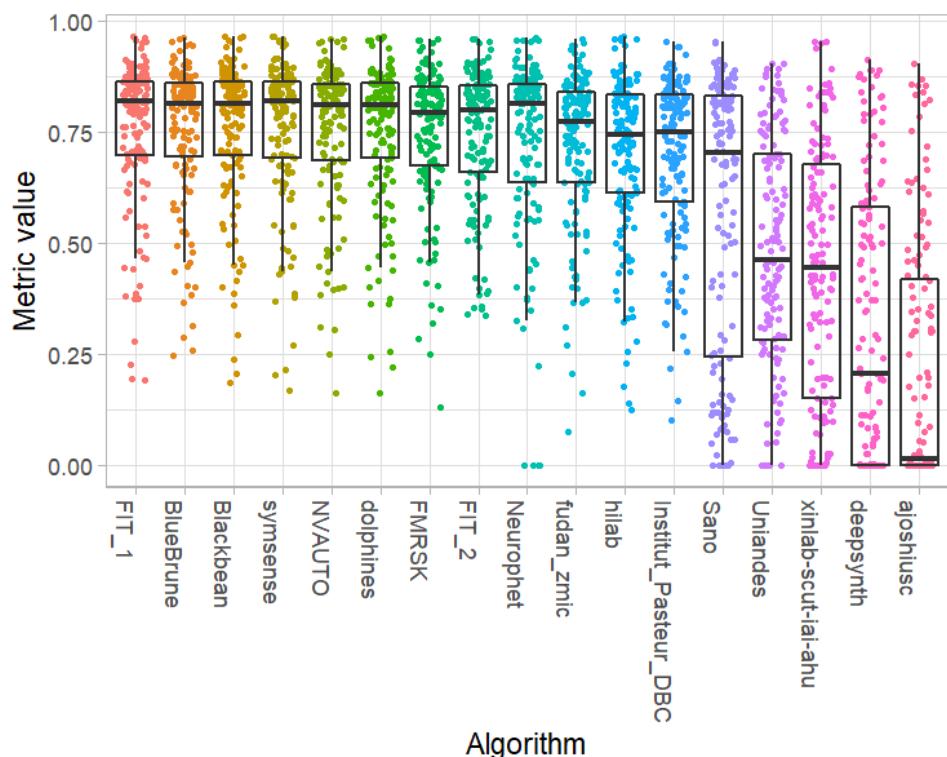
Ranking:

	Dice_mean	rank
FIT_1	0.7597518	1
BlueBrune	0.7573871	2
Blackbean	0.7569661	3
symsense	0.7566418	4
NVAUTO	0.7557553	5
dolphines	0.7508388	6
FMRSK	0.7452712	7
FIT_2	0.7395485	8
Neurophet	0.7234531	9
fudan_zmic	0.7191791	10
hilab	0.7021188	11
Institut_Pasteur_DBC	0.6984268	12
Sano	0.5670978	13
Uniandes	0.4770842	14
xinlab-scut-iai-ahu	0.4303897	15
deepsynth	0.3140195	16
ajoshiusc	0.2153574	17

## 19.2 Visualization of raw assessment data

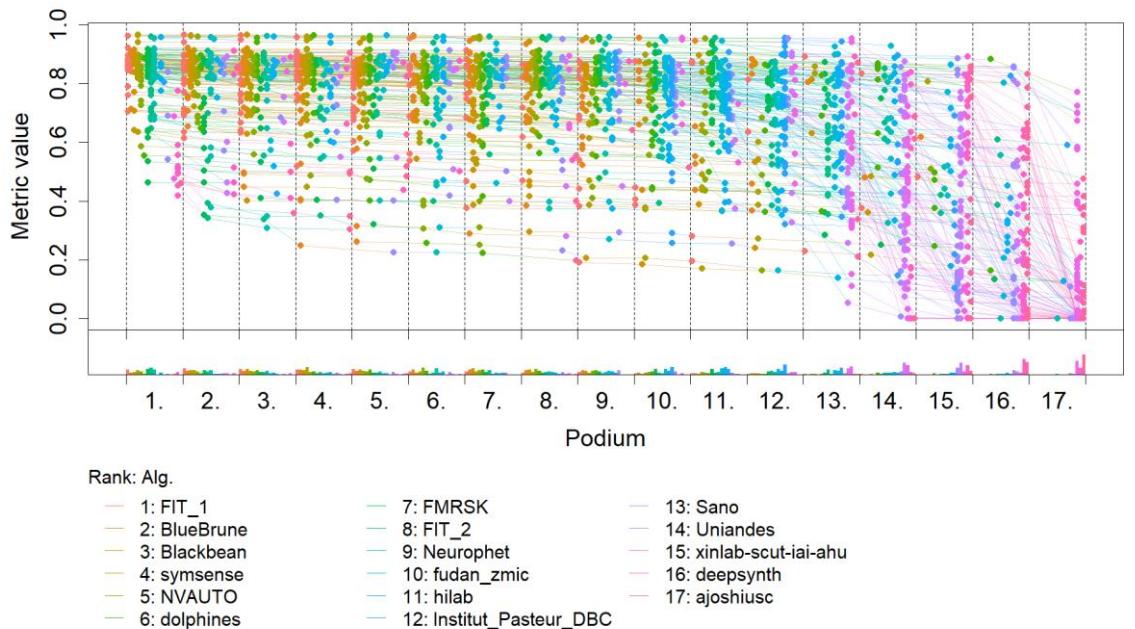
### Dot- and boxplot

*Dot- and boxplots* for visualizing raw assessment data separately for each algorithm. Boxplots representing descriptive statistics over all cases (median, quartiles and outliers) are combined with horizontally jittered dots representing individual cases.



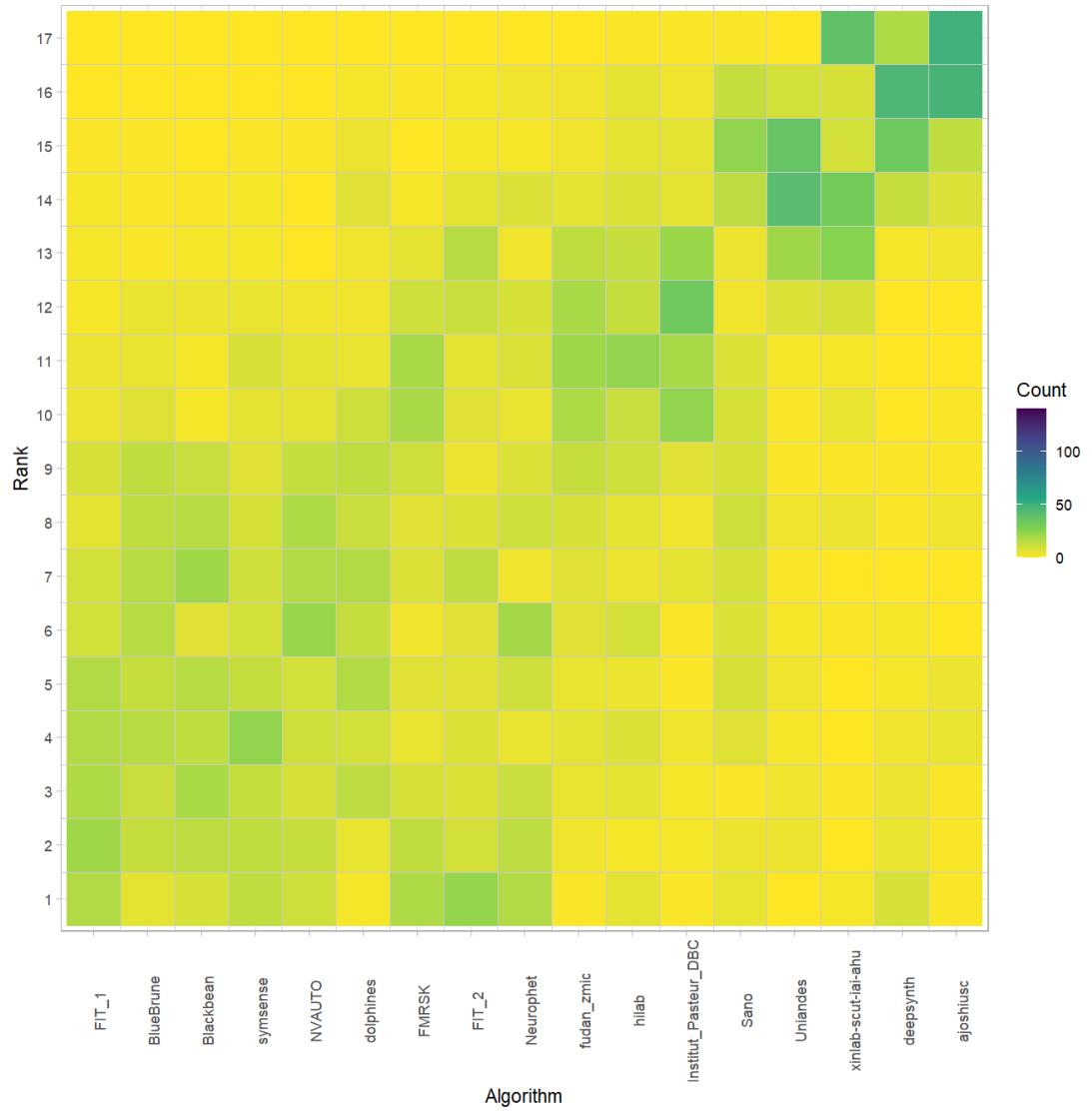
### Podium plot

*Podium plots* (see also Eugster et al., 2008) for visualizing raw assessment data. Upper part (spaghetti plot): Participating algorithms are color-coded, and each colored dot in the plot represents a metric value achieved with the respective algorithm. The actual metric value is encoded by the y-axis. Each podium (here:  $p=17$ ) represents one possible rank, ordered from best (1) to last (here: 17). The assignment of metric values (i.e. colored dots) to one of the podiums is based on the rank that the respective algorithm achieved on the corresponding case. Note that the plot part above each podium place is further subdivided into  $p$  “columns”, where each column represents one participating algorithm (here:  $p = 17$ ). Dots corresponding to identical cases are connected by a line, leading to the shown spaghetti structure. Lower part: Bar charts represent the relative frequency for each algorithm to achieve the rank encoded by the podium place.



## Ranking heatmap

*Ranking heatmaps* for visualizing raw assessment data. Each cell  $(i, A_j)$  shows the absolute frequency of cases in which algorithm  $A_j$  achieved rank  $i$ .

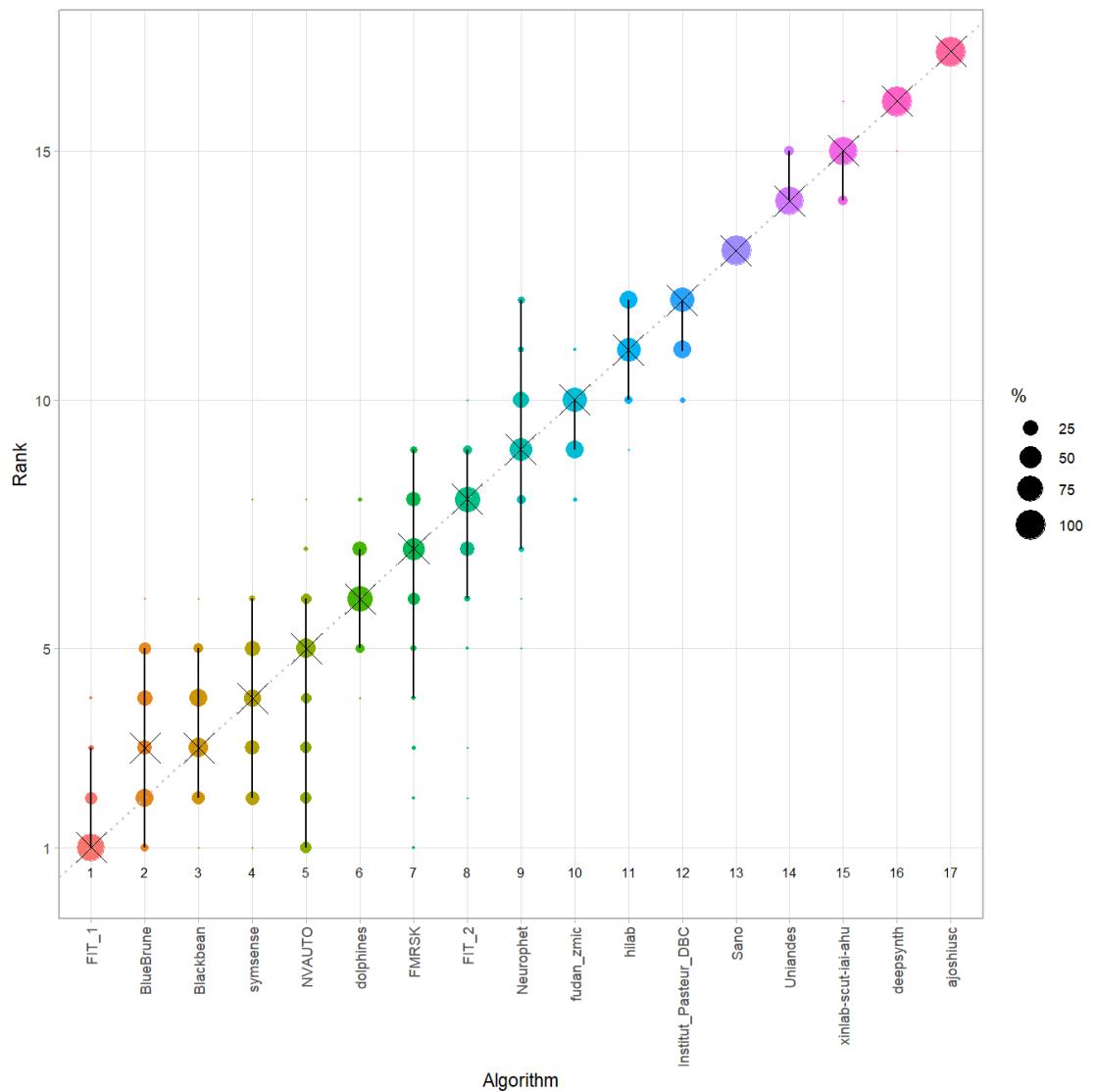


### Visualization of ranking stability

*Blob plot* for visualizing ranking stability based on bootstrap sampling

Algorithms are color-coded, and the area of each blob at position  $(A_i, \text{rank } j)$  is proportional to the relative frequency  $A_i$  achieved rank  $j$  across  $b = 1000$  bootstrap samples. The median rank for each algorithm is indicated by a black cross. 95% bootstrap intervals across bootstrap samples are indicated by black lines.

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.
```

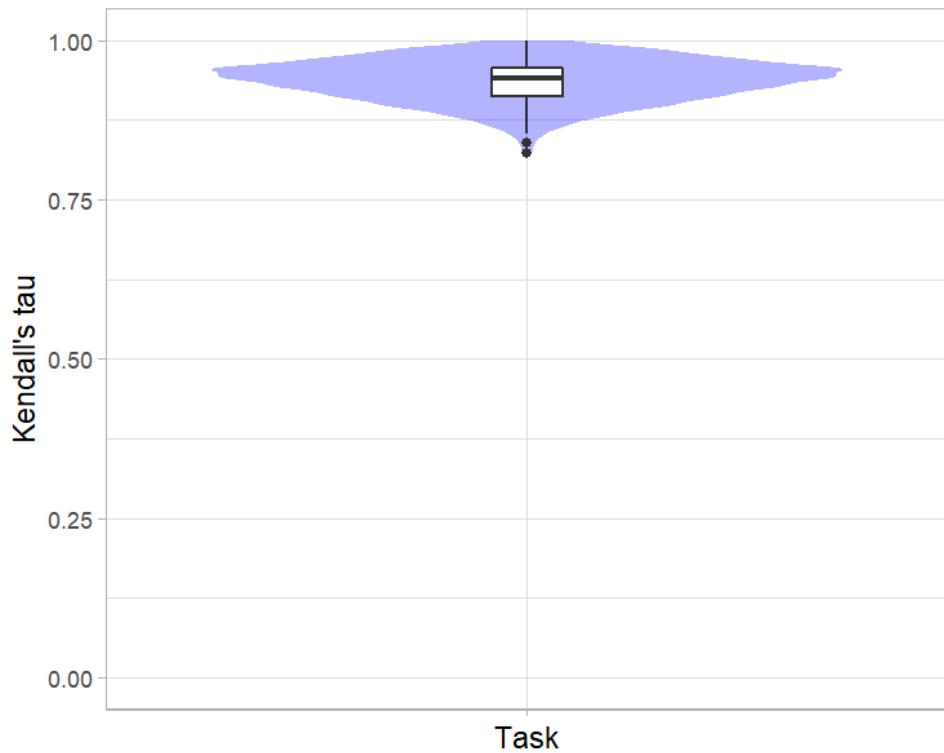


### ***Violin plot for visualizing ranking stability based on bootstrapping***

The ranking list based on the full assessment data is pairwise compared with the ranking lists based on the individual bootstrap samples (here  $b = 1000$  samples). For each pair of rankings, Kendall's  $\tau$  correlation is computed. Kendall's  $\tau$  is a scaled index determining the correlation between the lists. It is computed by evaluating the number of pairwise concordances and discordances between ranking lists and produces values between  $-1$  (for inverted order) and  $1$  (for identical order). A violin plot, which simultaneously depicts a boxplot and a density plot, is generated from the results.

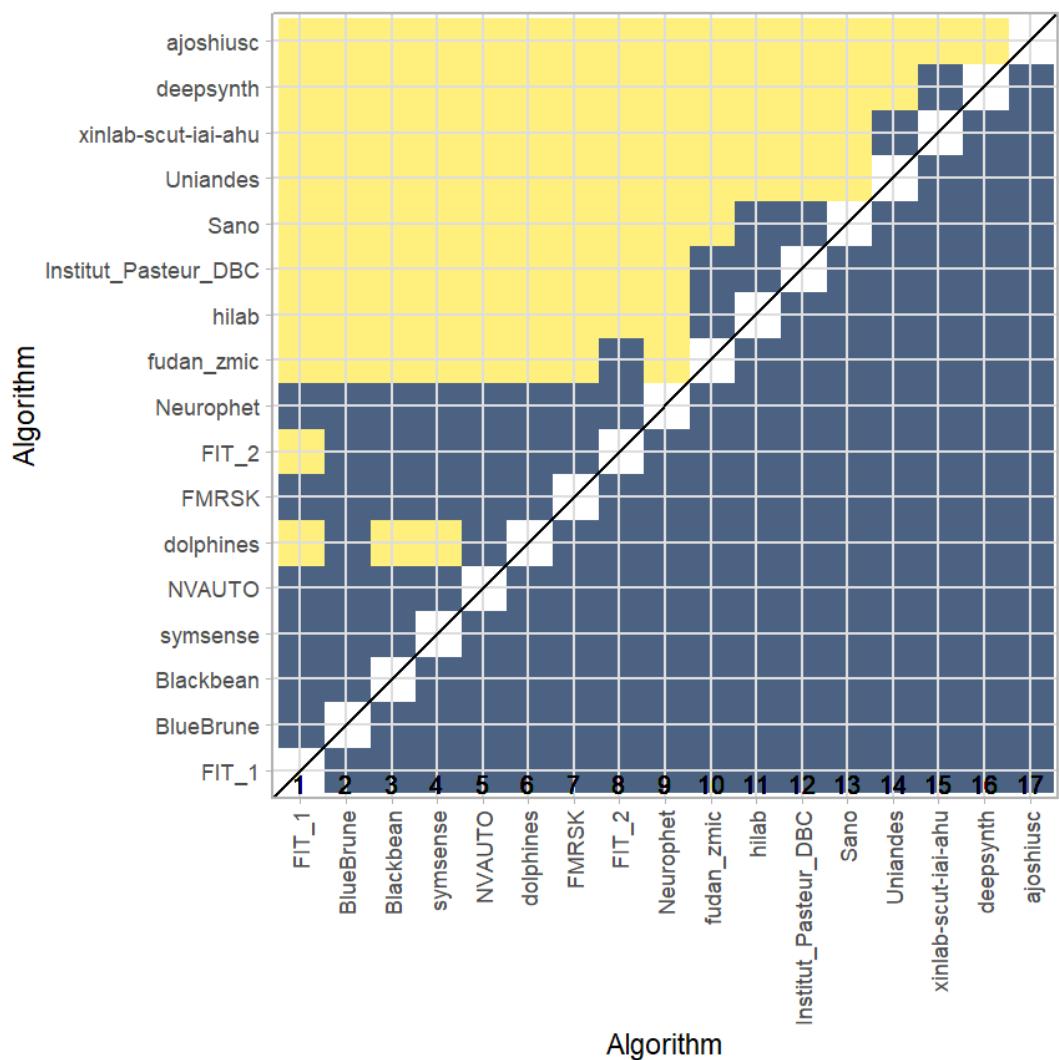
Summary Kendall's tau:

Task	mean	median	q25	q75
dummy-Task	0.9369118	0.9411765	0.9117647	0.9558824



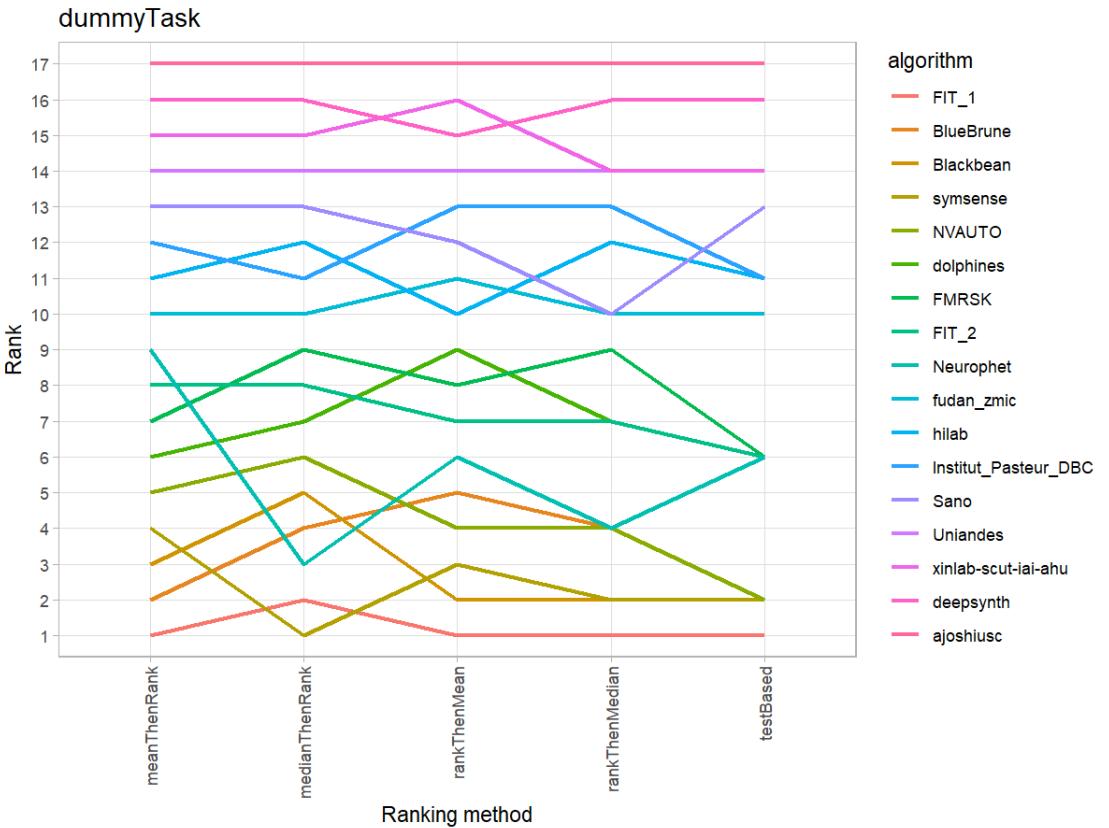
## *Significance maps* for visualizing ranking stability based on statistical significance

*Significance maps* depict incidence matrices of pairwise significant test results for the one-sided Wilcoxon signed rank test at a 5% significance level with adjustment for multiple testing according to Holm. Yellow shading indicates that metric values from the algorithm on the x-axis were significantly superior to those from the algorithm on the y-axis, blue color indicates no significant difference.



### Ranking robustness to ranking methods

*Line plots* for visualizing ranking robustness across different ranking methods. Each algorithm is represented by one colored line. For each ranking method encoded on the x-axis, the height of the line represents the corresponding rank. Horizontal lines indicate identical ranks for all methods.



### 19.3 References

Wiesenfarth, M., Reinke, A., Landman, B.A., Eisenmann, M., Aguilera Saiz, L., Cardoso, M.J., Maier-Hein, L. and Kopp-Schneider, A. Methods and open-source toolkit for analyzing and visualizing challenge results. *Sci Rep* **11**, 2369 (2021). <https://doi.org/10.1038/s41598-021-82017-6>

M. J. A. Eugster, T. Hothorn, and F. Leisch, “Exploratory and inferential analysis of benchmark experiments,” Institut fuer Statistik, Ludwig-Maximilians-Universitaet Muenchen, Germany, Technical Report 30, 2008. [Online]. Available: <http://epub.ub.uni-muenchen.de/4134/>.

## 20 Benchmarking report for Hausdorff Metrics – Poor Quality Reconstructions

created by challengeR v1.0.2  
29 September, 2023

This document presents a systematic report on the benchmark study “Hausdorff Metrics”. Input data comprises raw metric values for all algorithms and cases. Generated plots are:

- Visualization of assessment data: Dot- and boxplot, podium plot and ranking heatmap
- Visualization of ranking stability: Blob plot, violin plot and significance map, line plot

Details can be found in Wiesenfarth et al. (2021).

### 20.1 Ranking

Algorithms within a task are ranked according to the following ranking scheme:  
*aggregate using function (“mean”) then rank*

The analysis is based on 17 algorithms and 140 cases. 0 missing cases have been found in the data set.

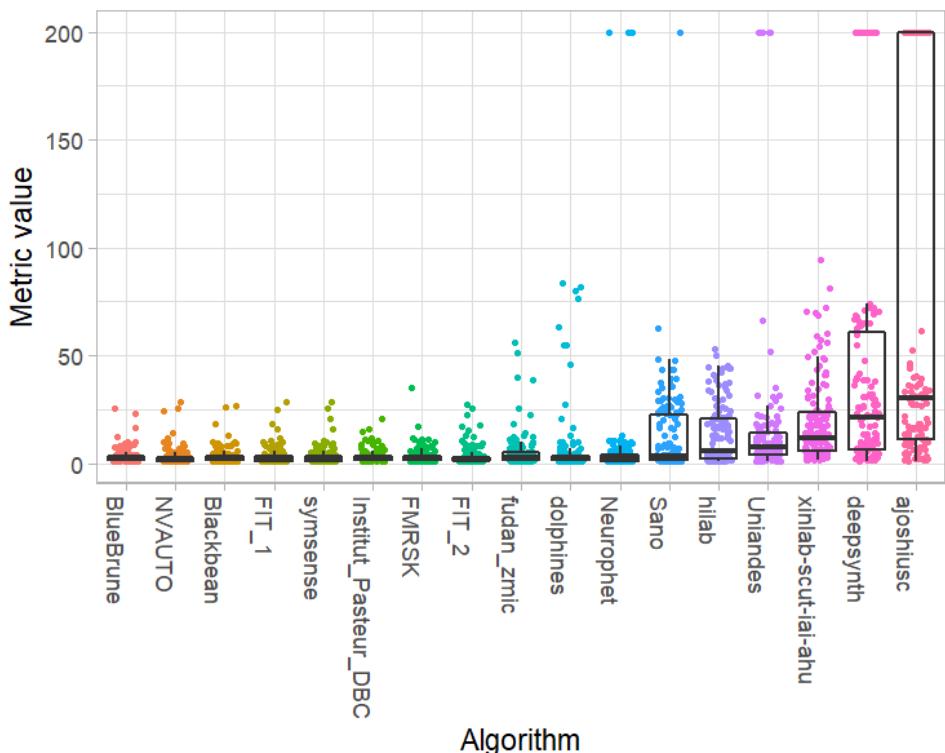
Ranking:

	Hausdorff_mean	rank
BlueBrune	3.097065	1
NVAUTO	3.161613	2
Blackbean	3.203836	3
FIT_1	3.259938	4
symsense	3.323026	5
Institut_Pasteur_DBC	3.354046	6
FMRSK	3.370501	7
FIT_2	3.526028	8
fudan_zmic	5.262537	9
dolphines	6.824333	10
Neurophet	8.791505	11
Sano	12.535329	12
hilab	13.510597	13
Uniandes	18.295327	14
xinlab-scut-iai-ahu	18.960886	15
deepsynth	46.613199	16
ajoshiusc	78.392945	17

## 20.2 Visualization of raw assessment data

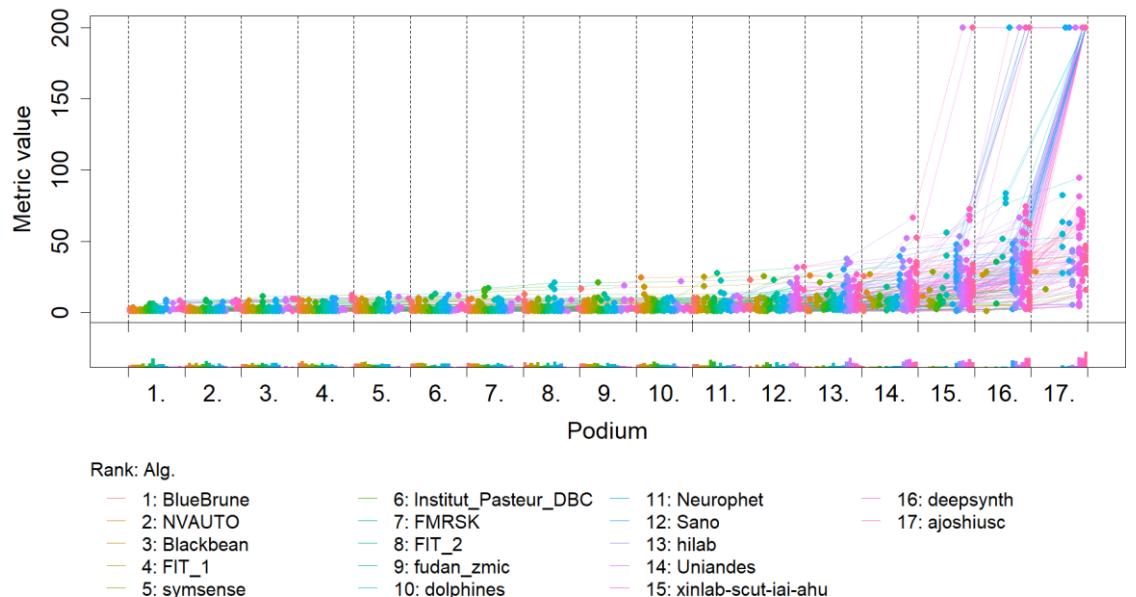
### Dot- and boxplot

*Dot- and boxplots* for visualizing raw assessment data separately for each algorithm. Boxplots representing descriptive statistics over all cases (median, quartiles and outliers) are combined with horizontally jittered dots representing individual cases.



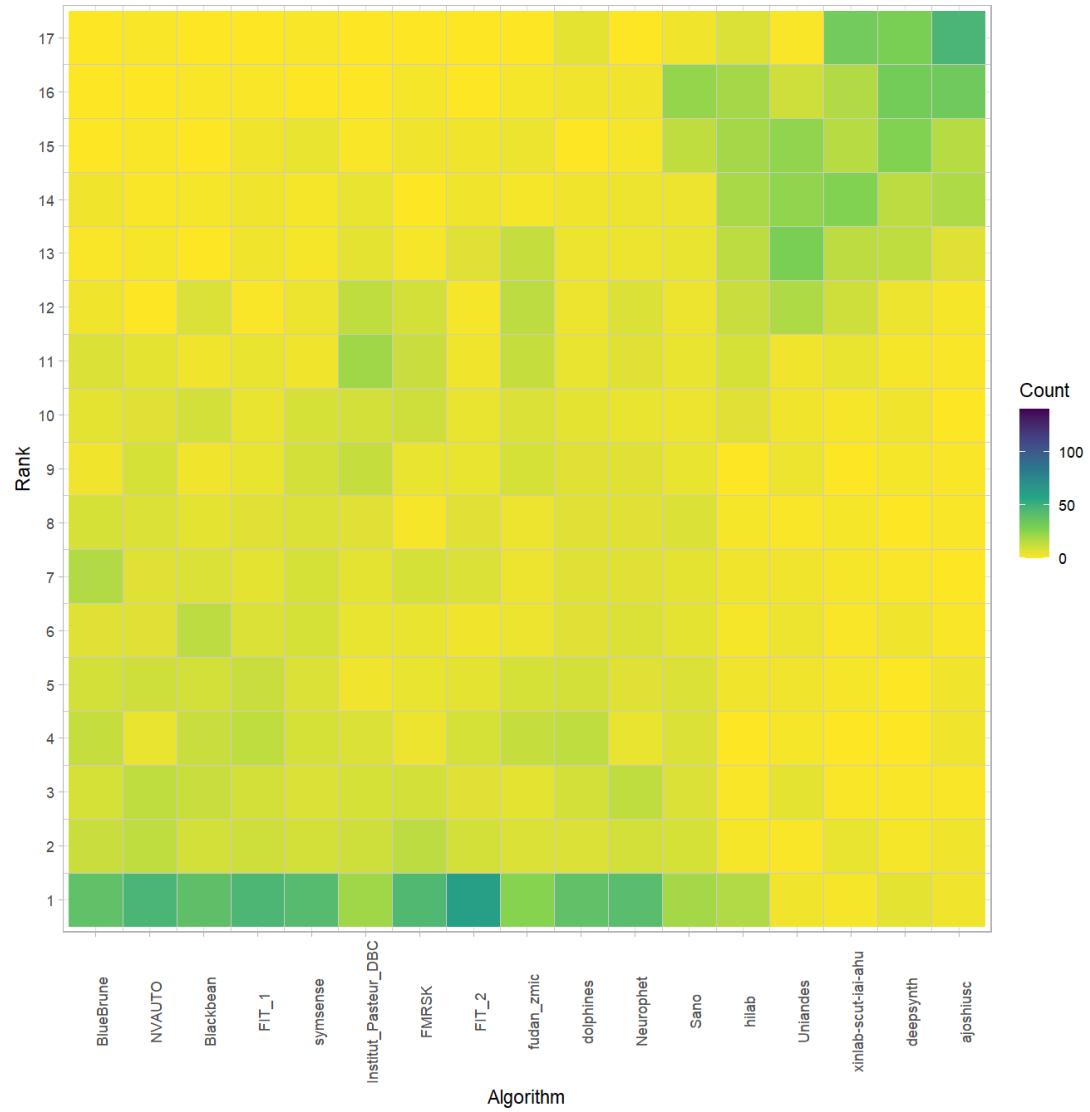
### Podium plot

*Podium plots* (see also Eugster et al., 2008) for visualizing raw assessment data. Upper part (spaghetti plot): Participating algorithms are color-coded, and each colored dot in the plot represents a metric value achieved with the respective algorithm. The actual metric value is encoded by the y-axis. Each podium (here:  $p=17$ ) represents one possible rank, ordered from best (1) to last (here: 17). The assignment of metric values (i.e. colored dots) to one of the podiums is based on the rank that the respective algorithm achieved on the corresponding case. Note that the plot part above each podium place is further subdivided into  $p$  “columns”, where each column represents one participating algorithm (here:  $p = 17$ ). Dots corresponding to identical cases are connected by a line, leading to the shown spaghetti structure. Lower part: Bar charts represent the relative frequency for each algorithm to achieve the rank encoded by the podium place.



## Ranking heatmap

*Ranking heatmaps* for visualizing raw assessment data. Each cell  $(i, A_j)$  shows the absolute frequency of cases in which algorithm  $A_j$  achieved rank  $i$ .

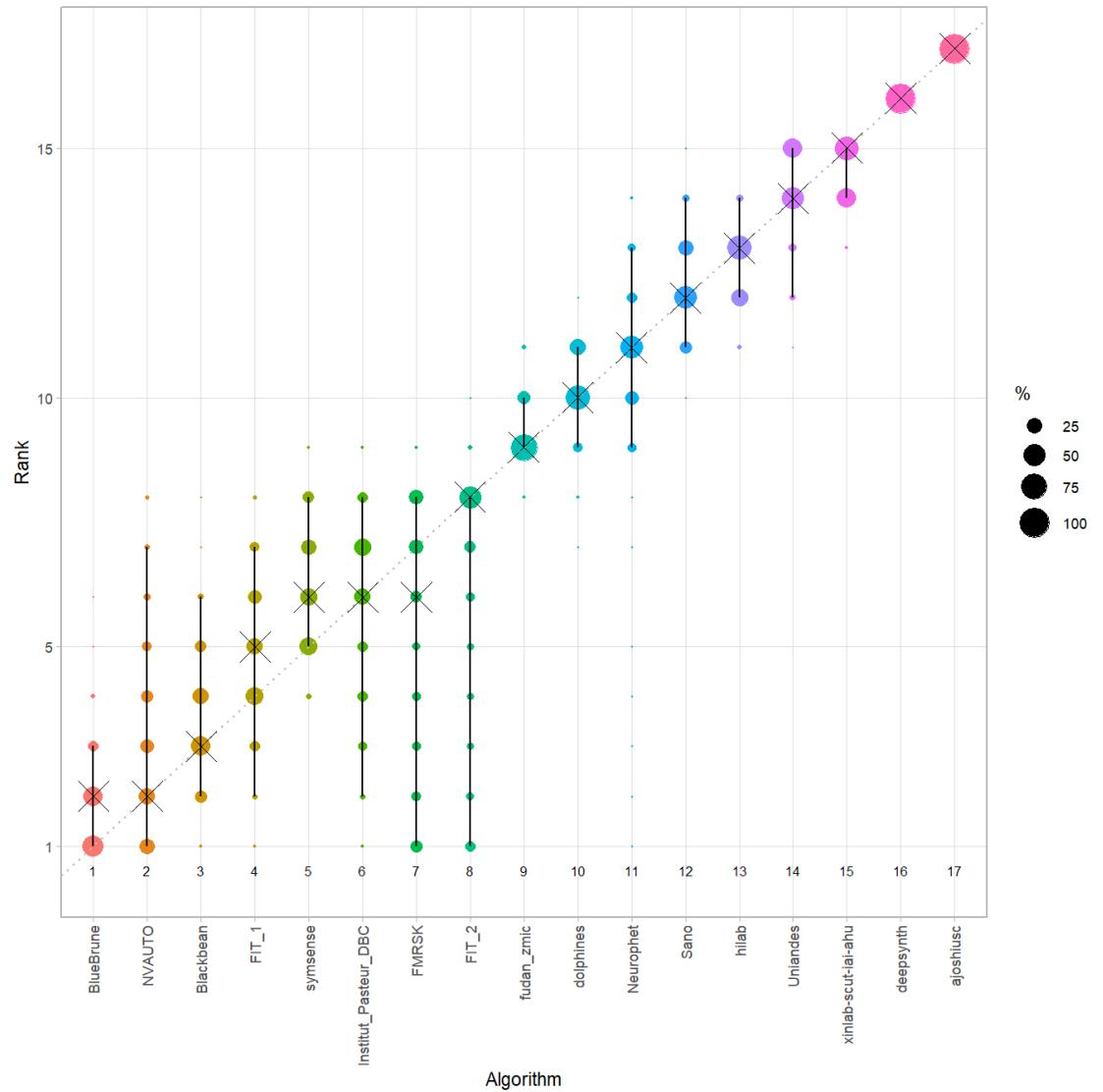


### 20.3 Visualization of ranking stability

*Blob plot* for visualizing ranking stability based on bootstrap sampling

Algorithms are color-coded, and the area of each blob at position  $(A_i, \text{rank } j)$  is proportional to the relative frequency  $A_i$  achieved rank  $j$  across  $b = 1000$  bootstrap samples. The median rank for each algorithm is indicated by a black cross. 95% bootstrap intervals across bootstrap samples are indicated by black lines.

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> = ## "none")` instead.
```

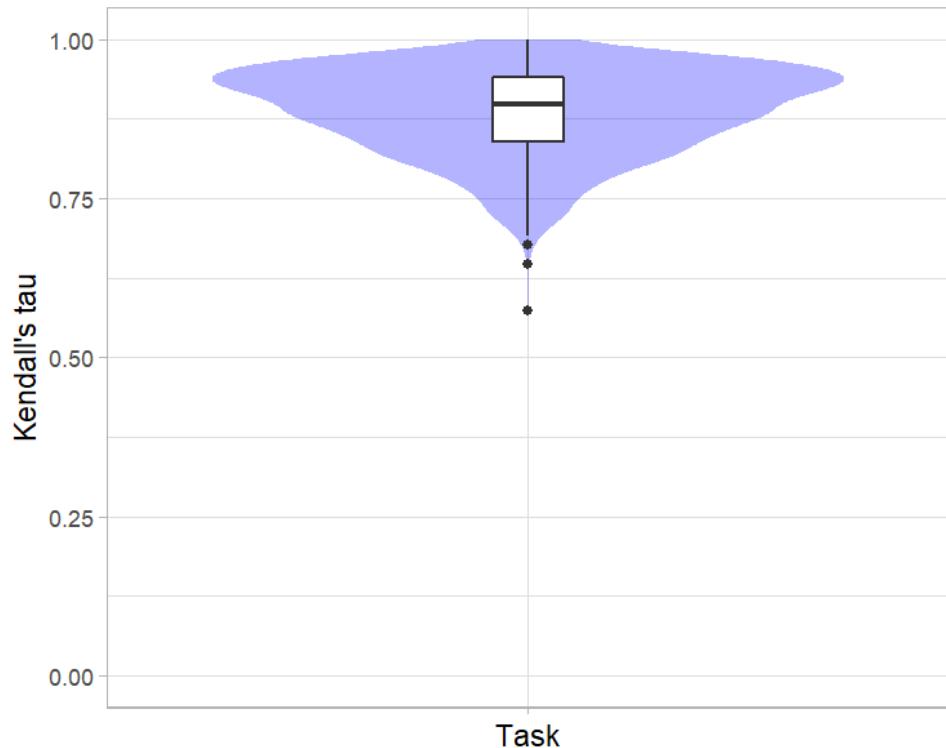


### ***Violin plot for visualizing ranking stability based on bootstrapping***

The ranking list based on the full assessment data is pairwise compared with the ranking lists based on the individual bootstrap samples (here  $b = 1000$  samples). For each pair of rankings, Kendall's  $\tau$  correlation is computed. Kendall's  $\tau$  is a scaled index determining the correlation between the lists. It is computed by evaluating the number of pairwise concordances and discordances between ranking lists and produces values between  $-1$  (for inverted order) and  $1$  (for identical order). A violin plot, which simultaneously depicts a boxplot and a density plot, is generated from the results.

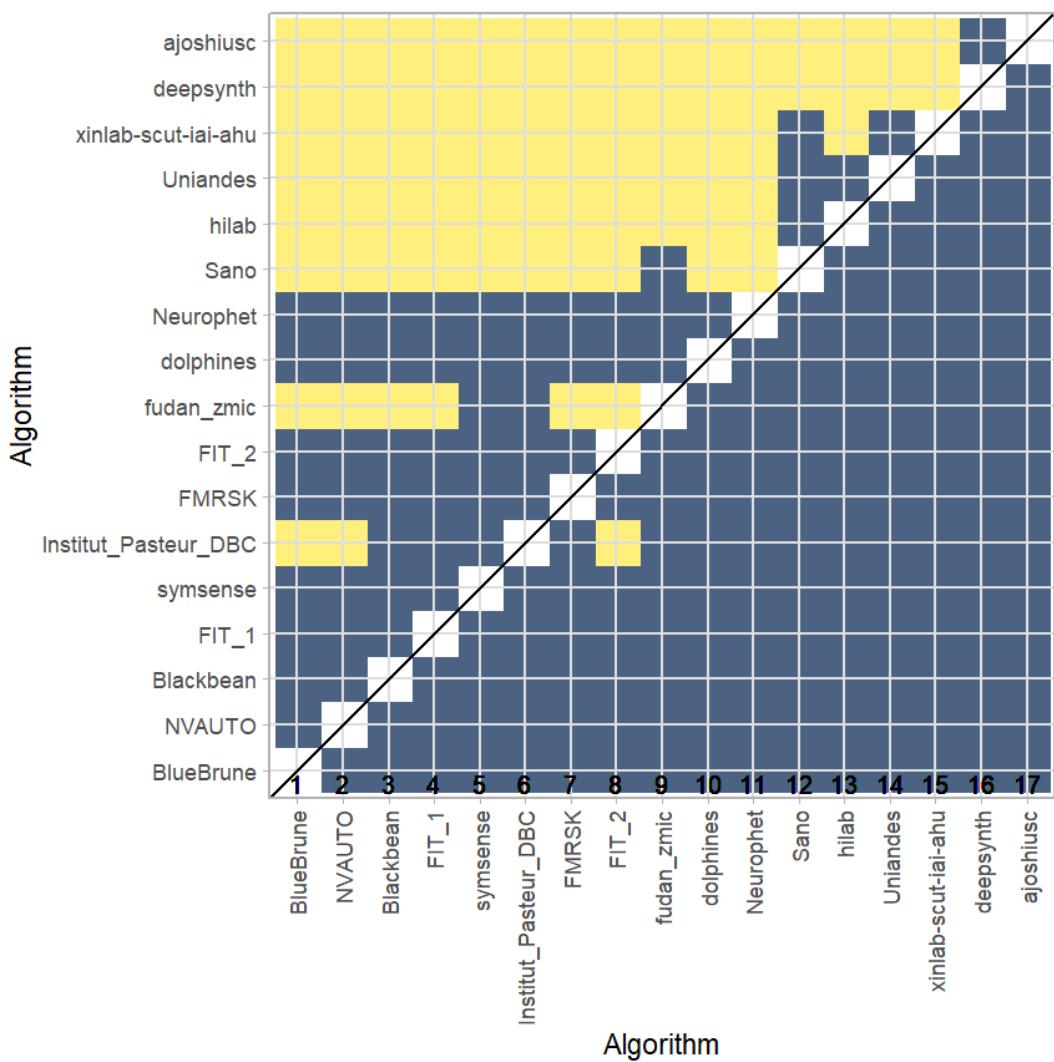
Summary Kendall's tau:

Task	mean	median	q25	q75
dummyTask	0.8869559	0.8970588	0.8382353	0.9411765



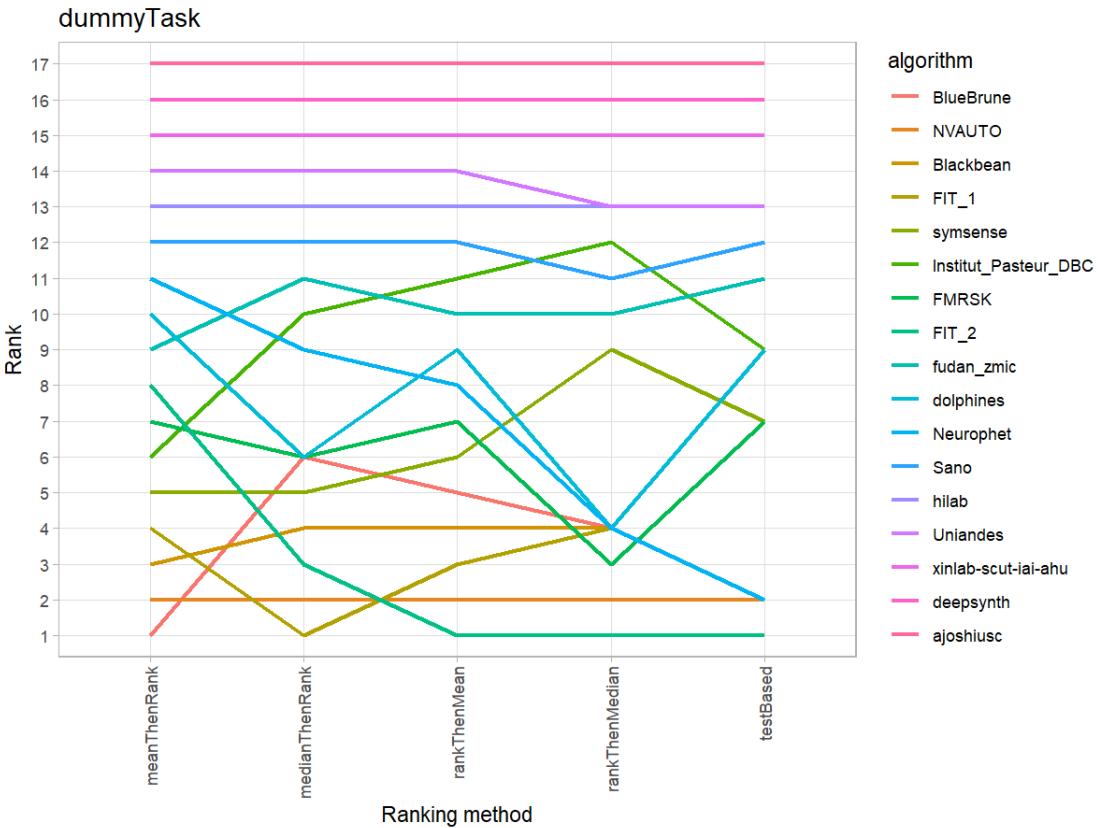
## *Significance maps* for visualizing ranking stability based on statistical significance

*Significance maps* depict incidence matrices of pairwise significant test results for the one-sided Wilcoxon signed rank test at a 5% significance level with adjustment for multiple testing according to Holm. Yellow shading indicates that metric values from the algorithm on the x-axis were significantly superior to those from the algorithm on the y-axis, blue color indicates no significant difference.



### Ranking robustness to ranking methods

*Line plots* for visualizing ranking robustness across different ranking methods. Each algorithm is represented by one colored line. For each ranking method encoded on the x-axis, the height of the line represents the corresponding rank. Horizontal lines indicate identical ranks for all methods.



## 20.4 References

Wiesenfarth, M., Reinke, A., Landman, B.A., Eisenmann, M., Aguilera Saiz, L., Cardoso, M.J., Maier-Hein, L. and Kopp-Schneider, A. Methods and open-source toolkit for analyzing and visualizing challenge results. *Sci Rep* **11**, 2369 (2021). <https://doi.org/10.1038/s41598-021-82017-6>

M. J. A. Eugster, T. Hothorn, and F. Leisch, “Exploratory and inferential analysis of benchmark experiments,” Institut fuer Statistik, Ludwig-Maximilians-Universitaet Muenchen, Germany, Technical Report 30, 2008. [Online]. Available: <http://epub.ub.uni-muenchen.de/4134/>.

## 21 Benchmarking report for Volume Similarity Metrics – Poor Quality Reconstructions

created by challengeR v1.0.2  
29 September, 2023

This document presents a systematic report on the benchmark study “Volume Similarity Metrics”. Input data comprises raw metric values for all algorithms and cases. Generated plots are:

- Visualization of assessment data: Dot- and boxplot, podium plot and ranking heatmap
- Visualization of ranking stability: Blob plot, violin plot and significance map, line plot

Details can be found in Wiesenfarth et al. (2021).

### 21.1 Ranking

Algorithms within a task are ranked according to the following ranking scheme:  
*aggregate using function (“mean”) then rank*

The analysis is based on 17 algorithms and 140 cases. 0 missing cases have been found in the data set.

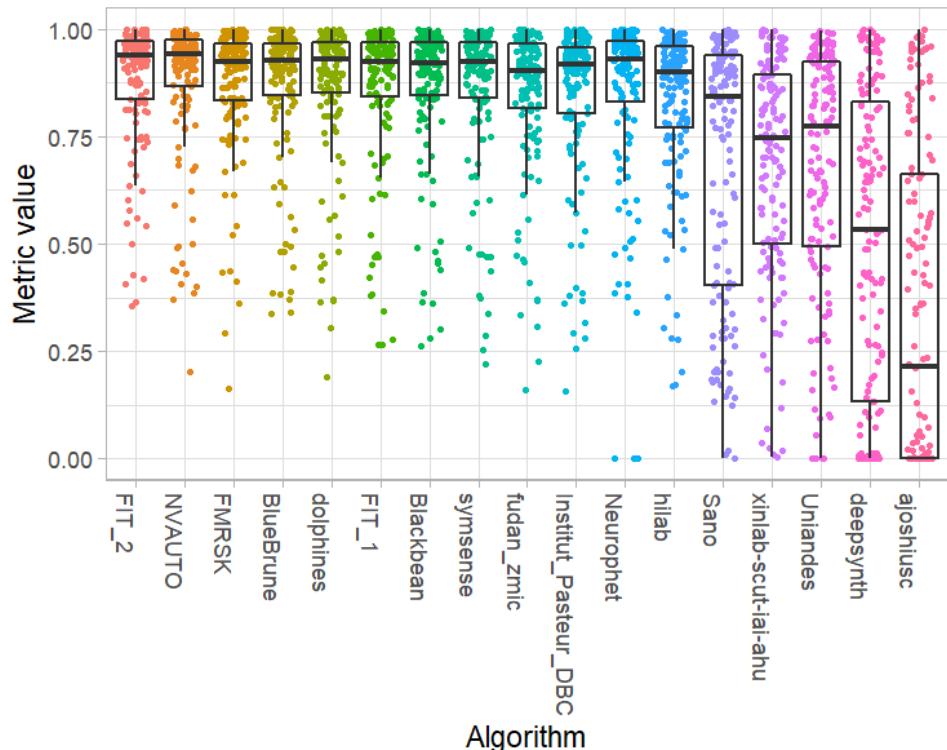
Ranking:

	Volume_Similarity_mean	rank
FIT_2	0.8766704	1
NVAUTO	0.8745644	2
FMRSK	0.8732227	3
BlueBrune	0.8685192	4
dolphins	0.8672304	5
FIT_1	0.8633964	6
Blackbean	0.8628127	7
symsense	0.8606914	8
fudan_zmic	0.8485512	9
Insti-tut_Pasteur_DBC	0.8452351	10
Neurophet	0.8429539	11
hilab	0.8271106	12
Sano	0.6892375	13
xinlab-scut-iai-ahu	0.6845423	14
Uniandes	0.6711314	15
deepsynth	0.5003960	16
ajoshiusc	0.3438202	17

## 21.2 Visualization of raw assessment data

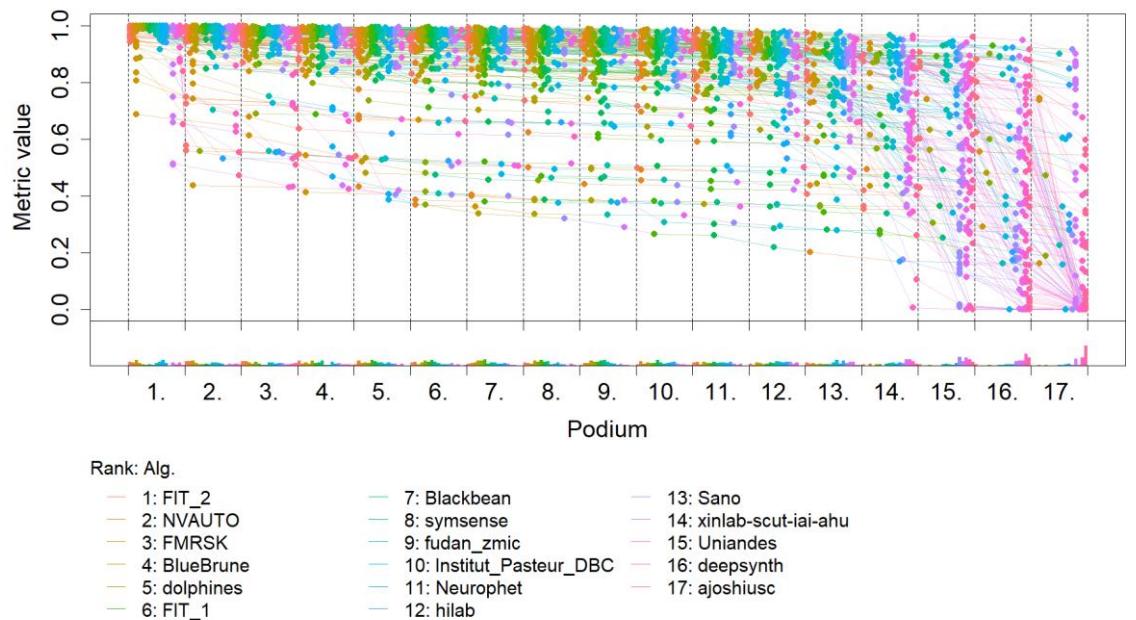
### Dot- and boxplot

*Dot- and boxplots* for visualizing raw assessment data separately for each algorithm. Boxplots representing descriptive statistics over all cases (median, quartiles and outliers) are combined with horizontally jittered dots representing individual cases.



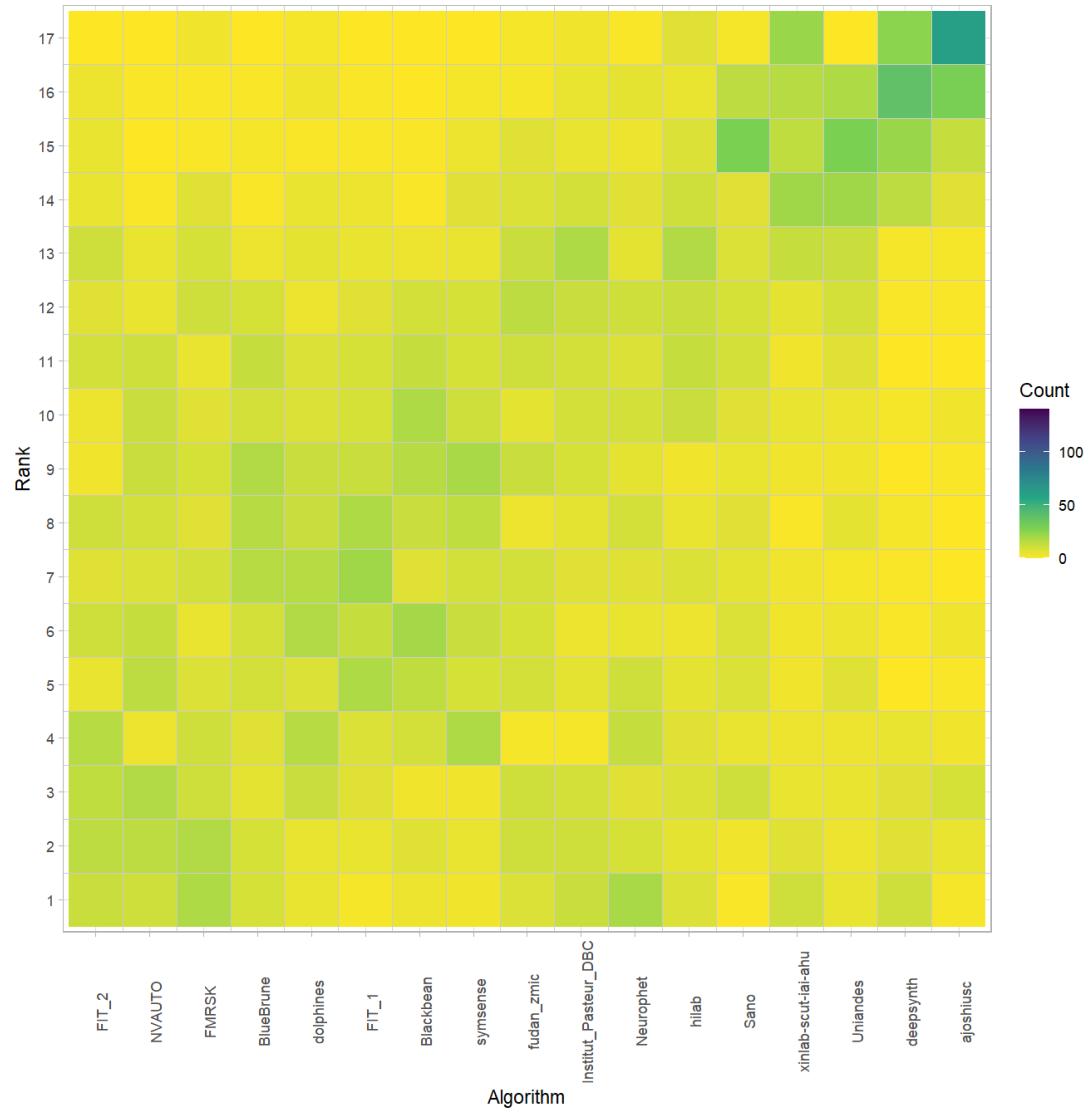
### Podium plot

*Podium plots* (see also Eugster et al., 2008) for visualizing raw assessment data. Upper part (spaghetti plot): Participating algorithms are color-coded, and each colored dot in the plot represents a metric value achieved with the respective algorithm. The actual metric value is encoded by the y-axis. Each podium (here:  $p=17$ ) represents one possible rank, ordered from best (1) to last (here: 17). The assignment of metric values (i.e. colored dots) to one of the podiums is based on the rank that the respective algorithm achieved on the corresponding case. Note that the plot part above each podium place is further subdivided into  $p$  “columns”, where each column represents one participating algorithm (here:  $p = 17$ ). Dots corresponding to identical cases are connected by a line, leading to the shown spaghetti structure. Lower part: Bar charts represent the relative frequency for each algorithm to achieve the rank encoded by the podium place.



## Ranking heatmap

*Ranking heatmaps* for visualizing raw assessment data. Each cell  $(i, A_j)$  shows the absolute frequency of cases in which algorithm  $A_j$  achieved rank  $i$ .

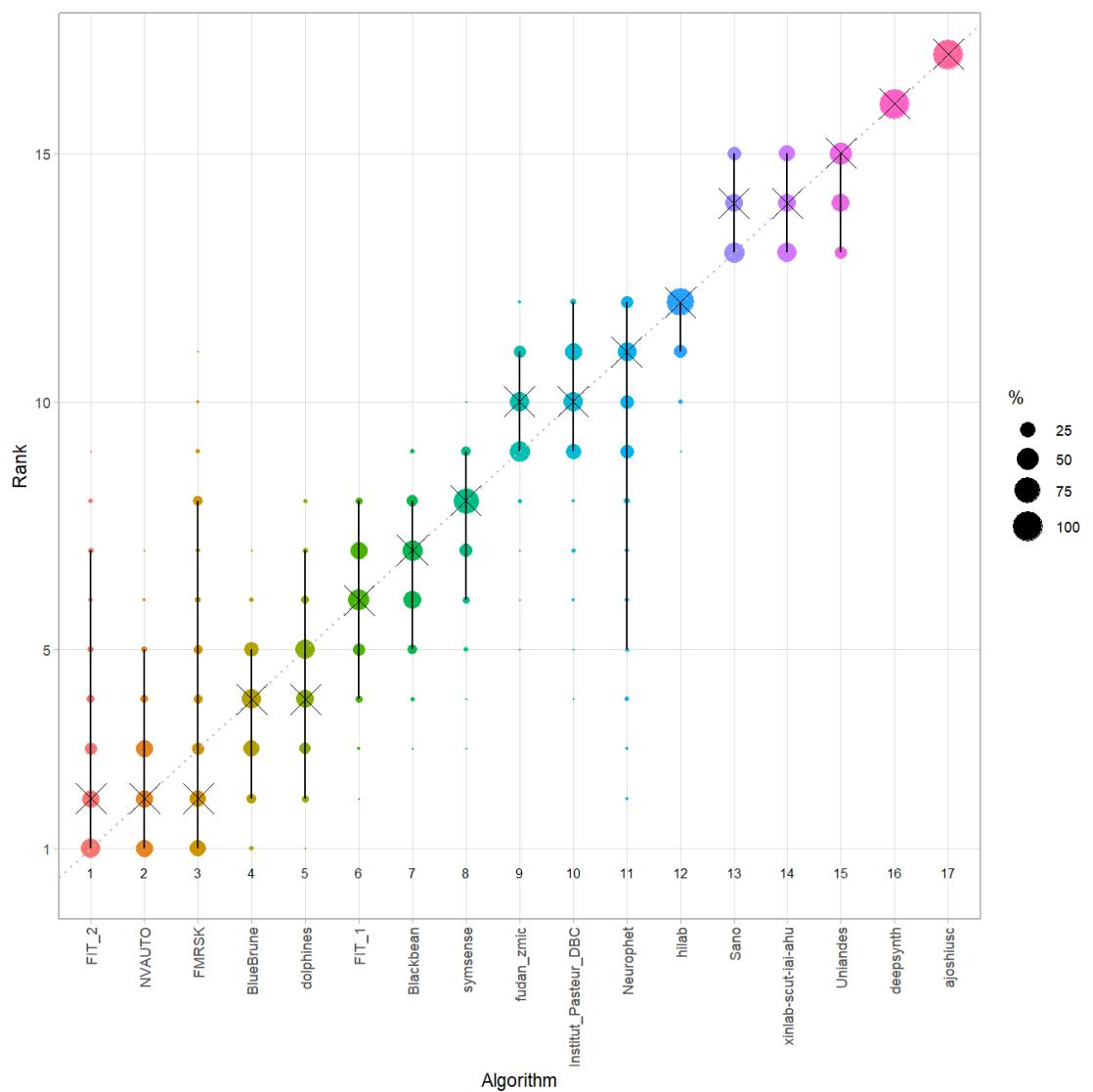


### Visualization of ranking stability

*Blob plot* for visualizing ranking stability based on bootstrap sampling

Algorithms are color-coded, and the area of each blob at position  $(A_i, \text{rank } j)$  is proportional to the relative frequency  $A_i$  achieved rank  $j$  across  $b = 1000$  bootstrap samples. The median rank for each algorithm is indicated by a black cross. 95% bootstrap intervals across bootstrap samples are indicated by black lines.

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.
```

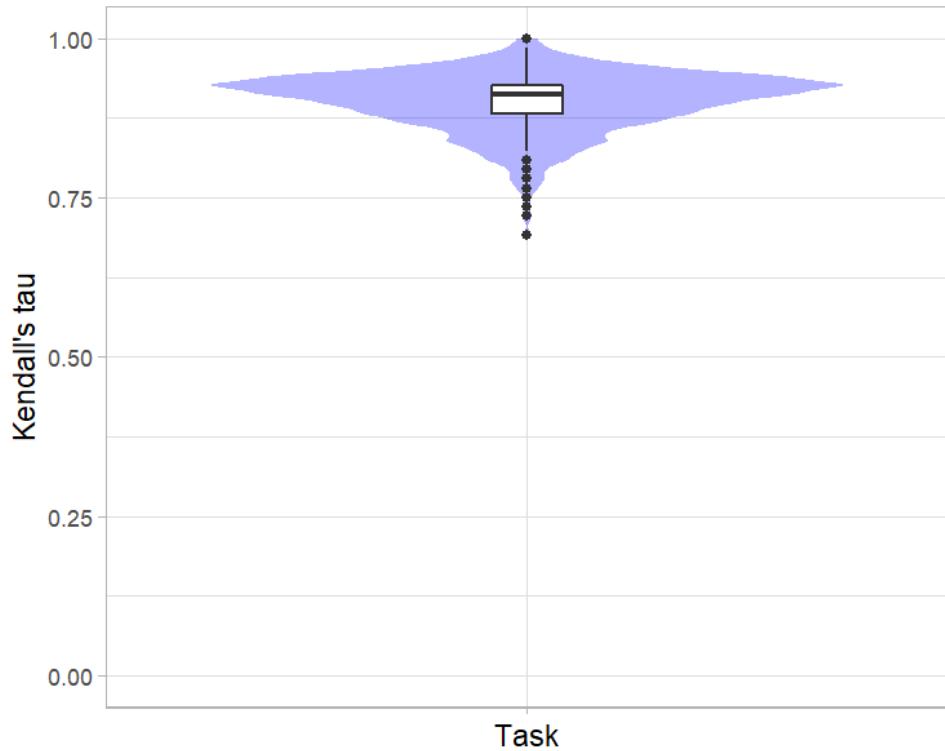


### ***Violin plot for visualizing ranking stability based on bootstrapping***

The ranking list based on the full assessment data is pairwise compared with the ranking lists based on the individual bootstrap samples (here  $b = 1000$  samples). For each pair of rankings, Kendall's  $\tau$  correlation is computed. Kendall's  $\tau$  is a scaled index determining the correlation between the lists. It is computed by evaluating the number of pairwise concordances and discordances between ranking lists and produces values between  $-1$  (for inverted order) and  $1$  (for identical order). A violin plot, which simultaneously depicts a boxplot and a density plot, is generated from the results.

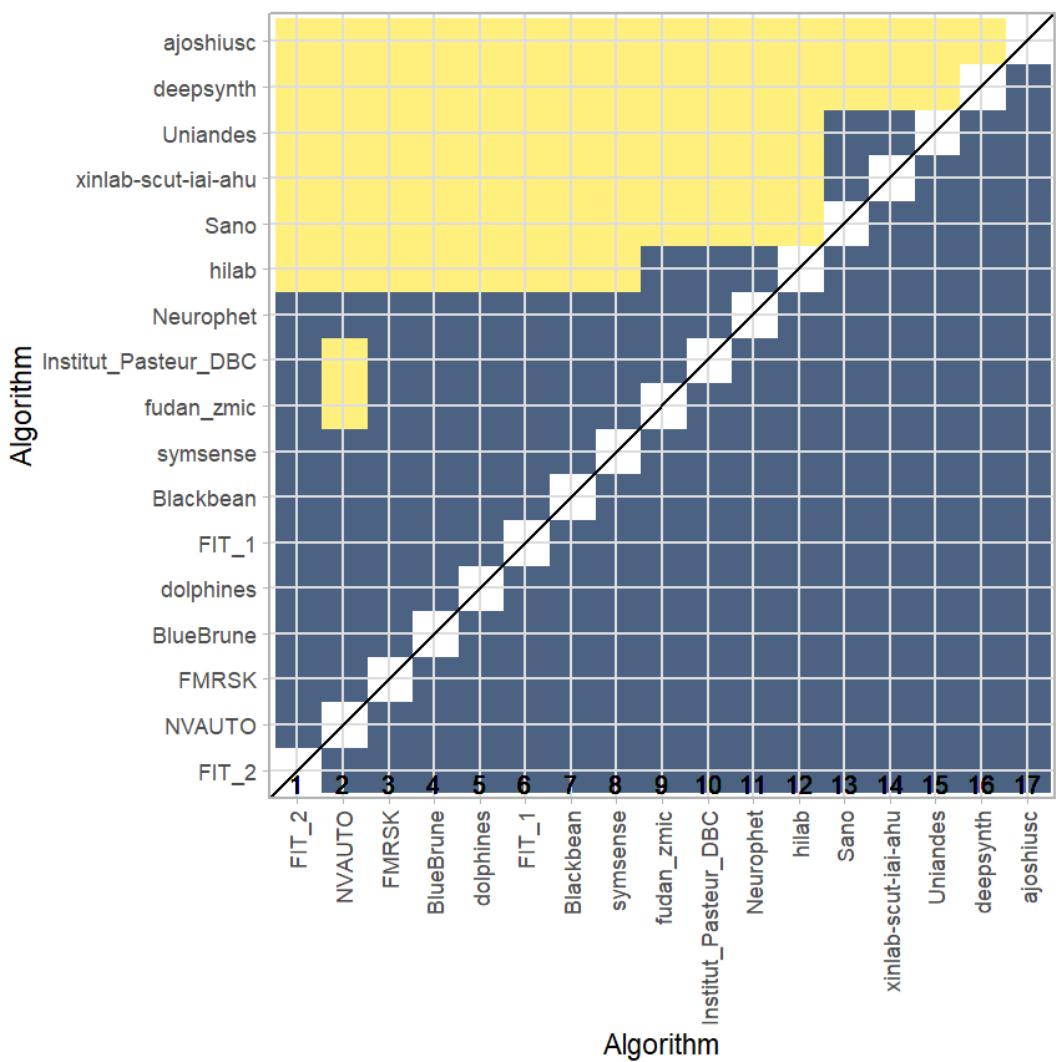
Summary Kendall's tau:

Task	mean	median	q25	q75
dummy-Task	0.9016765	0.9117647	0.8823529	0.9264706



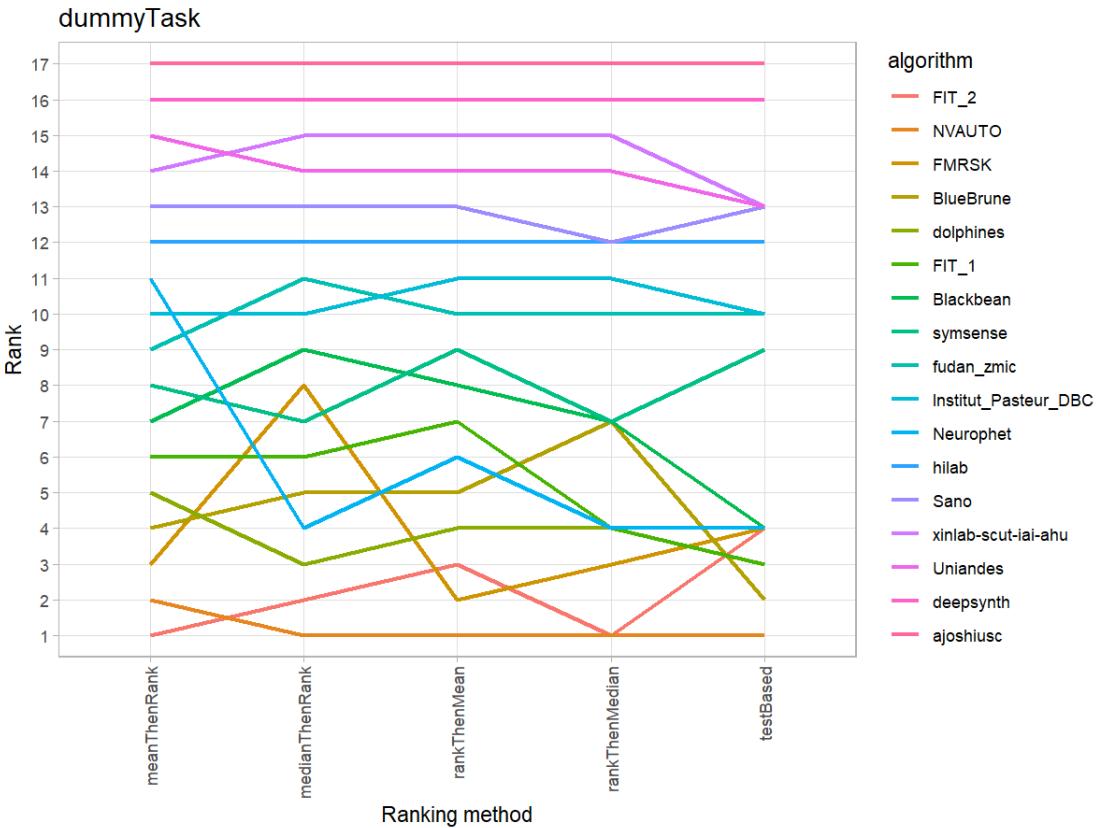
## *Significance maps* for visualizing ranking stability based on statistical significance

*Significance maps* depict incidence matrices of pairwise significant test results for the one-sided Wilcoxon signed rank test at a 5% significance level with adjustment for multiple testing according to Holm. Yellow shading indicates that metric values from the algorithm on the x-axis were significantly superior to those from the algorithm on the y-axis, blue color indicates no significant difference.



### Ranking robustness to ranking methods

*Line plots* for visualizing ranking robustness across different ranking methods. Each algorithm is represented by one colored line. For each ranking method encoded on the x-axis, the height of the line represents the corresponding rank. Horizontal lines indicate identical ranks for all methods.



### 21.3 References

Wiesenfarth, M., Reinke, A., Landman, B.A., Eisenmann, M., Aguilera Saiz, L., Cardoso, M.J., Maier-Hein, L. and Kopp-Schneider, A. Methods and open-source toolkit for analyzing and visualizing challenge results. *Sci Rep* **11**, 2369 (2021). <https://doi.org/10.1038/s41598-021-82017-6>

M. J. A. Eugster, T. Hothorn, and F. Leisch, “Exploratory and inferential analysis of benchmark experiments,” Institut fuer Statistik, Ludwig-Maximilians-Universitaet Muenchen, Germany, Technical Report 30, 2008. [Online]. Available: <http://epub.ub.uni-muenchen.de/4134/>.

## 22 Benchmarking report for Dice Metrics – Neurotypical Brains

created by challengeR v1.0.2

21 September, 2023

This document presents a systematic report on the benchmark study “Dice Metrics”. Input data comprises raw metric values for all algorithms and cases. Generated plots are:

- Visualization of assessment data: Dot- and boxplot, podium plot and ranking heatmap
- Visualization of ranking stability: Blob plot, violin plot and significance map, line plot

Details can be found in Wiesenfarth et al. (2021).

### 22.1 Ranking

Algorithms within a task are ranked according to the following ranking scheme:

*aggregate using function (“mean”) then rank*

The analysis is based on 17 algorithms and 476 cases. 0 missing cases have been found in the data set.

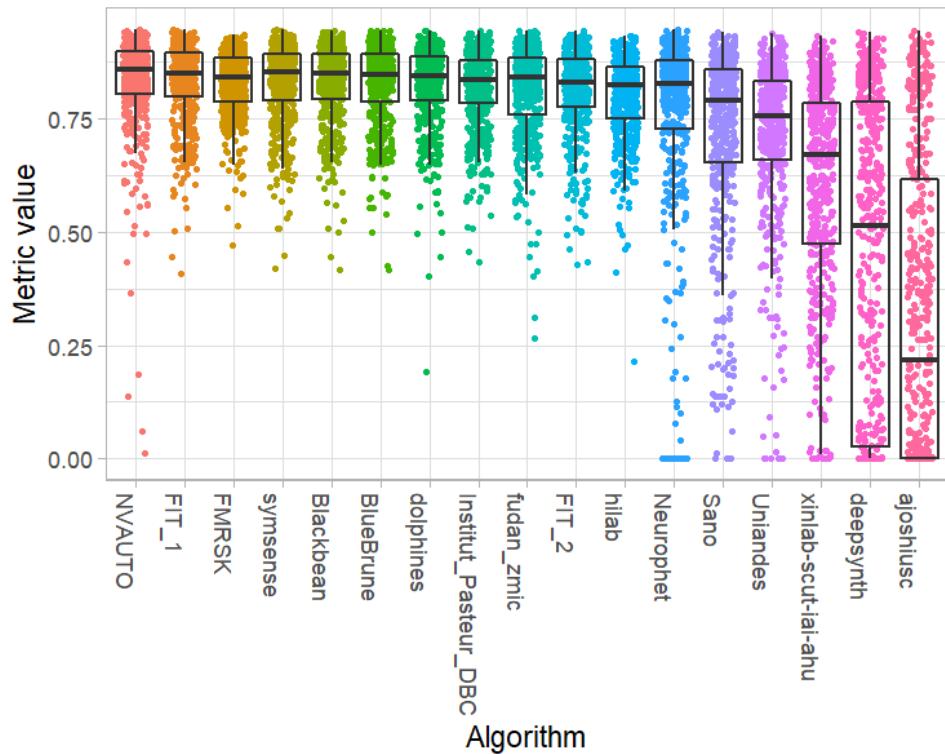
Ranking:

	Dice_mean	rank
NVAUTO	0.8306140	1
FIT_1	0.8288169	2
FMRSK	0.8277248	3
symsense	0.8264437	4
Blackbean	0.8261369	5
BlueBrune	0.8245527	6
dolphines	0.8212657	7
Institut_Pasteur_DBC	0.8169504	8
fudan_zmic	0.8125557	9
FIT_2	0.8111363	10
hilab	0.7979320	11
Neurophet	0.7504558	12
Sano	0.7187610	13
Uniandes	0.7108762	14
xinlab-scut-iai-ahu	0.6088211	15
deepsynth	0.4432369	16
ajoshiusc	0.3157481	17

## 22.2 Visualization of raw assessment data

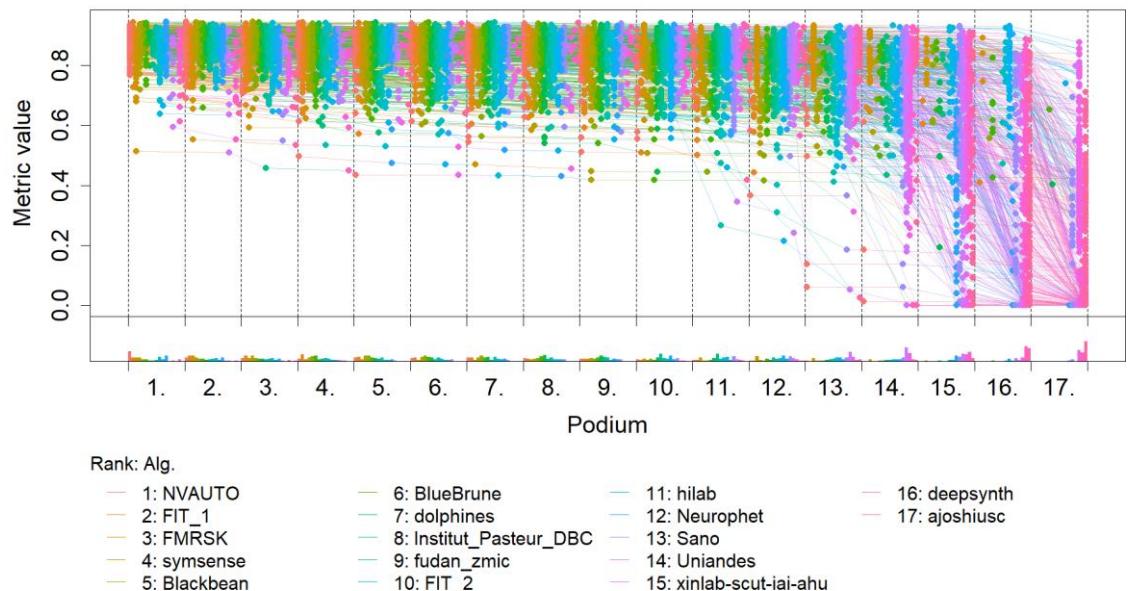
### Dot- and boxplot

*Dot- and boxplots* for visualizing raw assessment data separately for each algorithm. Boxplots representing descriptive statistics over all cases (median, quartiles and outliers) are combined with horizontally jittered dots representing individual cases.



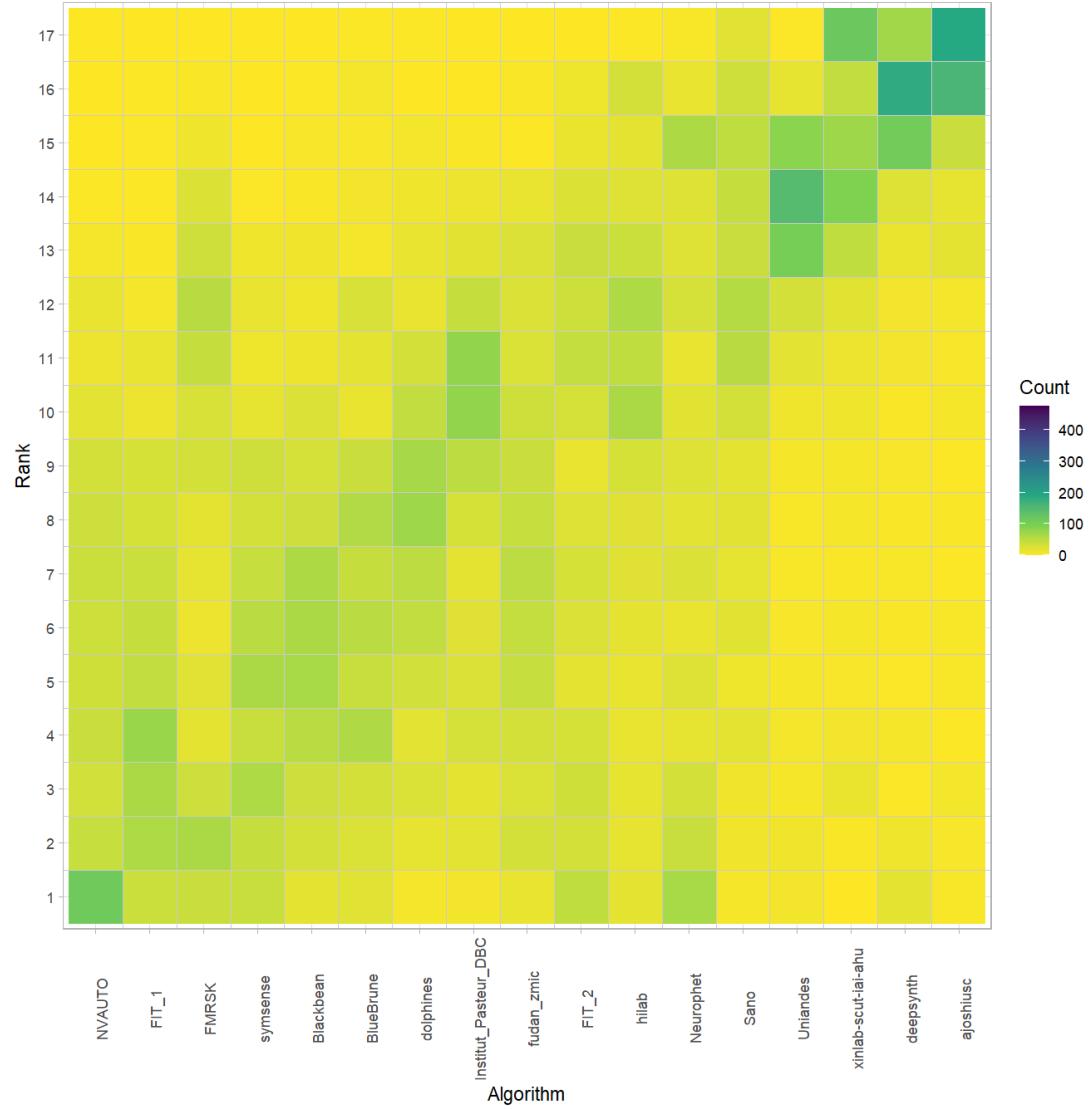
### Podium plot

*Podium plots* (see also Eugster et al., 2008) for visualizing raw assessment data. Upper part (spaghetti plot): Participating algorithms are color-coded, and each colored dot in the plot represents a metric value achieved with the respective algorithm. The actual metric value is encoded by the y-axis. Each podium (here:  $p=17$ ) represents one possible rank, ordered from best (1) to last (here: 17). The assignment of metric values (i.e. colored dots) to one of the podiums is based on the rank that the respective algorithm achieved on the corresponding case. Note that the plot part above each podium place is further subdivided into  $p$  “columns”, where each column represents one participating algorithm (here:  $p = 17$ ). Dots corresponding to identical cases are connected by a line, leading to the shown spaghetti structure. Lower part: Bar charts represent the relative frequency for each algorithm to achieve the rank encoded by the podium place.



## Ranking heatmap

*Ranking heatmaps* for visualizing raw assessment data. Each cell  $(i, A_j)$  shows the absolute frequency of cases in which algorithm  $A_j$  achieved rank  $i$ .

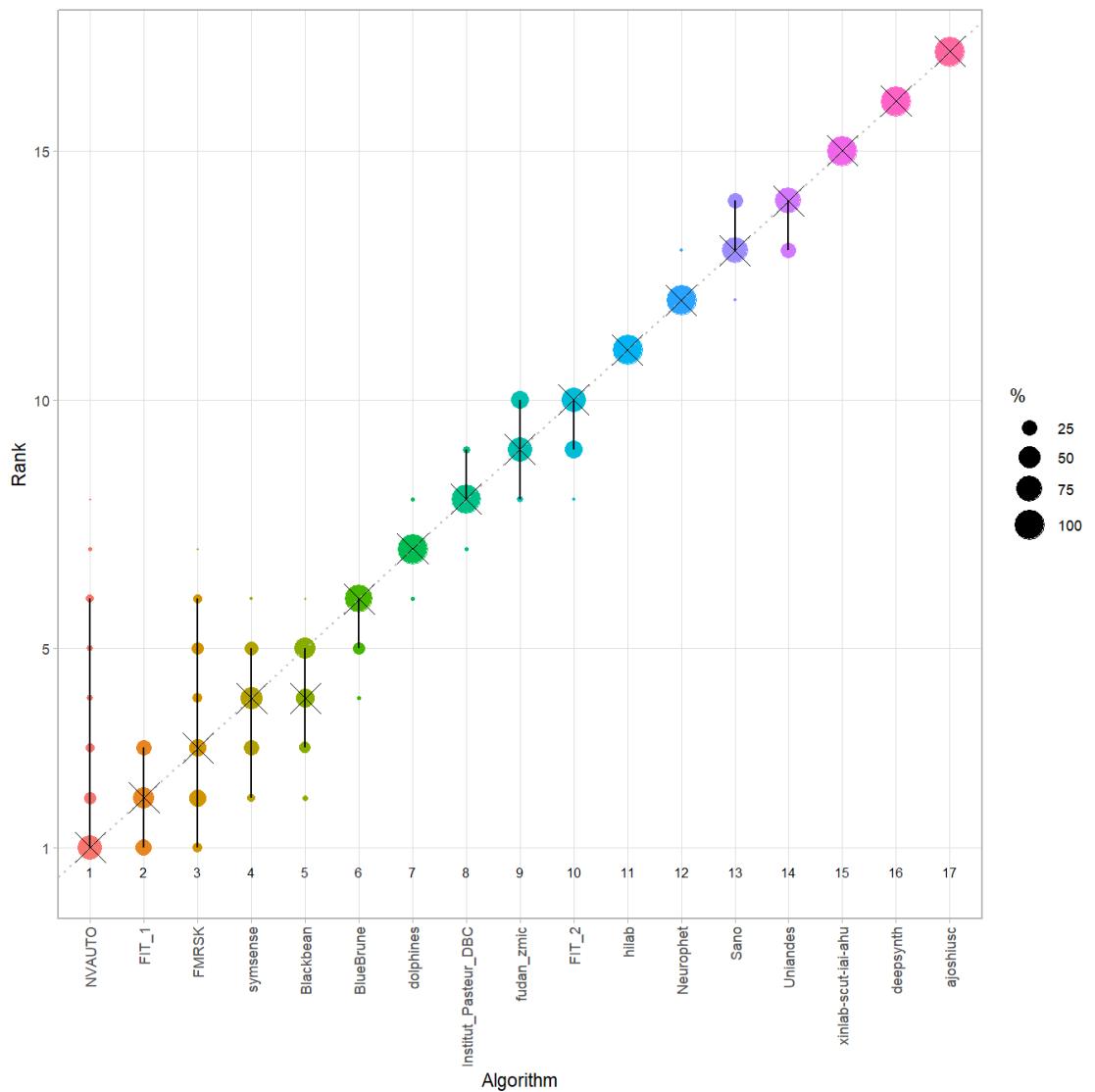


### Visualization of ranking stability

*Blob plot* for visualizing ranking stability based on bootstrap sampling

Algorithms are color-coded, and the area of each blob at position  $(A_i, \text{rank } j)$  is proportional to the relative frequency  $A_i$  achieved rank  $j$  across  $b = 1000$  bootstrap samples. The median rank for each algorithm is indicated by a black cross. 95% bootstrap intervals across bootstrap samples are indicated by black lines.

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.
```

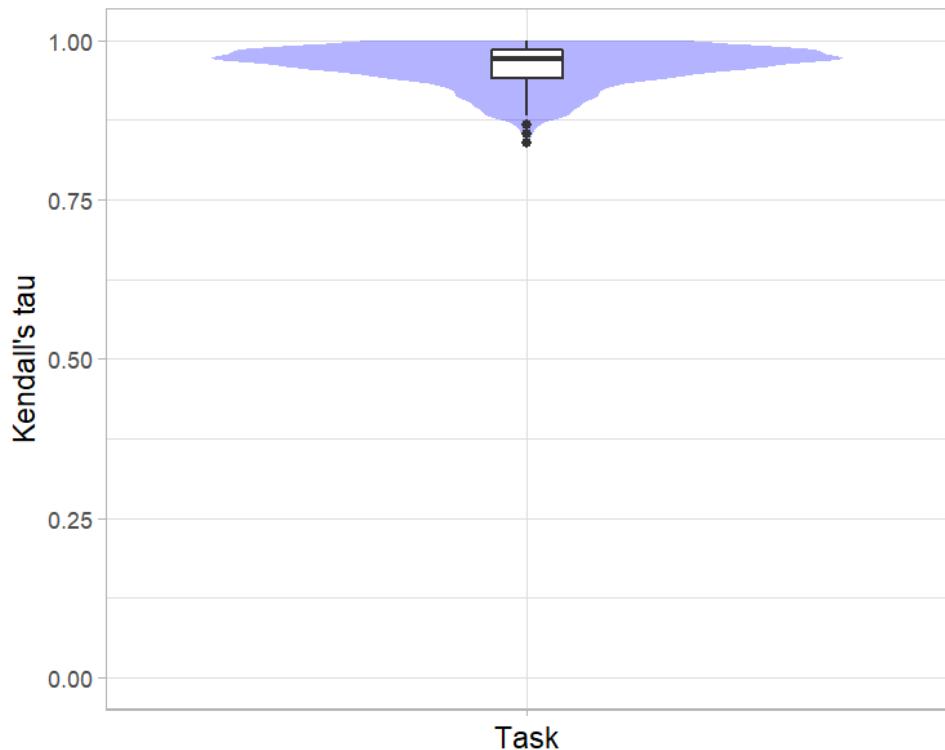


### ***Violin plot for visualizing ranking stability based on bootstrapping***

The ranking list based on the full assessment data is pairwise compared with the ranking lists based on the individual bootstrap samples (here  $b = 1000$  samples). For each pair of rankings, Kendall's  $\tau$  correlation is computed. Kendall's  $\tau$  is a scaled index determining the correlation between the lists. It is computed by evaluating the number of pairwise concordances and discordances between ranking lists and produces values between  $-1$  (for inverted order) and  $1$  (for identical order). A violin plot, which simultaneously depicts a boxplot and a density plot, is generated from the results.

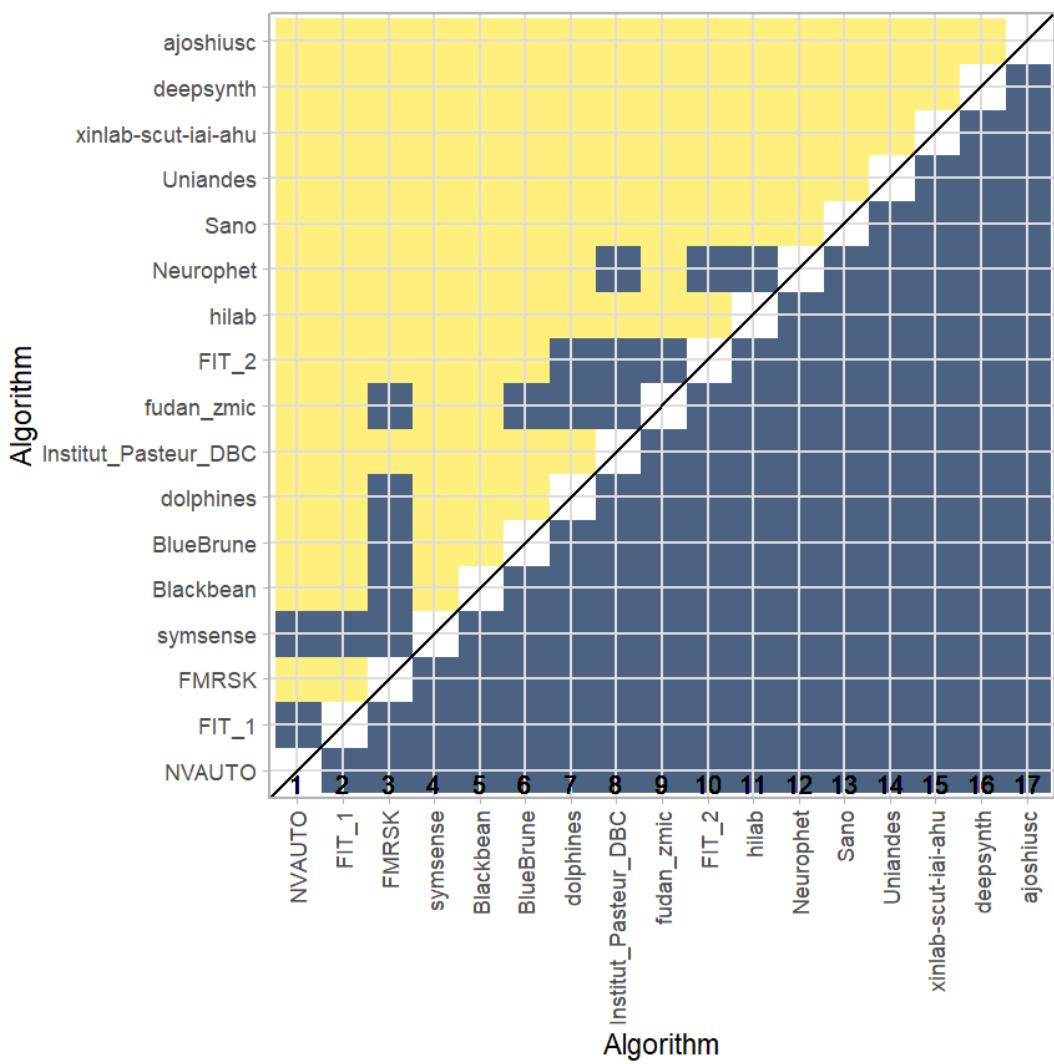
Summary Kendall's tau:

Task	mean	median	q25	q75
dummyTask	0.9591765	0.9705882	0.9411765	0.9852941



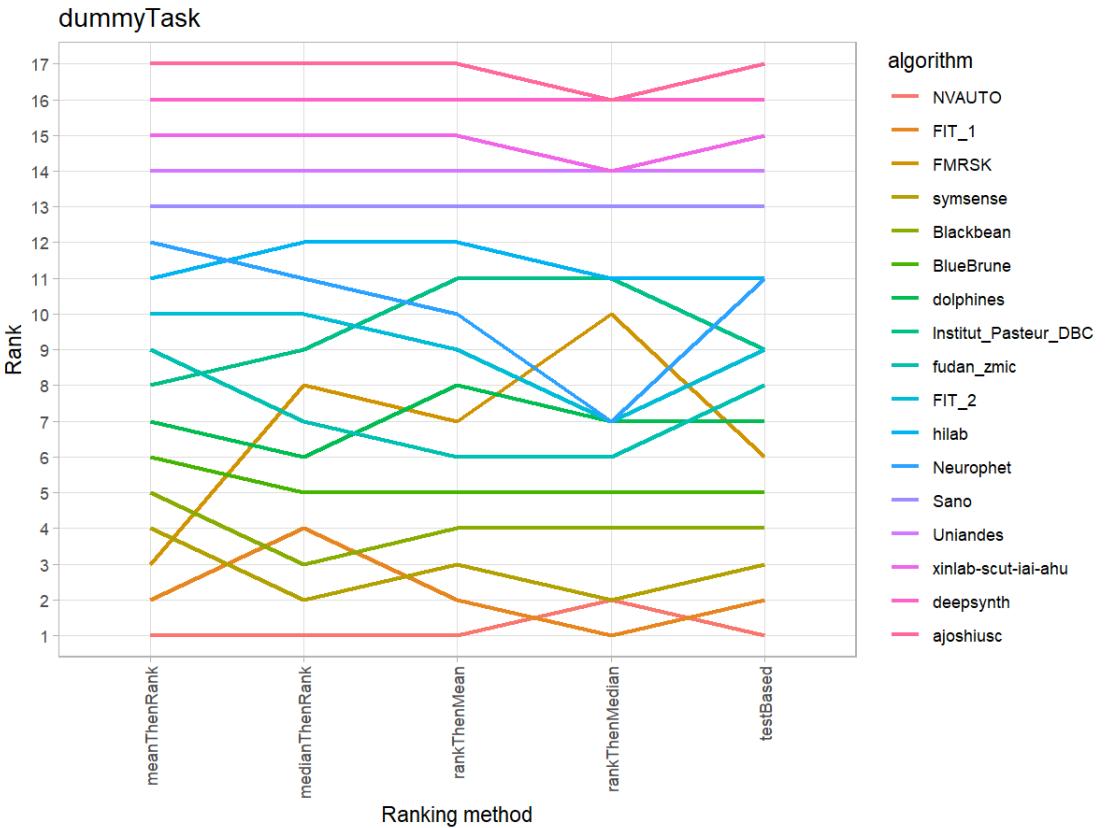
## *Significance maps* for visualizing ranking stability based on statistical significance

*Significance maps* depict incidence matrices of pairwise significant test results for the one-sided Wilcoxon signed rank test at a 5% significance level with adjustment for multiple testing according to Holm. Yellow shading indicates that metric values from the algorithm on the x-axis were significantly superior to those from the algorithm on the y-axis, blue color indicates no significant difference.



### Ranking robustness to ranking methods

*Line plots* for visualizing ranking robustness across different ranking methods. Each algorithm is represented by one colored line. For each ranking method encoded on the x-axis, the height of the line represents the corresponding rank. Horizontal lines indicate identical ranks for all methods.



### 22.3 References

Wiesenfarth, M., Reinke, A., Landman, B.A., Eisenmann, M., Aguilera Saiz, L., Cardoso, M.J., Maier-Hein, L. and Kopp-Schneider, A. Methods and open-source toolkit for analyzing and visualizing challenge results. *Sci Rep* **11**, 2369 (2021). <https://doi.org/10.1038/s41598-021-82017-6>

M. J. A. Eugster, T. Hothorn, and F. Leisch, “Exploratory and inferential analysis of benchmark experiments,” Institut fuer Statistik, Ludwig-Maximilians-Universitaet Muenchen, Germany, Technical Report 30, 2008. [Online]. Available: <http://epub.ub.uni-muenchen.de/4134/>.

## 23 Benchmarking report for Hausdorff Metrics – Neurotypical Brains

created by challengeR v1.0.2  
21 September, 2023

This document presents a systematic report on the benchmark study “Hausdorff Metrics”. Input data comprises raw metric values for all algorithms and cases. Generated plots are:

- Visualization of assessment data: Dot- and boxplot, podium plot and ranking heatmap
- Visualization of ranking stability: Blob plot, violin plot and significance map, line plot

Details can be found in Wiesenfarth et al. (2021).

### 23.1 Ranking

Algorithms within a task are ranked according to the following ranking scheme:  
*aggregate using function (“mean”) then rank*

The analysis is based on 17 algorithms and 476 cases. 0 missing cases have been found in the data set.

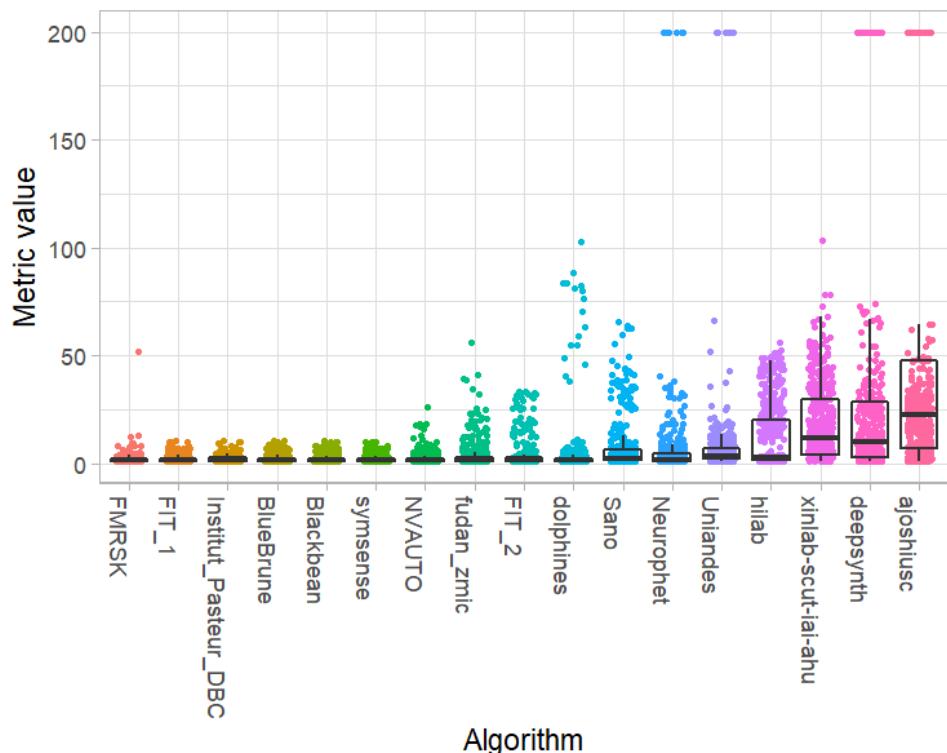
Ranking:

	Hausdorff_mean	rank
FMRSK	2.010238	1
FIT_1	2.062637	2
Institut_Pasteur_DBC	2.069963	3
BlueBrune	2.086025	4
Blackbean	2.091519	5
symsense	2.093226	6
NVAUTO	2.136397	7
fudan_zmic	3.729507	8
FIT_2	4.010603	9
dolphines	4.416571	10
Sano	7.287937	11
Neurophet	7.730027	12
Uniandes	7.860998	13
hilab	12.556709	14
xinlab-scut-iai-ahu	18.092100	15
deepsynth	38.013180	16
ajoshiusc	57.103198	17

## 23.2 Visualization of raw assessment data

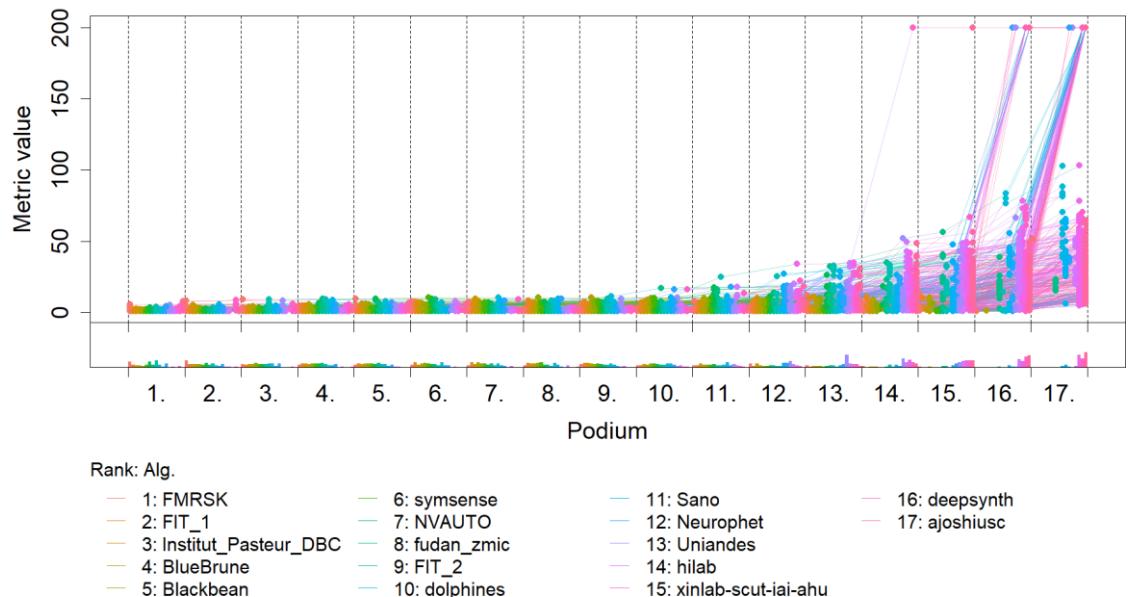
### Dot- and boxplot

*Dot- and boxplots* for visualizing raw assessment data separately for each algorithm. Boxplots representing descriptive statistics over all cases (median, quartiles and outliers) are combined with horizontally jittered dots representing individual cases.



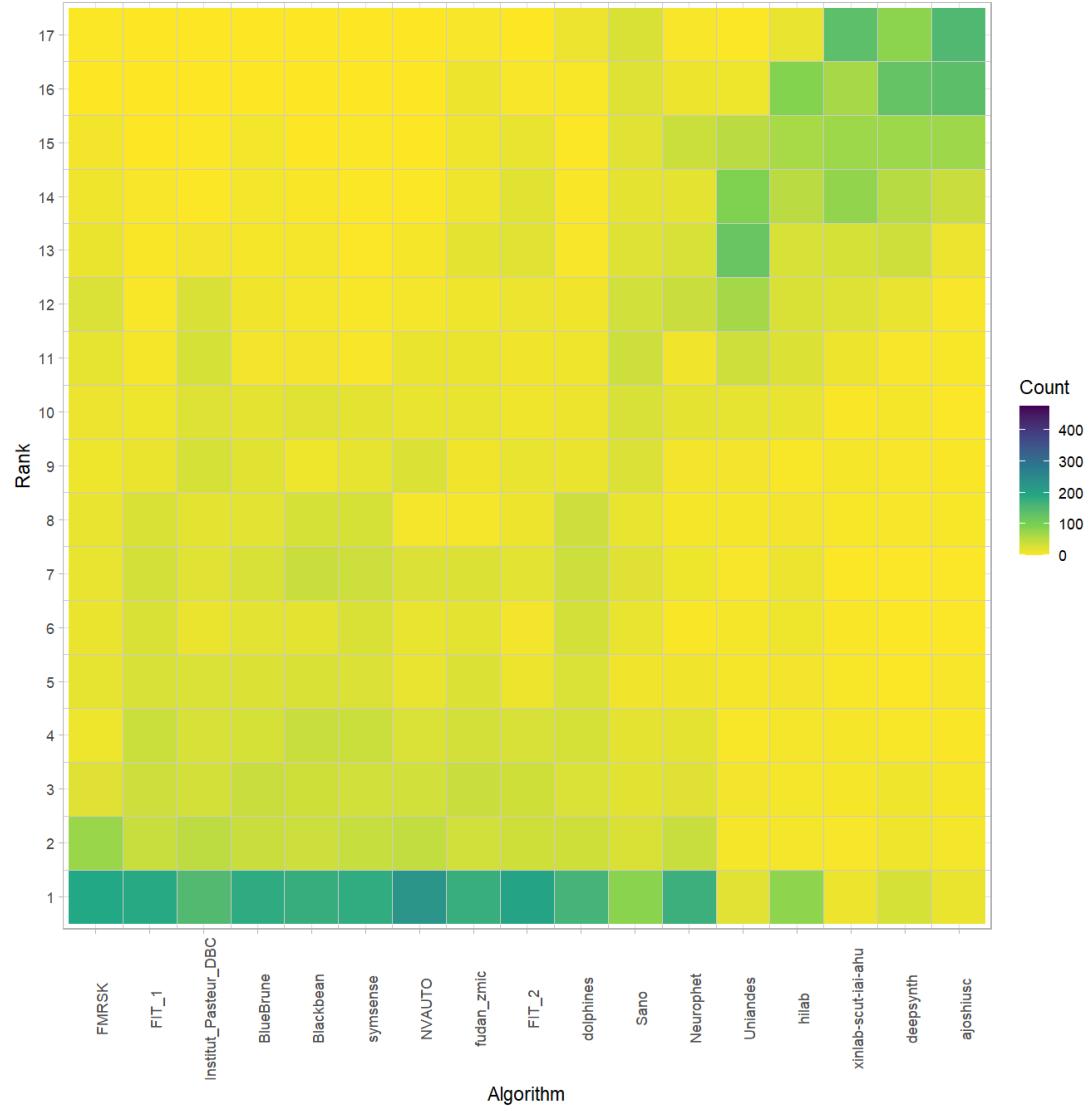
### Podium plot

*Podium plots* (see also Eugster et al., 2008) for visualizing raw assessment data. Upper part (spaghetti plot): Participating algorithms are color-coded, and each colored dot in the plot represents a metric value achieved with the respective algorithm. The actual metric value is encoded by the y-axis. Each podium (here:  $p=17$ ) represents one possible rank, ordered from best (1) to last (here: 17). The assignment of metric values (i.e. colored dots) to one of the podiums is based on the rank that the respective algorithm achieved on the corresponding case. Note that the plot part above each podium place is further subdivided into  $p$  “columns”, where each column represents one participating algorithm (here:  $p = 17$ ). Dots corresponding to identical cases are connected by a line, leading to the shown spaghetti structure. Lower part: Bar charts represent the relative frequency for each algorithm to achieve the rank encoded by the podium place.



## Ranking heatmap

*Ranking heatmaps* for visualizing raw assessment data. Each cell  $(i, A_j)$  shows the absolute frequency of cases in which algorithm  $A_j$  achieved rank  $i$ .

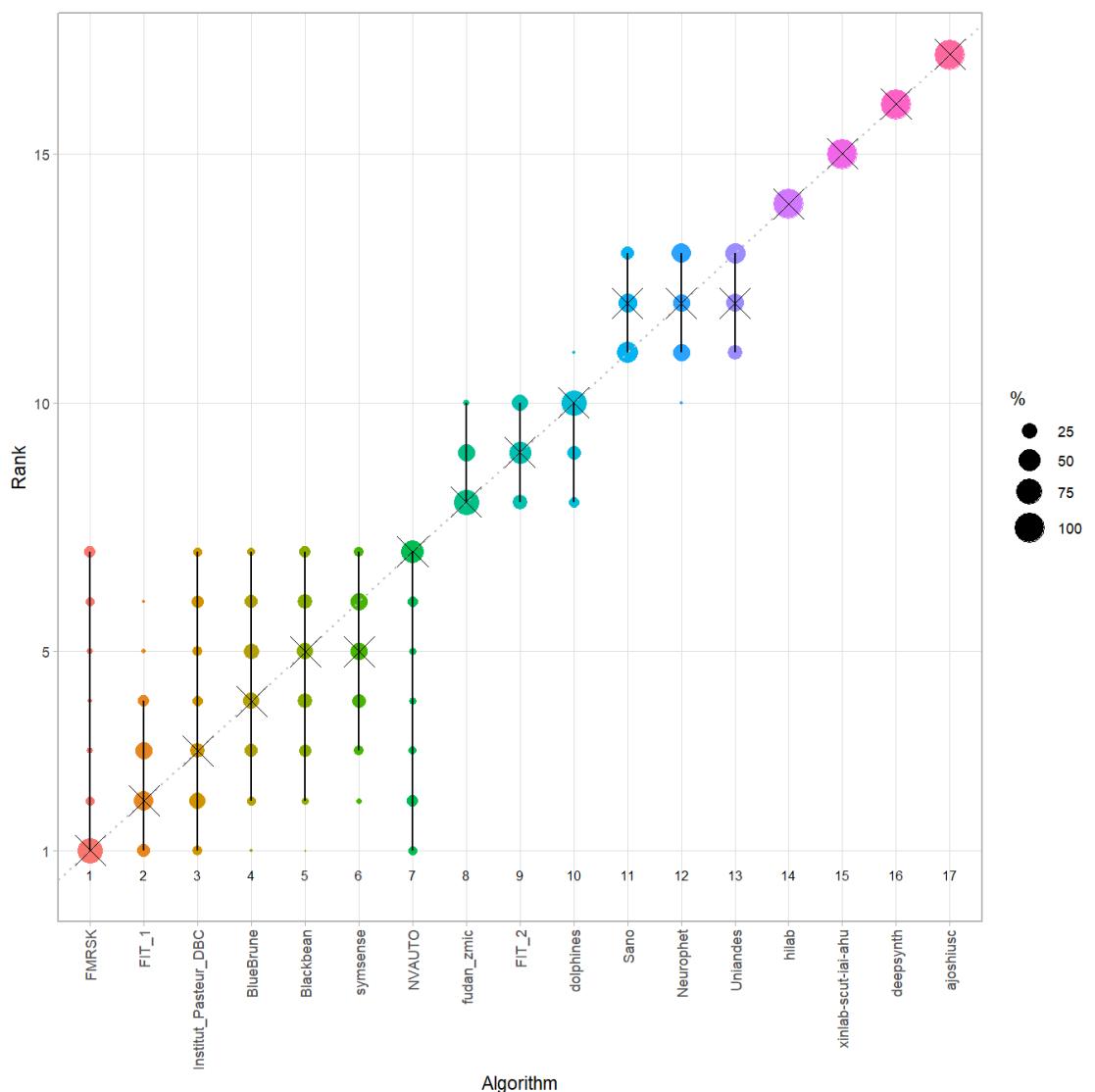


### Visualization of ranking stability

#### *Blob plot for visualizing ranking stability based on bootstrap sampling*

Algorithms are color-coded, and the area of each blob at position ( $A_i$ , rank  $j$ ) is proportional to the relative frequency  $A_i$  achieved rank  $j$  across  $b = 1000$  bootstrap samples. The median rank for each algorithm is indicated by a black cross. 95% bootstrap intervals across bootstrap samples are indicated by black lines.

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.
```

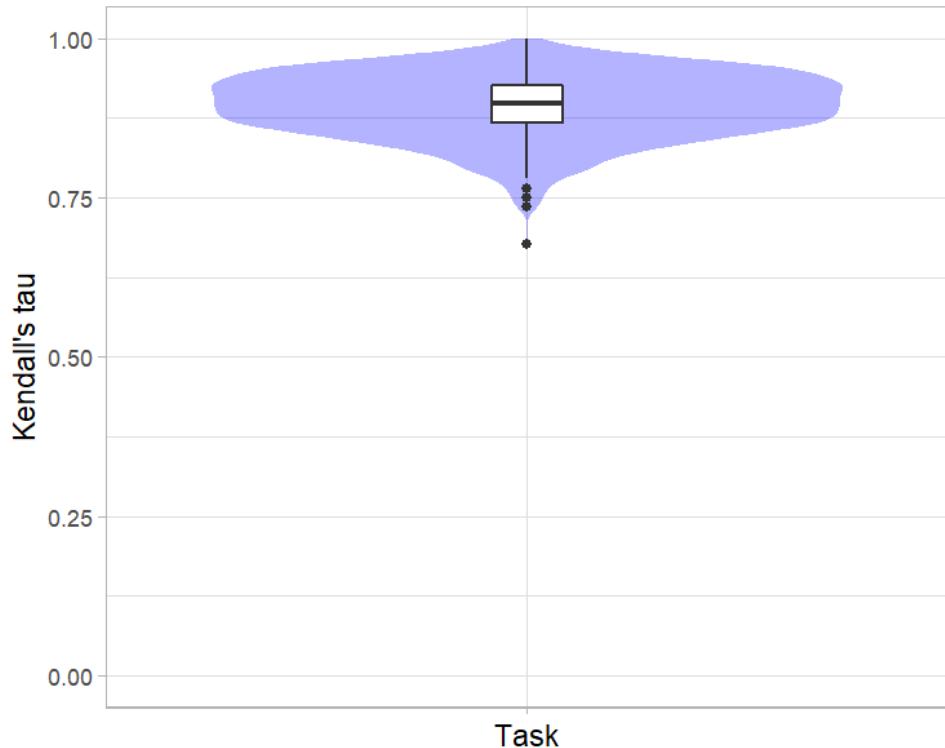


### ***Violin plot for visualizing ranking stability based on bootstrapping***

The ranking list based on the full assessment data is pairwise compared with the ranking lists based on the individual bootstrap samples (here  $b = 1000$  samples). For each pair of rankings, Kendall's  $\tau$  correlation is computed. Kendall's  $\tau$  is a scaled index determining the correlation between the lists. It is computed by evaluating the number of pairwise concordances and discordances between ranking lists and produces values between  $-1$  (for inverted order) and  $1$  (for identical order). A violin plot, which simultaneously depicts a boxplot and a density plot, is generated from the results.

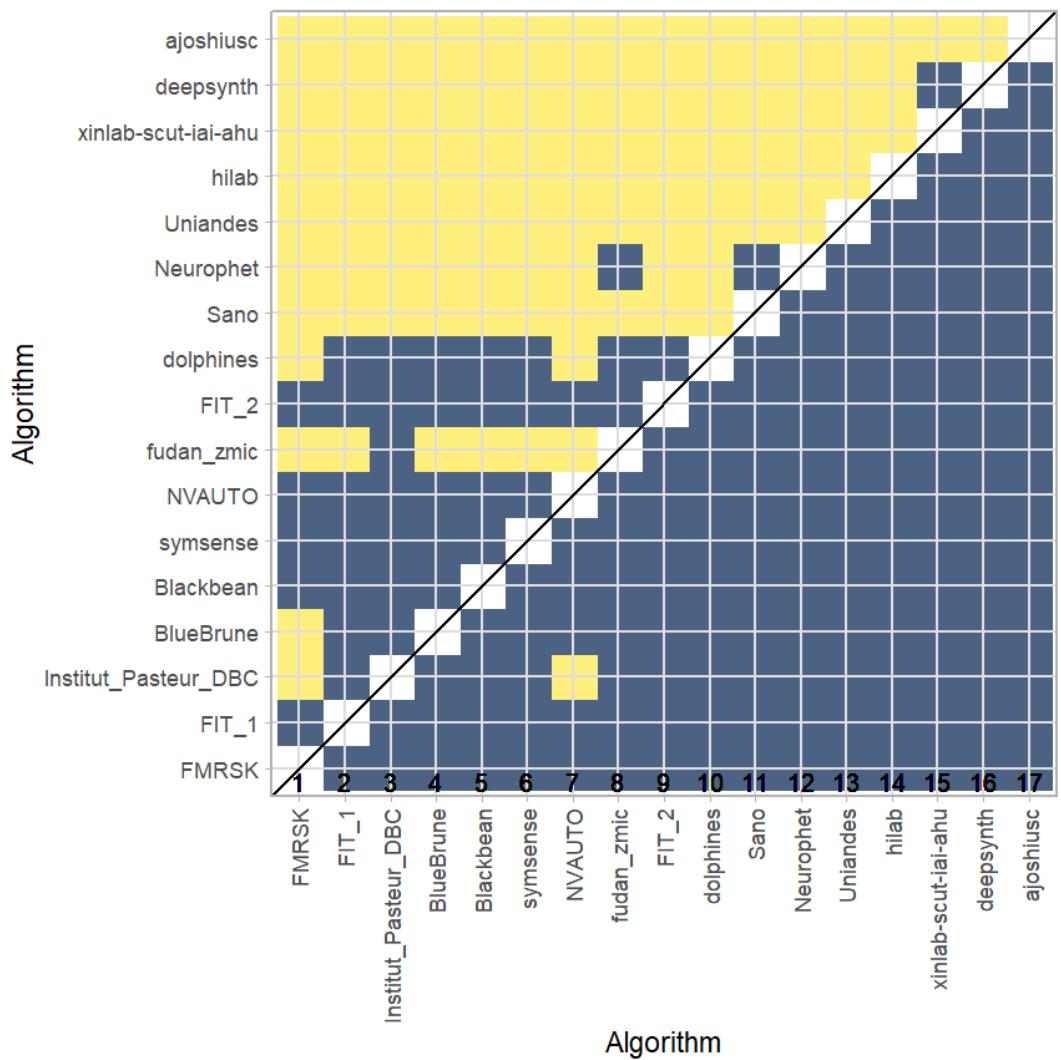
Summary Kendall's tau:

Task	mean	median	q25	q75
dummyTask	0.8938235	0.8970588	0.8676471	0.9264706



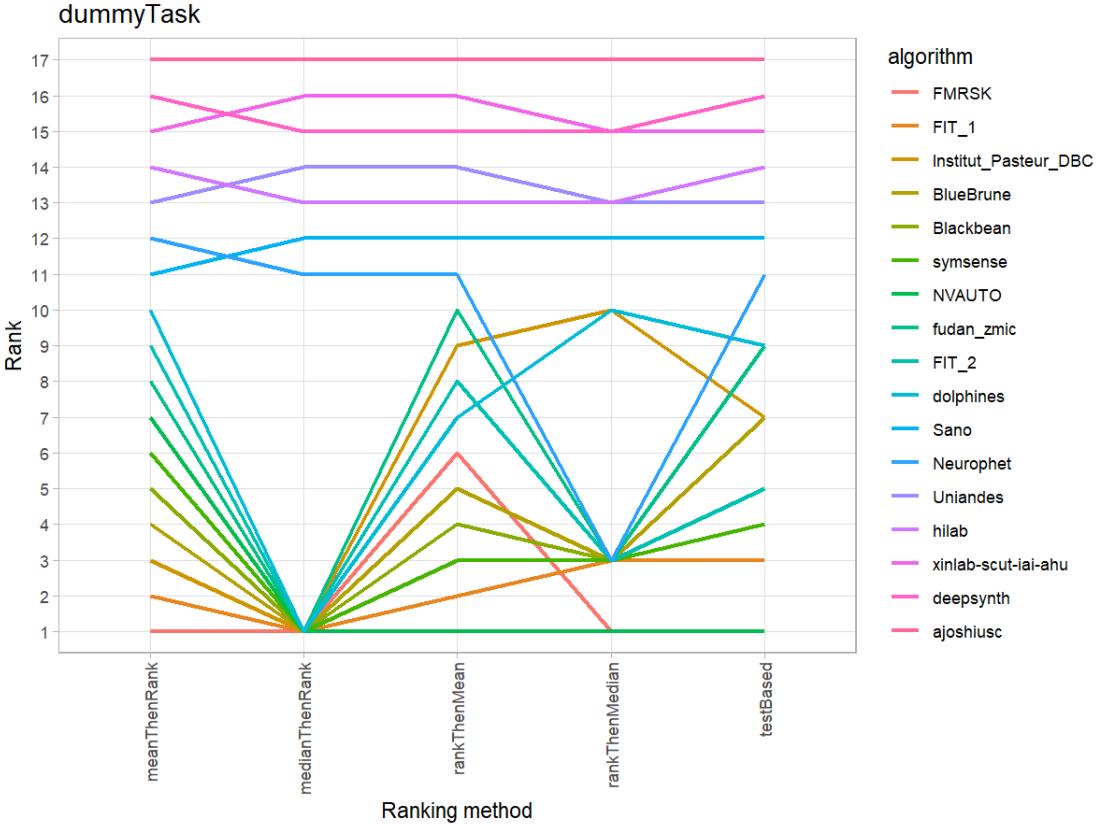
**Significance maps for visualizing ranking stability based on statistical significance**

*Significance maps* depict incidence matrices of pairwise significant test results for the one-sided Wilcoxon signed rank test at a 5% significance level with adjustment for multiple testing according to Holm. Yellow shading indicates that metric values from the algorithm on the x-axis were significantly superior to those from the algorithm on the y-axis, blue color indicates no significant difference.



### Ranking robustness to ranking methods

*Line plots* for visualizing ranking robustness across different ranking methods. Each algorithm is represented by one colored line. For each ranking method encoded on the x-axis, the height of the line represents the corresponding rank. Horizontal lines indicate identical ranks for all methods.



### 23.3 References

Wiesenfarth, M., Reinke, A., Landman, B.A., Eisenmann, M., Aguilera Saiz, L., Cardoso, M.J., Maier-Hein, L. and Kopp-Schneider, A. Methods and open-source toolkit for analyzing and visualizing challenge results. *Sci Rep* **11**, 2369 (2021). <https://doi.org/10.1038/s41598-021-82017-6>

M. J. A. Eugster, T. Hothorn, and F. Leisch, “Exploratory and inferential analysis of benchmark experiments,” Institut fuer Statistik, Ludwig-Maximilians-Universitaet Muenchen, Germany, Technical Report 30, 2008. [Online]. Available: <http://epub.ub.uni-muenchen.de/4134/>.

## 24 Benchmarking report for Volume Similarity Metrics – Neurotypical Brains

created by challengeR v1.0.2  
21 September, 2023

This document presents a systematic report on the benchmark study “Volume Similarity Metrics”. Input data comprises raw metric values for all algorithms and cases. Generated plots are:

Visualization of assessment data: Dot- and boxplot, podium plot and ranking heatmap

Visualization of ranking stability: Blob plot, violin plot and significance map, line plot

Details can be found in Wiesenfarth et al. (2021).

### 24.1 Ranking

Algorithms within a task are ranked according to the following ranking scheme:

*aggregate using function (“mean”) then rank*

The analysis is based on 17 algorithms and 476 cases. 0 missing cases have been found in the data set.

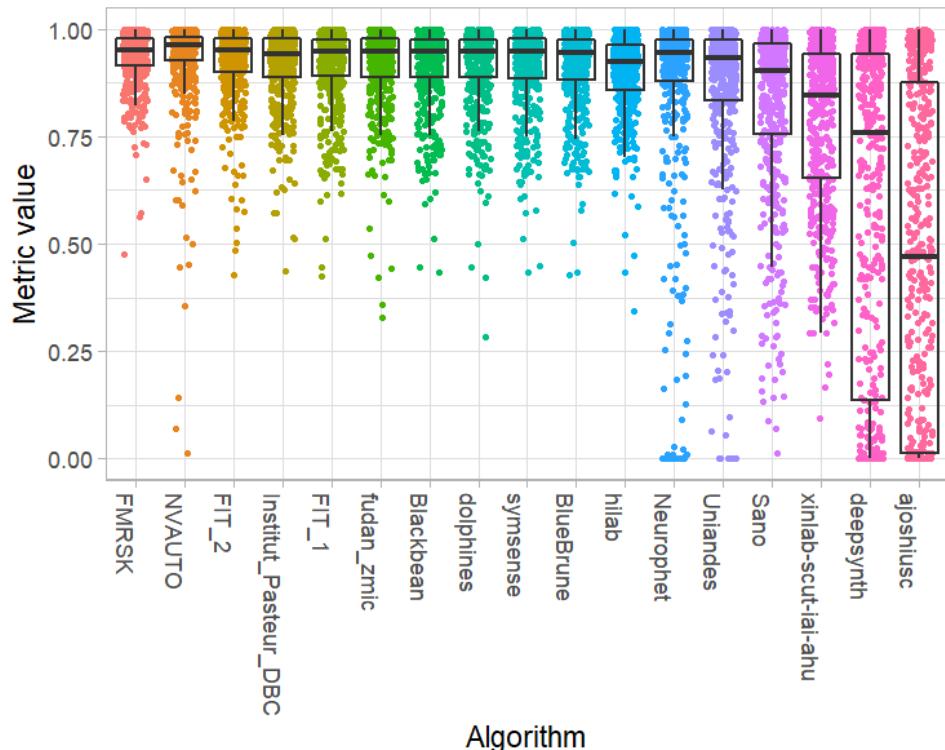
Ranking:

	Volume_Similarity_mean	rank
FMRSK	0.9327262	1
NVAUTO	0.9300129	2
FIT_2	0.9235789	3
Institut_Pasteur_DBC	0.9160553	4
FIT_1	0.9145272	5
fudan_zmic	0.9131280	6
Blackbean	0.9129656	7
dolphines	0.9120900	8
symsense	0.9115349	9
BlueBrune	0.9104248	10
hilab	0.8982362	11
Neurophet	0.8559453	12
Uniandes	0.8545801	13
Sano	0.8195460	14
xinlab-scut-iai-ahu	0.7826698	15
deepsynth	0.5894117	16
ajoshiusc	0.4655215	17

## 24.2 Visualization of raw assessment data

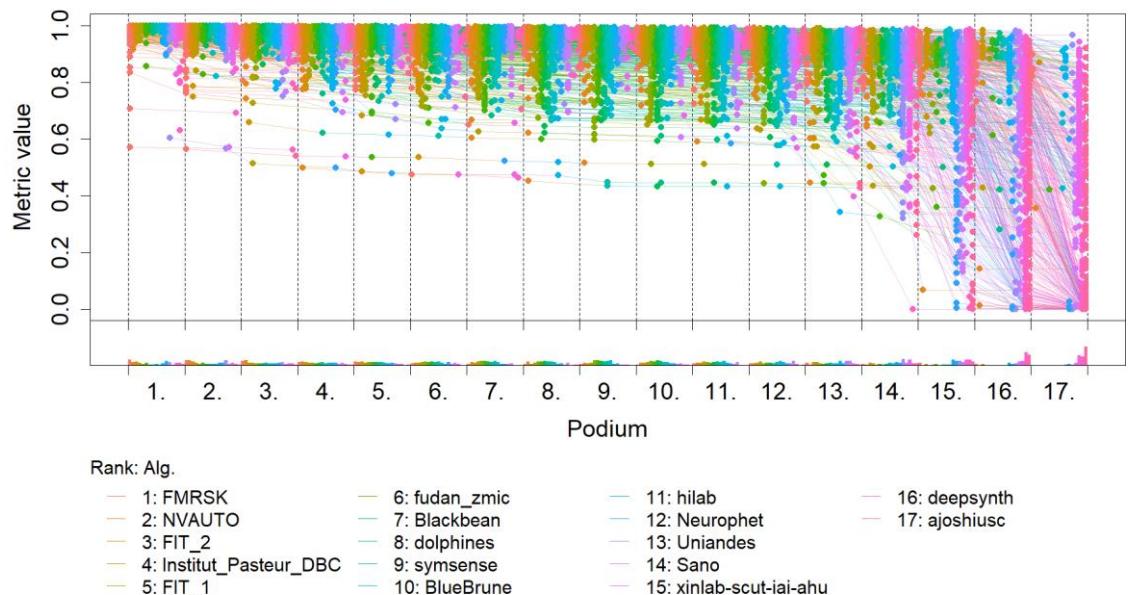
### Dot- and boxplot

*Dot- and boxplots* for visualizing raw assessment data separately for each algorithm. Boxplots representing descriptive statistics over all cases (median, quartiles and outliers) are combined with horizontally jittered dots representing individual cases.



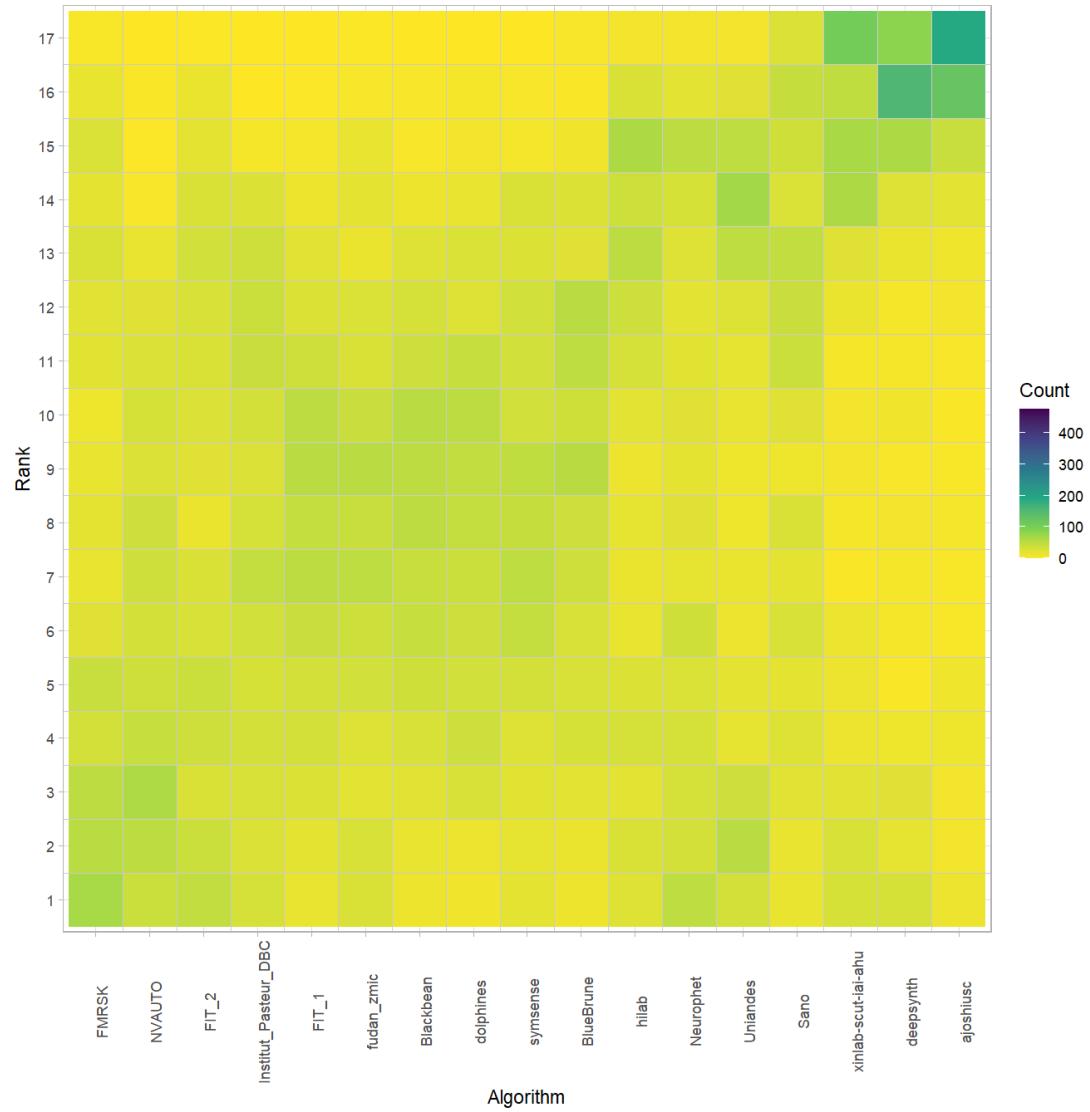
## Podium plot

*Podium plots* (see also Eugster et al., 2008) for visualizing raw assessment data. Upper part (spaghetti plot): Participating algorithms are color-coded, and each colored dot in the plot represents a metric value achieved with the respective algorithm. The actual metric value is encoded by the y-axis. Each podium (here:  $p=17$ ) represents one possible rank, ordered from best (1) to last (here: 17). The assignment of metric values (i.e. colored dots) to one of the podiums is based on the rank that the respective algorithm achieved on the corresponding case. Note that the plot part above each podium place is further subdivided into  $p$  “columns”, where each column represents one participating algorithm (here:  $p = 17$ ). Dots corresponding to identical cases are connected by a line, leading to the shown spaghetti structure. Lower part: Bar charts represent the relative frequency for each algorithm to achieve the rank encoded by the podium place.



## Ranking heatmap

*Ranking heatmaps* for visualizing raw assessment data. Each cell  $(i, A_j)$  shows the absolute frequency of cases in which algorithm  $A_j$  achieved rank  $i$ .

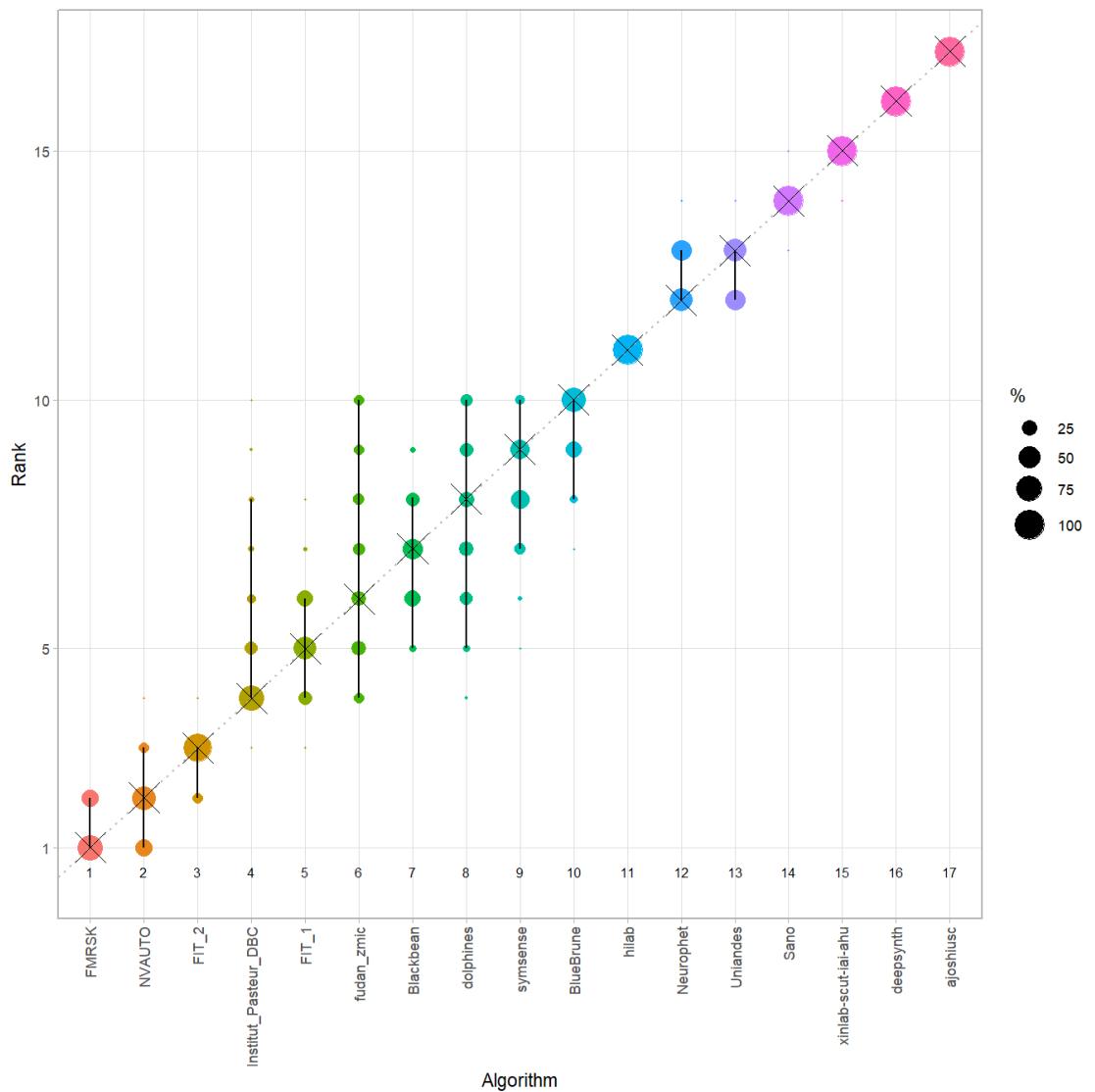


### Visualization of ranking stability

*Blob plot* for visualizing ranking stability based on bootstrap sampling

Algorithms are color-coded, and the area of each blob at position  $(A_i, \text{rank } j)$  is proportional to the relative frequency  $A_i$  achieved rank  $j$  across  $b = 1000$  bootstrap samples. The median rank for each algorithm is indicated by a black cross. 95% bootstrap intervals across bootstrap samples are indicated by black lines.

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.
```

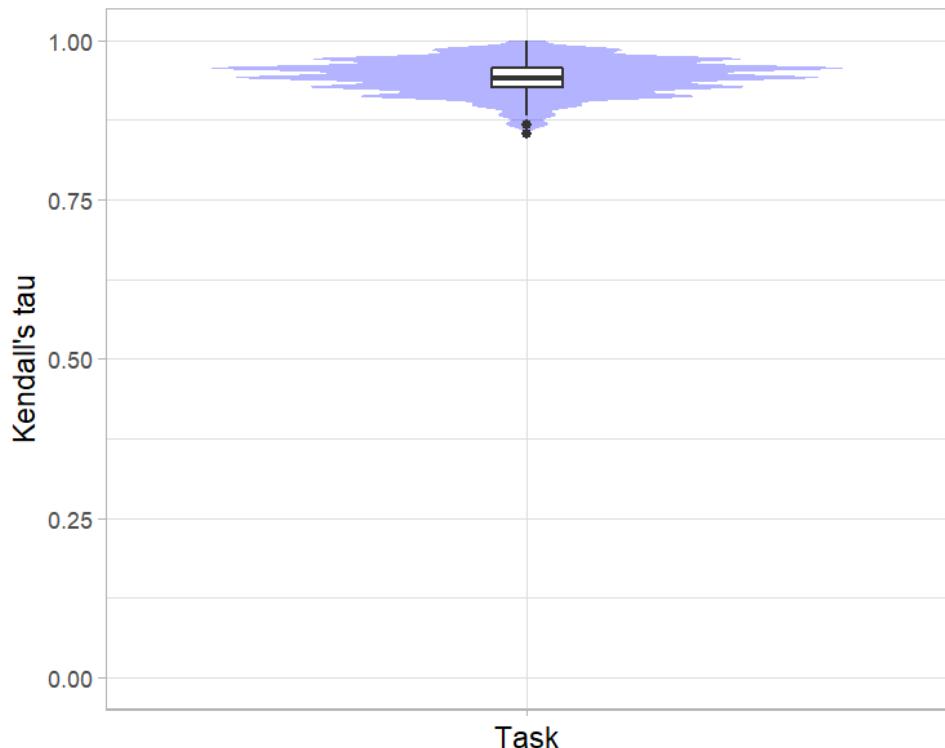


### ***Violin plot for visualizing ranking stability based on bootstrapping***

The ranking list based on the full assessment data is pairwise compared with the ranking lists based on the individual bootstrap samples (here  $b = 1000$  samples). For each pair of rankings, Kendall's  $\tau$  correlation is computed. Kendall's  $\tau$  is a scaled index determining the correlation between the lists. It is computed by evaluating the number of pairwise concordances and discordances between ranking lists and produces values between  $-1$  (for inverted order) and  $1$  (for identical order). A violin plot, which simultaneously depicts a boxplot and a density plot, is generated from the results.

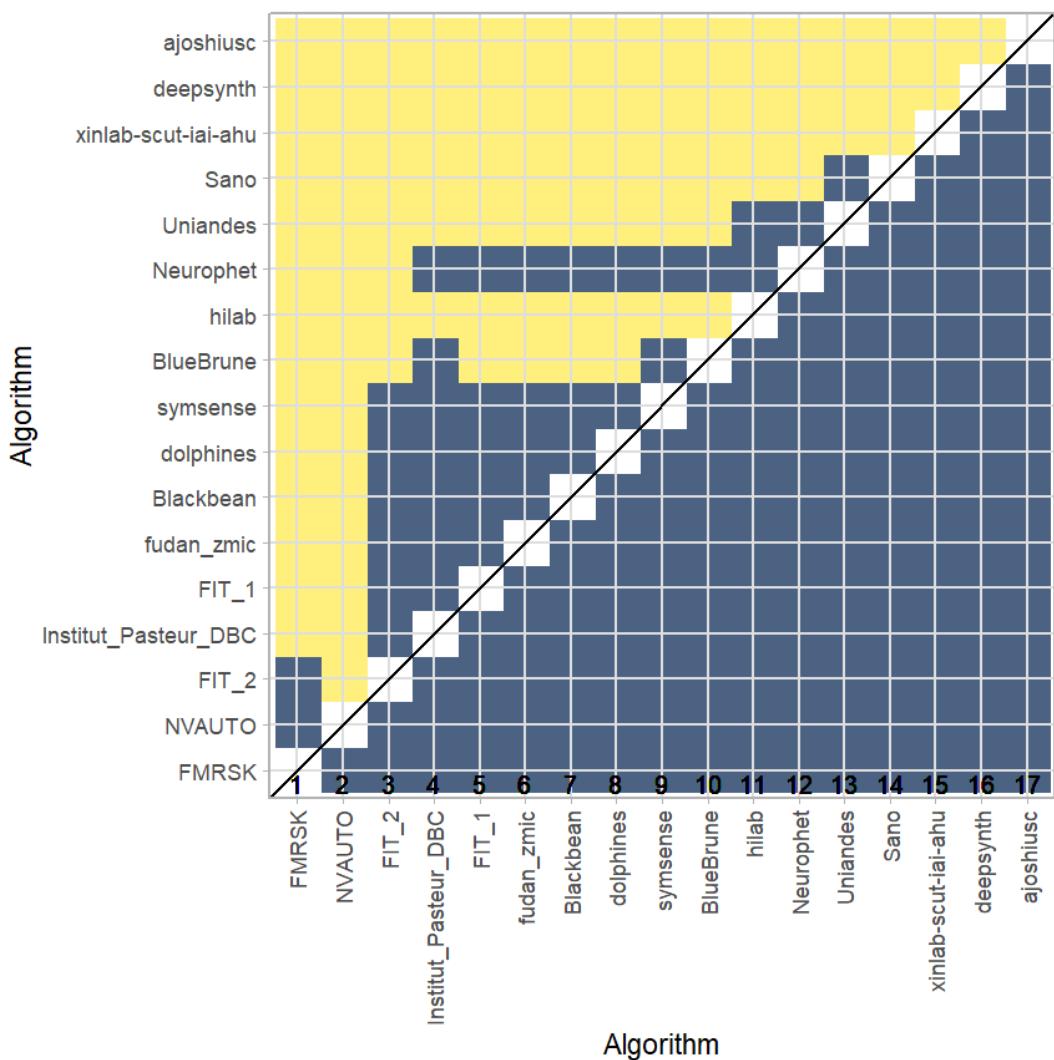
Summary Kendall's tau:

Task	mean	median	q25	q75
dummyTask	0.9426176	0.9411765	0.9264706	0.9558824



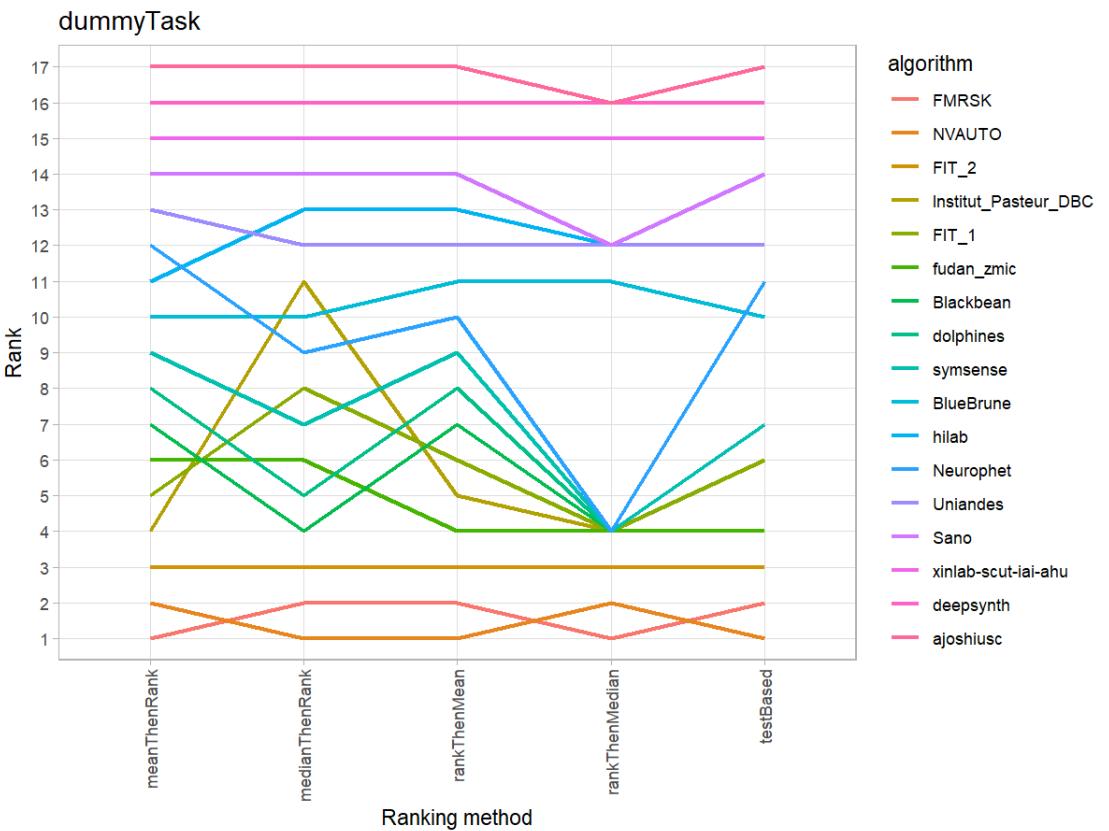
## *Significance maps* for visualizing ranking stability based on statistical significance

*Significance maps* depict incidence matrices of pairwise significant test results for the one-sided Wilcoxon signed rank test at a 5% significance level with adjustment for multiple testing according to Holm. Yellow shading indicates that metric values from the algorithm on the x-axis were significantly superior to those from the algorithm on the y-axis, blue color indicates no significant difference.



### Ranking robustness to ranking methods

*Line plots* for visualizing ranking robustness across different ranking methods. Each algorithm is represented by one colored line. For each ranking method encoded on the x-axis, the height of the line represents the corresponding rank. Horizontal lines indicate identical ranks for all methods.



### 24.3 References

Wiesenfarth, M., Reinke, A., Landman, B.A., Eisenmann, M., Aguilera Saiz, L., Cardoso, M.J., Maier-Hein, L. and Kopp-Schneider, A. Methods and open-source toolkit for analyzing and visualizing challenge results. *Sci Rep* **11**, 2369 (2021). <https://doi.org/10.1038/s41598-021-82017-6>

M. J. A. Eugster, T. Hothorn, and F. Leisch, “Exploratory and inferential analysis of benchmark experiments,” Institut fuer Statistik, Ludwig-Maximilians-Universitaet Muenchen, Germany, Technical Report 30, 2008. [Online]. Available: <http://epub.ub.uni-muenchen.de/4134/>.

## 25 Benchmarking report for Dice Metrics – Pathological Brains

created by challengeR v1.0.2  
21 September, 2023

This document presents a systematic report on the benchmark study “Dice Metrics”. Input data comprises raw metric values for all algorithms and cases. Generated plots are:

- Visualization of assessment data: Dot- and boxplot, podium plot and ranking heatmap
- Visualization of ranking stability: Blob plot, violin plot and significance map, line plot

Details can be found in Wiesenfarth et al. (2021).

### 25.1 Ranking

Algorithms within a task are ranked according to the following ranking scheme:

*aggregate using function (“mean”) then rank*

The analysis is based on 17 algorithms and 644 cases. 0 missing cases have been found in the data set.

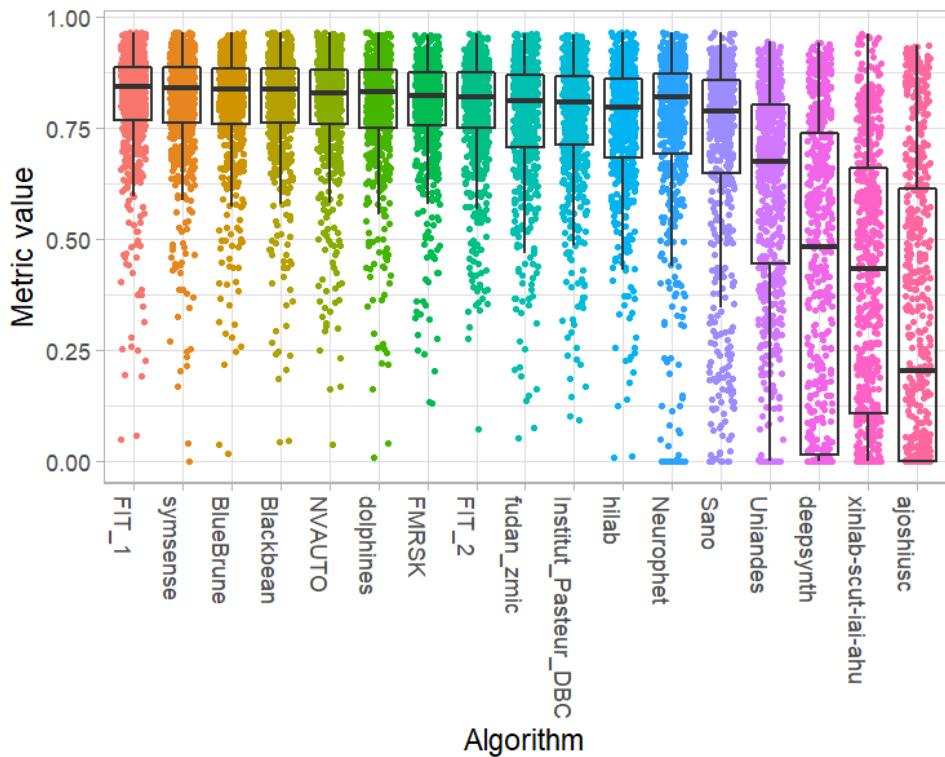
Ranking:

	Dice_mean	rank
FIT_1	0.8065845	1
symsense	0.8036268	2
BlueBrune	0.8027996	3
Blackbean	0.8016119	4
NVAUTO	0.7950238	5
dolphines	0.7946791	6
FMRSK	0.7938482	7
FIT_2	0.7884127	8
fudan_zmic	0.7701964	9
Institut_Pasteur_DBC	0.7677456	10
hilab	0.7555400	11
Neurophet	0.7312100	12
Sano	0.7024996	13
Uniandes	0.6086597	14
deepsynth	0.4262748	15
xinlab-scut-iai-ahu	0.4092711	16
ajoshiusc	0.3205442	17

## 25.2 Visualization of raw assessment data

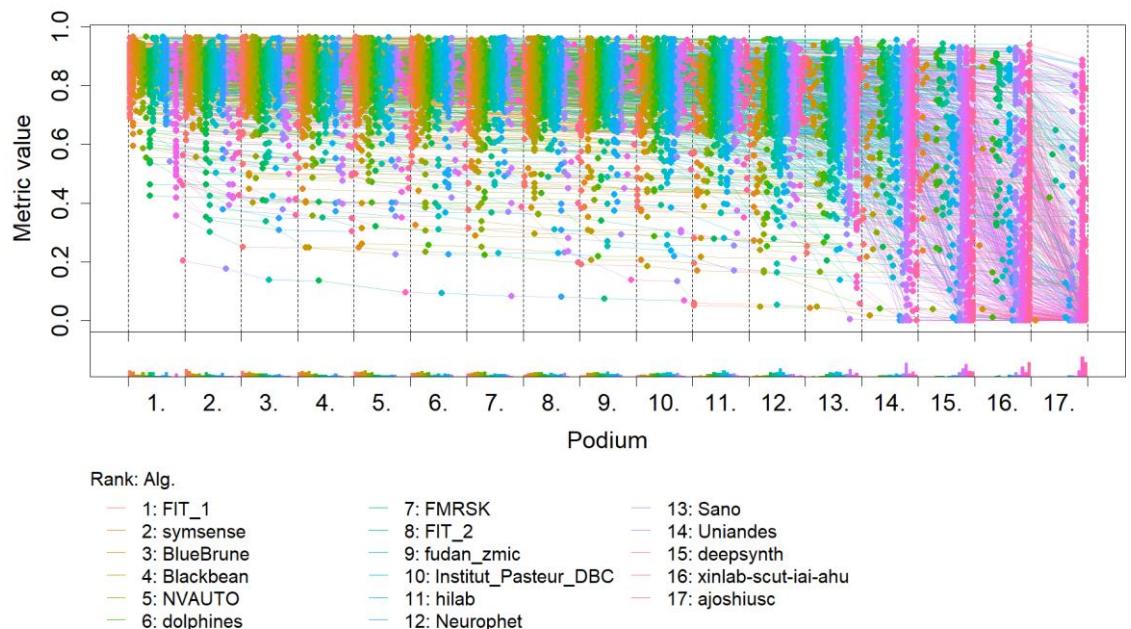
### Dot- and boxplot

*Dot- and boxplots* for visualizing raw assessment data separately for each algorithm. Boxplots representing descriptive statistics over all cases (median, quartiles and outliers) are combined with horizontally jittered dots representing individual cases.



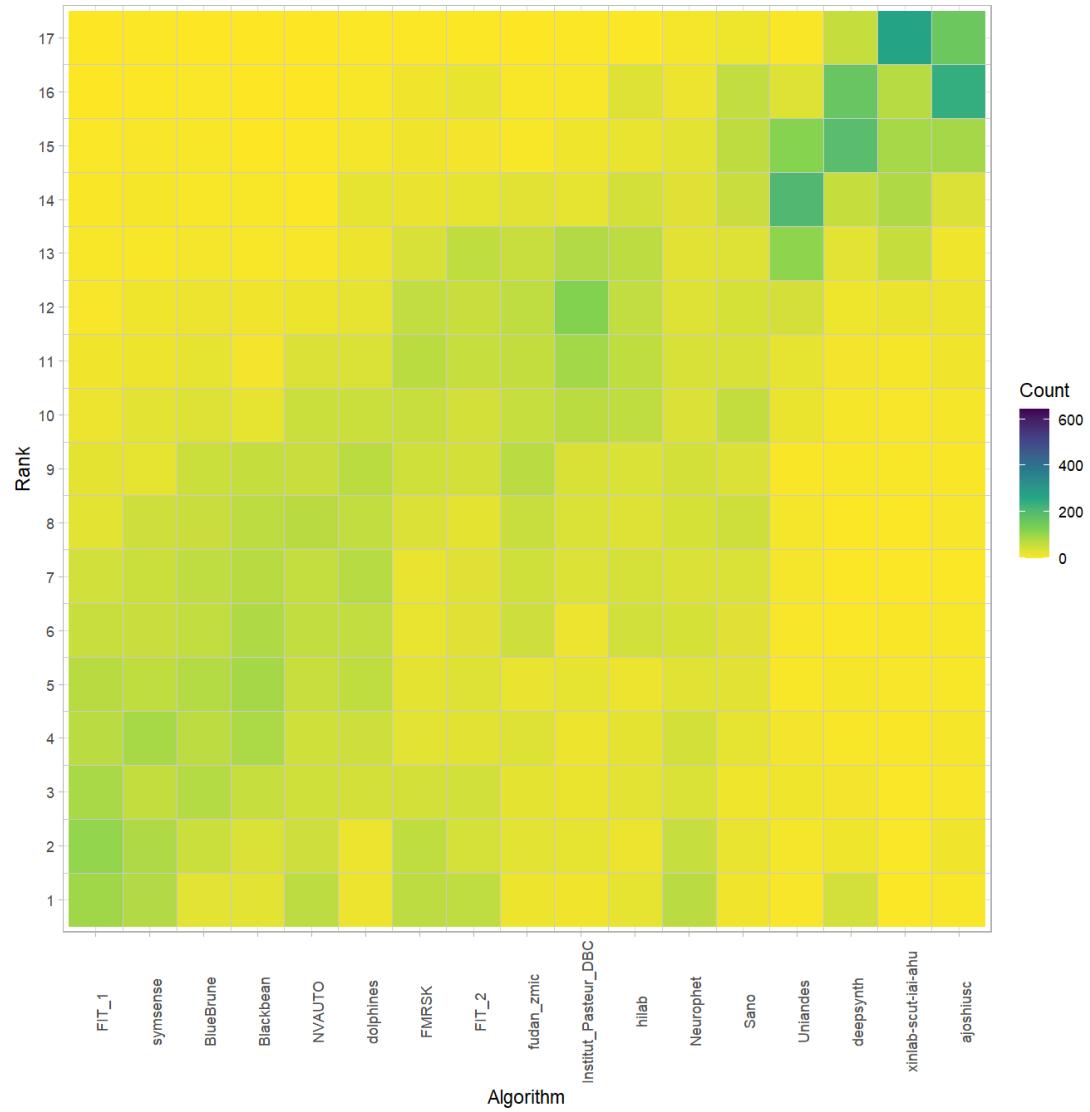
### Podium plot

*Podium plots* (see also Eugster et al., 2008) for visualizing raw assessment data. Upper part (spaghetti plot): Participating algorithms are color-coded, and each colored dot in the plot represents a metric value achieved with the respective algorithm. The actual metric value is encoded by the y-axis. Each podium (here:  $p=17$ ) represents one possible rank, ordered from best (1) to last (here: 17). The assignment of metric values (i.e. colored dots) to one of the podiums is based on the rank that the respective algorithm achieved on the corresponding case. Note that the plot part above each podium place is further subdivided into  $p$  “columns”, where each column represents one participating algorithm (here:  $p = 17$ ). Dots corresponding to identical cases are connected by a line, leading to the shown spaghetti structure. Lower part: Bar charts represent the relative frequency for each algorithm to achieve the rank encoded by the podium place.



## Ranking heatmap

*Ranking heatmaps* for visualizing raw assessment data. Each cell  $(i, A_j)$  shows the absolute frequency of cases in which algorithm  $A_j$  achieved rank  $i$ .

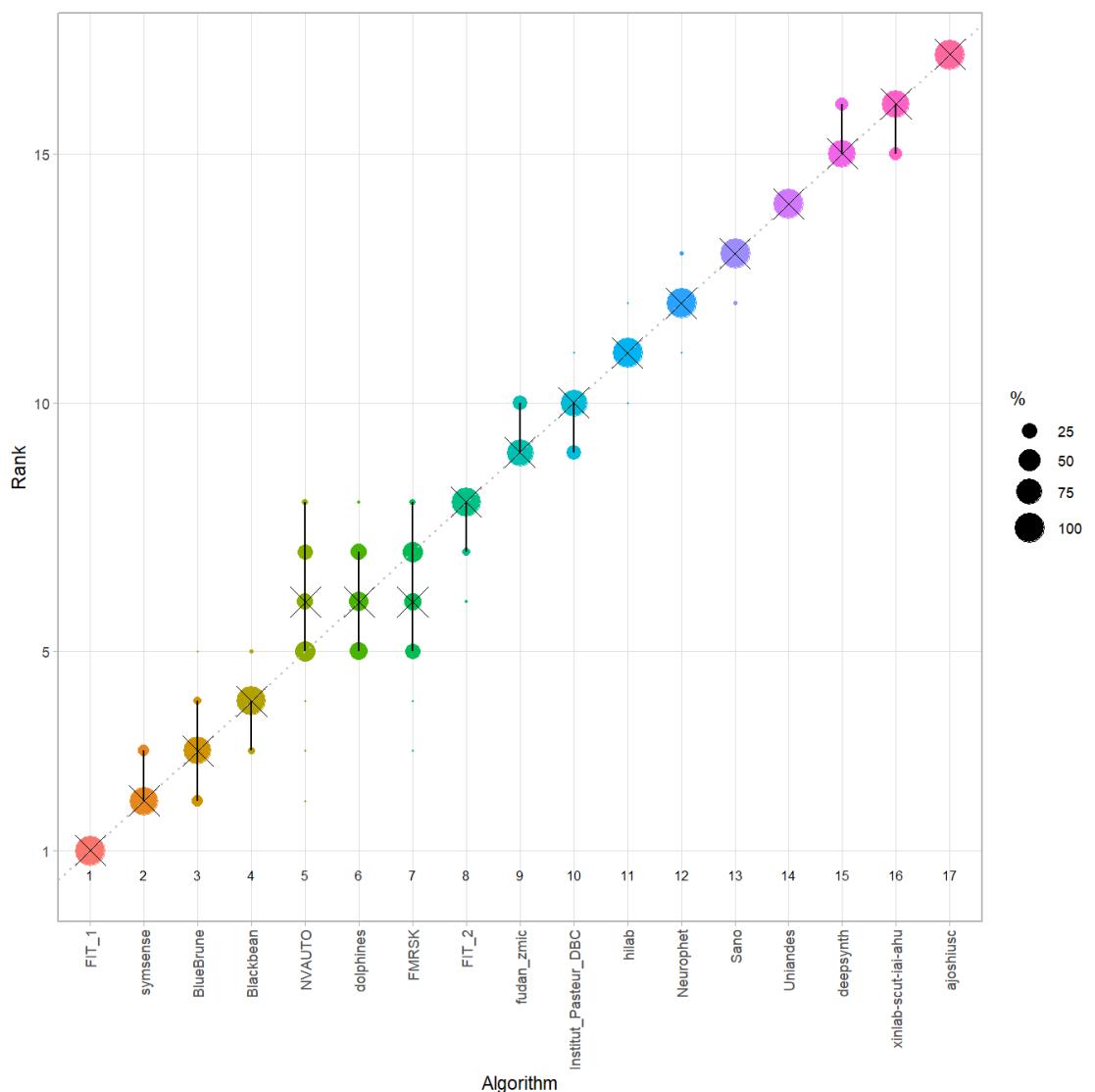


## Visualization of ranking stability

## *Blob plot* for visualizing ranking stability based on bootstrap sampling

Algorithms are color-coded, and the area of each blob at position ( $A_i$ , rank  $j$ ) is proportional to the relative frequency  $A_i$  achieved rank  $j$  across  $b = 1000$  bootstrap samples. The median rank for each algorithm is indicated by a black cross. 95% bootstrap intervals across bootstrap samples are indicated by black lines.

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> = "none")` instead.
```

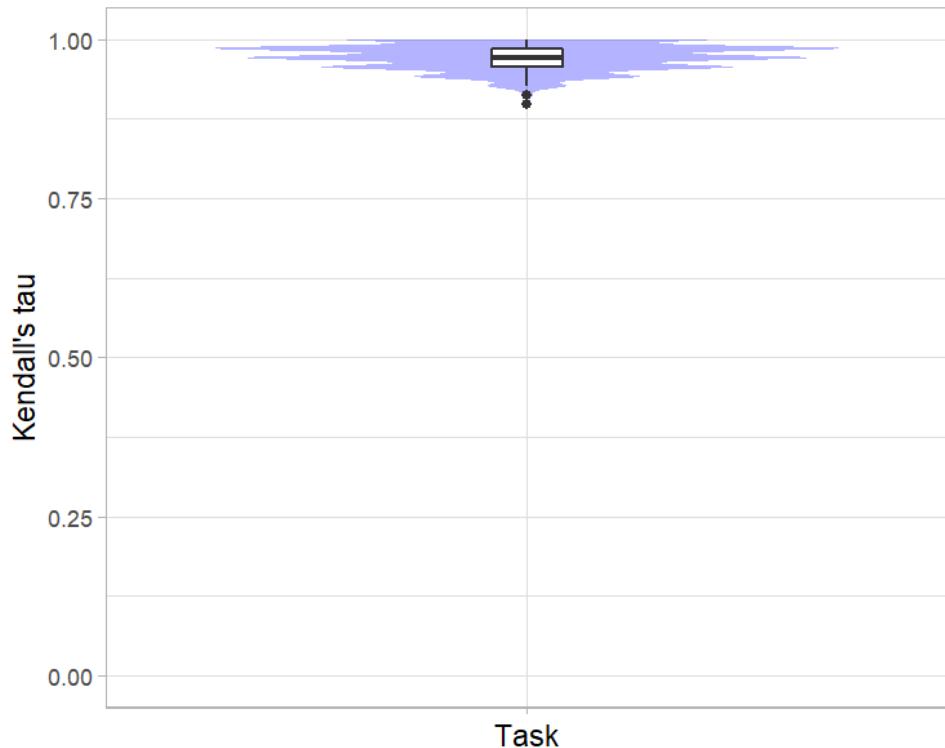


### **Violin plot for visualizing ranking stability based on bootstrapping**

The ranking list based on the full assessment data is pairwise compared with the ranking lists based on the individual bootstrap samples (here  $b = 1000$  samples). For each pair of rankings, Kendall's  $\tau$  correlation is computed. Kendall's  $\tau$  is a scaled index determining the correlation between the lists. It is computed by evaluating the number of pairwise concordances and discordances between ranking lists and produces values between  $-1$  (for inverted order) and  $1$  (for identical order). A violin plot, which simultaneously depicts a boxplot and a density plot, is generated from the results.

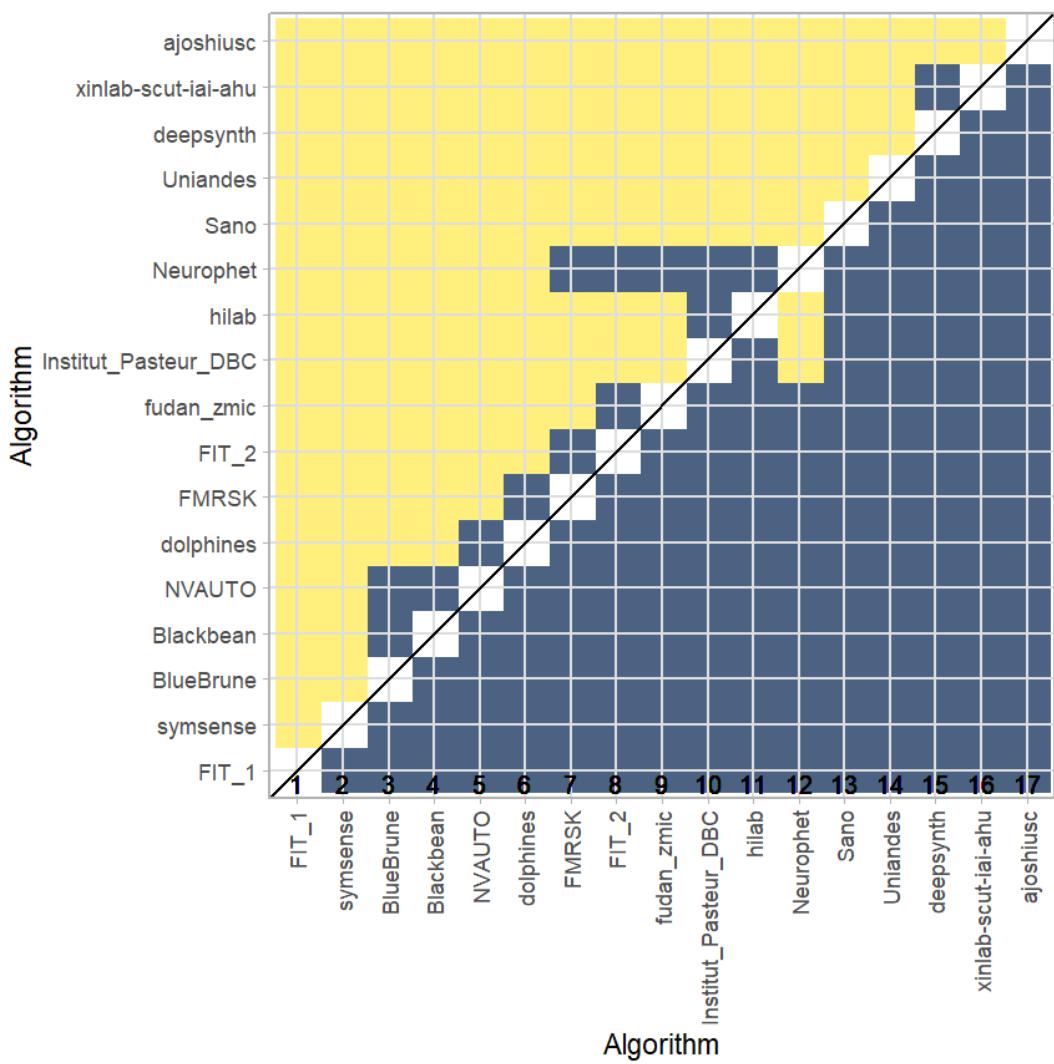
Summary Kendall's tau:

Task	mean	median	q25	q75
dummyTask	0.9720735	0.9705882	0.9558824	0.9852941



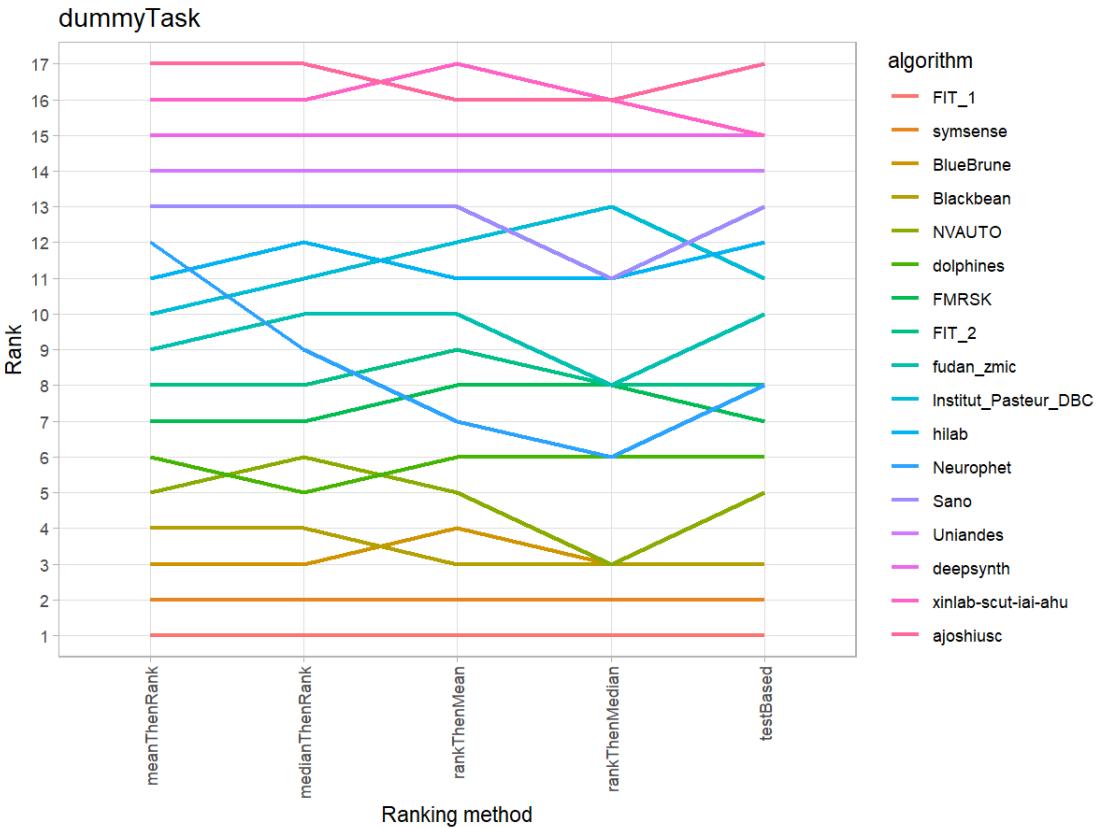
## *Significance maps* for visualizing ranking stability based on statistical significance

*Significance maps* depict incidence matrices of pairwise significant test results for the one-sided Wilcoxon signed rank test at a 5% significance level with adjustment for multiple testing according to Holm. Yellow shading indicates that metric values from the algorithm on the x-axis were significantly superior to those from the algorithm on the y-axis, blue color indicates no significant difference.



### Ranking robustness to ranking methods

*Line plots* for visualizing ranking robustness across different ranking methods. Each algorithm is represented by one colored line. For each ranking method encoded on the x-axis, the height of the line represents the corresponding rank. Horizontal lines indicate identical ranks for all methods.



### 25.3 References

Wiesenfarth, M., Reinke, A., Landman, B.A., Eisenmann, M., Aguilera Saiz, L., Cardoso, M.J., Maier-Hein, L. and Kopp-Schneider, A. Methods and open-source toolkit for analyzing and visualizing challenge results. *Sci Rep* **11**, 2369 (2021). <https://doi.org/10.1038/s41598-021-82017-6>

M. J. A. Eugster, T. Hothorn, and F. Leisch, “Exploratory and inferential analysis of benchmark experiments,” Institut fuer Statistik, Ludwig-Maximilians-Universitaet Muenchen, Germany, Technical Report 30, 2008. [Online]. Available: <http://epub.ub.uni-muenchen.de/4134/>.

## 26 Benchmarking report for Hausdorff Metrics – Pathological Brains

created by challengeR v1.0.2  
21 September, 2023

This document presents a systematic report on the benchmark study “Hausdorff Metrics”. Input data comprises raw metric values for all algorithms and cases. Generated plots are:

- Visualization of assessment data: Dot- and boxplot, podium plot and ranking heatmap
- Visualization of ranking stability: Blob plot, violin plot and significance map, line plot

Details can be found in Wiesenfarth et al. (2021).

### 26.1 Ranking

Algorithms within a task are ranked according to the following ranking scheme:  
*aggregate using function (“mean”) then rank*

The analysis is based on 17 algorithms and 644 cases. 0 missing cases have been found in the data set.

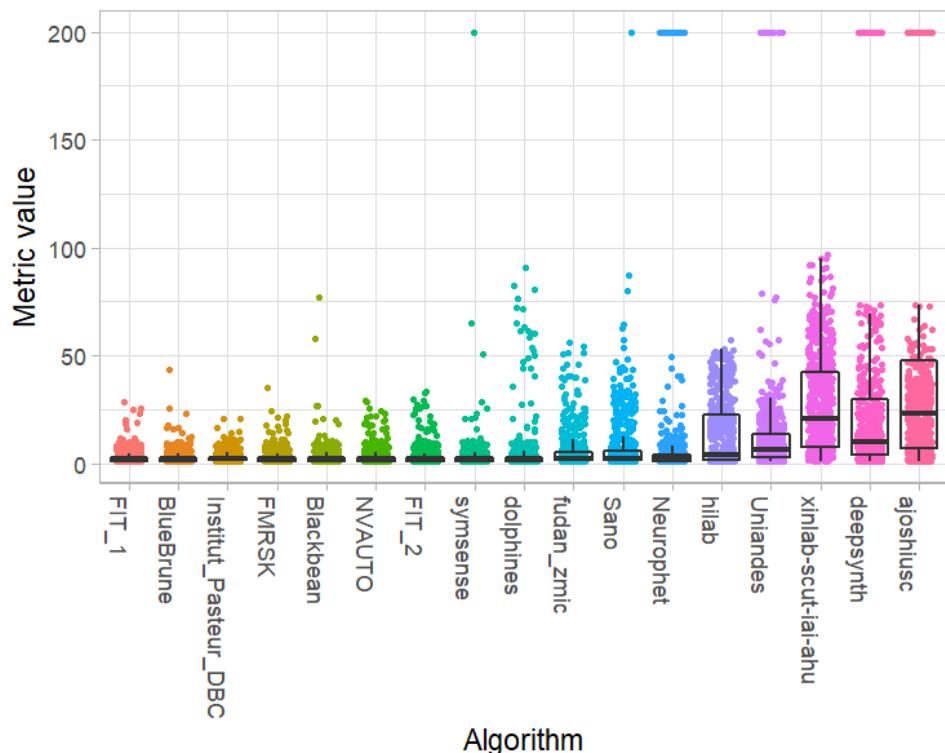
Ranking:

	Hausdorff_mean	rank
FIT_1	2.556429	1
BlueBrune	2.592852	2
Institut_Pasteur_DBC	2.620900	3
FMRSK	2.678659	4
Blackbean	2.813099	5
NVAUTO	2.956343	6
FIT_2	2.985061	7
symsense	3.079684	8
dolphines	4.597607	9
fudan_zmic	5.452280	10
Sano	7.085424	11
Neurophet	12.178524	12
hilab	13.341449	13
Uniandes	13.956941	14
xinlab-scut-iai-ahu	26.887995	15
deepsynth	35.647450	16
ajoshiusc	56.224501	17

## 26.2 Visualization of raw assessment data

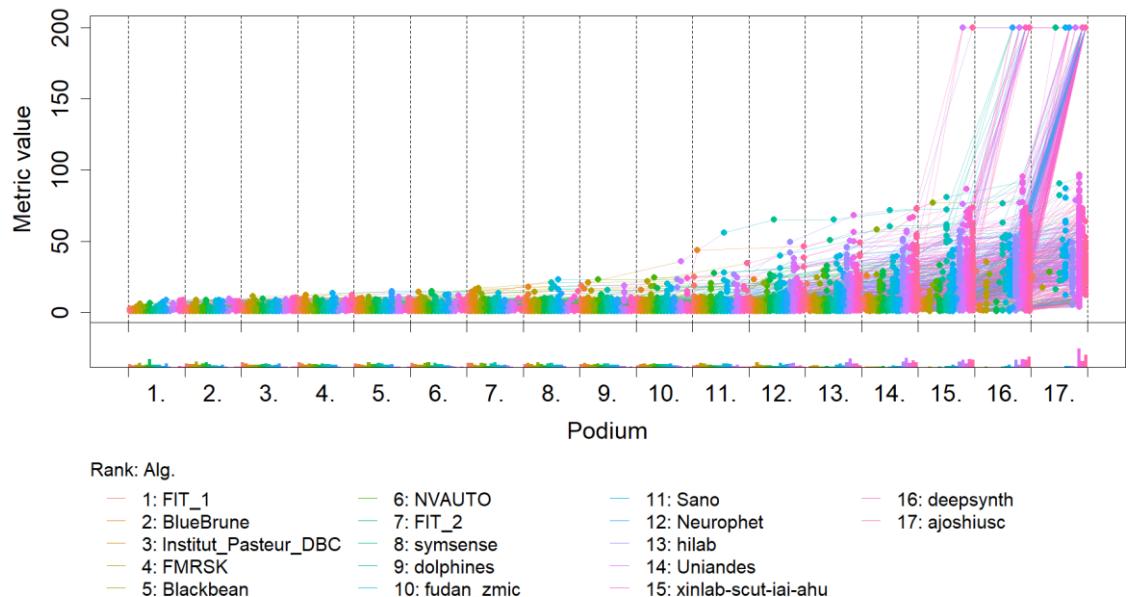
### Dot- and boxplot

*Dot- and boxplots* for visualizing raw assessment data separately for each algorithm. Boxplots representing descriptive statistics over all cases (median, quartiles and outliers) are combined with horizontally jittered dots representing individual cases.



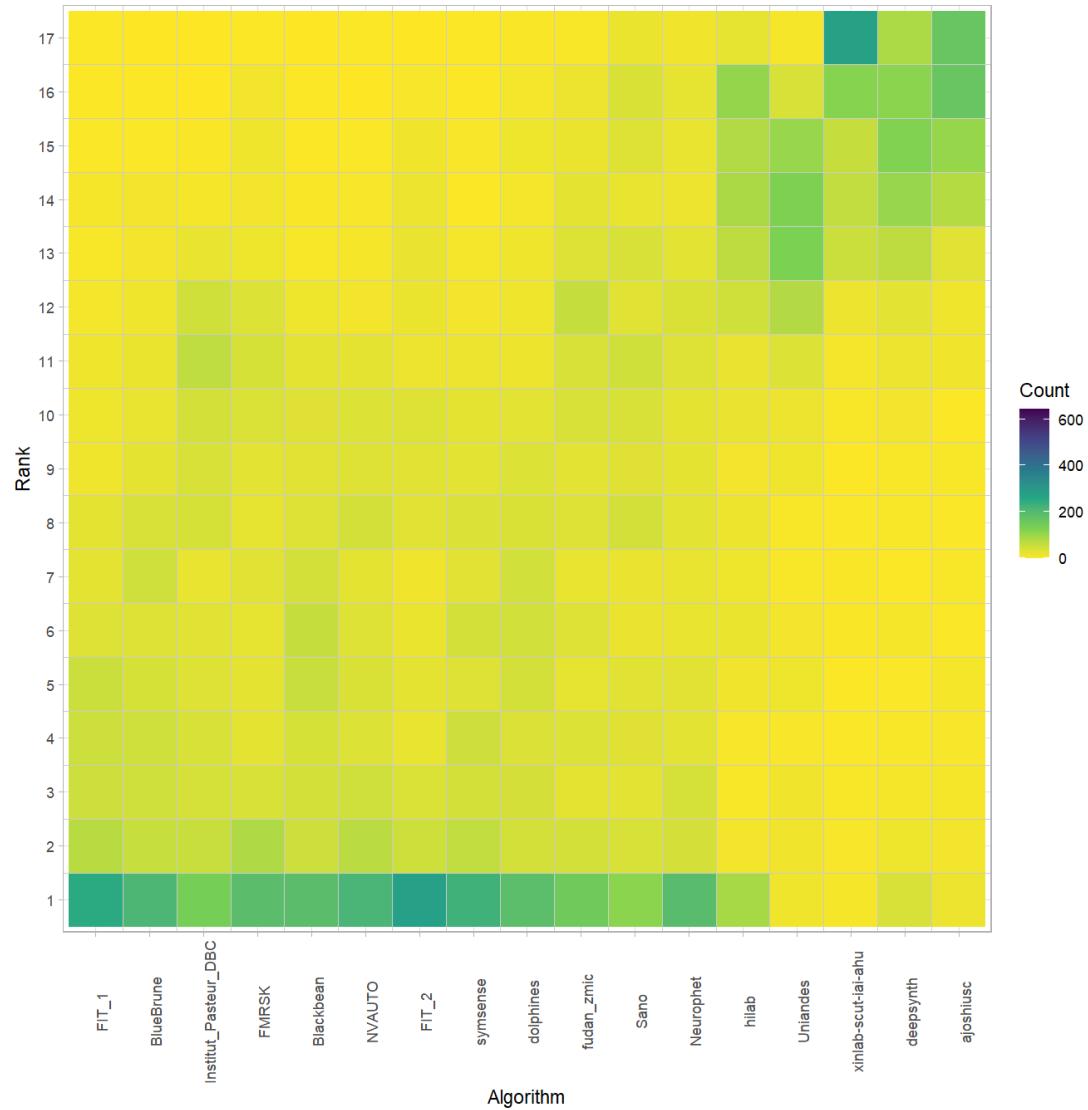
### Podium plot

*Podium plots* (see also Eugster et al., 2008) for visualizing raw assessment data. Upper part (spaghetti plot): Participating algorithms are color-coded, and each colored dot in the plot represents a metric value achieved with the respective algorithm. The actual metric value is encoded by the y-axis. Each podium (here:  $p=17$ ) represents one possible rank, ordered from best (1) to last (here: 17). The assignment of metric values (i.e. colored dots) to one of the podiums is based on the rank that the respective algorithm achieved on the corresponding case. Note that the plot part above each podium place is further subdivided into  $p$  “columns”, where each column represents one participating algorithm (here:  $p = 17$ ). Dots corresponding to identical cases are connected by a line, leading to the shown spaghetti structure. Lower part: Bar charts represent the relative frequency for each algorithm to achieve the rank encoded by the podium place.



## Ranking heatmap

*Ranking heatmaps* for visualizing raw assessment data. Each cell  $(i, A_j)$  shows the absolute frequency of cases in which algorithm  $A_j$  achieved rank  $i$ .

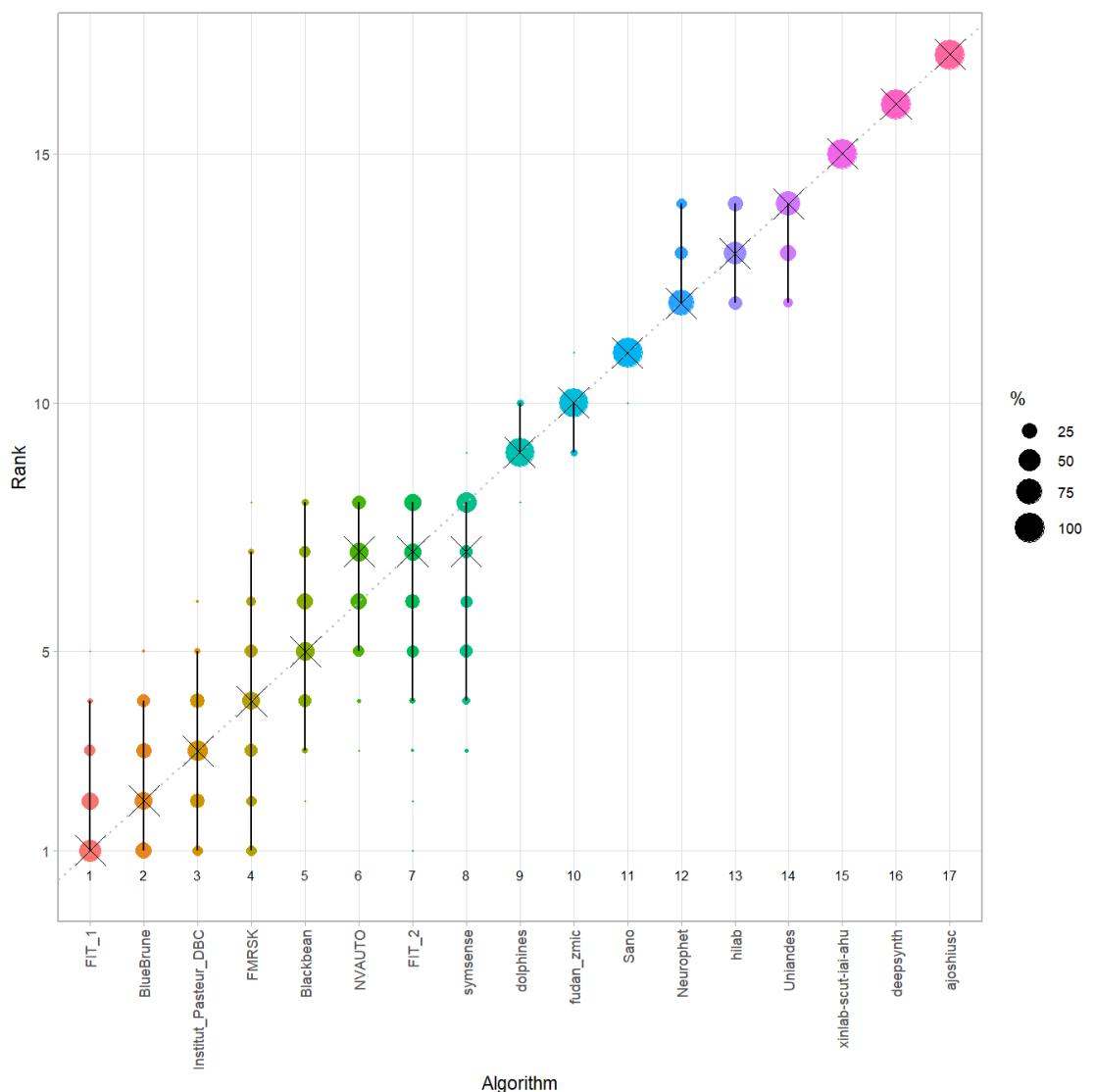


## Visualization of ranking stability

## **Blob plot for visualizing ranking stability based on bootstrap sampling**

Algorithms are color-coded, and the area of each blob at position ( $A_i$ , rank  $j$ ) is proportional to the relative frequency  $A_i$  achieved rank  $j$  across  $b = 1000$  bootstrap samples. The median rank for each algorithm is indicated by a black cross. 95% bootstrap intervals across bootstrap samples are indicated by black lines.

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> = ## "none")` instead.
```

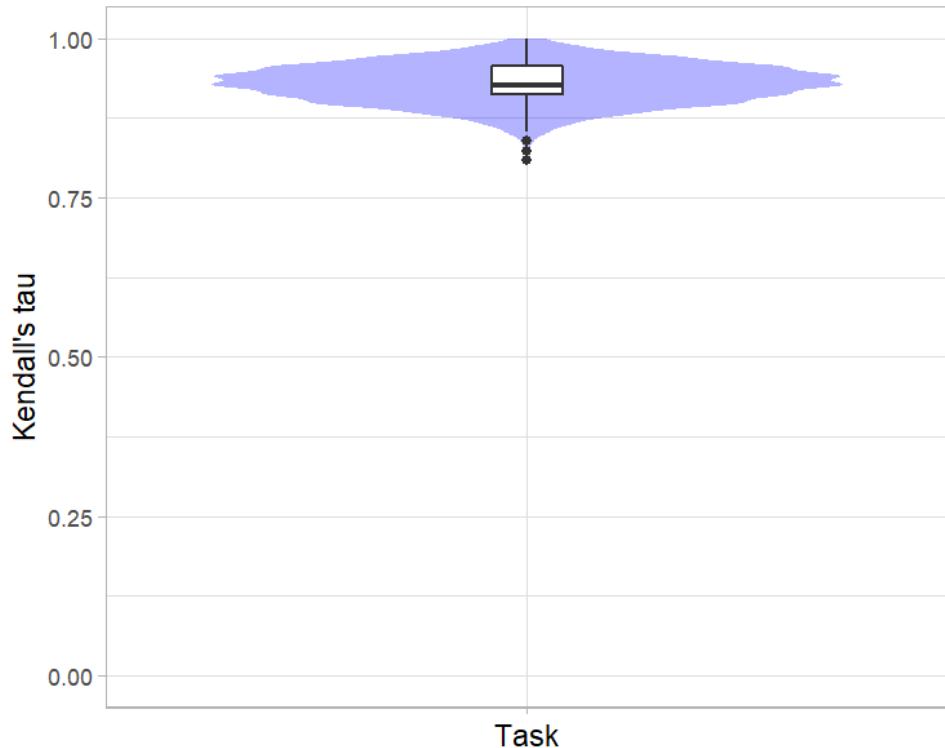


### ***Violin plot for visualizing ranking stability based on bootstrapping***

The ranking list based on the full assessment data is pairwise compared with the ranking lists based on the individual bootstrap samples (here  $b = 1000$  samples). For each pair of rankings, Kendall's  $\tau$  correlation is computed. Kendall's  $\tau$  is a scaled index determining the correlation between the lists. It is computed by evaluating the number of pairwise concordances and discordances between ranking lists and produces values between  $-1$  (for inverted order) and  $1$  (for identical order). A violin plot, which simultaneously depicts a boxplot and a density plot, is generated from the results.

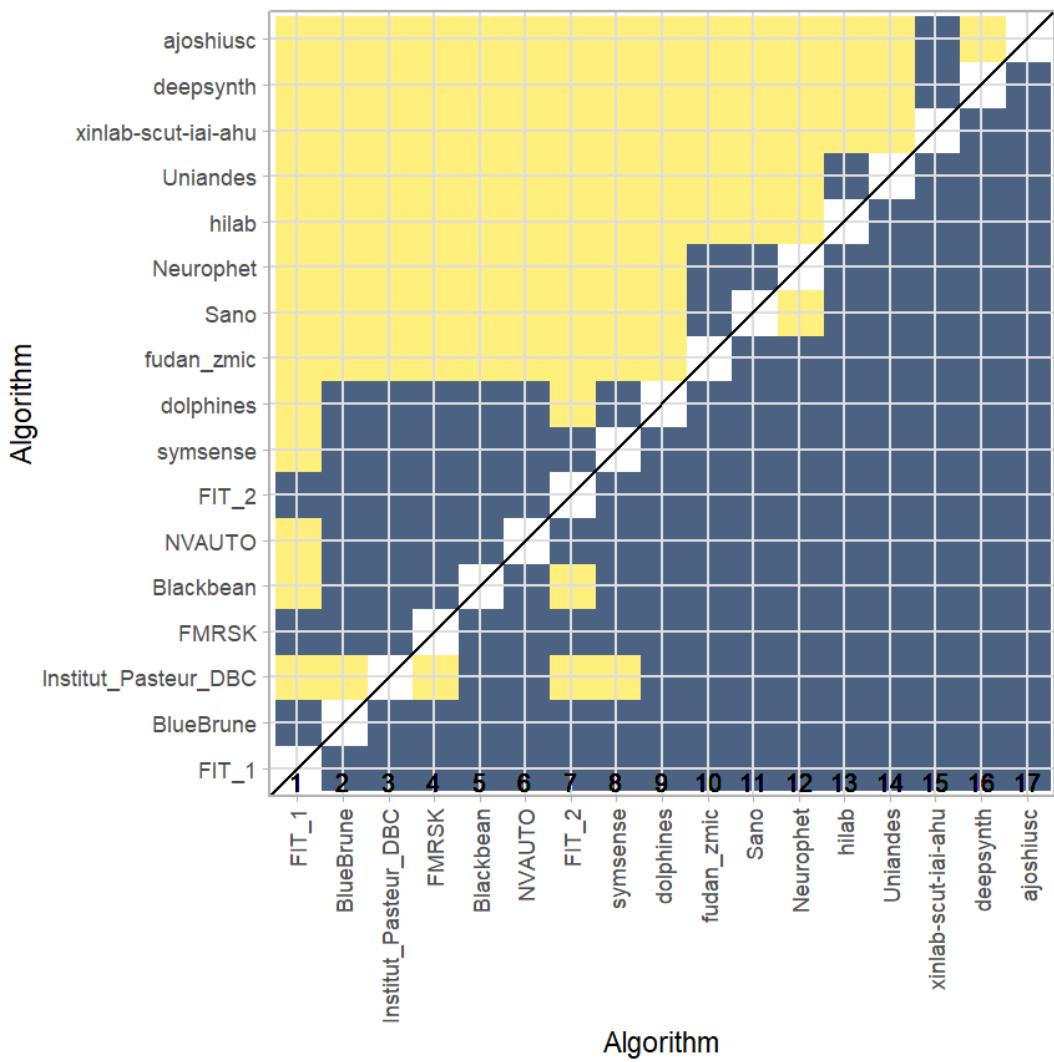
Summary Kendall's tau:

Task	mean	median	q25	q75
dummyTask	0.9283529	0.9264706	0.9117647	0.9558824



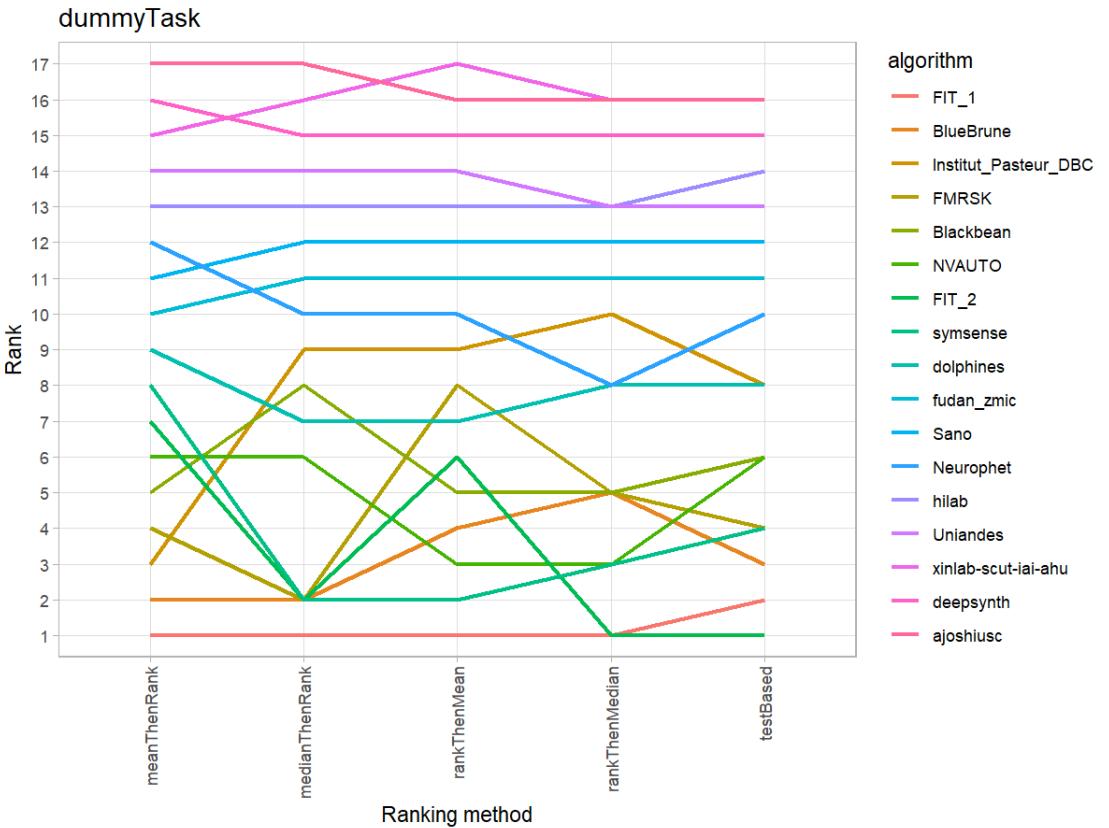
## *Significance maps* for visualizing ranking stability based on statistical significance

*Significance maps* depict incidence matrices of pairwise significant test results for the one-sided Wilcoxon signed rank test at a 5% significance level with adjustment for multiple testing according to Holm. Yellow shading indicates that metric values from the algorithm on the x-axis were significantly superior to those from the algorithm on the y-axis, blue color indicates no significant difference.



### Ranking robustness to ranking methods

*Line plots* for visualizing ranking robustness across different ranking methods. Each algorithm is represented by one colored line. For each ranking method encoded on the x-axis, the height of the line represents the corresponding rank. Horizontal lines indicate identical ranks for all methods.



### 26.3 References

Wiesenfarth, M., Reinke, A., Landman, B.A., Eisenmann, M., Aguilera Saiz, L., Cardoso, M.J., Maier-Hein, L. and Kopp-Schneider, A. Methods and open-source toolkit for analyzing and visualizing challenge results. *Sci Rep* **11**, 2369 (2021). <https://doi.org/10.1038/s41598-021-82017-6>

M. J. A. Eugster, T. Hothorn, and F. Leisch, “Exploratory and inferential analysis of benchmark experiments,” Institut fuer Statistik, Ludwig-Maximilians-Universitaet Muenchen, Germany, Technical Report 30, 2008. [Online]. Available: <http://epub.ub.uni-muenchen.de/4134/>.

## 27 Benchmarking report for Volume Similarity Metrics – Pathological Brains

created by challengeR v1.0.2  
21 September, 2023

This document presents a systematic report on the benchmark study “Volume Similarity Metrics”. Input data comprises raw metric values for all algorithms and cases. Generated plots are:

- Visualization of assessment data: Dot- and boxplot, podium plot and ranking heatmap
- Visualization of ranking stability: Blob plot, violin plot and significance map, line plot

Details can be found in Wiesenfarth et al. (2021).

### 27.1 Ranking

Algorithms within a task are ranked according to the following ranking scheme:  
*aggregate using function (“mean”) then rank*

The analysis is based on 17 algorithms and 644 cases. 0 missing cases have been found in the data set.

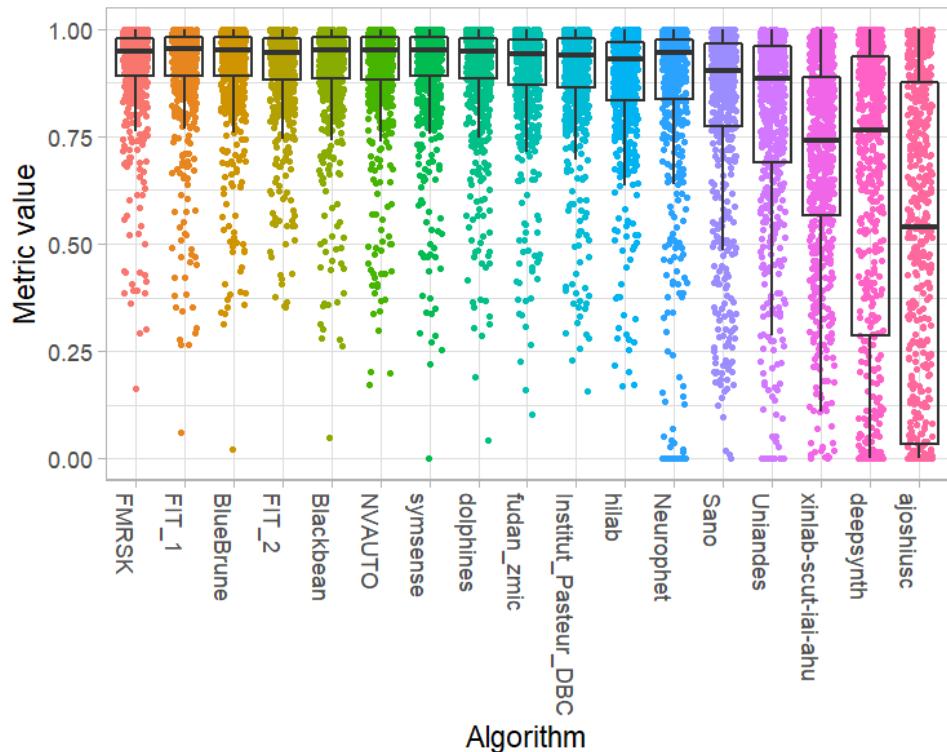
Ranking:

	Volume_Similarity_mean	rank
FMRSK	0.9099045	1
FIT_1	0.9068104	2
BlueBrune	0.9062367	3
FIT_2	0.9056555	4
Blackbean	0.9053491	5
NVAUTO	0.9042042	6
symsense	0.9038989	7
dolphines	0.9001010	8
fudan_zmic	0.8954260	9
Institut_Pasteur_DBC	0.8903602	10
hilab	0.8778469	11
Neurophet	0.8343312	12
Sano	0.8157597	13
Uniandes	0.7836403	14
xinlab-scut-iai-ahu	0.6925521	15
deepsynth	0.6150683	16
ajoshiusc	0.4902697	17

## 27.2 Visualization of raw assessment data

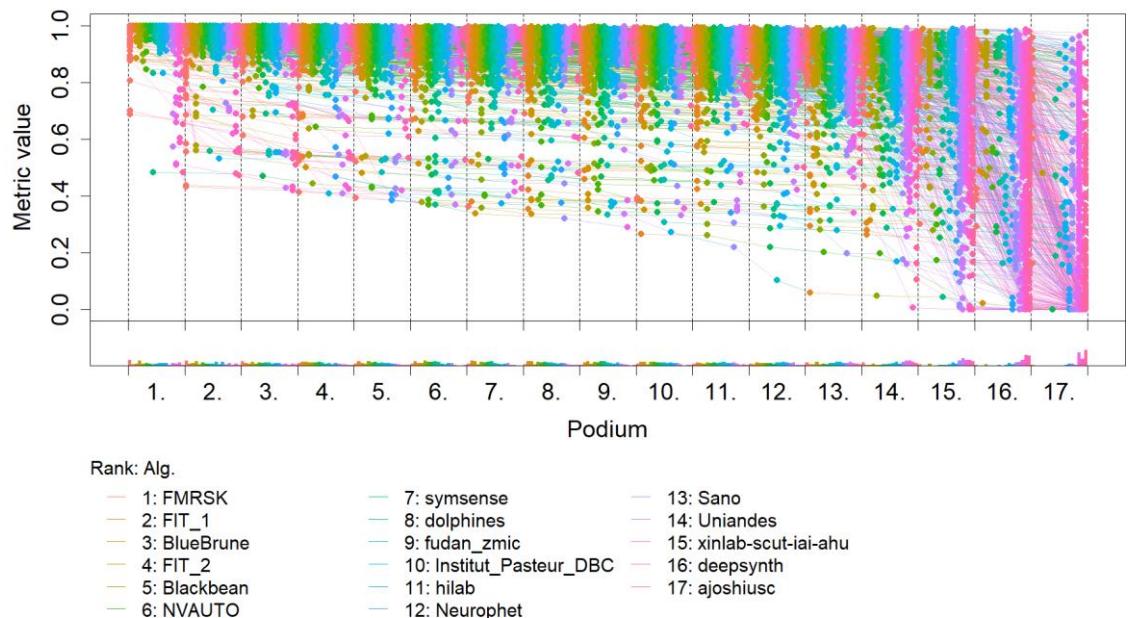
### Dot- and boxplot

*Dot- and boxplots* for visualizing raw assessment data separately for each algorithm. Boxplots representing descriptive statistics over all cases (median, quartiles and outliers) are combined with horizontally jittered dots representing individual cases.



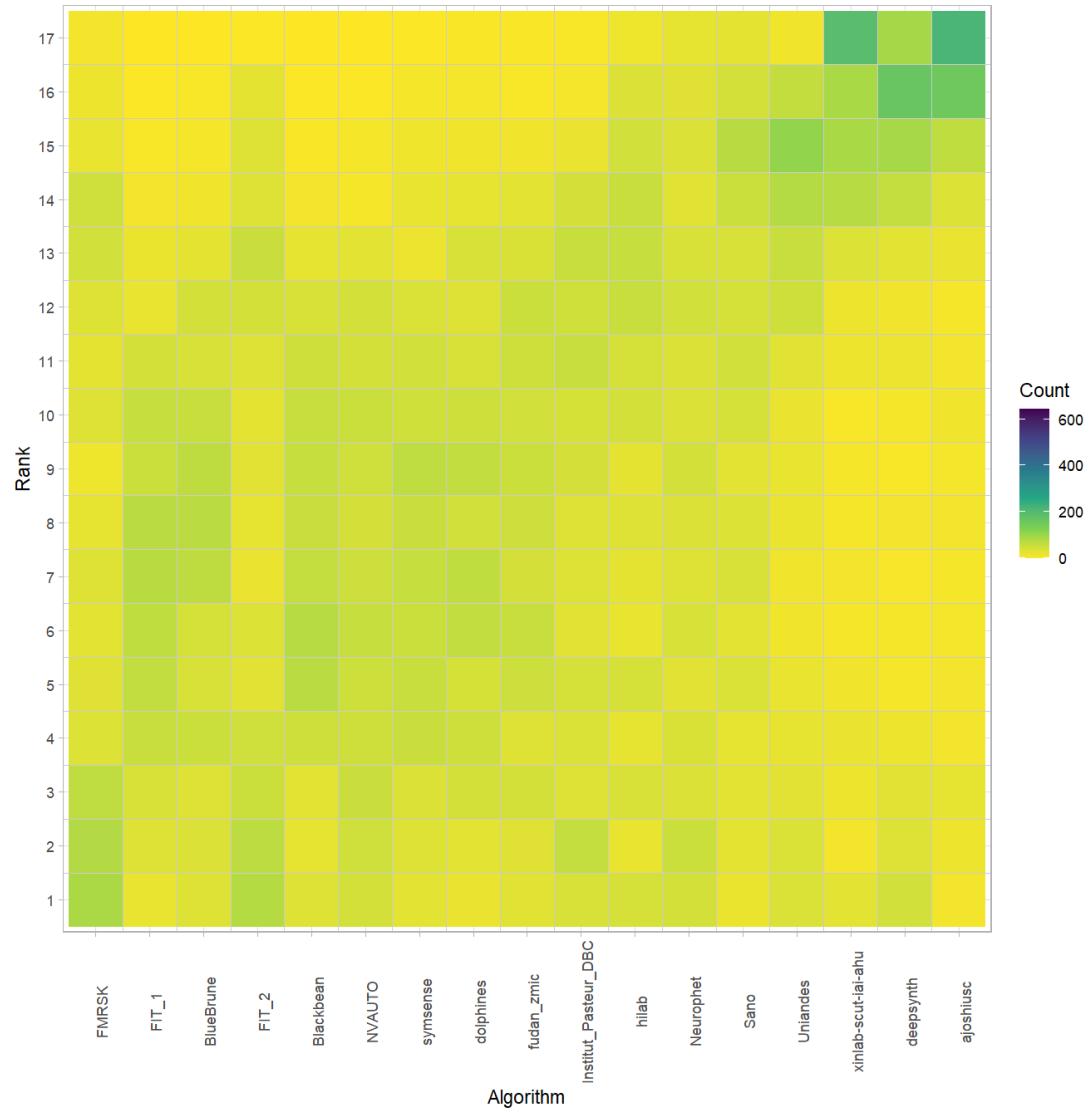
### Podium plot

*Podium plots* (see also Eugster et al., 2008) for visualizing raw assessment data. Upper part (spaghetti plot): Participating algorithms are color-coded, and each colored dot in the plot represents a metric value achieved with the respective algorithm. The actual metric value is encoded by the y-axis. Each podium (here:  $p=17$ ) represents one possible rank, ordered from best (1) to last (here: 17). The assignment of metric values (i.e. colored dots) to one of the podiums is based on the rank that the respective algorithm achieved on the corresponding case. Note that the plot part above each podium place is further subdivided into  $p$  “columns”, where each column represents one participating algorithm (here:  $p = 17$ ). Dots corresponding to identical cases are connected by a line, leading to the shown spaghetti structure. Lower part: Bar charts represent the relative frequency for each algorithm to achieve the rank encoded by the podium place.



## Ranking heatmap

*Ranking heatmaps* for visualizing raw assessment data. Each cell  $(i, A_j)$  shows the absolute frequency of cases in which algorithm  $A_j$  achieved rank  $i$ .

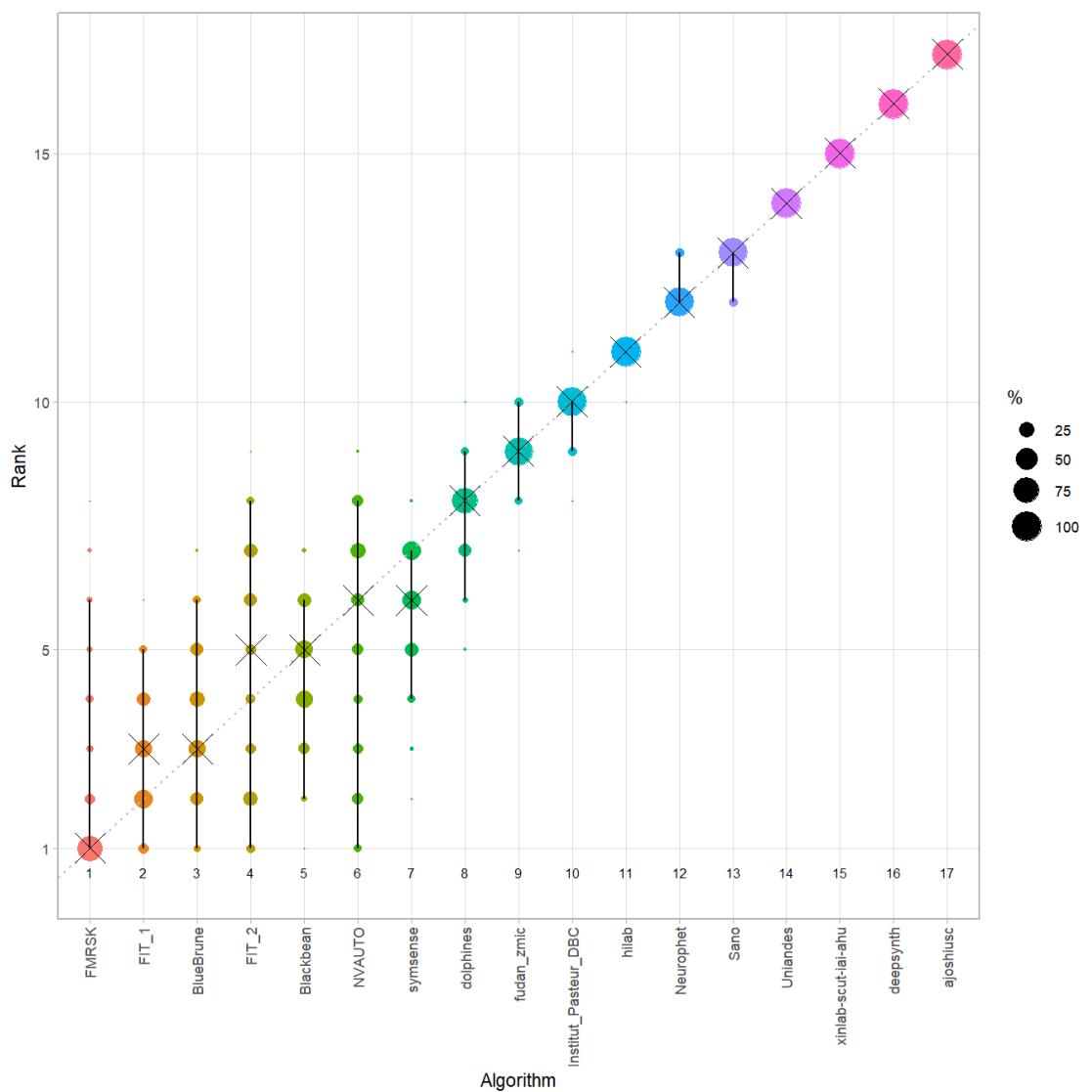


### Visualization of ranking stability

#### *Blob plot for visualizing ranking stability based on bootstrap sampling*

Algorithms are color-coded, and the area of each blob at position ( $A_i$ , rank  $j$ ) is proportional to the relative frequency  $A_i$  achieved rank  $j$  across  $b = 1000$  bootstrap samples. The median rank for each algorithm is indicated by a black cross. 95% bootstrap intervals across bootstrap samples are indicated by black lines.

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.
```

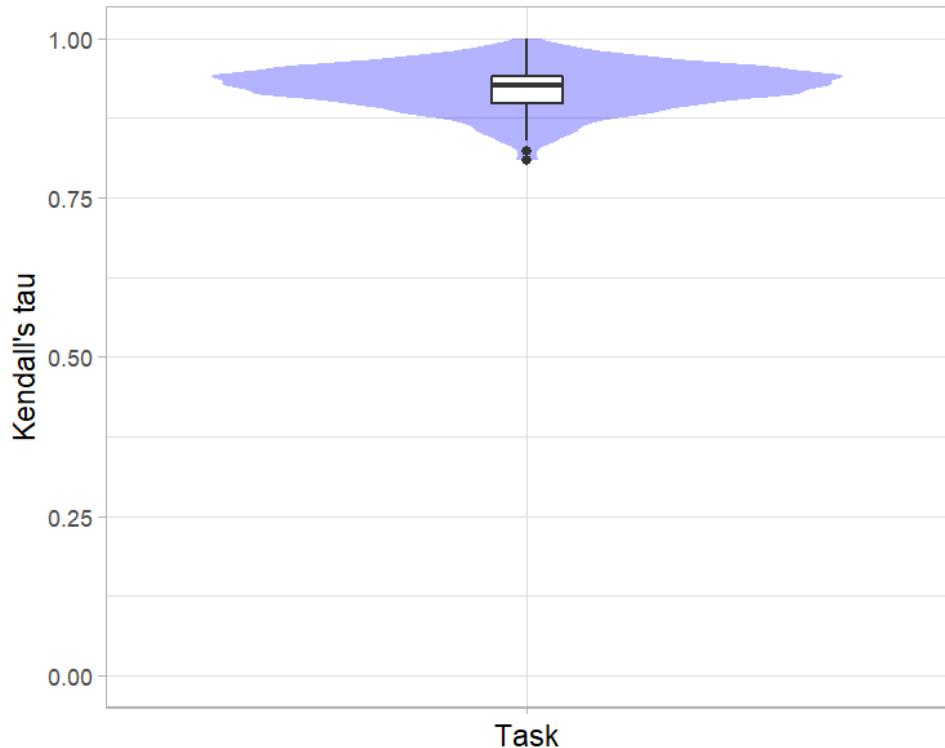


### ***Violin plot for visualizing ranking stability based on bootstrapping***

The ranking list based on the full assessment data is pairwise compared with the ranking lists based on the individual bootstrap samples (here  $b = 1000$  samples). For each pair of rankings, Kendall's  $\tau$  correlation is computed. Kendall's  $\tau$  is a scaled index determining the correlation between the lists. It is computed by evaluating the number of pairwise concordances and discordances between ranking lists and produces values between  $-1$  (for inverted order) and  $1$  (for identical order). A violin plot, which simultaneously depicts a boxplot and a density plot, is generated from the results.

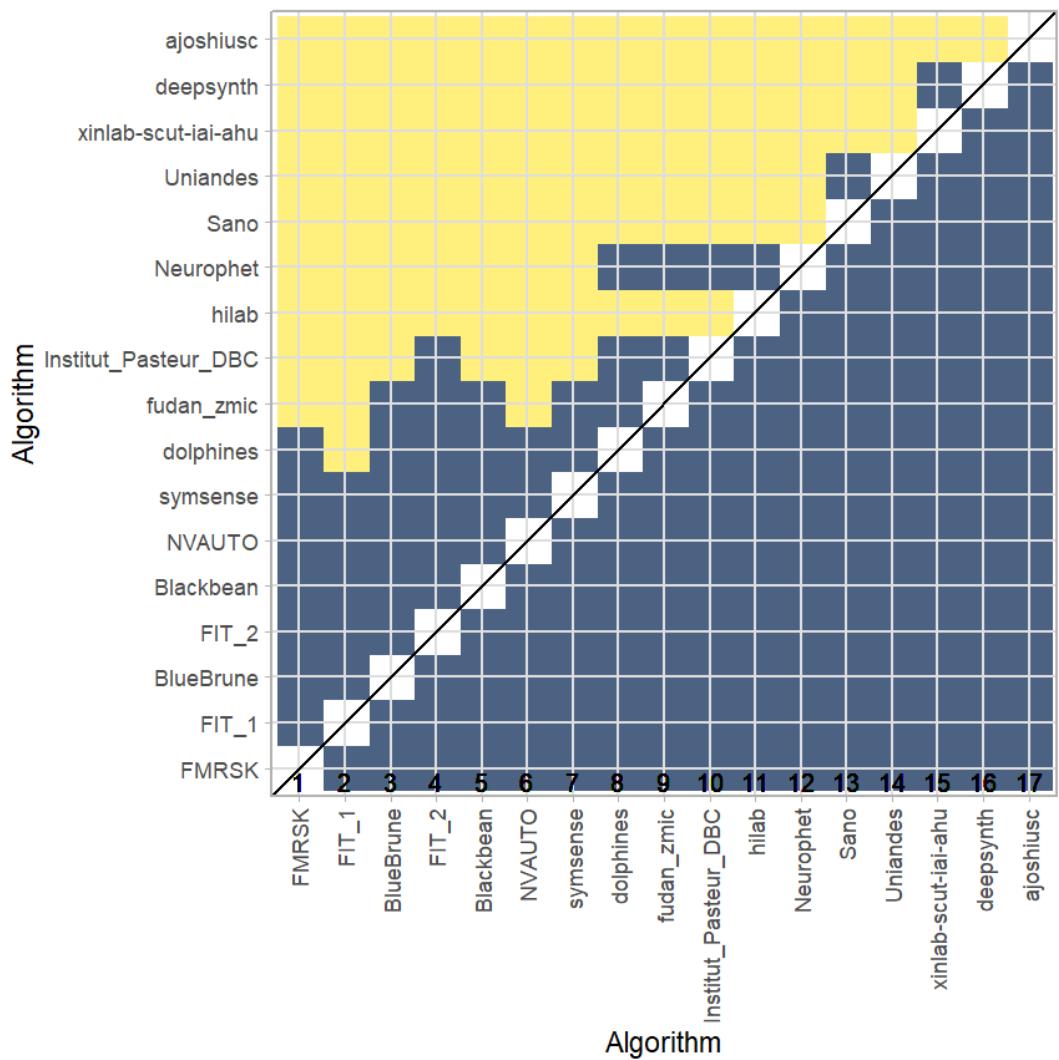
Summary Kendall's tau:

Task	mean	median	q25	q75
dummyTask	0.9232206	0.9264706	0.8970588	0.9411765



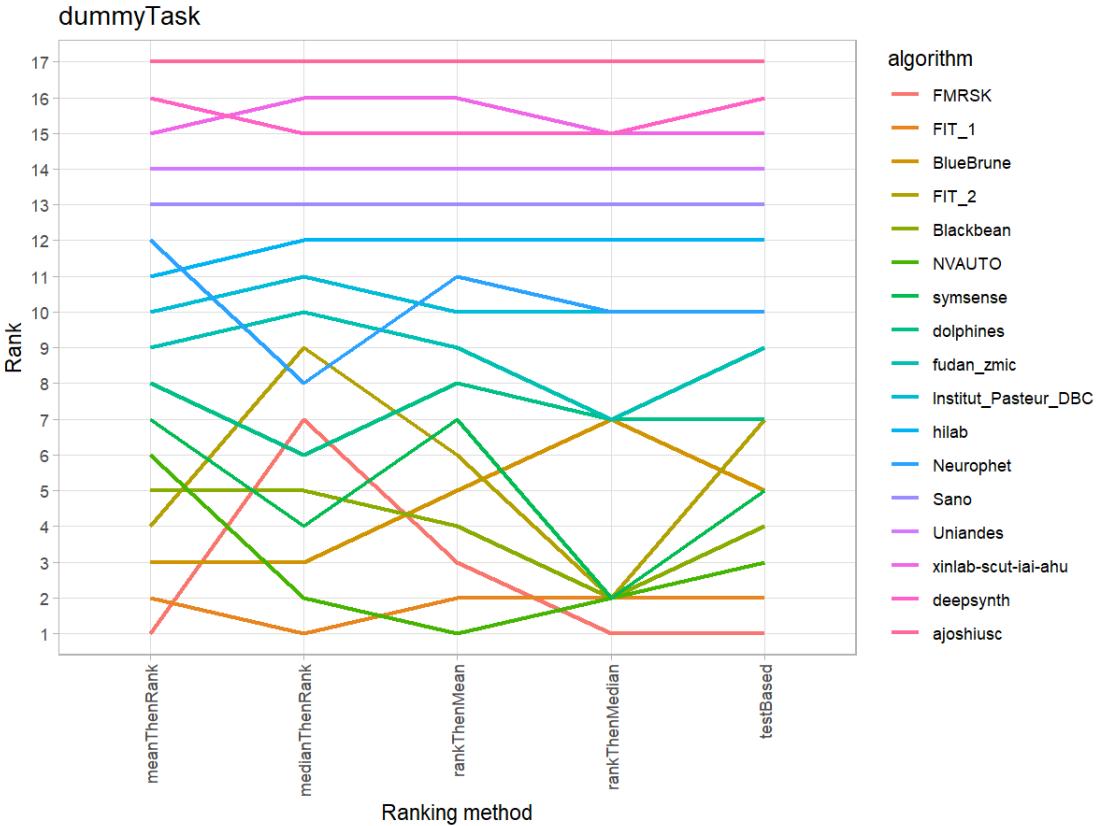
**Significance maps for visualizing ranking stability based on statistical significance**

*Significance maps* depict incidence matrices of pairwise significant test results for the one-sided Wilcoxon signed rank test at a 5% significance level with adjustment for multiple testing according to Holm. Yellow shading indicates that metric values from the algorithm on the x-axis were significantly superior to those from the algorithm on the y-axis, blue color indicates no significant difference.



### Ranking robustness to ranking methods

*Line plots* for visualizing ranking robustness across different ranking methods. Each algorithm is represented by one colored line. For each ranking method encoded on the x-axis, the height of the line represents the corresponding rank. Horizontal lines indicate identical ranks for all methods.



### 27.3 References

Wiesenfarth, M., Reinke, A., Landman, B.A., Eisenmann, M., Aguilera Saiz, L., Cardoso, M.J., Maier-Hein, L. and Kopp-Schneider, A. Methods and open-source toolkit for analyzing and visualizing challenge results. *Sci Rep* **11**, 2369 (2021). <https://doi.org/10.1038/s41598-021-82017-6>

M. J. A. Eugster, T. Hothorn, and F. Leisch, “Exploratory and inferential analysis of benchmark experiments,” Institut fuer Statistik, Ludwig-Maximilians-Universitaet Muenchen, Germany, Technical Report 30, 2008. [Online]. Available: <http://epub.ub.uni-muenchen.de/4134/>.

## 28 Benchmarking report for Dice Metrics – irtkSimple Reconstruction Method

created by challengeR v1.0.2  
23 October, 2023

This document presents a systematic report on the benchmark study “Dice Metrics”. Input data comprises raw metric values for all algorithms and cases. Generated plots are:

- Visualization of assessment data: Dot- and boxplot, podium plot and ranking heatmap
- Visualization of ranking stability: Blob plot, violin plot and significance map, line plot

Details can be found in Wiesenfarth et al. (2021).

### 28.1 Ranking

Algorithms within a task are ranked according to the following ranking scheme:  
*aggregate using function (“mean”) then rank*

The analysis is based on 17 algorithms and 140 cases. 0 missing cases have been found in the data set.

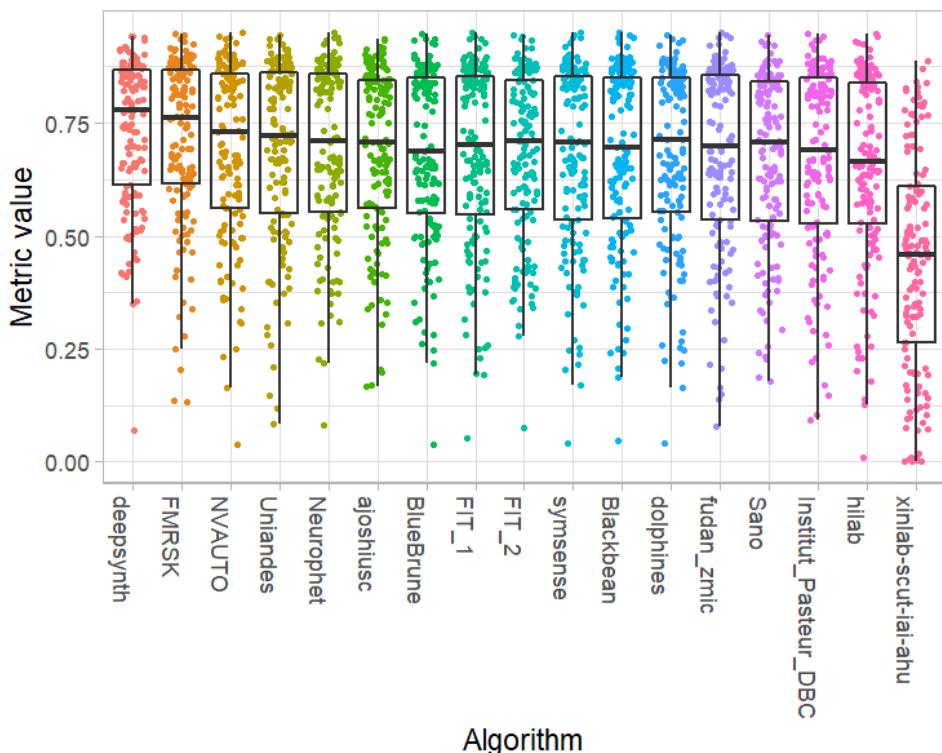
Ranking:

	Dice_m ean	r ank
deepsynth	0.7327846	1
FMRSK	0.7136193	2
NVAUTO	0.6934984	3
Uniandes	0.6926523	4
Neurophet	0.6890230	5
ajoshiusc	0.6856601	6
BlueBrune	0.6812812	7
FIT_1	0.6810933	8
FIT_2	0.6800074	9
symsense	0.6791905	10
Blackbean	0.6789500	11
dolphines	0.6787659	12
fudan_zmic	0.6771299	13
Sano	0.6712588	14
Institut_Pasteur_DBC	0.6687211	15
hilab	0.6537900	16
xinlab-scut-iai-ahu	0.4443192	17

## 28.2 Visualization of raw assessment data

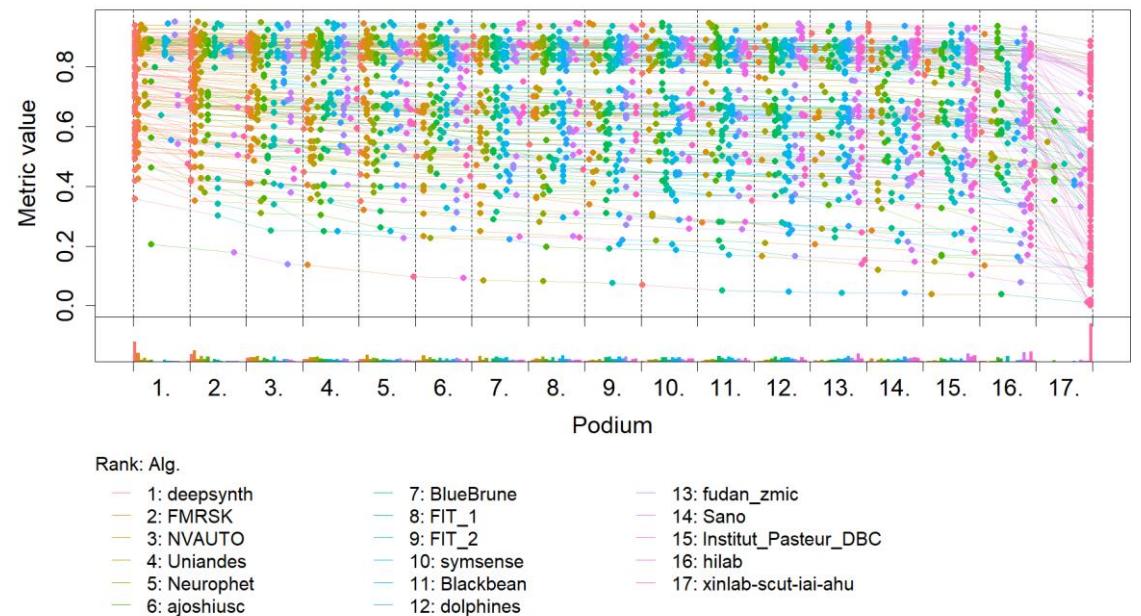
### Dot- and boxplot

*Dot- and boxplots* for visualizing raw assessment data separately for each algorithm. Boxplots representing descriptive statistics over all cases (median, quartiles and outliers) are combined with horizontally jittered dots representing individual cases.



### Podium plot

*Podium plots* (see also Eugster et al., 2008) for visualizing raw assessment data. Upper part (spaghetti plot): Participating algorithms are color-coded, and each colored dot in the plot represents a metric value achieved with the respective algorithm. The actual metric value is encoded by the y-axis. Each podium (here:  $p=17$ ) represents one possible rank, ordered from best (1) to last (here: 17). The assignment of metric values (i.e. colored dots) to one of the podiums is based on the rank that the respective algorithm achieved on the corresponding case. Note that the plot part above each podium place is further subdivided into  $p$  “columns”, where each column represents one participating algorithm (here:  $p = 17$ ). Dots corresponding to identical cases are connected by a line, leading to the shown spaghetti structure. Lower part: Bar charts represent the relative frequency for each algorithm to achieve the rank encoded by the podium place.



## Ranking heatmap

*Ranking heatmaps* for visualizing raw assessment data. Each cell  $(i, A_j)$  shows the absolute frequency of cases in which algorithm  $A_j$  achieved rank  $i$ .

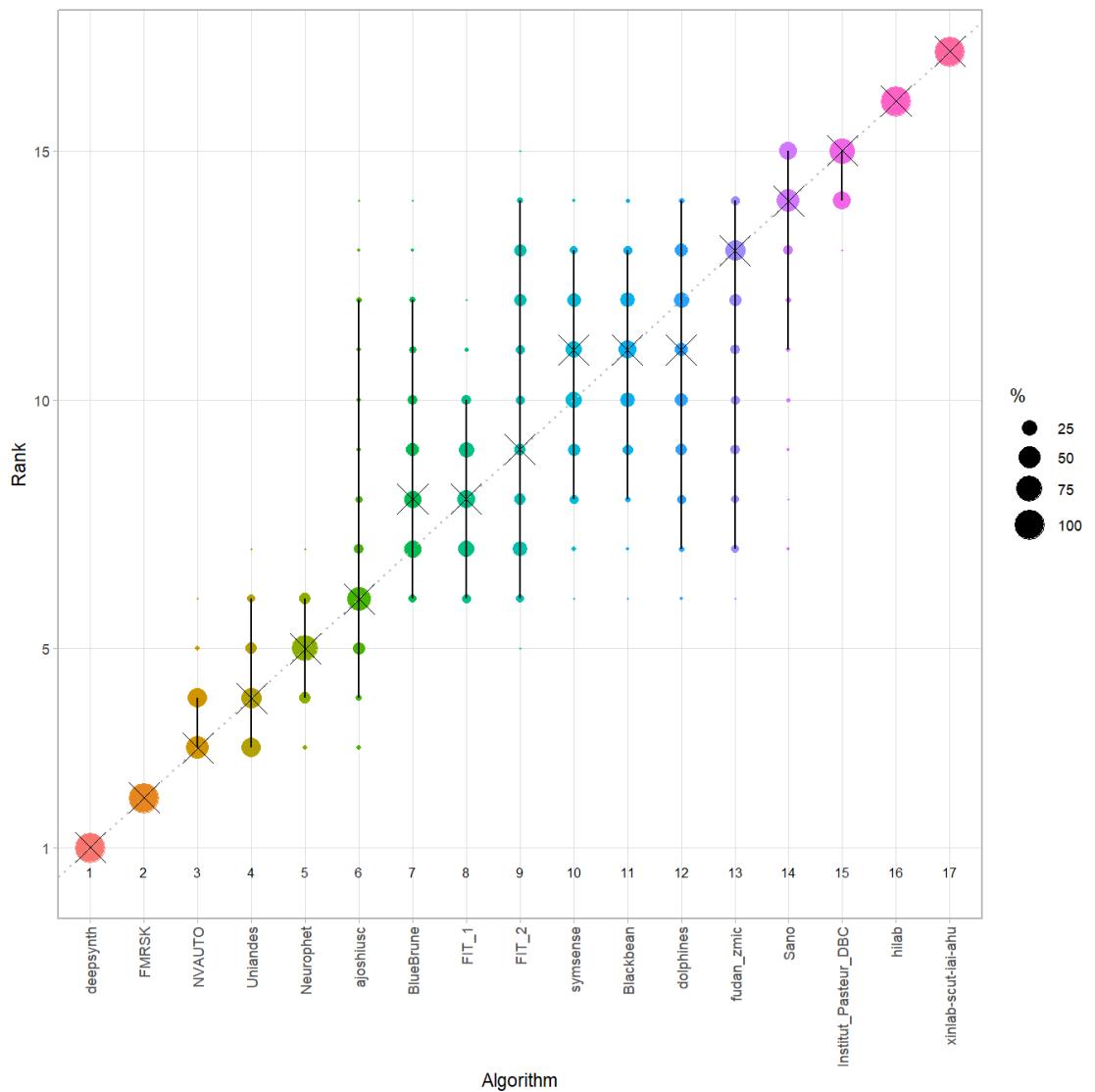


### Visualization of ranking stability

#### *Blob plot for visualizing ranking stability based on bootstrap sampling*

Algorithms are color-coded, and the area of each blob at position ( $A_i, \text{rank } j$ ) is proportional to the relative frequency  $A_i$  achieved rank  $j$  across  $b = 1000$  bootstrap samples. The median rank for each algorithm is indicated by a black cross. 95% bootstrap intervals across bootstrap samples are indicated by black lines.

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.
```

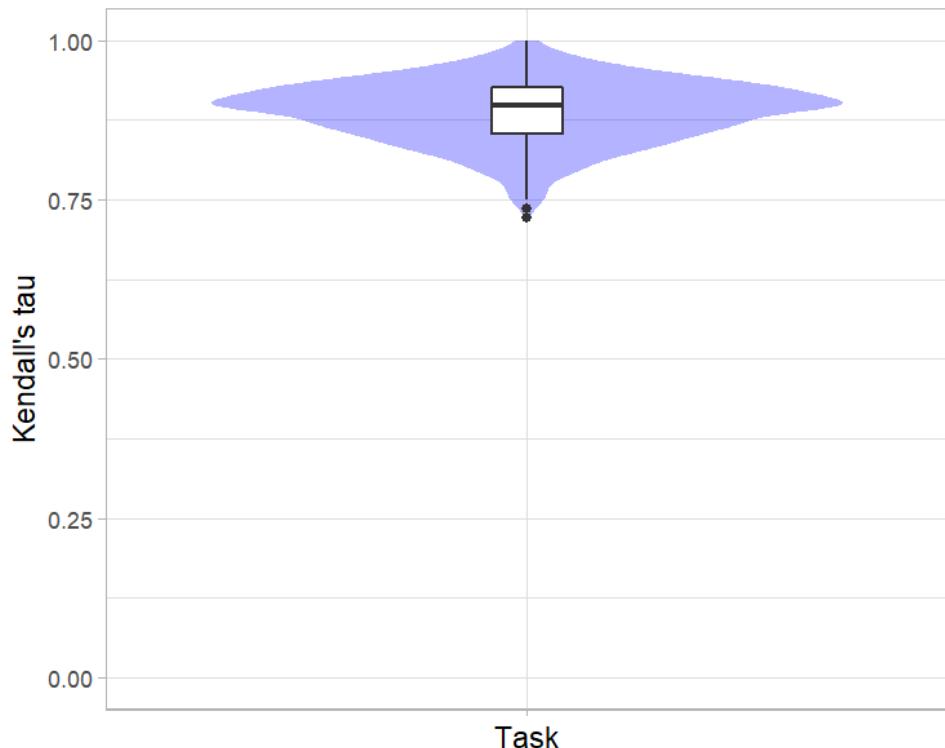


### ***Violin plot for visualizing ranking stability based on bootstrapping***

The ranking list based on the full assessment data is pairwise compared with the ranking lists based on the individual bootstrap samples (here  $b = 1000$  samples). For each pair of rankings, Kendall's  $\tau$  correlation is computed. Kendall's  $\tau$  is a scaled index determining the correlation between the lists. It is computed by evaluating the number of pairwise concordances and discordances between ranking lists and produces values between  $-1$  (for inverted order) and  $1$  (for identical order). A violin plot, which simultaneously depicts a boxplot and a density plot, is generated from the results.

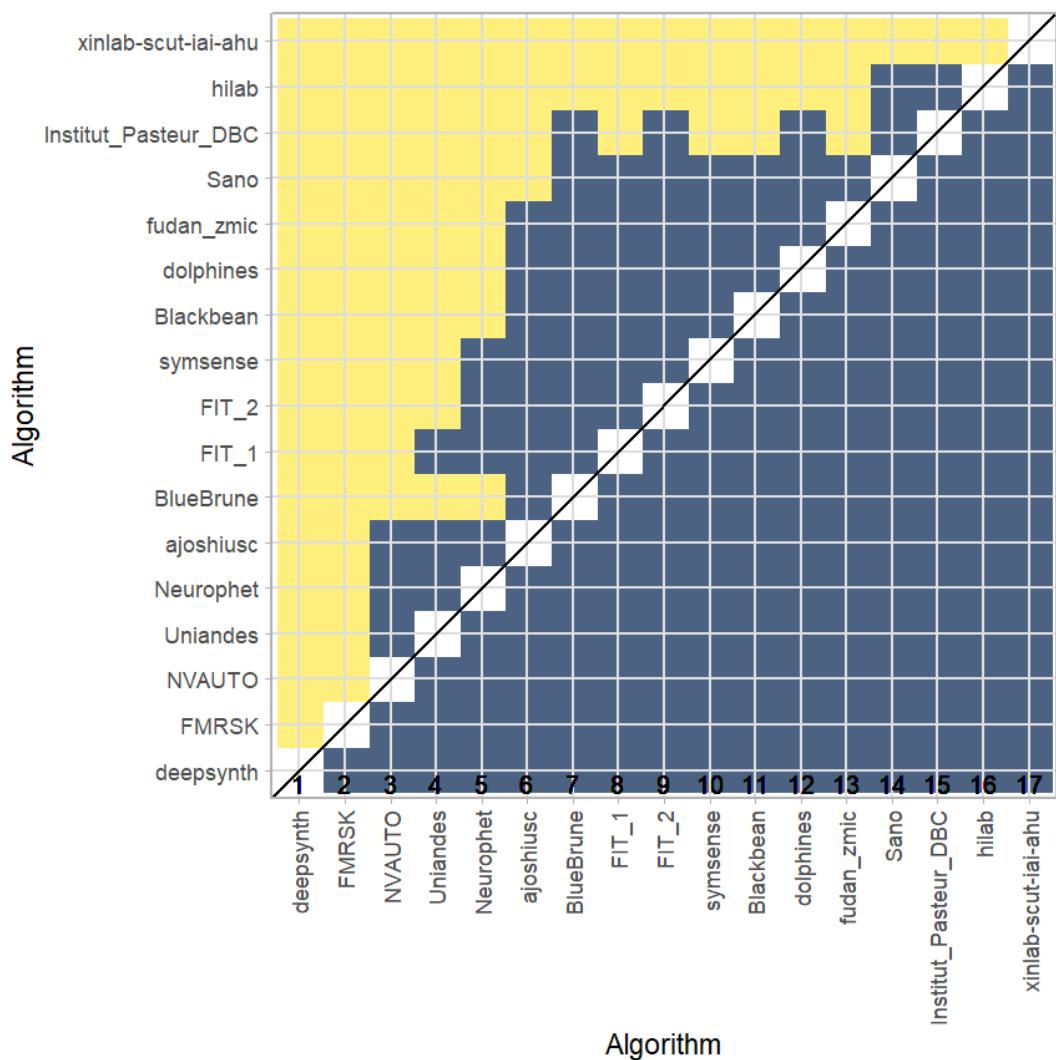
Summary Kendall's tau:

Task	mean	median	q25	q75
dummyTask	0.8881618	0.8970588	0.8529412	0.9264706



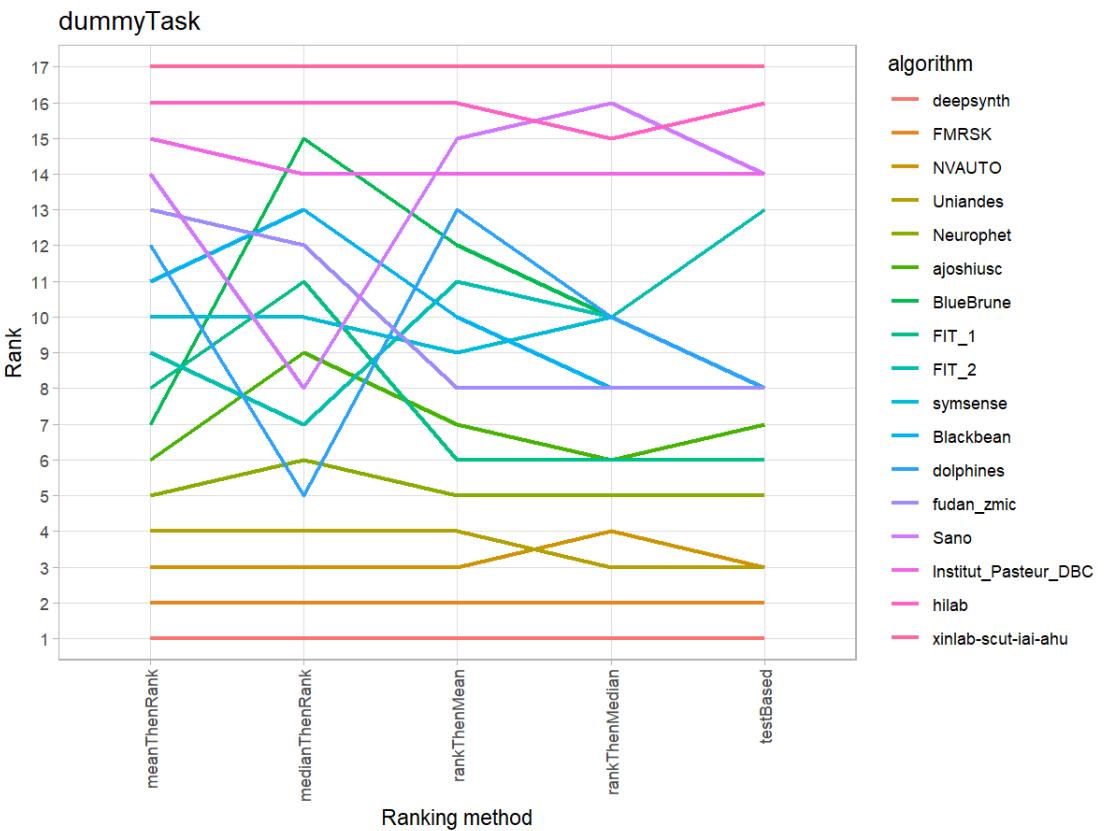
## *Significance maps* for visualizing ranking stability based on statistical significance

*Significance maps* depict incidence matrices of pairwise significant test results for the one-sided Wilcoxon signed rank test at a 5% significance level with adjustment for multiple testing according to Holm. Yellow shading indicates that metric values from the algorithm on the x-axis were significantly superior to those from the algorithm on the y-axis, blue color indicates no significant difference.



### Ranking robustness to ranking methods

*Line plots* for visualizing ranking robustness across different ranking methods. Each algorithm is represented by one colored line. For each ranking method encoded on the x-axis, the height of the line represents the corresponding rank. Horizontal lines indicate identical ranks for all methods.



### 28.3 References

Wiesenfarth, M., Reinke, A., Landman, B.A., Eisenmann, M., Aguilera Saiz, L., Cardoso, M.J., Maier-Hein, L. and Kopp-Schneider, A. Methods and open-source toolkit for analyzing and visualizing challenge results. *Sci Rep* **11**, 2369 (2021). <https://doi.org/10.1038/s41598-021-82017-6>

M. J. A. Eugster, T. Hothorn, and F. Leisch, “Exploratory and inferential analysis of benchmark experiments,” Institut fuer Statistik, Ludwig-Maximilians-Universitaet Muenchen, Germany, Technical Report 30, 2008. [Online]. Available: <http://epub.ub.uni-muenchen.de/4134/>.

## 29 Benchmarking report for Hausdorff Metrics – `irtkSimple Reconstruction Method`

created by challengeR v1.0.2  
23 October, 2023

This document presents a systematic report on the benchmark study “Hausdorff Metrics”. Input data comprises raw metric values for all algorithms and cases. Generated plots are:

- Visualization of assessment data: Dot- and boxplot, podium plot and ranking heatmap
- Visualization of ranking stability: Blob plot, violin plot and significance map, line plot

Details can be found in Wiesenfarth et al. (2021).

### 29.1 Ranking

Algorithms within a task are ranked according to the following ranking scheme:  
*aggregate using function (“mean”) then rank*

The analysis is based on 17 algorithms and 140 cases. 0 missing cases have been found in the data set.

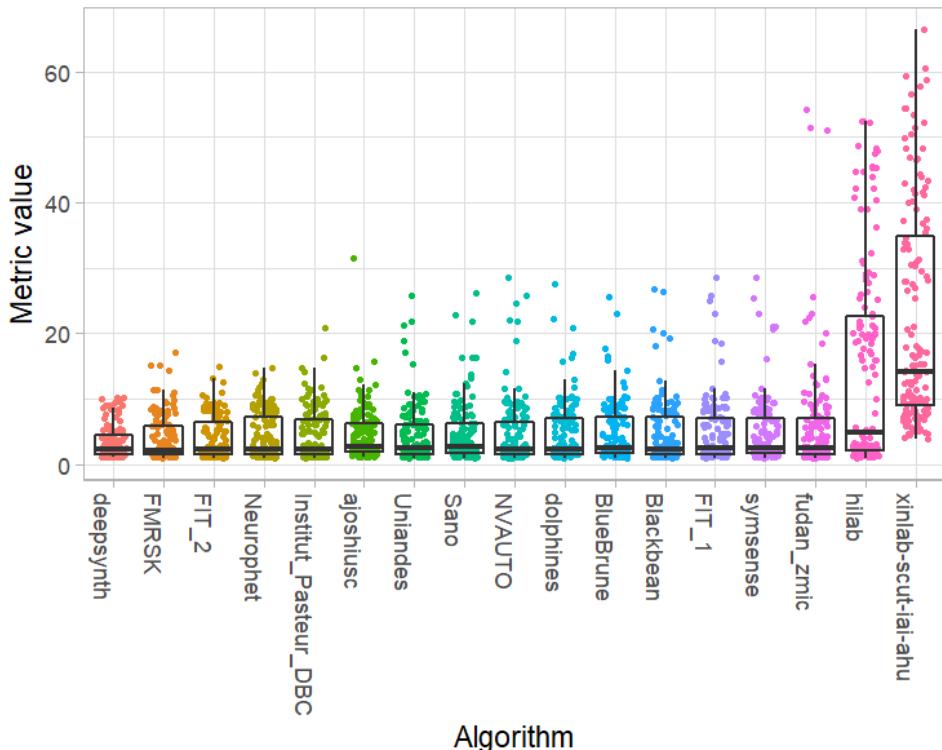
Ranking:

	Hausdorff_mean	rank
deepsynth	3.367647	1
FMRSK	3.828919	2
FIT_2	3.911252	3
Neurophet	4.189914	4
Institut_Pasteur_DBC	4.335186	5
ajoshiusc	4.362200	6
Uniandes	4.442928	7
Sano	4.544371	8
NVAUTO	4.554302	9
dolphines	4.564495	10
BlueBrune	4.618616	11
Blackbean	4.670160	12
FIT_1	4.763669	13
symsense	4.775997	14
fudan_zmic	5.842642	15
hilab	14.574006	16
xinlab-scut-iai-ahu	22.087781	17

## 29.2 Visualization of raw assessment data

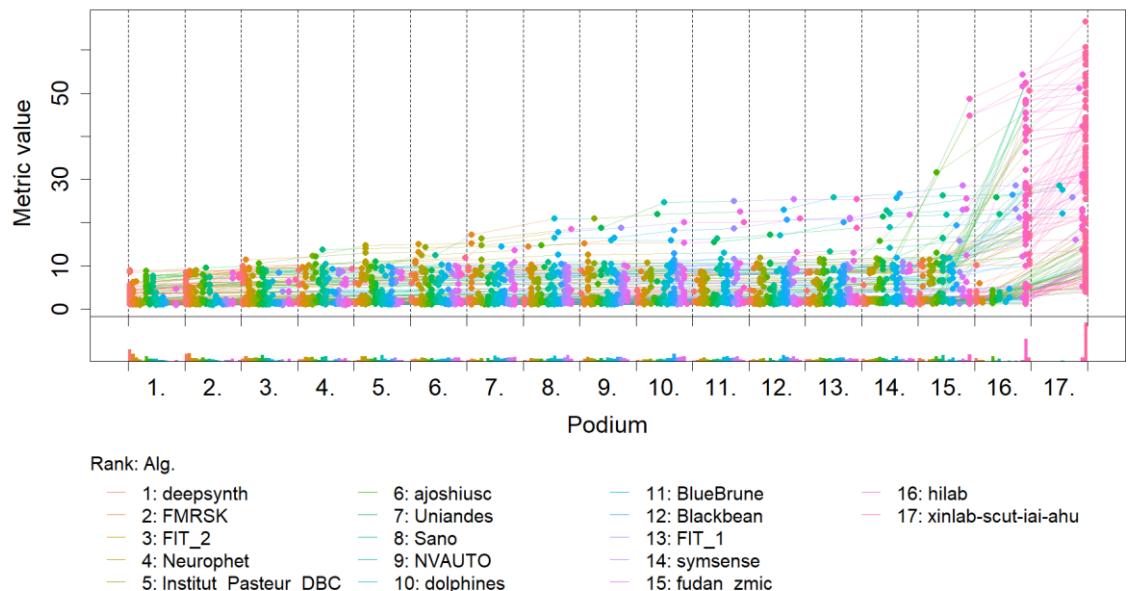
### Dot- and boxplot

*Dot- and boxplots* for visualizing raw assessment data separately for each algorithm. Boxplots representing descriptive statistics over all cases (median, quartiles and outliers) are combined with horizontally jittered dots representing individual cases.



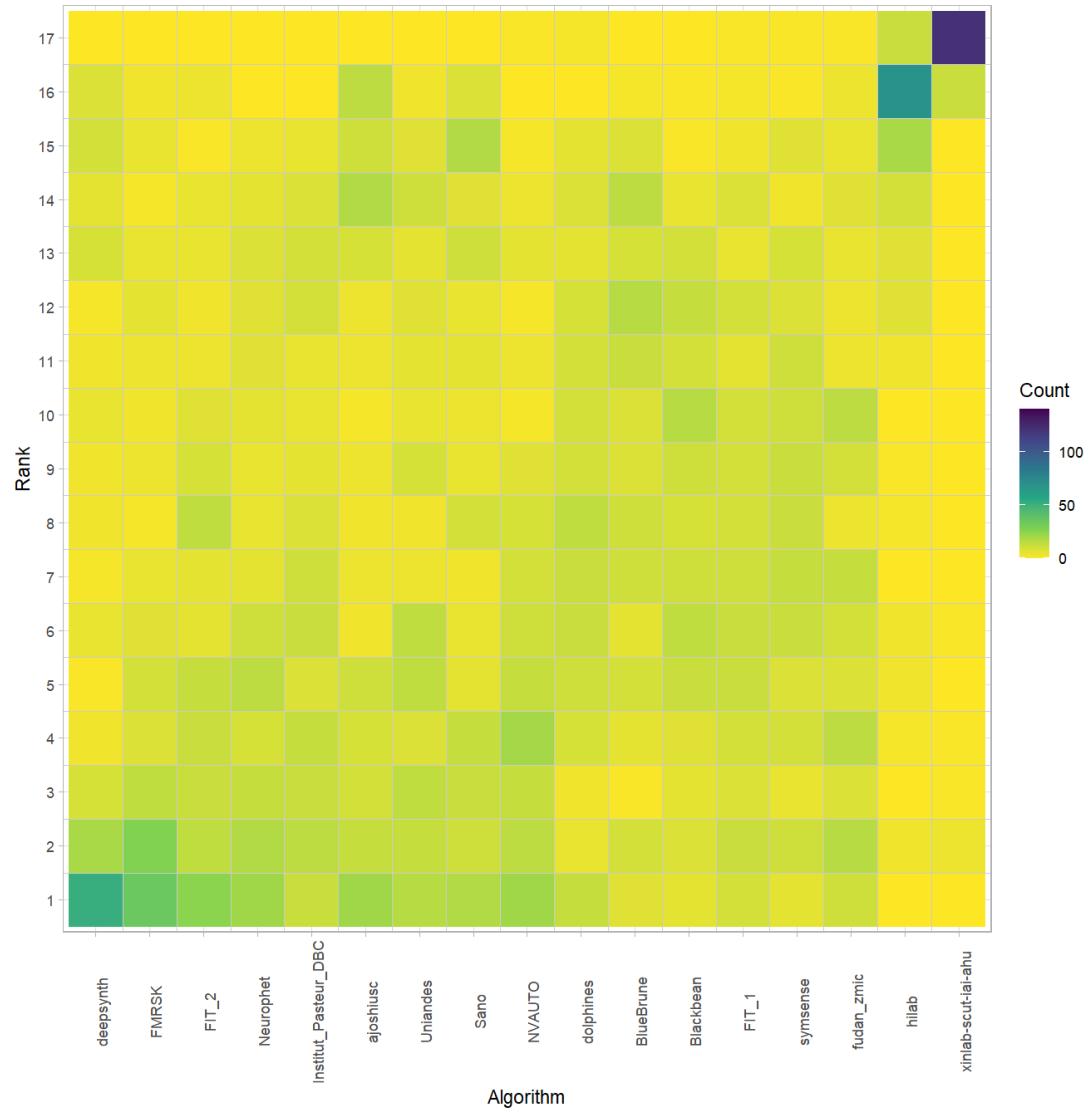
### Podium plot

*Podium plots* (see also Eugster et al., 2008) for visualizing raw assessment data. Upper part (spaghetti plot): Participating algorithms are color-coded, and each colored dot in the plot represents a metric value achieved with the respective algorithm. The actual metric value is encoded by the y-axis. Each podium (here:  $p=17$ ) represents one possible rank, ordered from best (1) to last (here: 17). The assignment of metric values (i.e. colored dots) to one of the podiums is based on the rank that the respective algorithm achieved on the corresponding case. Note that the plot part above each podium place is further subdivided into  $p$  “columns”, where each column represents one participating algorithm (here:  $p = 17$ ). Dots corresponding to identical cases are connected by a line, leading to the shown spaghetti structure. Lower part: Bar charts represent the relative frequency for each algorithm to achieve the rank encoded by the podium place.



## Ranking heatmap

*Ranking heatmaps* for visualizing raw assessment data. Each cell  $(i, A_j)$  shows the absolute frequency of cases in which algorithm  $A_j$  achieved rank  $i$ .

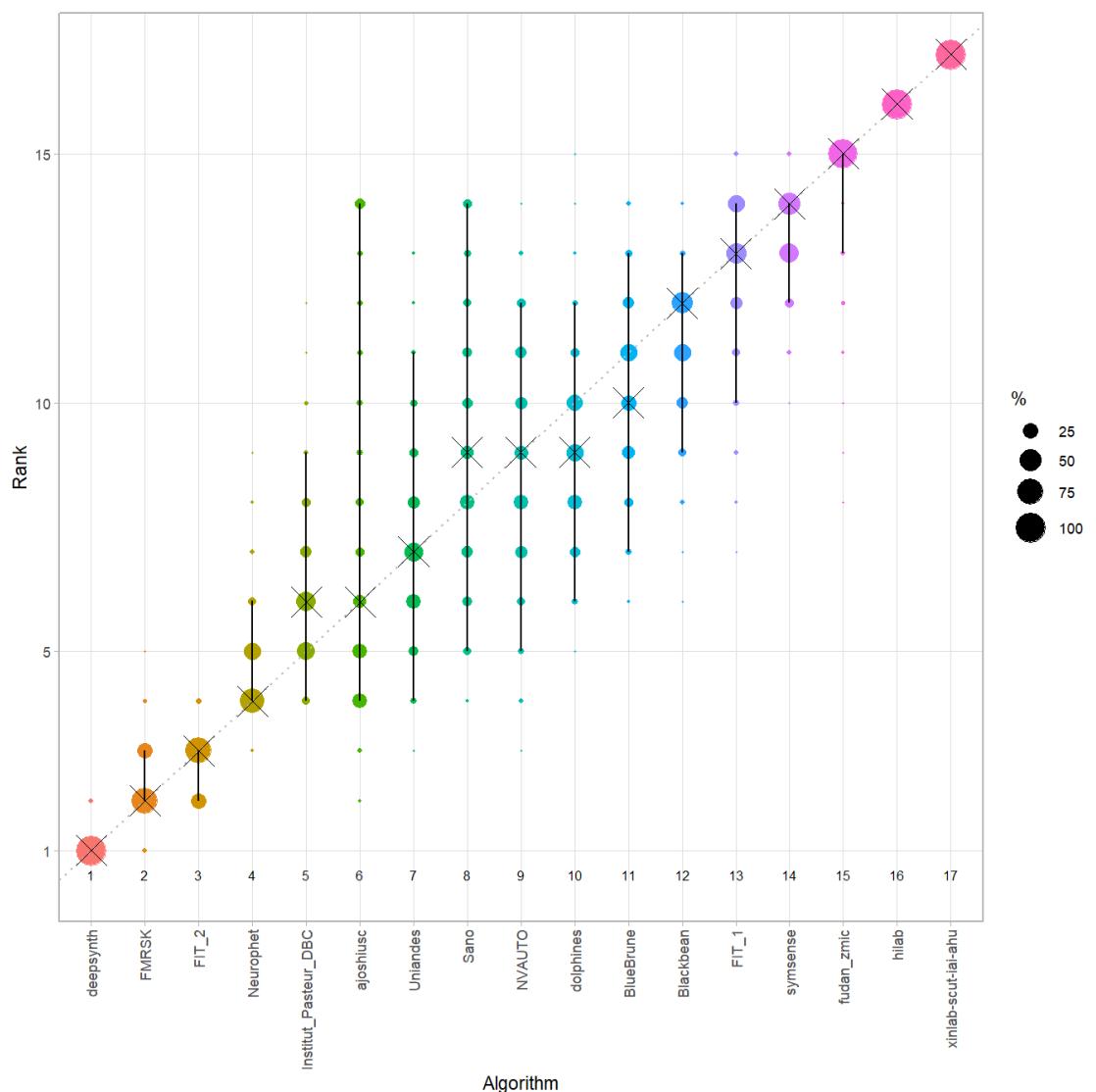


### Visualization of ranking stability

#### *Blob plot for visualizing ranking stability based on bootstrap sampling*

Algorithms are color-coded, and the area of each blob at position ( $A_i$ , rank  $j$ ) is proportional to the relative frequency  $A_i$  achieved rank  $j$  across  $b = 1000$  bootstrap samples. The median rank for each algorithm is indicated by a black cross. 95% bootstrap intervals across bootstrap samples are indicated by black lines.

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.
```

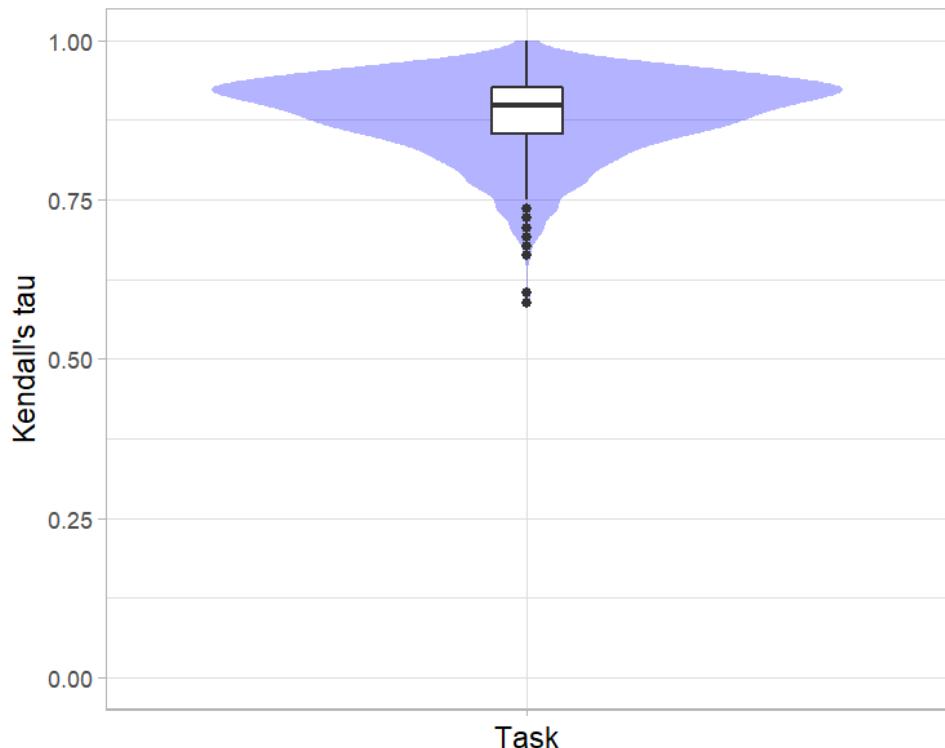


### ***Violin plot for visualizing ranking stability based on bootstrapping***

The ranking list based on the full assessment data is pairwise compared with the ranking lists based on the individual bootstrap samples (here  $b = 1000$  samples). For each pair of rankings, Kendall's  $\tau$  correlation is computed. Kendall's  $\tau$  is a scaled index determining the correlation between the lists. It is computed by evaluating the number of pairwise concordances and discordances between ranking lists and produces values between  $-1$  (for inverted order) and  $1$  (for identical order). A violin plot, which simultaneously depicts a boxplot and a density plot, is generated from the results.

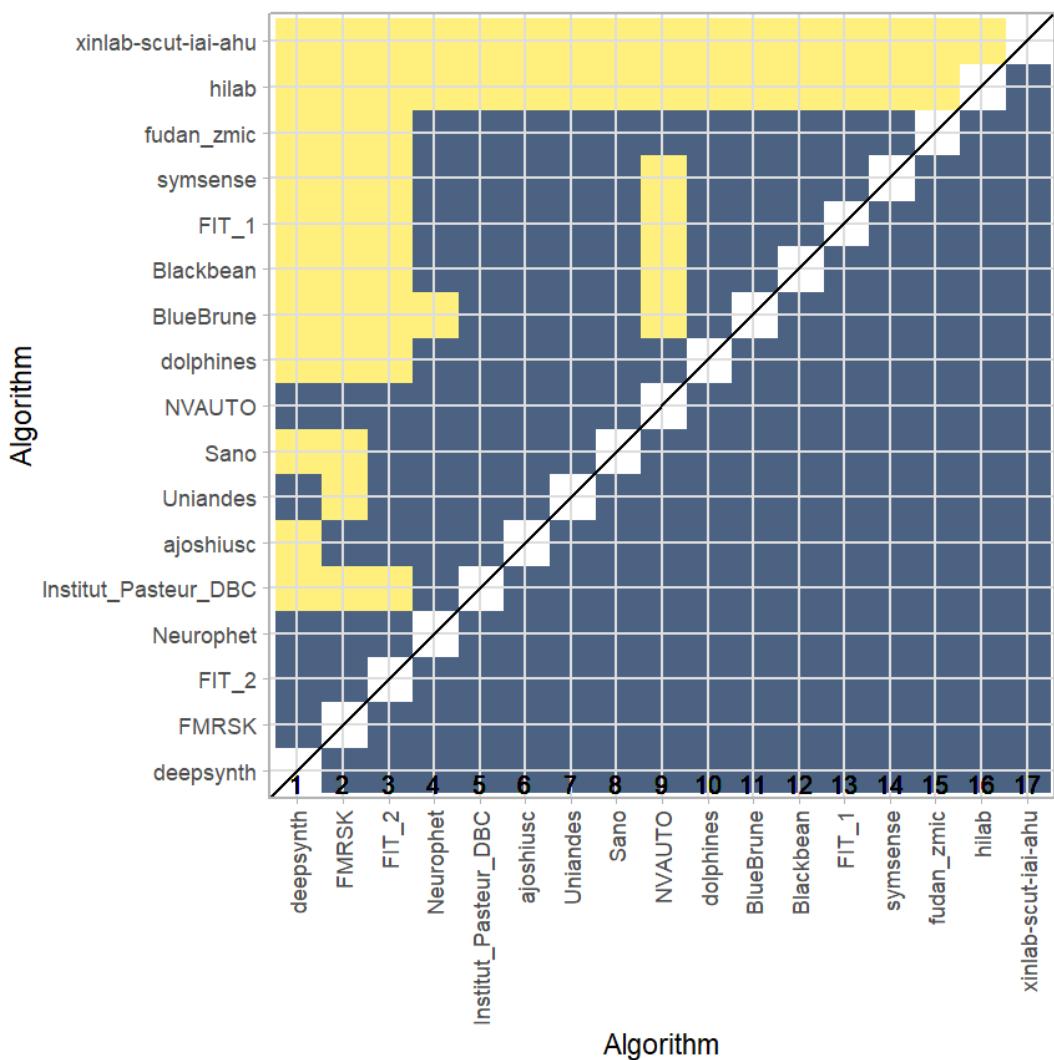
Summary Kendall's tau:

Task	mean	median	q25	q75
dummyTask	0.8834853	0.8970588	0.8529412	0.9264706



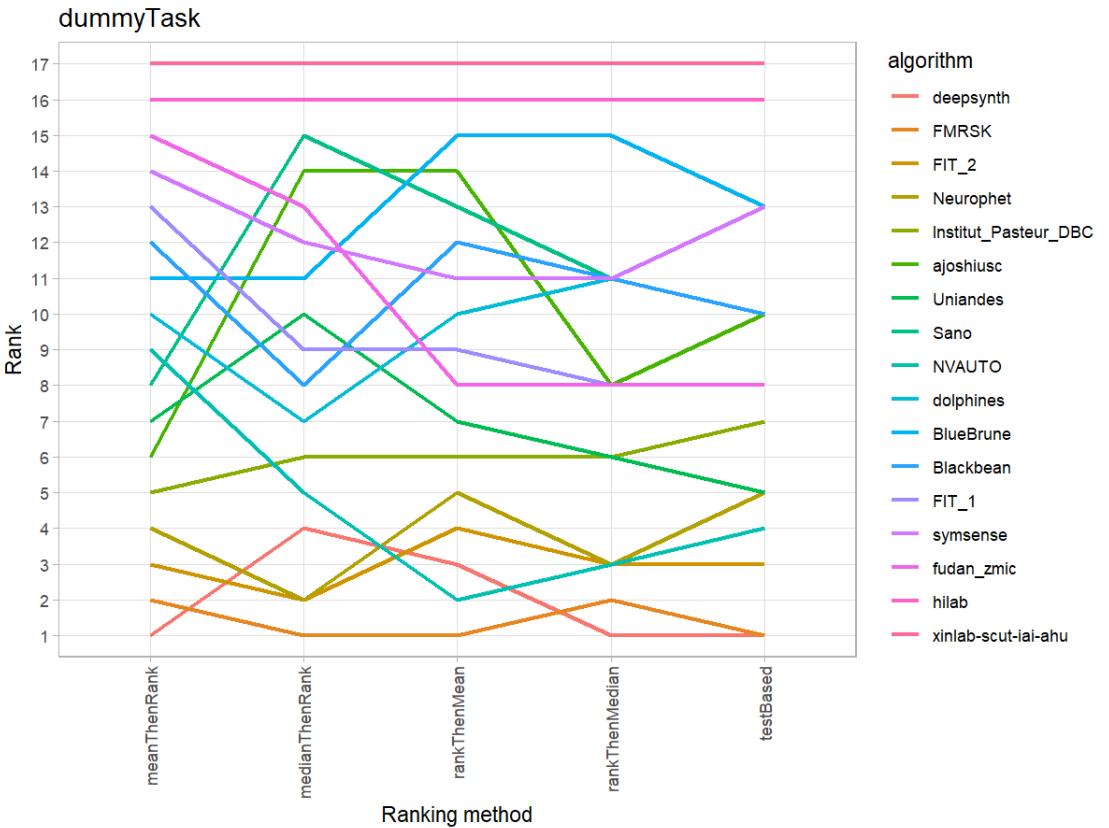
## *Significance maps* for visualizing ranking stability based on statistical significance

*Significance maps* depict incidence matrices of pairwise significant test results for the one-sided Wilcoxon signed rank test at a 5% significance level with adjustment for multiple testing according to Holm. Yellow shading indicates that metric values from the algorithm on the x-axis were significantly superior to those from the algorithm on the y-axis, blue color indicates no significant difference.



### Ranking robustness to ranking methods

*Line plots* for visualizing ranking robustness across different ranking methods. Each algorithm is represented by one colored line. For each ranking method encoded on the x-axis, the height of the line represents the corresponding rank. Horizontal lines indicate identical ranks for all methods.



### 29.3 References

Wiesenfarth, M., Reinke, A., Landman, B.A., Eisenmann, M., Aguilera Saiz, L., Cardoso, M.J., Maier-Hein, L. and Kopp-Schneider, A. Methods and open-source toolkit for analyzing and visualizing challenge results. *Sci Rep* **11**, 2369 (2021). <https://doi.org/10.1038/s41598-021-82017-6>

M. J. A. Eugster, T. Hothorn, and F. Leisch, “Exploratory and inferential analysis of benchmark experiments,” Institut fuer Statistik, Ludwig-Maximilians-Universitaet Muenchen, Germany, Technical Report 30, 2008. [Online]. Available: <http://epub.ub.uni-muenchen.de/4134/>.

## 30 Benchmarking report for Volume Similarity Metrics – irtkSimple Reconstruction Method

created by challengeR v1.0.2  
23 October, 2023

This document presents a systematic report on the benchmark study “Volume Similarity Metrics”. Input data comprises raw metric values for all algorithms and cases. Generated plots are:

- Visualization of assessment data: Dot- and boxplot, podium plot and ranking heatmap
- Visualization of ranking stability: Blob plot, violin plot and significance map, line plot

Details can be found in Wiesenfarth et al. (2021).

### 30.1 Ranking

Algorithms within a task are ranked according to the following ranking scheme:  
*aggregate using function (“mean”) then rank*

The analysis is based on 17 algorithms and 140 cases. 0 missing cases have been found in the data set.

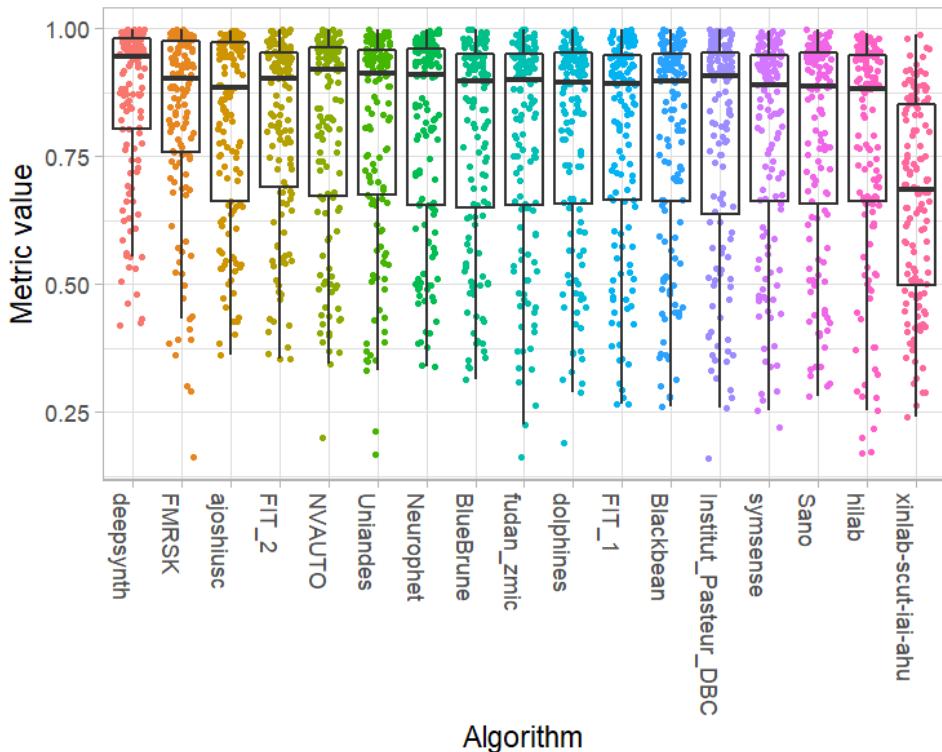
Ranking:

	Volume_Similarity_mean	rank
deepsynth	0.8649748	1
FMRSK	0.8339122	2
ajoshiusc	0.8159130	3
FIT_2	0.8128959	4
NVAUTO	0.8050503	5
Uniandes	0.8037555	6
Neurophet	0.8029831	7
BlueBrune	0.7933006	8
fudan_zmic	0.7913864	9
dolphines	0.7908812	10
FIT_1	0.7880491	11
Blackbean	0.7879563	12
Institut_Pasteur_DBC	0.7870338	13
symsense	0.7869364	14
Sano	0.7866331	15
hilab	0.7719335	16
xinlab-scut-iai-ahu	0.6660453	17

### 30.2 Visualization of raw assessment data

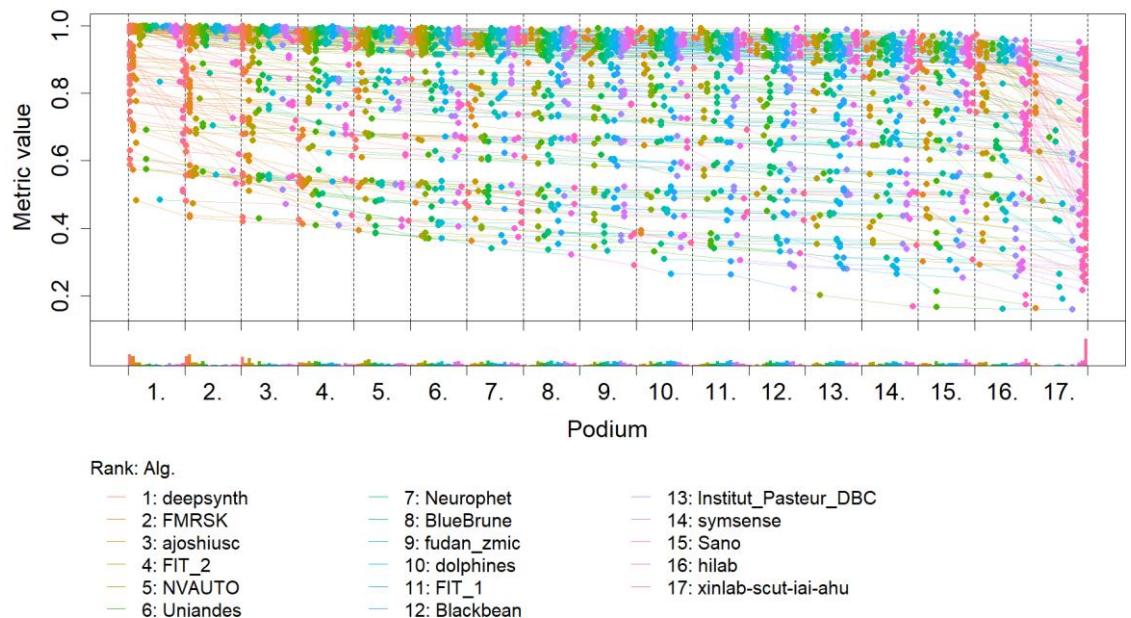
#### Dot- and boxplot

*Dot- and boxplots* for visualizing raw assessment data separately for each algorithm. Boxplots representing descriptive statistics over all cases (median, quartiles and outliers) are combined with horizontally jittered dots representing individual cases.



## Podium plot

*Podium plots* (see also Eugster et al., 2008) for visualizing raw assessment data. Upper part (spaghetti plot): Participating algorithms are color-coded, and each colored dot in the plot represents a metric value achieved with the respective algorithm. The actual metric value is encoded by the y-axis. Each podium (here:  $p=17$ ) represents one possible rank, ordered from best (1) to last (here: 17). The assignment of metric values (i.e. colored dots) to one of the podiums is based on the rank that the respective algorithm achieved on the corresponding case. Note that the plot part above each podium place is further subdivided into  $p$  “columns”, where each column represents one participating algorithm (here:  $p = 17$ ). Dots corresponding to identical cases are connected by a line, leading to the shown spaghetti structure. Lower part: Bar charts represent the relative frequency for each algorithm to achieve the rank encoded by the podium place.



## Ranking heatmap

*Ranking heatmaps* for visualizing raw assessment data. Each cell  $(i, A_j)$  shows the absolute frequency of cases in which algorithm  $A_j$  achieved rank  $i$ .

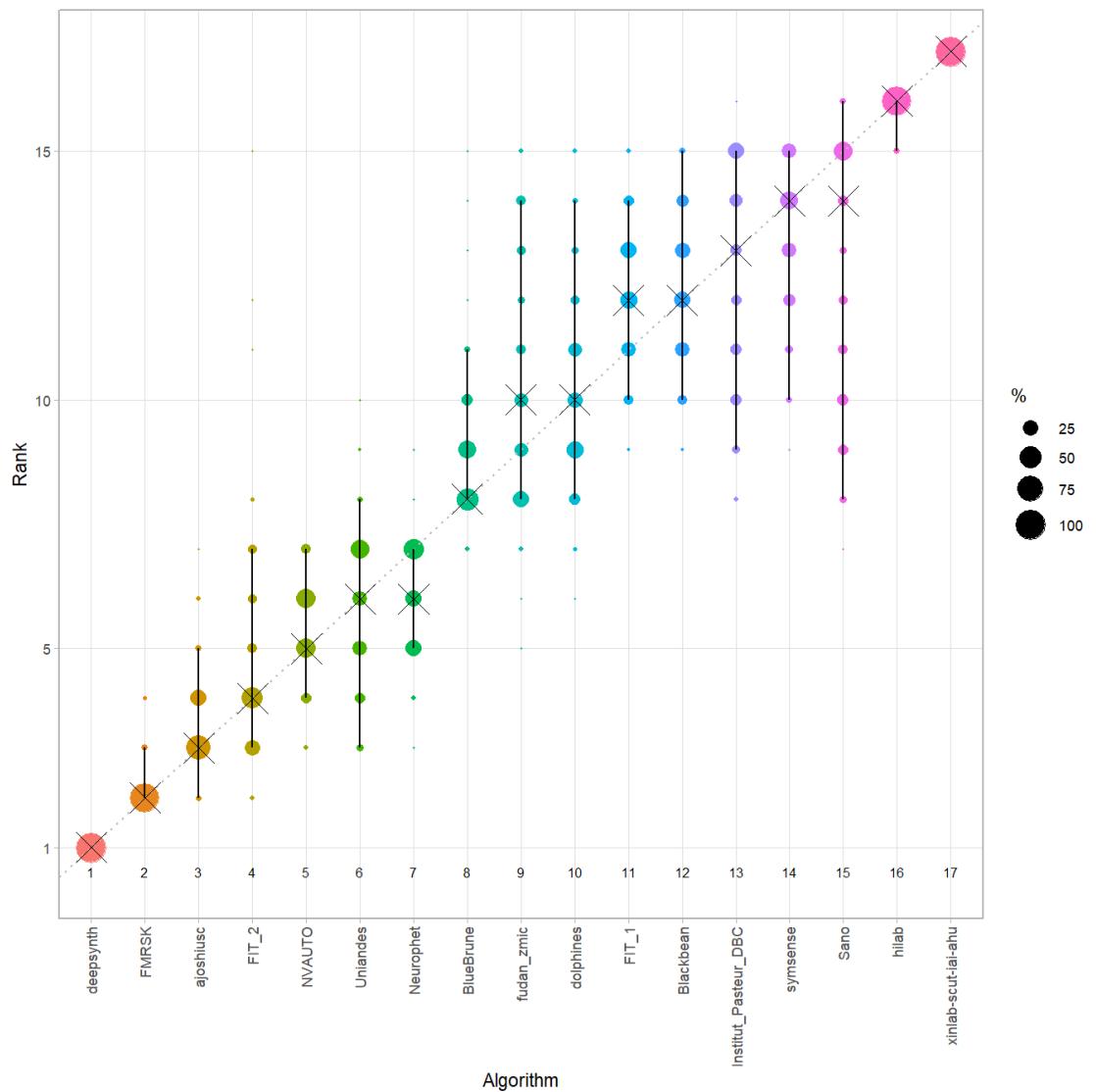


### Visualization of ranking stability

*Blob plot* for visualizing ranking stability based on bootstrap sampling

Algorithms are color-coded, and the area of each blob at position  $(A_i, \text{rank } j)$  is proportional to the relative frequency  $A_i$  achieved rank  $j$  across  $b = 1000$  bootstrap samples. The median rank for each algorithm is indicated by a black cross. 95% bootstrap intervals across bootstrap samples are indicated by black lines.

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.
```

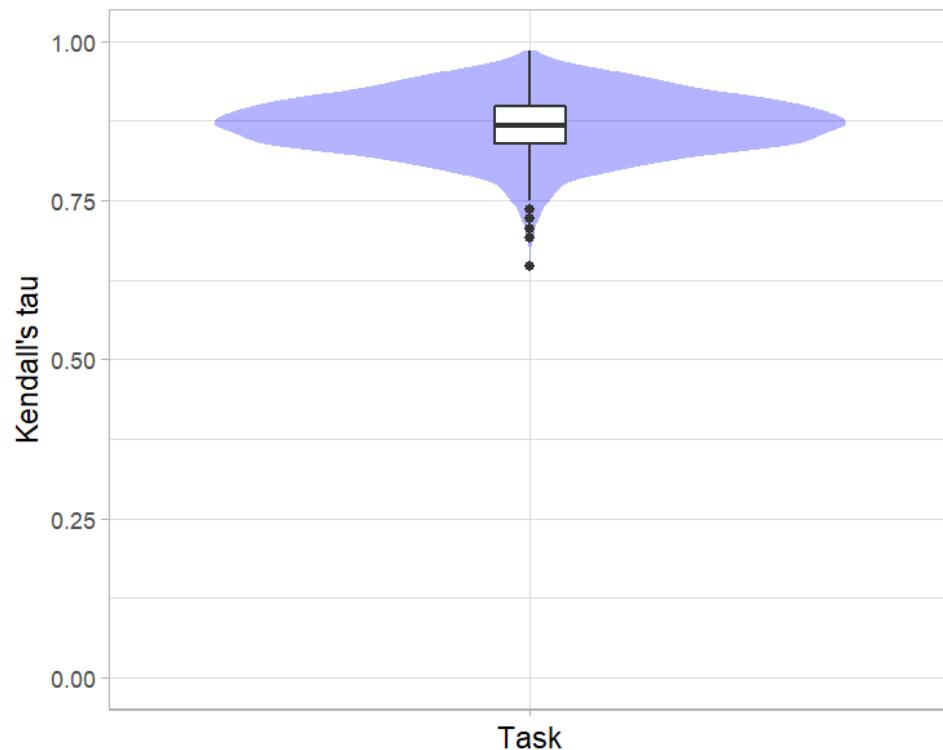


### **Violin plot for visualizing ranking stability based on bootstrapping**

The ranking list based on the full assessment data is pairwise compared with the ranking lists based on the individual bootstrap samples (here  $b = 1000$  samples). For each pair of rankings, Kendall's  $\tau$  correlation is computed. Kendall's  $\tau$  is a scaled index determining the correlation between the lists. It is computed by evaluating the number of pairwise concordances and discordances between ranking lists and produces values between  $-1$  (for inverted order) and  $1$  (for identical order). A violin plot, which simultaneously depicts a boxplot and a density plot, is generated from the results.

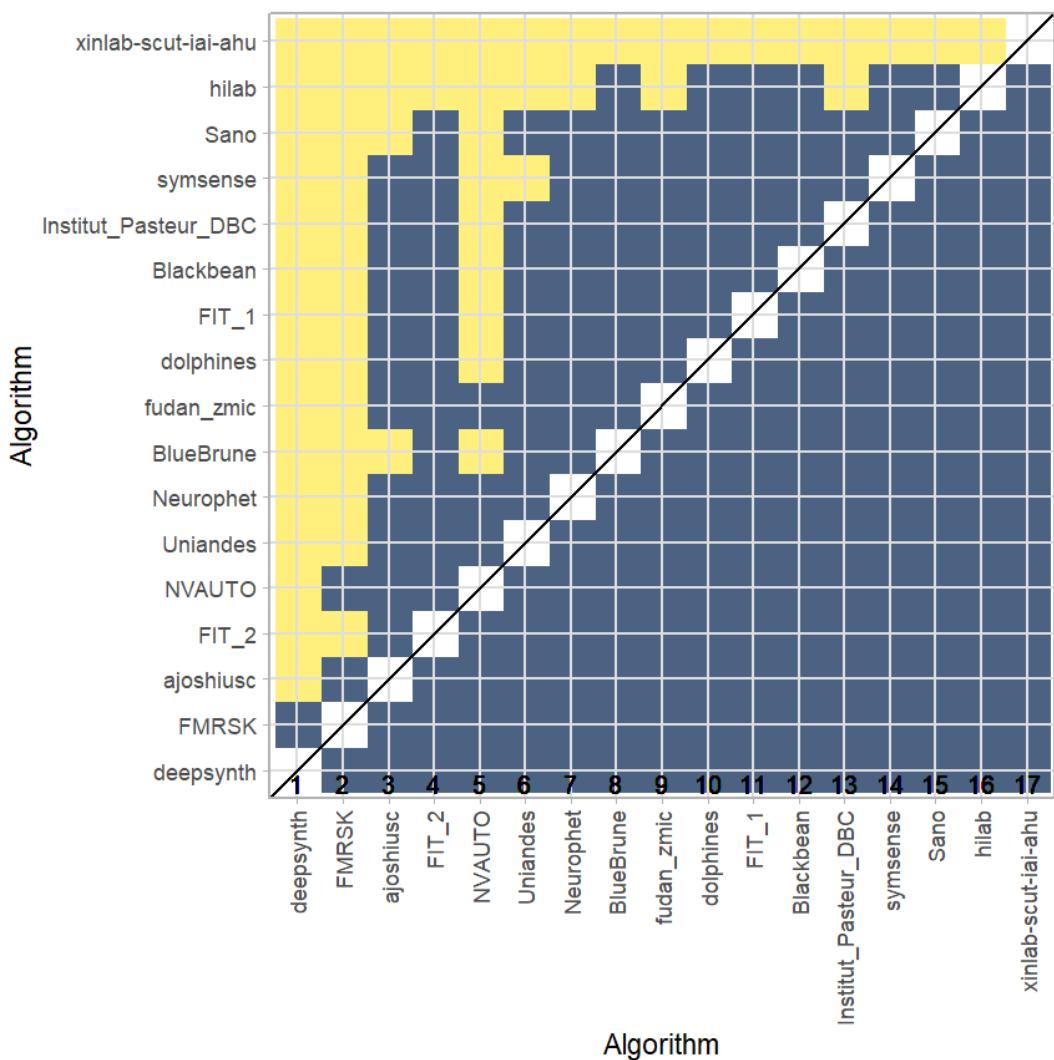
Summary Kendall's tau:

Task	mean	median	q25	q75
dummyTask	0.8693088	0.8676471	0.8382353	0.8970588



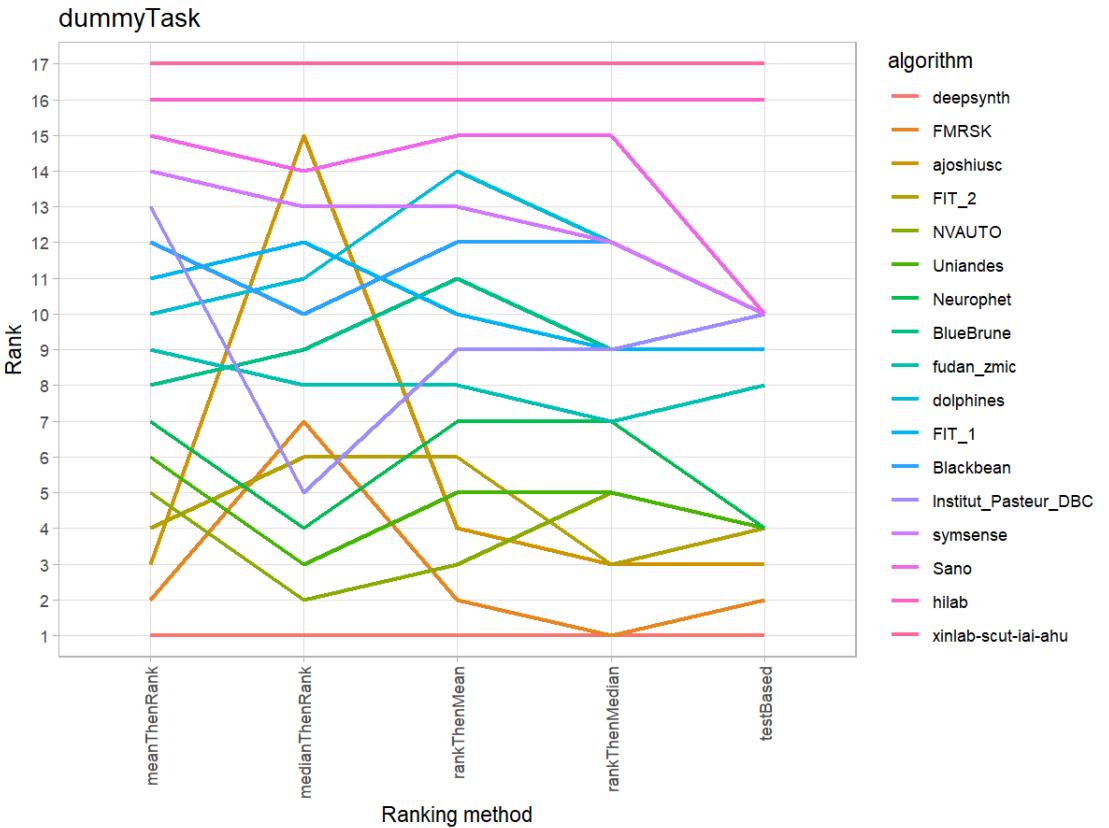
## *Significance maps* for visualizing ranking stability based on statistical significance

*Significance maps* depict incidence matrices of pairwise significant test results for the one-sided Wilcoxon signed rank test at a 5% significance level with adjustment for multiple testing according to Holm. Yellow shading indicates that metric values from the algorithm on the x-axis were significantly superior to those from the algorithm on the y-axis, blue color indicates no significant difference.



### Ranking robustness to ranking methods

*Line plots* for visualizing ranking robustness across different ranking methods. Each algorithm is represented by one colored line. For each ranking method encoded on the x-axis, the height of the line represents the corresponding rank. Horizontal lines indicate identical ranks for all methods.



### 30.3 References

Wiesenfarth, M., Reinke, A., Landman, B.A., Eisenmann, M., Aguilera Saiz, L., Cardoso, M.J., Maier-Hein, L. and Kopp-Schneider, A. Methods and open-source toolkit for analyzing and visualizing challenge results. *Sci Rep* **11**, 2369 (2021). <https://doi.org/10.1038/s41598-021-82017-6>

M. J. A. Eugster, T. Hothorn, and F. Leisch, “Exploratory and inferential analysis of benchmark experiments,” Institut fuer Statistik, Ludwig-Maximilians-Universitaet Muenchen, Germany, Technical Report 30, 2008. [Online]. Available: <http://epub.ub.uni-muenchen.de/4134/>.

## 31 Benchmarking report for Dice Metrics – mial-srtk Reconstruction Method

created by challengeR v1.0.2  
23 October, 2023

This document presents a systematic report on the benchmark study “Dice Metrics”. Input data comprises raw metric values for all algorithms and cases. Generated plots are:

- Visualization of assessment data: Dot- and boxplot, podium plot and ranking heatmap
- Visualization of ranking stability: Blob plot, violin plot and significance map, line plot

Details can be found in Wiesenfarth et al. (2021).

### 31.1 Ranking

Algorithms within a task are ranked according to the following ranking scheme:  
*aggregate using function (“mean”) then rank*

The analysis is based on 17 algorithms and 420 cases. 0 missing cases have been found in the data set.

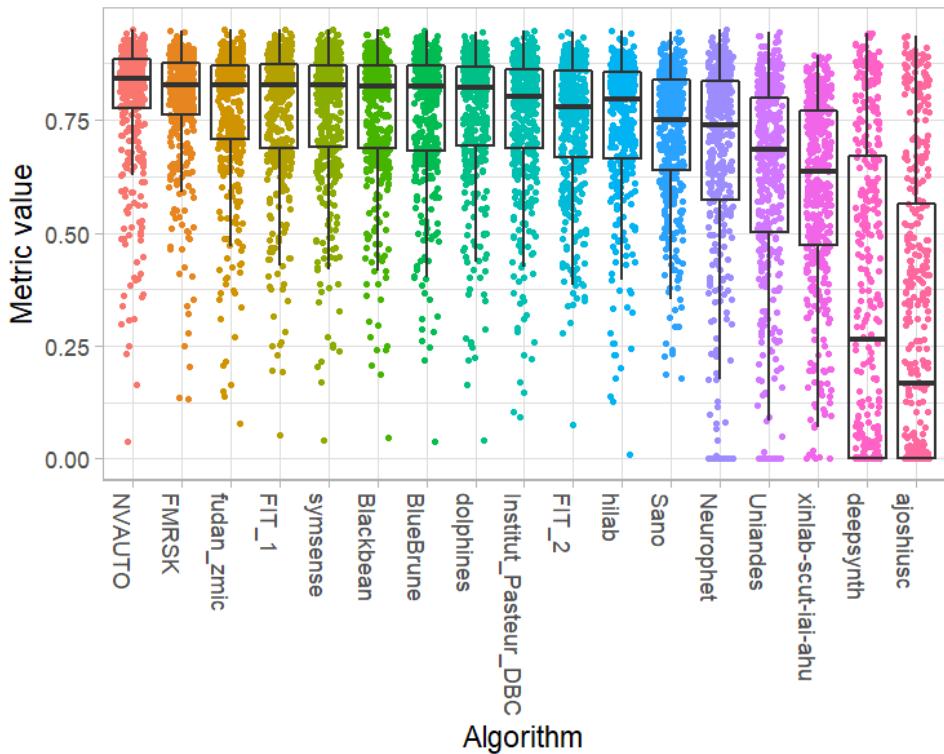
Ranking:

	Dice_mean	rank
NVAUTO	0.7965113	1
FMRSK	0.7932151	2
fudan_zmic	0.7702846	3
FIT_1	0.7693562	4
symsense	0.7677151	5
Blackbean	0.7668915	6
BlueBrune	0.7664517	7
dolphines	0.7660195	8
Institut_Pasteur_DBC	0.7528768	9
FIT_2	0.7426605	10
hilab	0.7424022	11
Sano	0.7155146	12
Neurophet	0.6500694	13
Uniandes	0.6166908	14
xinlab-scut-iai-ahu	0.5946697	15
deepsynth	0.3460063	16
ajoshiusc	0.2965834	17

### 31.2 Visualization of raw assessment data

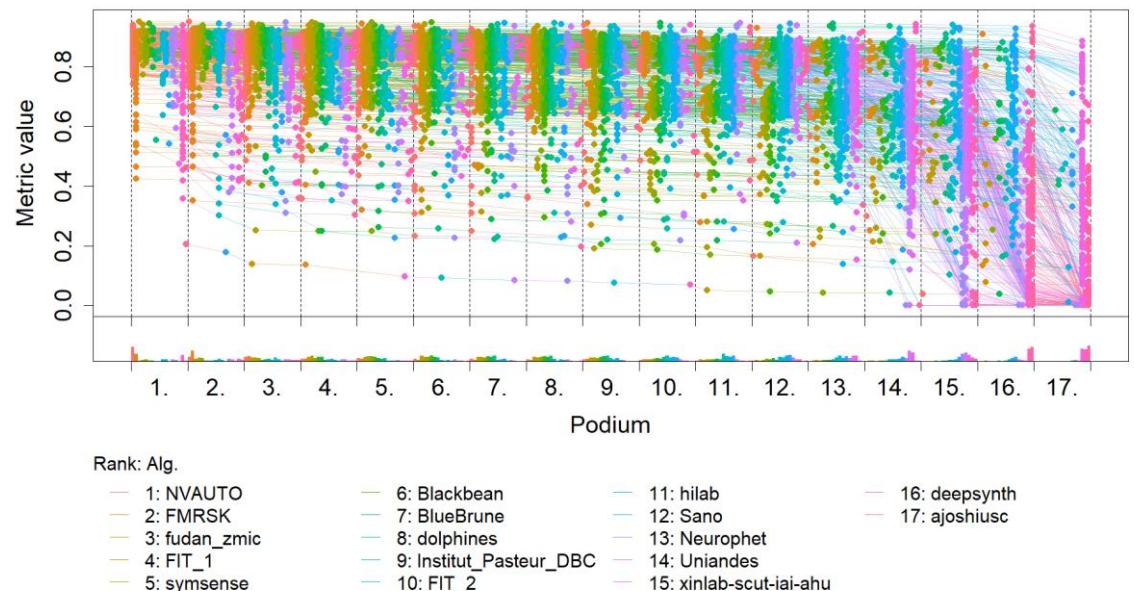
#### Dot- and boxplot

*Dot- and boxplots* for visualizing raw assessment data separately for each algorithm. Boxplots representing descriptive statistics over all cases (median, quartiles and outliers) are combined with horizontally jittered dots representing individual cases.



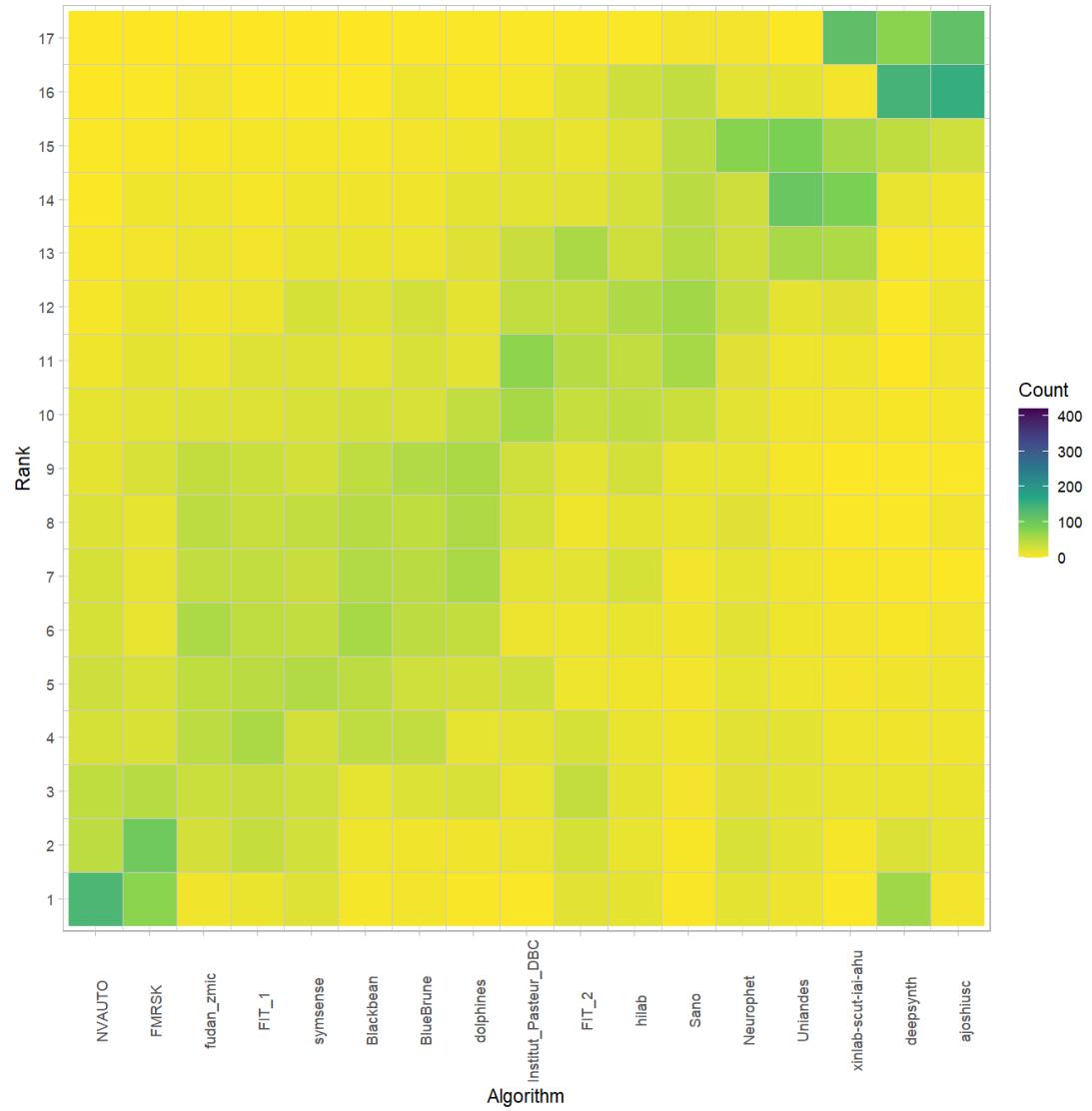
## Podium plot

*Podium plots* (see also Eugster et al., 2008) for visualizing raw assessment data. Upper part (spaghetti plot): Participating algorithms are color-coded, and each colored dot in the plot represents a metric value achieved with the respective algorithm. The actual metric value is encoded by the y-axis. Each podium (here:  $p=17$ ) represents one possible rank, ordered from best (1) to last (here: 17). The assignment of metric values (i.e. colored dots) to one of the podiums is based on the rank that the respective algorithm achieved on the corresponding case. Note that the plot part above each podium place is further subdivided into  $p$  “columns”, where each column represents one participating algorithm (here:  $p = 17$ ). Dots corresponding to identical cases are connected by a line, leading to the shown spaghetti structure. Lower part: Bar charts represent the relative frequency for each algorithm to achieve the rank encoded by the podium place.



## Ranking heatmap

*Ranking heatmaps* for visualizing raw assessment data. Each cell  $(i, A_j)$  shows the absolute frequency of cases in which algorithm  $A_j$  achieved rank  $i$ .

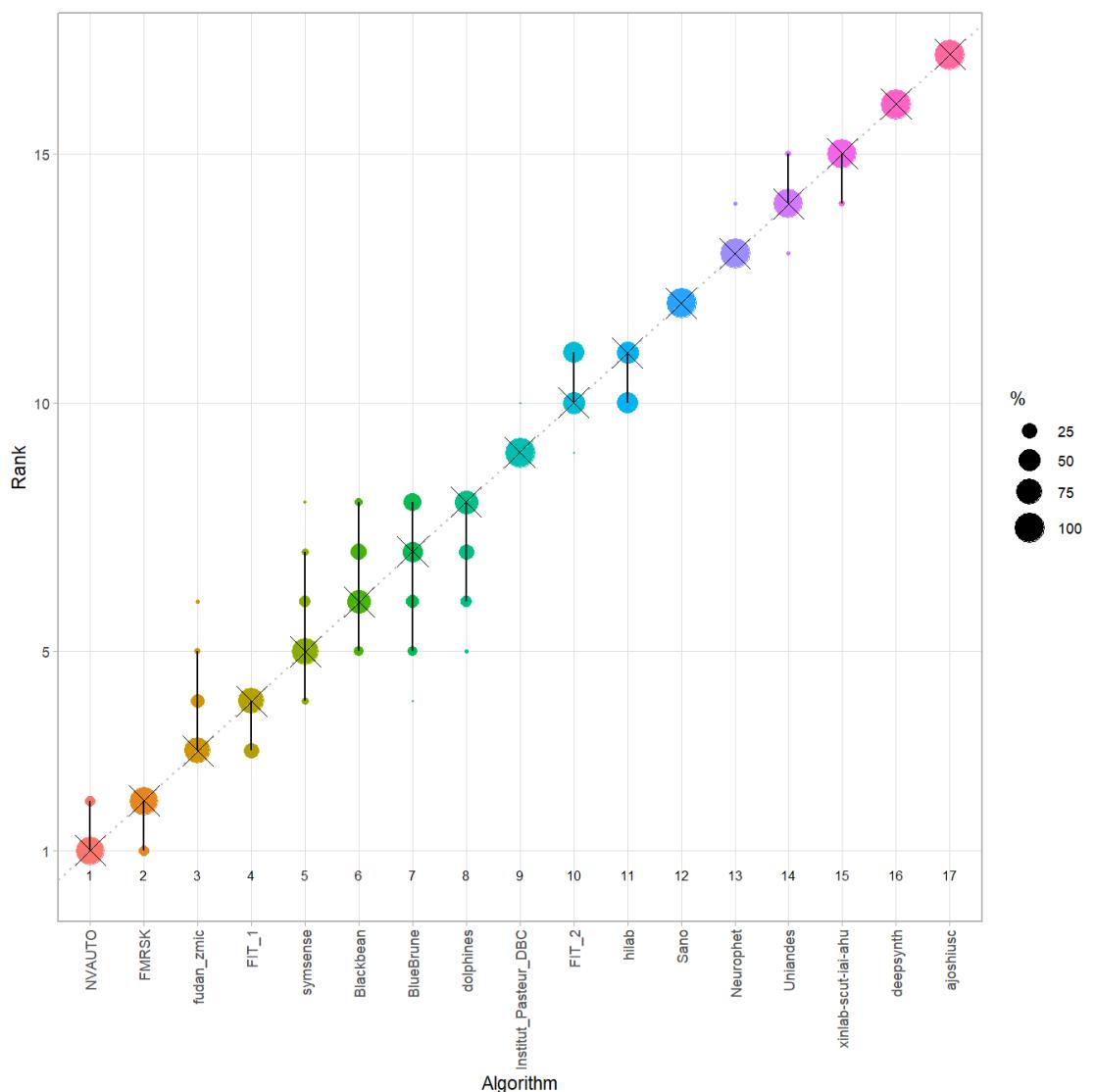


### Visualization of ranking stability

#### *Blob plot for visualizing ranking stability based on bootstrap sampling*

Algorithms are color-coded, and the area of each blob at position ( $A_i$ , rank  $j$ ) is proportional to the relative frequency  $A_i$  achieved rank  $j$  across  $b = 1000$  bootstrap samples. The median rank for each algorithm is indicated by a black cross. 95% bootstrap intervals across bootstrap samples are indicated by black lines.

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.
```

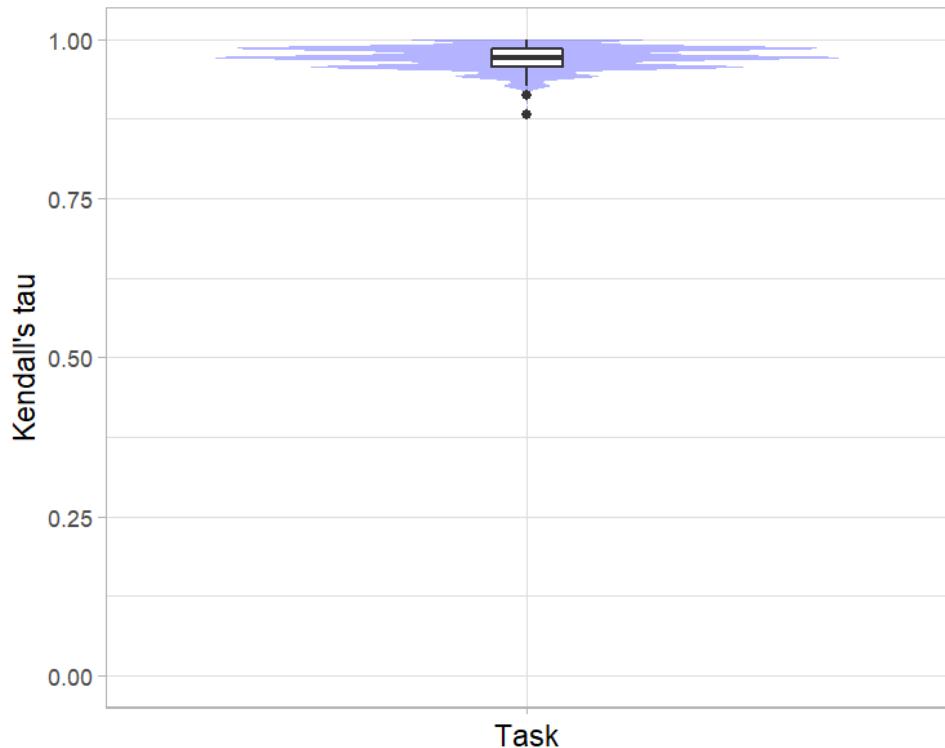


### ***Violin plot for visualizing ranking stability based on bootstrapping***

The ranking list based on the full assessment data is pairwise compared with the ranking lists based on the individual bootstrap samples (here  $b = 1000$  samples). For each pair of rankings, Kendall's  $\tau$  correlation is computed. Kendall's  $\tau$  is a scaled index determining the correlation between the lists. It is computed by evaluating the number of pairwise concordances and discordances between ranking lists and produces values between  $-1$  (for inverted order) and  $1$  (for identical order). A violin plot, which simultaneously depicts a boxplot and a density plot, is generated from the results.

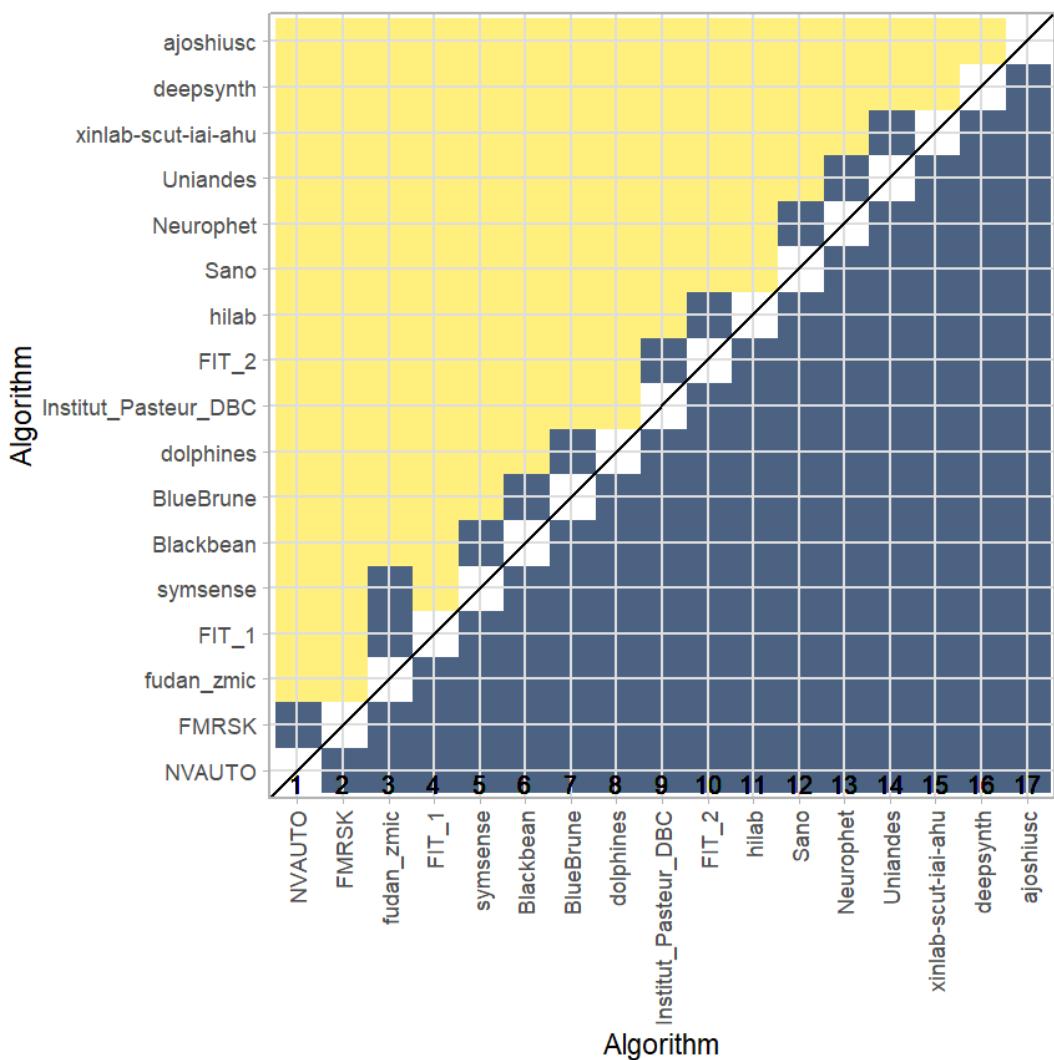
Summary Kendall's tau:

Task	mean	median	q25	q75
dummyTask	0.9719118	0.9705882	0.9558824	0.9852941



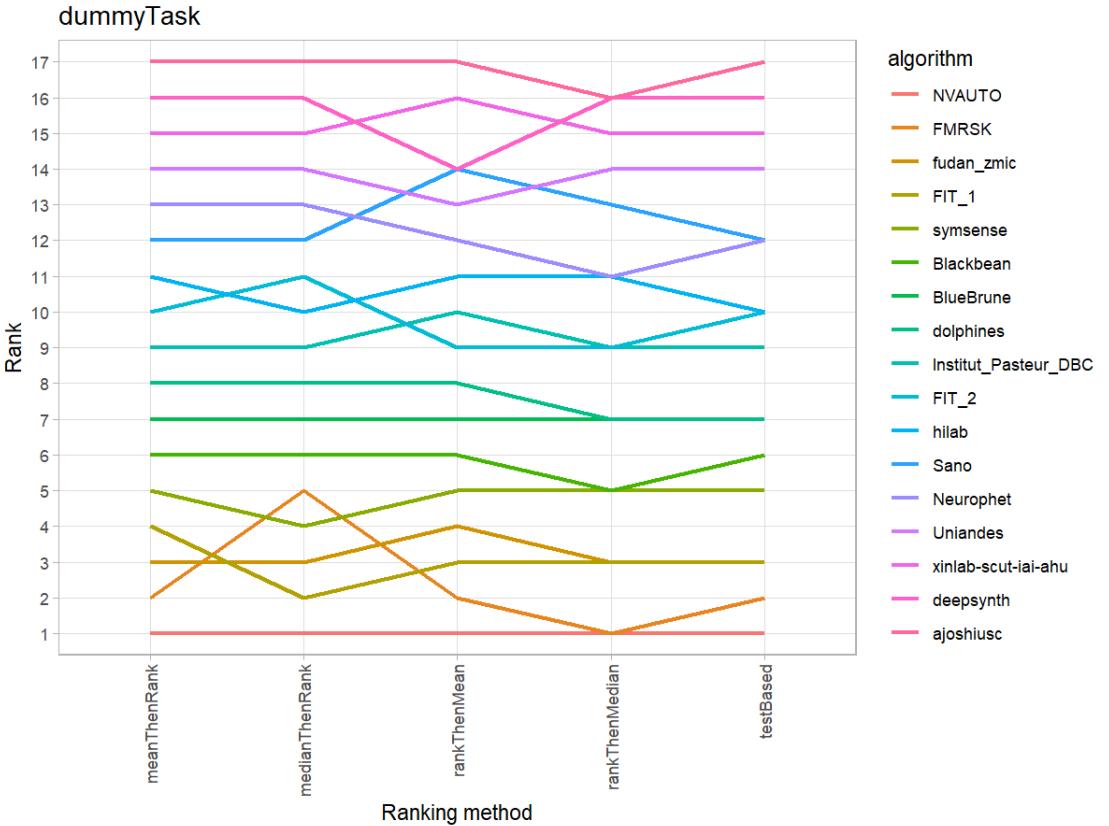
## *Significance maps* for visualizing ranking stability based on statistical significance

*Significance maps* depict incidence matrices of pairwise significant test results for the one-sided Wilcoxon signed rank test at a 5% significance level with adjustment for multiple testing according to Holm. Yellow shading indicates that metric values from the algorithm on the x-axis were significantly superior to those from the algorithm on the y-axis, blue color indicates no significant difference.



### Ranking robustness to ranking methods

*Line plots* for visualizing ranking robustness across different ranking methods. Each algorithm is represented by one colored line. For each ranking method encoded on the x-axis, the height of the line represents the corresponding rank. Horizontal lines indicate identical ranks for all methods.



### 31.3 References

Wiesenfarth, M., Reinke, A., Landman, B.A., Eisenmann, M., Aguilera Saiz, L., Cardoso, M.J., Maier-Hein, L. and Kopp-Schneider, A. Methods and open-source toolkit for analyzing and visualizing challenge results. *Sci Rep* **11**, 2369 (2021). <https://doi.org/10.1038/s41598-021-82017-6>

M. J. A. Eugster, T. Hothorn, and F. Leisch, “Exploratory and inferential analysis of benchmark experiments,” Institut fuer Statistik, Ludwig-Maximilians-Universitaet Muenchen, Germany, Technical Report 30, 2008. [Online]. Available: <http://epub.ub.uni-muenchen.de/4134/>.

## 32 Benchmarking report for Hausdorff Metrics – mial-srtk Reconstruction Method

created by challengeR v1.0.2  
23 October, 2023

This document presents a systematic report on the benchmark study “Hausdorff Metrics”. Input data comprises raw metric values for all algorithms and cases. Generated plots are:

- Visualization of assessment data: Dot- and boxplot, podium plot and ranking heatmap
- Visualization of ranking stability: Blob plot, violin plot and significance map, line plot

Details can be found in Wiesenfarth et al. (2021).

### 32.1 Ranking

Algorithms within a task are ranked according to the following ranking scheme:  
*aggregate using function (“mean”) then rank*

The analysis is based on 17 algorithms and 420 cases. 0 missing cases have been found in the data set.

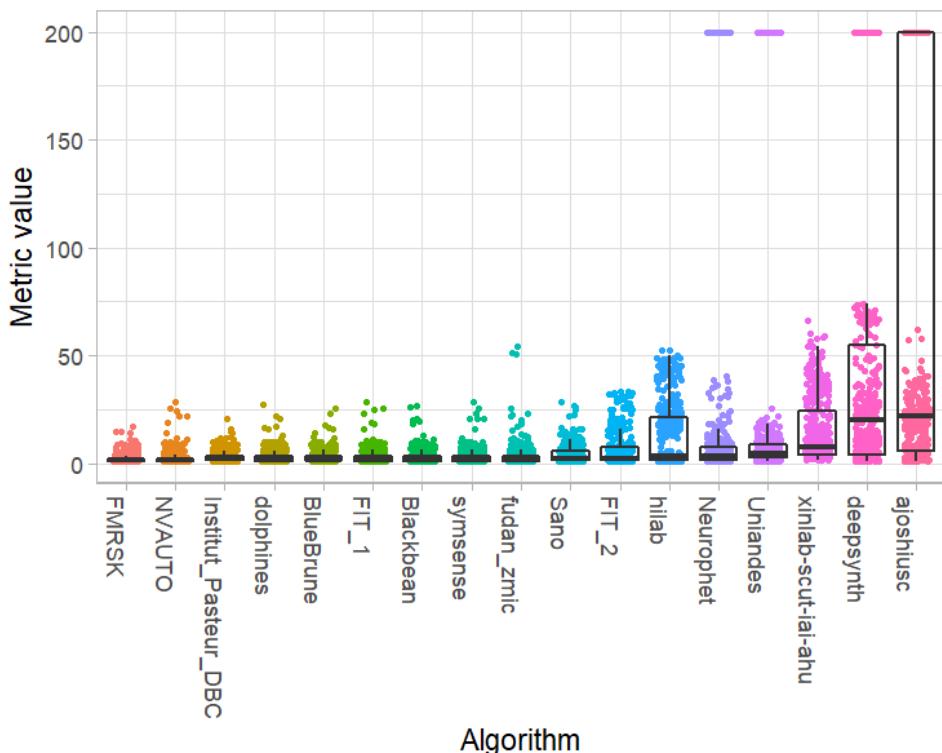
Ranking:

	Hausdorff_m	r
ean		ank
FMRSK	2.445920	1
NVAUTO	2.670251	2
Institut_Pasteur_DBC	3.075040	3
dolphines	3.082171	4
BlueBrune	3.121191	5
FIT_1	3.140513	6
Blackbean	3.141349	7
symsense	3.158501	8
fudan_zmic	3.473781	9
Sano	4.256106	10
FIT_2	6.118941	11
hilab	13.404766	12
Neurophet	14.004507	13
Uniandes	14.157257	14
xinlab-scut-iai-ahu	15.190746	15
deepsynth	48.140913	16
ajoshiusc	63.917304	17

### 32.2 Visualization of raw assessment data

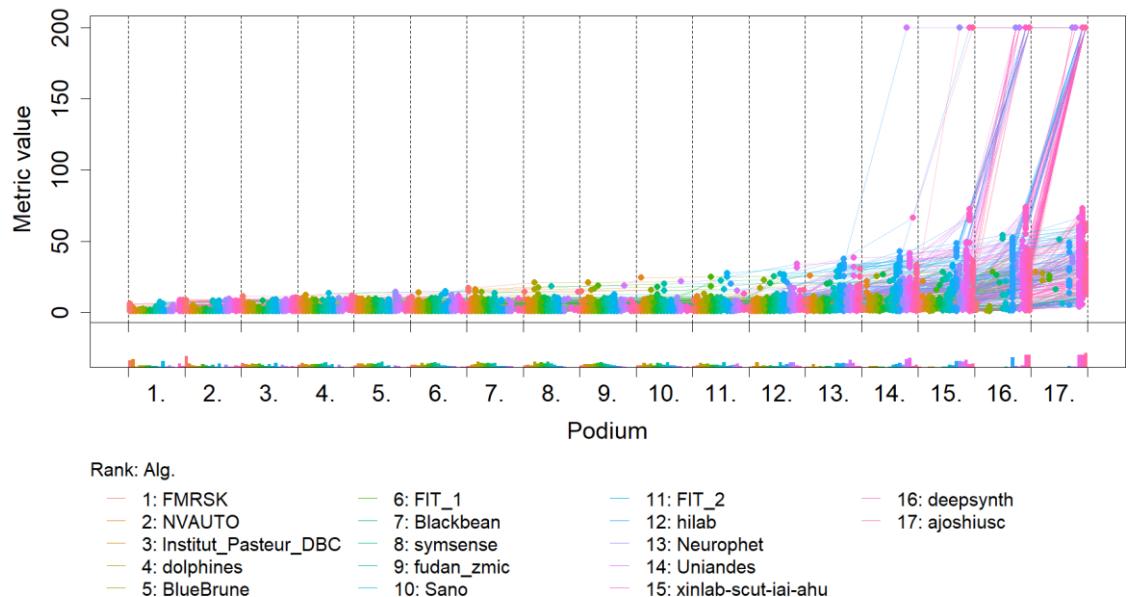
#### Dot- and boxplot

*Dot- and boxplots* for visualizing raw assessment data separately for each algorithm. Boxplots representing descriptive statistics over all cases (median, quartiles and outliers) are combined with horizontally jittered dots representing individual cases.



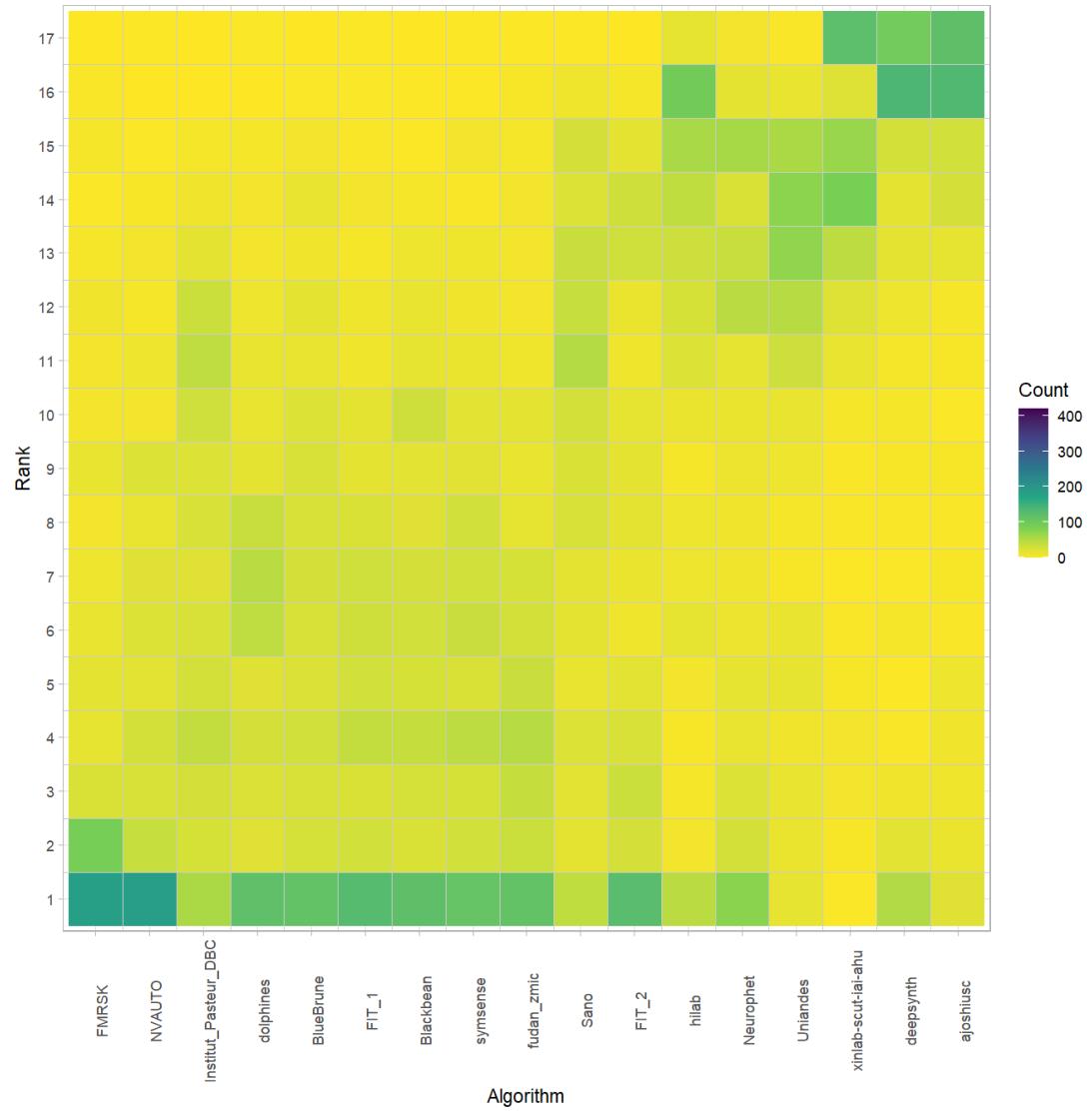
## Podium plot

*Podium plots* (see also Eugster et al., 2008) for visualizing raw assessment data. Upper part (spaghetti plot): Participating algorithms are color-coded, and each colored dot in the plot represents a metric value achieved with the respective algorithm. The actual metric value is encoded by the y-axis. Each podium (here:  $p=17$ ) represents one possible rank, ordered from best (1) to last (here: 17). The assignment of metric values (i.e. colored dots) to one of the podiums is based on the rank that the respective algorithm achieved on the corresponding case. Note that the plot part above each podium place is further subdivided into  $p$  “columns”, where each column represents one participating algorithm (here:  $p = 17$ ). Dots corresponding to identical cases are connected by a line, leading to the shown spaghetti structure. Lower part: Bar charts represent the relative frequency for each algorithm to achieve the rank encoded by the podium place.



## Ranking heatmap

*Ranking heatmaps* for visualizing raw assessment data. Each cell  $(i, A_j)$  shows the absolute frequency of cases in which algorithm  $A_j$  achieved rank  $i$ .

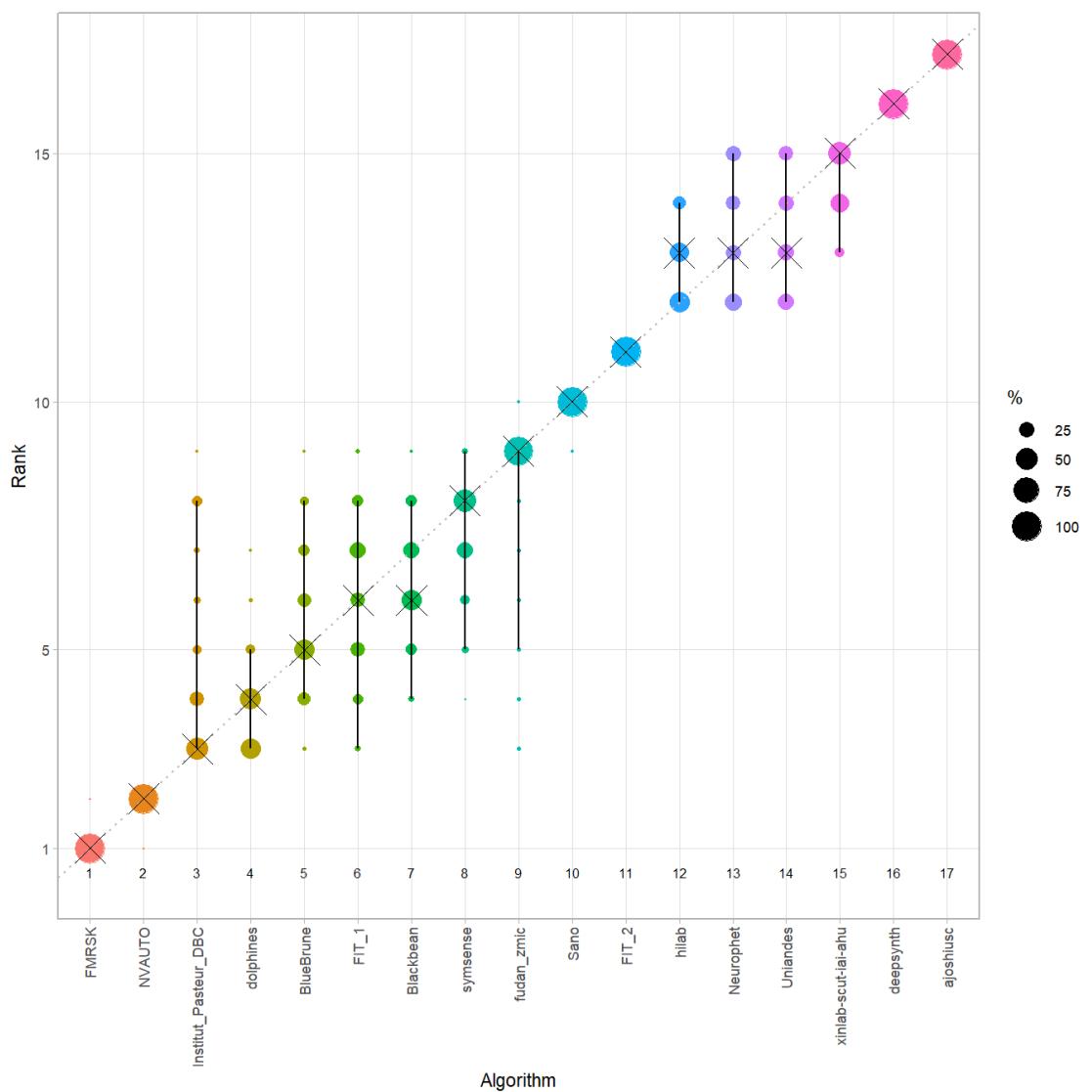


### Visualization of ranking stability

#### *Blob plot for visualizing ranking stability based on bootstrap sampling*

Algorithms are color-coded, and the area of each blob at position ( $A_i, \text{rank } j$ ) is proportional to the relative frequency  $A_i$  achieved rank  $j$  across  $b = 1000$  bootstrap samples. The median rank for each algorithm is indicated by a black cross. 95% bootstrap intervals across bootstrap samples are indicated by black lines.

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.
```

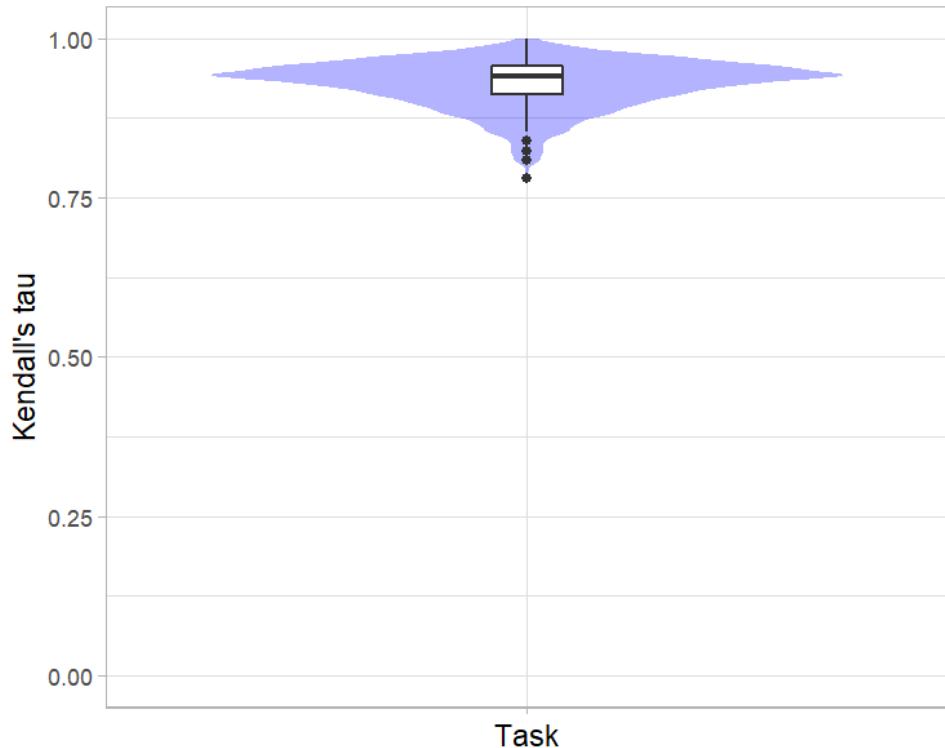


### ***Violin plot for visualizing ranking stability based on bootstrapping***

The ranking list based on the full assessment data is pairwise compared with the ranking lists based on the individual bootstrap samples (here  $b = 1000$  samples). For each pair of rankings, Kendall's  $\tau$  correlation is computed. Kendall's  $\tau$  is a scaled index determining the correlation between the lists. It is computed by evaluating the number of pairwise concordances and discordances between ranking lists and produces values between  $-1$  (for inverted order) and  $1$  (for identical order). A violin plot, which simultaneously depicts a boxplot and a density plot, is generated from the results.

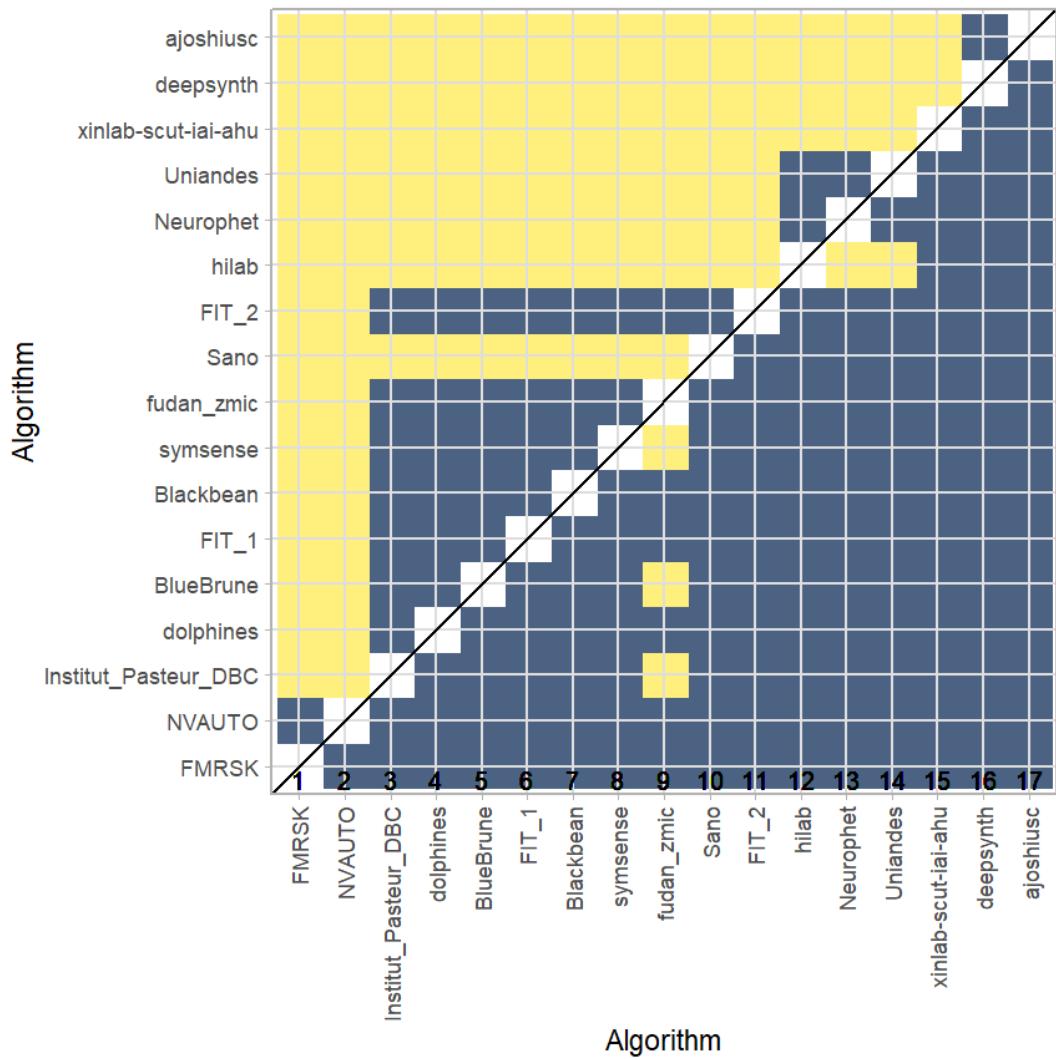
Summary Kendall's tau:

Task	mean	median	q25	q75
dummyTask	0.9278235	0.9411765	0.9117647	0.9558824



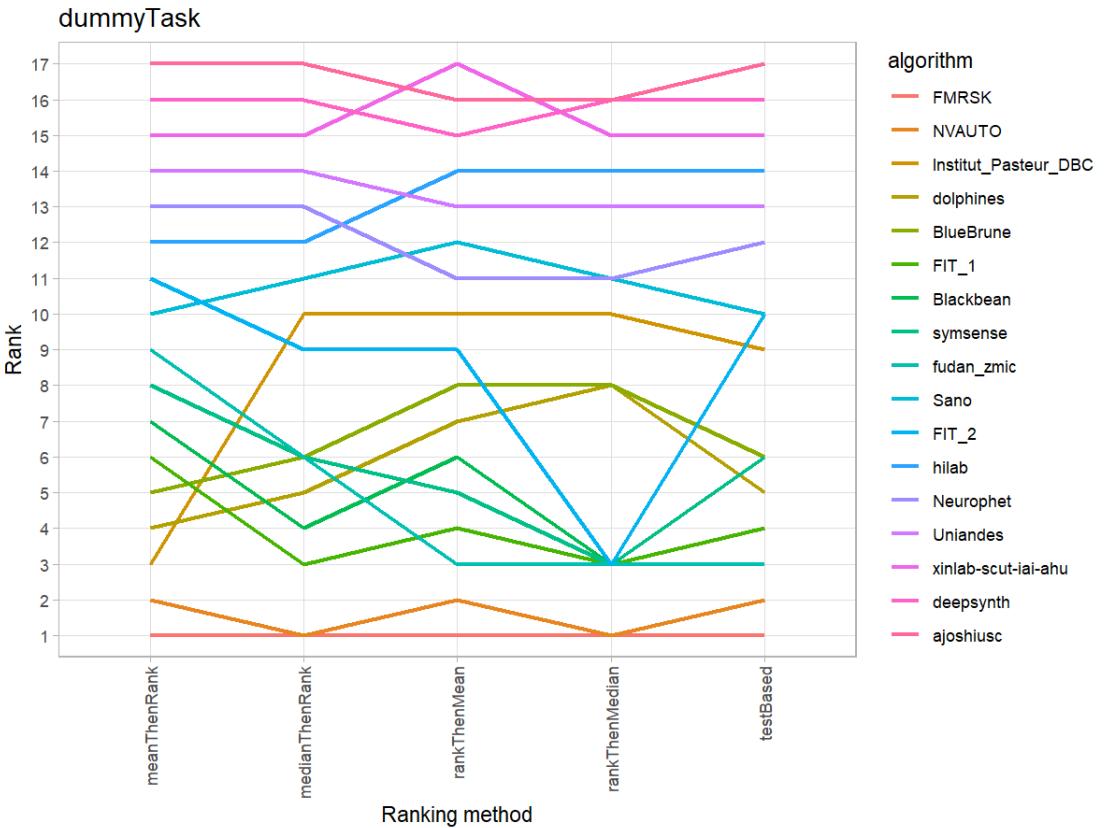
**Significance maps for visualizing ranking stability based on statistical significance**

*Significance maps* depict incidence matrices of pairwise significant test results for the one-sided Wilcoxon signed rank test at a 5% significance level with adjustment for multiple testing according to Holm. Yellow shading indicates that metric values from the algorithm on the x-axis were significantly superior to those from the algorithm on the y-axis, blue color indicates no significant difference.



### Ranking robustness to ranking methods

*Line plots* for visualizing ranking robustness across different ranking methods. Each algorithm is represented by one colored line. For each ranking method encoded on the x-axis, the height of the line represents the corresponding rank. Horizontal lines indicate identical ranks for all methods.



### 32.3 References

Wiesenfarth, M., Reinke, A., Landman, B.A., Eisenmann, M., Aguilera Saiz, L., Cardoso, M.J., Maier-Hein, L. and Kopp-Schneider, A. Methods and open-source toolkit for analyzing and visualizing challenge results. *Sci Rep* **11**, 2369 (2021). <https://doi.org/10.1038/s41598-021-82017-6>

M. J. A. Eugster, T. Hothorn, and F. Leisch, “Exploratory and inferential analysis of benchmark experiments,” Institut fuer Statistik, Ludwig-Maximilians-Universitaet Muenchen, Germany, Technical Report 30, 2008. [Online]. Available: <http://epub.ub.uni-muenchen.de/4134/>.

### 33 Benchmarking report for Volume Similarity Metrics – mial-srtk Reconstruction Method

created by challengeR v1.0.2  
23 October, 2023

This document presents a systematic report on the benchmark study “Volume Similarity Metrics”. Input data comprises raw metric values for all algorithms and cases. Generated plots are:

- Visualization of assessment data: Dot- and boxplot, podium plot and ranking heatmap
- Visualization of ranking stability: Blob plot, violin plot and significance map, line plot

Details can be found in Wiesenfarth et al. (2021).

#### 33.1 Ranking

Algorithms within a task are ranked according to the following ranking scheme:  
*aggregate using function (“mean”) then rank*

The analysis is based on 17 algorithms and 420 cases. 0 missing cases have been found in the data set.

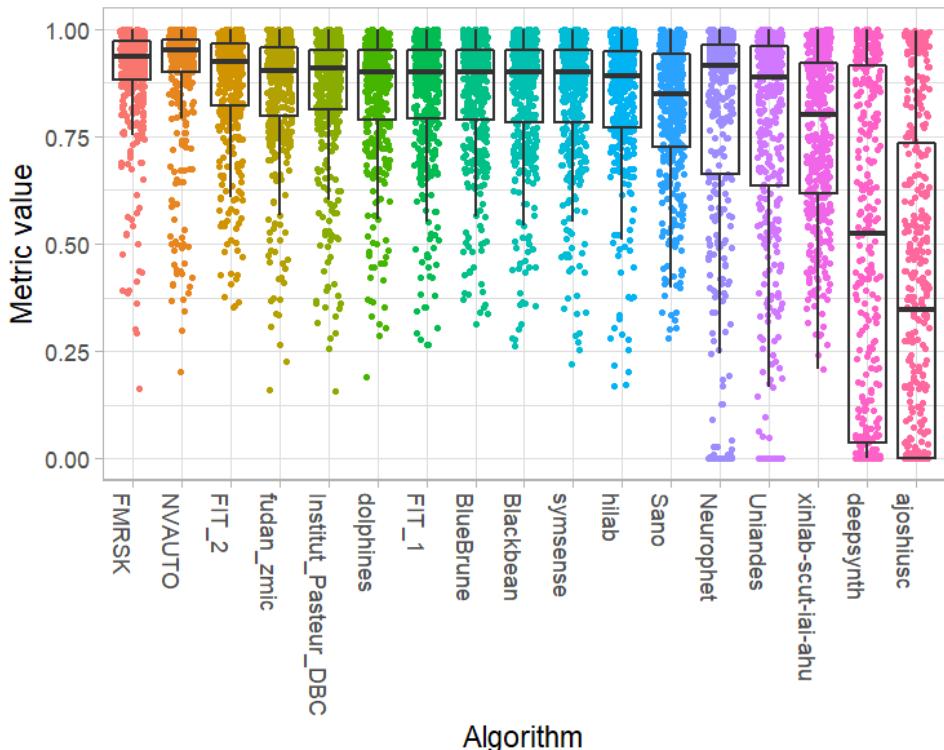
Ranking:

	Volume_Similarity_mean	rank
FMRSK	0.8984594	1
NVAUTO	0.8971790	2
FIT_2	0.8712877	3
fudan_zmic	0.8553545	4
Institut_Pasteur_DBC	0.8520549	5
dolphins	0.8481905	6
FIT_1	0.8474980	7
BlueBrune	0.8459830	8
Blackbean	0.8458801	9
symsense	0.8452604	10
hilab	0.8386639	11
Sano	0.8094217	12
Neurophet	0.7707029	13
Uniandes	0.7598427	14
xinlab-scut-iai-ahu	0.7556871	15
deepsynth	0.4865213	16
ajoshiusc	0.3915736	17

### 33.2 Visualization of raw assessment data

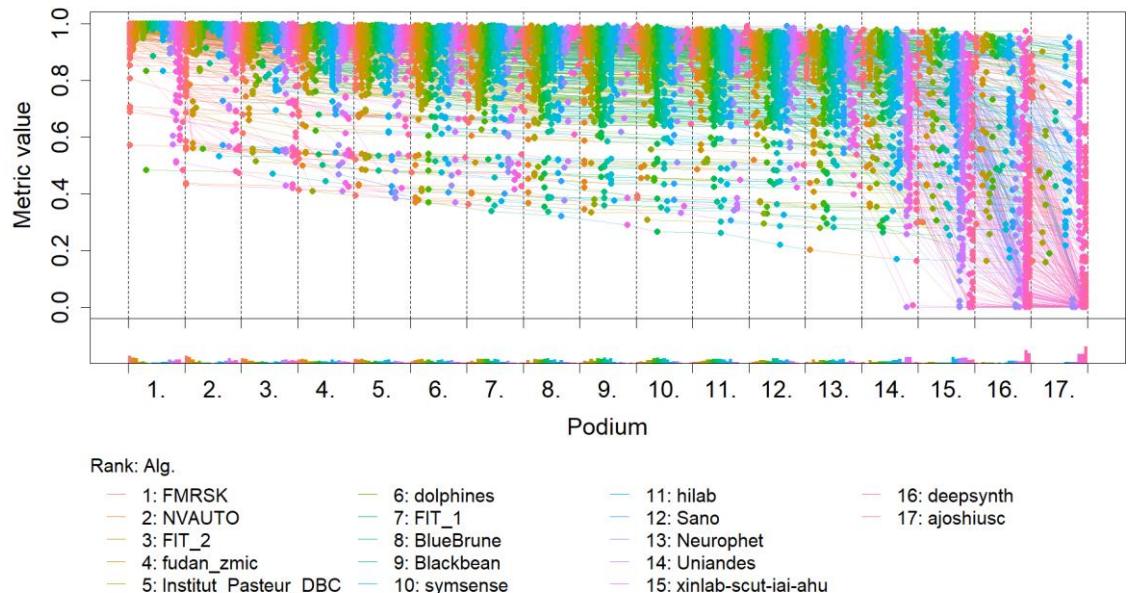
#### Dot- and boxplot

*Dot- and boxplots* for visualizing raw assessment data separately for each algorithm. Boxplots representing descriptive statistics over all cases (median, quartiles and outliers) are combined with horizontally jittered dots representing individual cases.



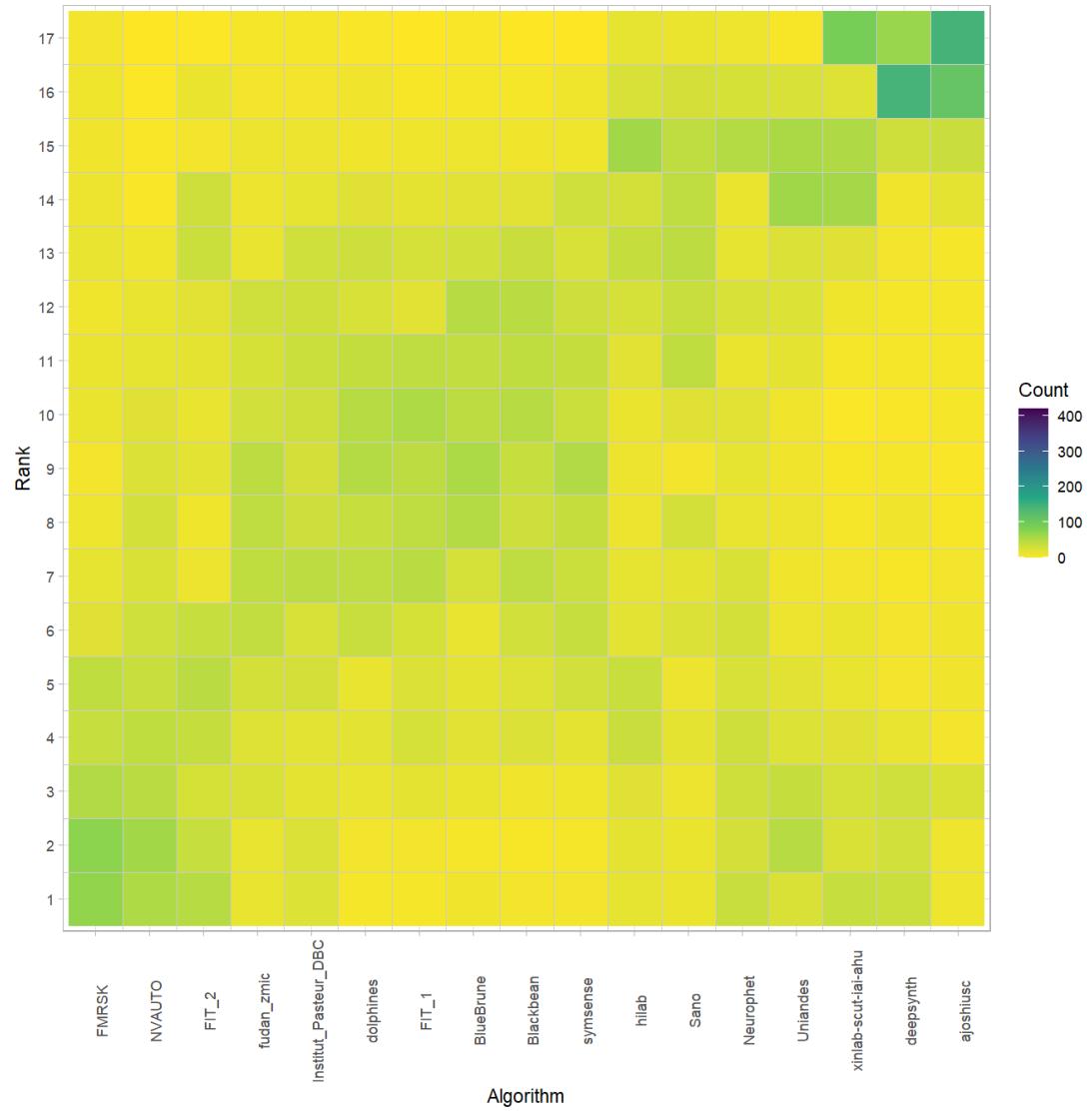
### Podium plot

*Podium plots* (see also Eugster et al., 2008) for visualizing raw assessment data. Upper part (spaghetti plot): Participating algorithms are color-coded, and each colored dot in the plot represents a metric value achieved with the respective algorithm. The actual metric value is encoded by the y-axis. Each podium (here:  $p=17$ ) represents one possible rank, ordered from best (1) to last (here: 17). The assignment of metric values (i.e. colored dots) to one of the podiums is based on the rank that the respective algorithm achieved on the corresponding case. Note that the plot part above each podium place is further subdivided into  $p$  “columns”, where each column represents one participating algorithm (here:  $p = 17$ ). Dots corresponding to identical cases are connected by a line, leading to the shown spaghetti structure. Lower part: Bar charts represent the relative frequency for each algorithm to achieve the rank encoded by the podium place.



## Ranking heatmap

*Ranking heatmaps* for visualizing raw assessment data. Each cell  $(i, A_j)$  shows the absolute frequency of cases in which algorithm  $A_j$  achieved rank  $i$ .

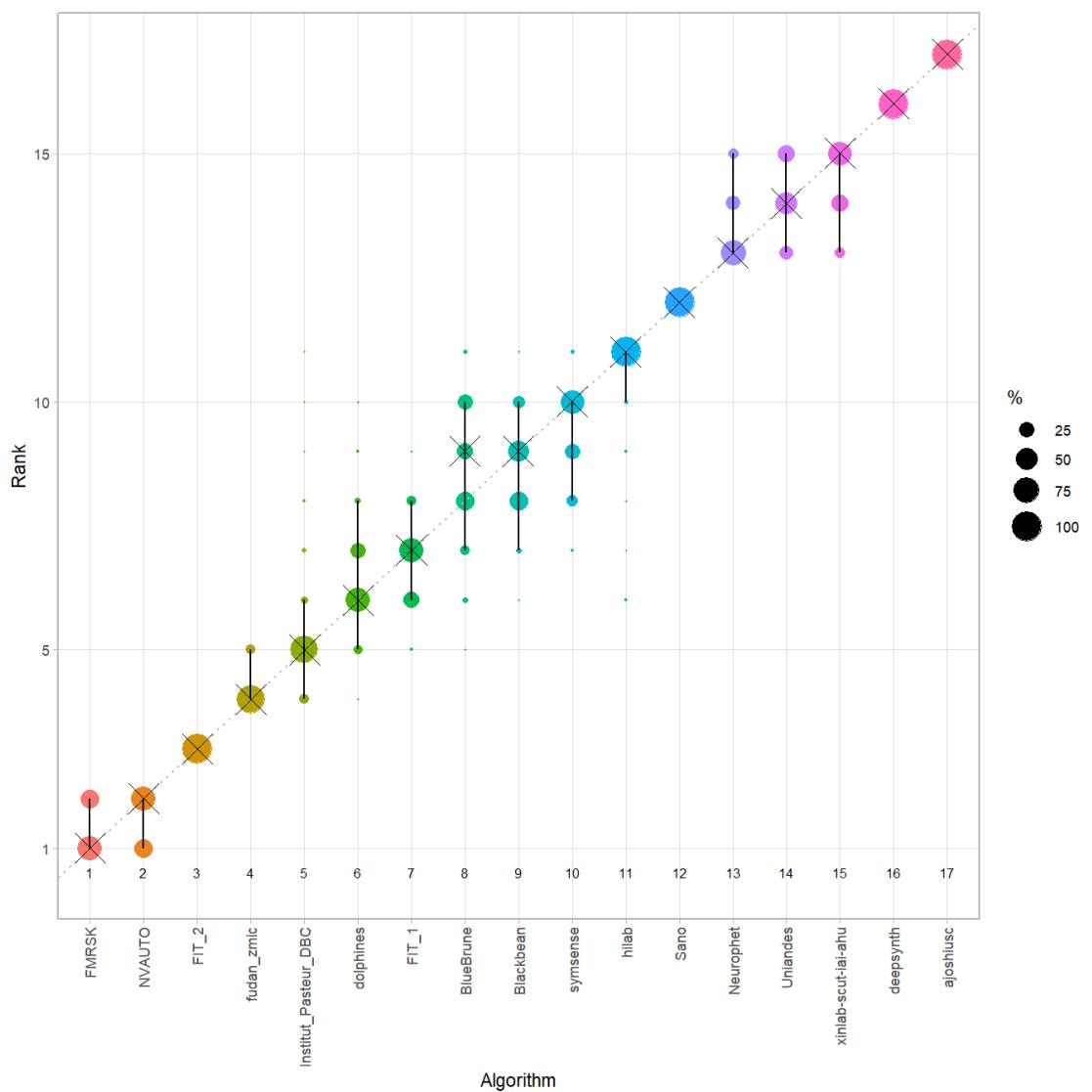


### Visualization of ranking stability

#### *Blob plot for visualizing ranking stability based on bootstrap sampling*

Algorithms are color-coded, and the area of each blob at position ( $A_i$ , rank  $j$ ) is proportional to the relative frequency  $A_i$  achieved rank  $j$  across  $b = 1000$  bootstrap samples. The median rank for each algorithm is indicated by a black cross. 95% bootstrap intervals across bootstrap samples are indicated by black lines.

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.
```

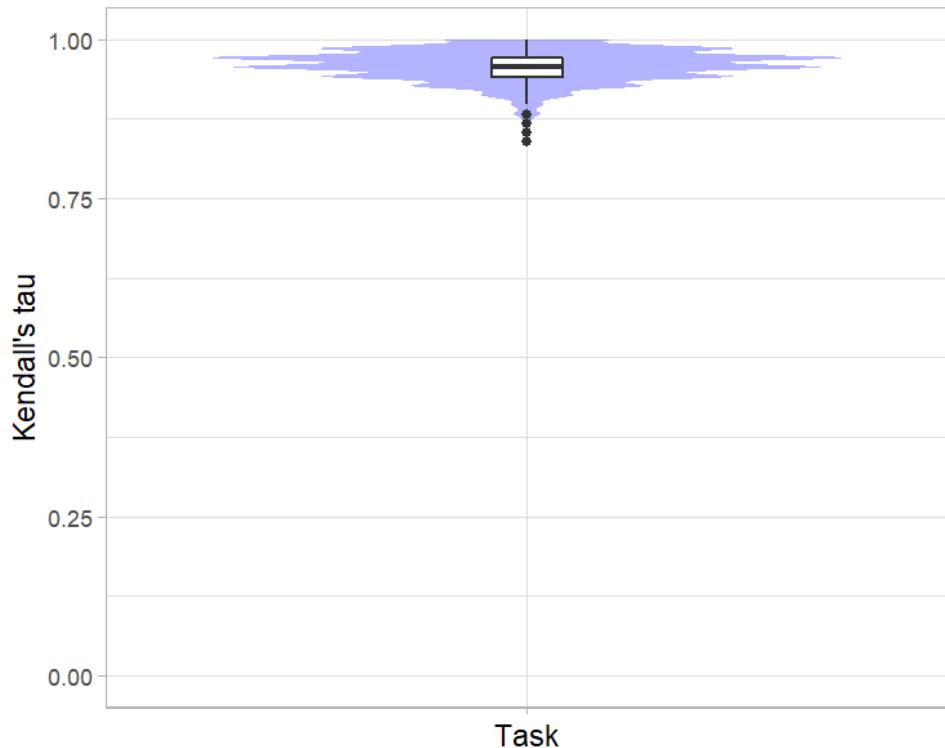


### ***Violin plot for visualizing ranking stability based on bootstrapping***

The ranking list based on the full assessment data is pairwise compared with the ranking lists based on the individual bootstrap samples (here  $b = 1000$  samples). For each pair of rankings, Kendall's  $\tau$  correlation is computed. Kendall's  $\tau$  is a scaled index determining the correlation between the lists. It is computed by evaluating the number of pairwise concordances and discordances between ranking lists and produces values between  $-1$  (for inverted order) and  $1$  (for identical order). A violin plot, which simultaneously depicts a boxplot and a density plot, is generated from the results.

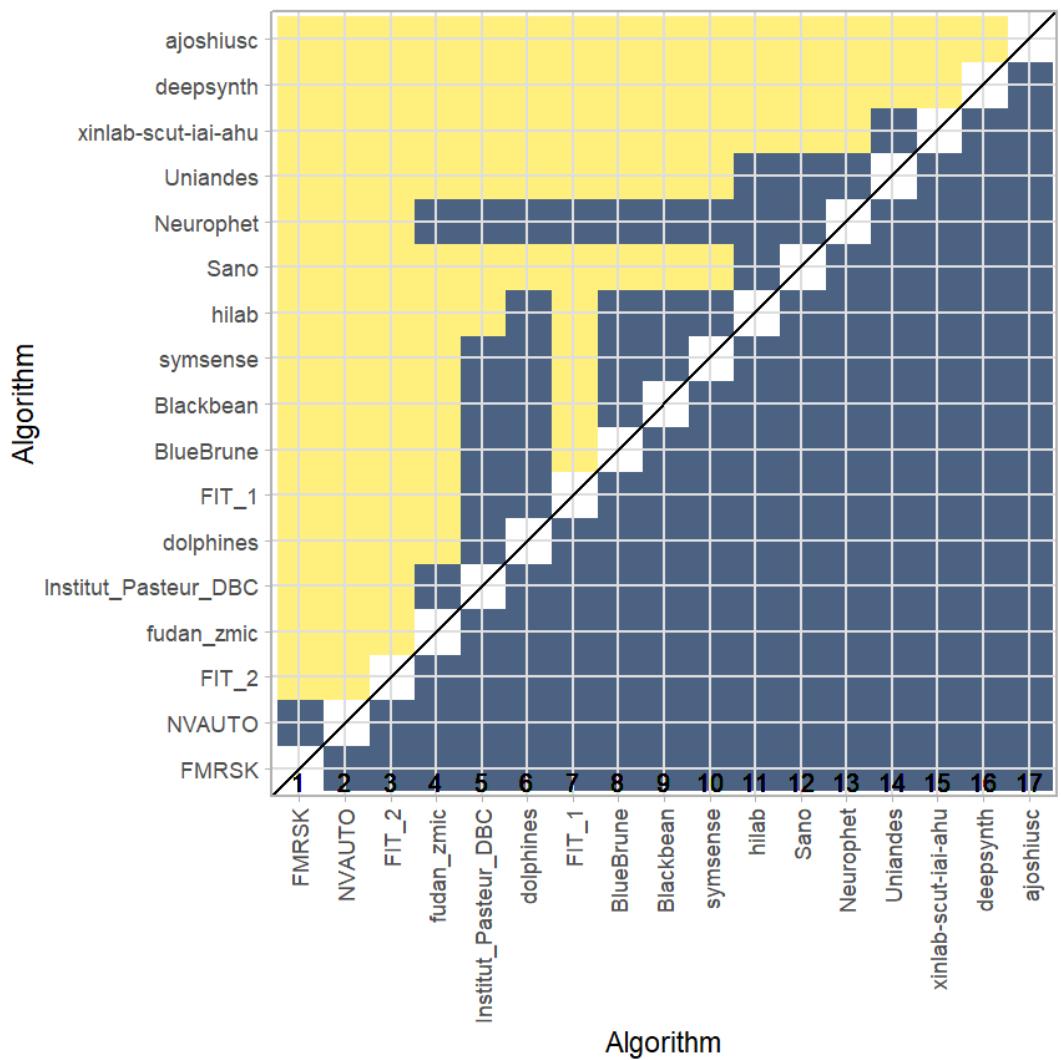
Summary Kendall's tau:

Task	mean	median	q25	q75
dummyTask	0.9582647	0.9558824	0.9411765	0.9705882



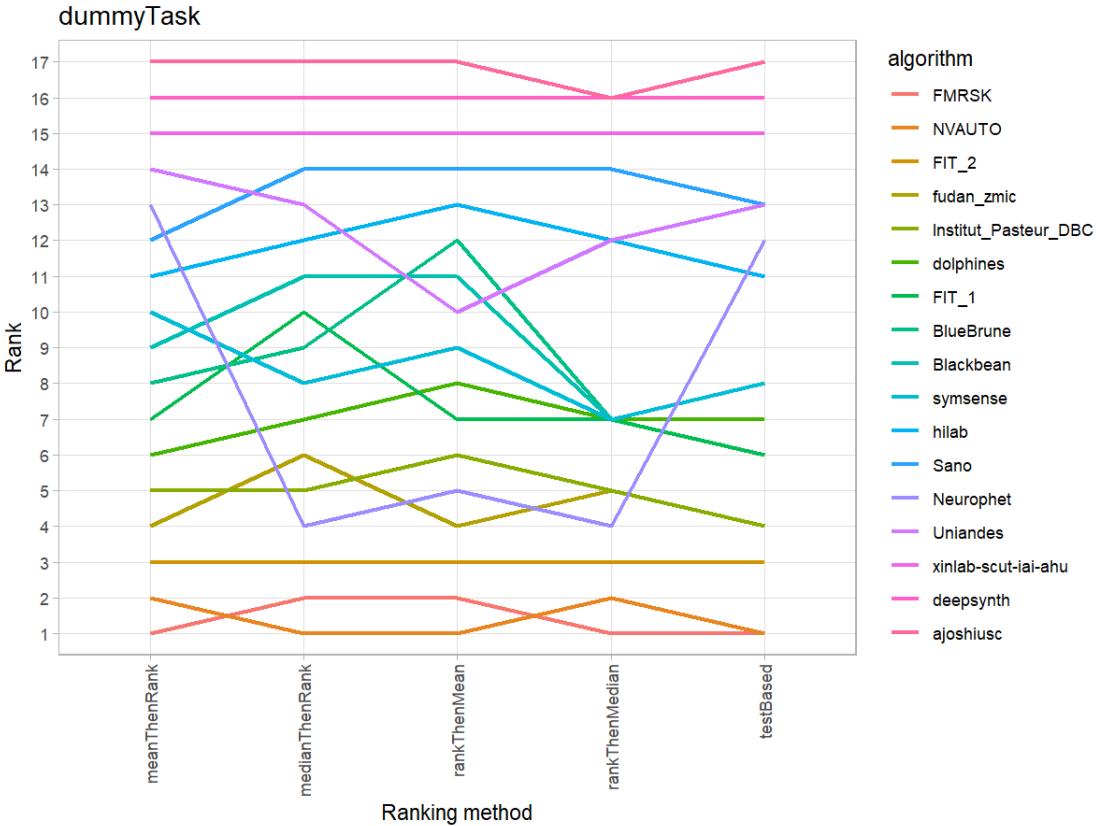
## *Significance maps* for visualizing ranking stability based on statistical significance

*Significance maps* depict incidence matrices of pairwise significant test results for the one-sided Wilcoxon signed rank test at a 5% significance level with adjustment for multiple testing according to Holm. Yellow shading indicates that metric values from the algorithm on the x-axis were significantly superior to those from the algorithm on the y-axis, blue color indicates no significant difference.



### Ranking robustness to ranking methods

*Line plots* for visualizing ranking robustness across different ranking methods. Each algorithm is represented by one colored line. For each ranking method encoded on the x-axis, the height of the line represents the corresponding rank. Horizontal lines indicate identical ranks for all methods.



### 33.3 References

Wiesenfarth, M., Reinke, A., Landman, B.A., Eisenmann, M., Aguilera Saiz, L., Cardoso, M.J., Maier-Hein, L. and Kopp-Schneider, A. Methods and open-source toolkit for analyzing and visualizing challenge results. *Sci Rep* **11**, 2369 (2021). <https://doi.org/10.1038/s41598-021-82017-6>

M. J. A. Eugster, T. Hothorn, and F. Leisch, “Exploratory and inferential analysis of benchmark experiments,” Institut fuer Statistik, Ludwig-Maximilians-Universitaet Muenchen, Germany, Technical Report 30, 2008. [Online]. Available: <http://epub.ub.uni-muenchen.de/4134/>.

## 34 Benchmarking report for Dice Metrics – NiftyMIC Reconstruction Method

created by challengeR v1.0.2  
23 October, 2023

This document presents a systematic report on the benchmark study “Dice Metrics”. Input data comprises raw metric values for all algorithms and cases. Generated plots are:

- Visualization of assessment data: Dot- and boxplot, podium plot and ranking heatmap
- Visualization of ranking stability: Blob plot, violin plot and significance map, line plot

Details can be found in Wiesenfarth et al. (2021).

### 34.1 Ranking

Algorithms within a task are ranked according to the following ranking scheme:  
*aggregate using function (“mean”) then rank*

The analysis is based on 17 algorithms and 343 cases. 0 missing cases have been found in the data set.

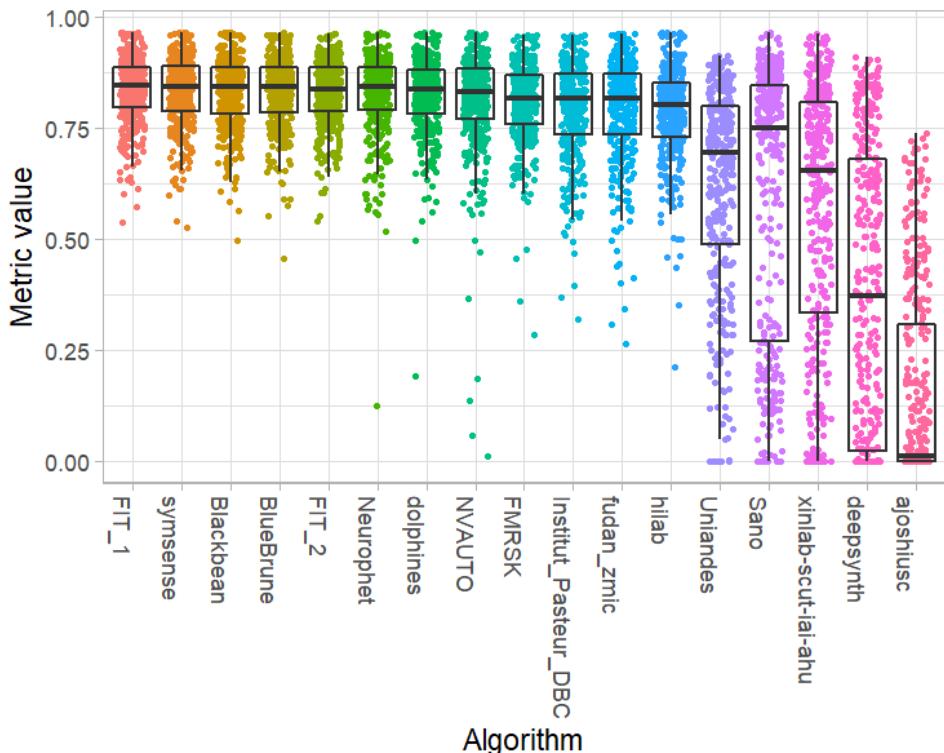
Ranking:

	Dice_m ean	r ank
FIT_1	0.8360724	1
symsense	0.8322206	2
Blackbean	0.8310190	3
BlueBrune	0.8300694	4
FIT_2	0.8299381	5
Neurophet	0.8263272	6
dolphines	0.8246378	7
NVAUTO	0.8089763	8
FMRSK	0.8073792	9
Institut_Pasteur_DBC	0.7931034	10
fudan_zmic	0.7908866	11
hilab	0.7835601	12
Uniandes	0.6160903	13
Sano	0.5992771	14
xinlab-scut-iai-ahu	0.5690613	15
deepsynth	0.3779188	16
ajoshiusc	0.1597532	17

### 34.2 Visualization of raw assessment data

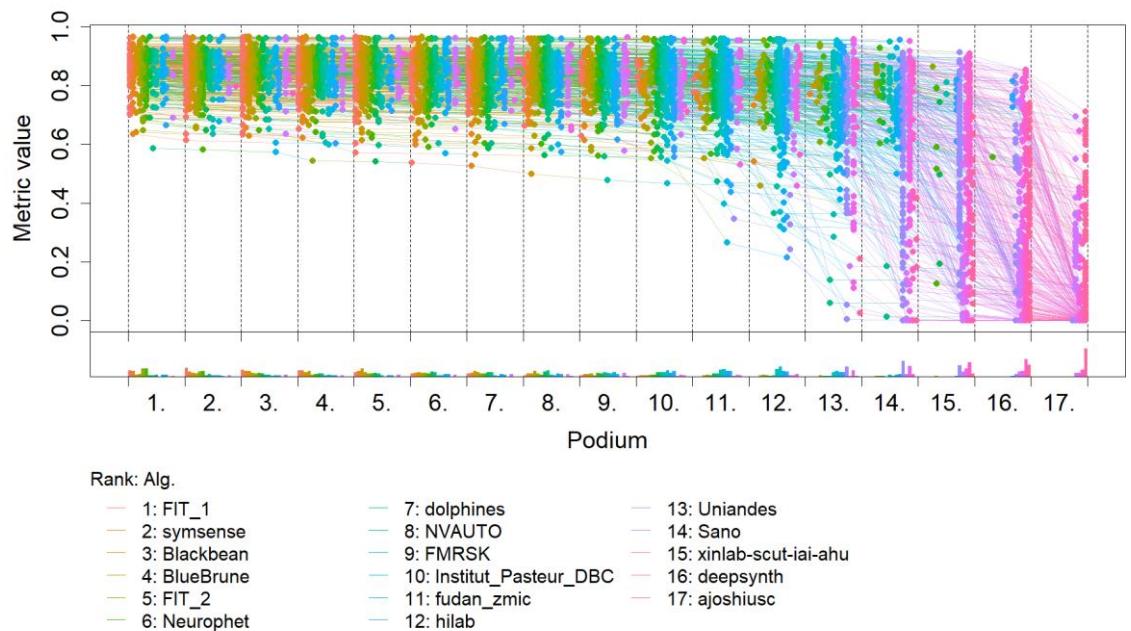
#### Dot- and boxplot

*Dot- and boxplots* for visualizing raw assessment data separately for each algorithm. Boxplots representing descriptive statistics over all cases (median, quartiles and outliers) are combined with horizontally jittered dots representing individual cases.



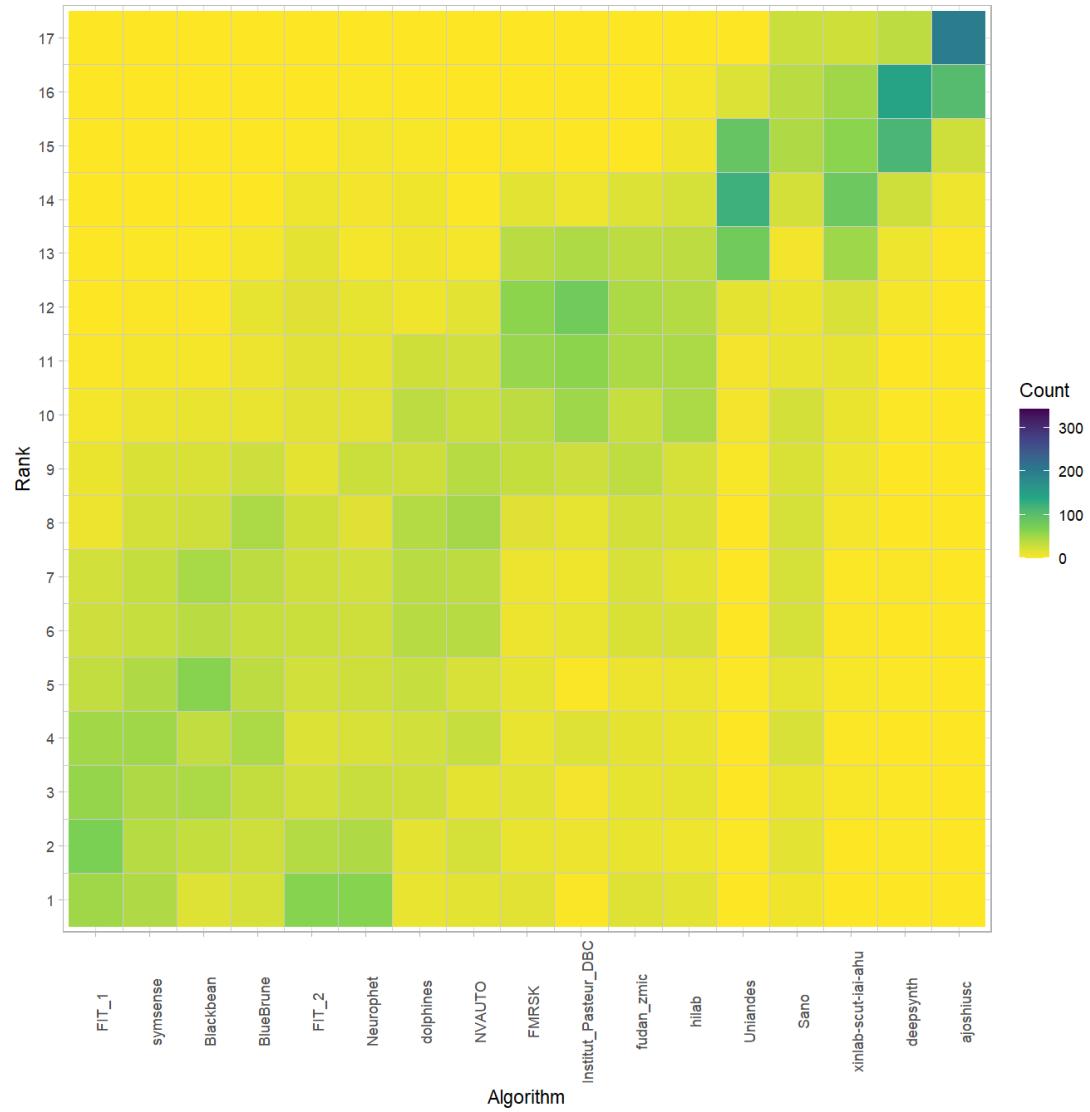
### Podium plot

*Podium plots* (see also Eugster et al., 2008) for visualizing raw assessment data. Upper part (spaghetti plot): Participating algorithms are color-coded, and each colored dot in the plot represents a metric value achieved with the respective algorithm. The actual metric value is encoded by the y-axis. Each podium (here:  $p=17$ ) represents one possible rank, ordered from best (1) to last (here: 17). The assignment of metric values (i.e. colored dots) to one of the podiums is based on the rank that the respective algorithm achieved on the corresponding case. Note that the plot part above each podium place is further subdivided into  $p$  “columns”, where each column represents one participating algorithm (here:  $p = 17$ ). Dots corresponding to identical cases are connected by a line, leading to the shown spaghetti structure. Lower part: Bar charts represent the relative frequency for each algorithm to achieve the rank encoded by the podium place.



## Ranking heatmap

*Ranking heatmaps* for visualizing raw assessment data. Each cell  $(i, A_j)$  shows the absolute frequency of cases in which algorithm  $A_j$  achieved rank  $i$ .

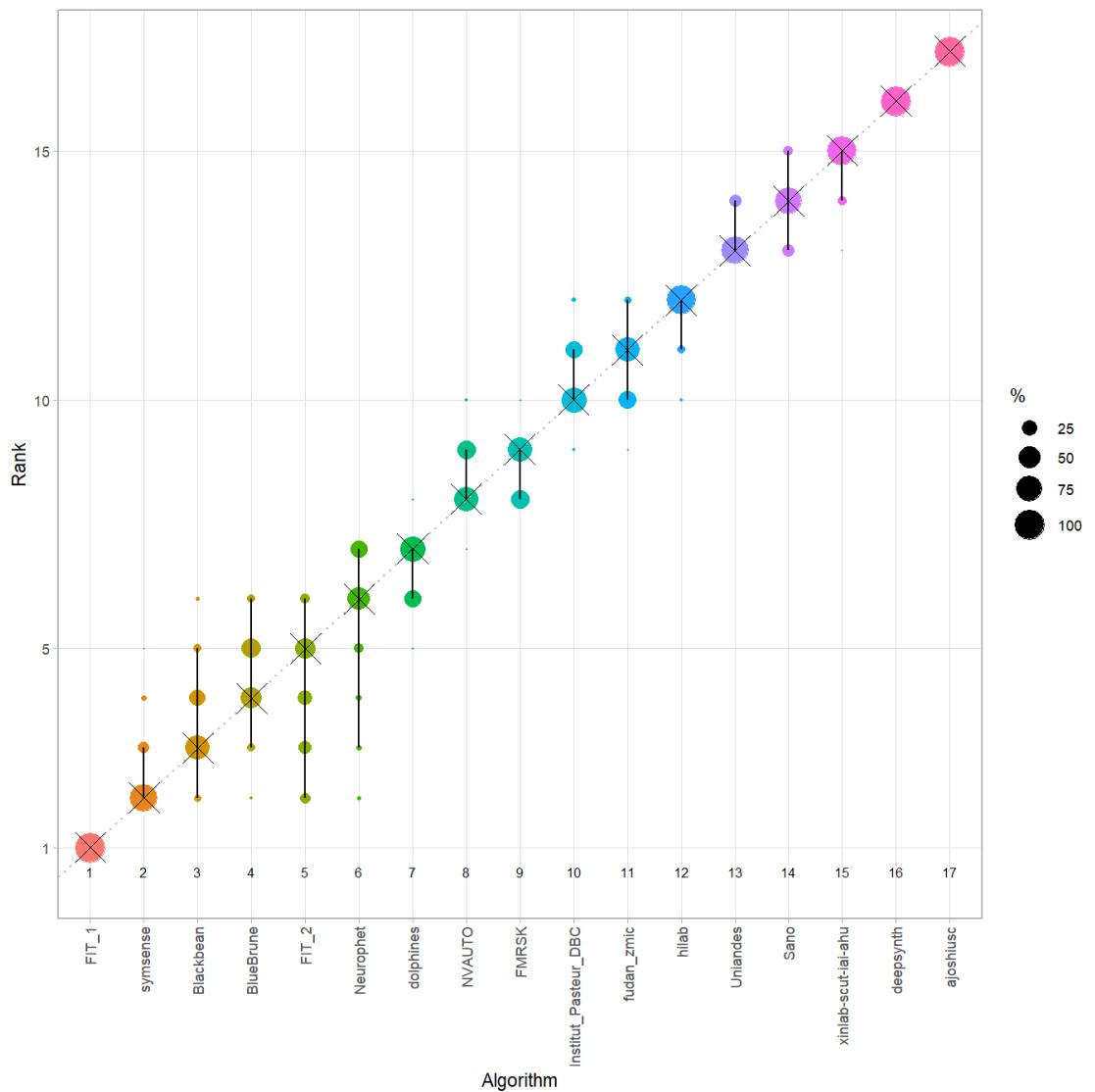


### Visualization of ranking stability

#### *Blob plot for visualizing ranking stability based on bootstrap sampling*

Algorithms are color-coded, and the area of each blob at position ( $A_i, \text{rank } j$ ) is proportional to the relative frequency  $A_i$  achieved rank  $j$  across  $b = 1000$  bootstrap samples. The median rank for each algorithm is indicated by a black cross. 95% bootstrap intervals across bootstrap samples are indicated by black lines.

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.
```

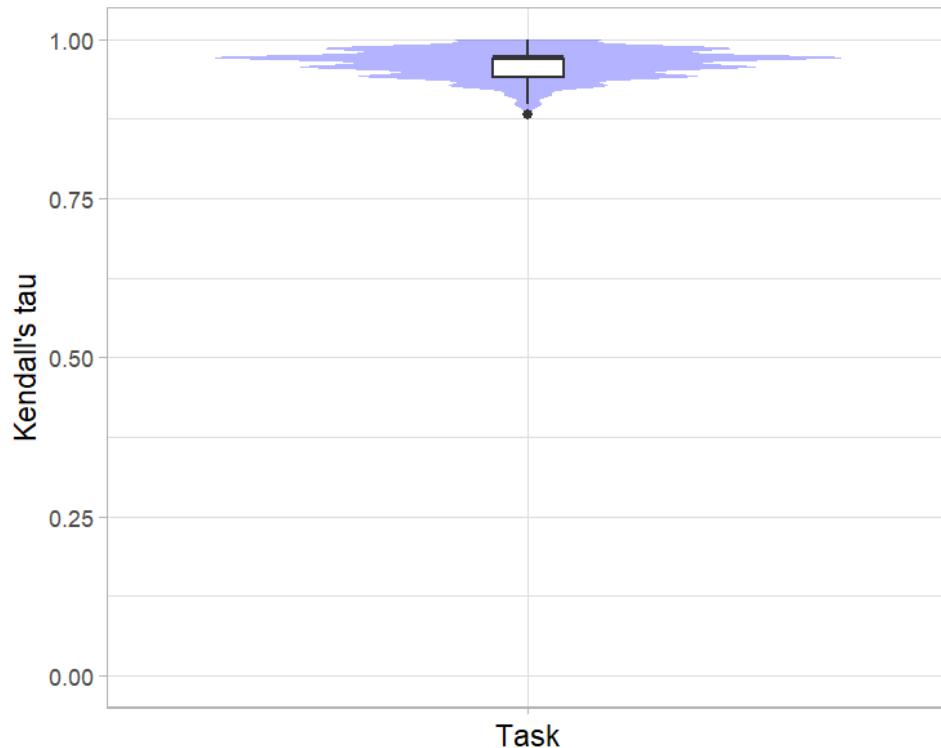


### ***Violin plot for visualizing ranking stability based on bootstrapping***

The ranking list based on the full assessment data is pairwise compared with the ranking lists based on the individual bootstrap samples (here  $b = 1000$  samples). For each pair of rankings, Kendall's  $\tau$  correlation is computed. Kendall's  $\tau$  is a scaled index determining the correlation between the lists. It is computed by evaluating the number of pairwise concordances and discordances between ranking lists and produces values between  $-1$  (for inverted order) and  $1$  (for identical order). A violin plot, which simultaneously depicts a boxplot and a density plot, is generated from the results.

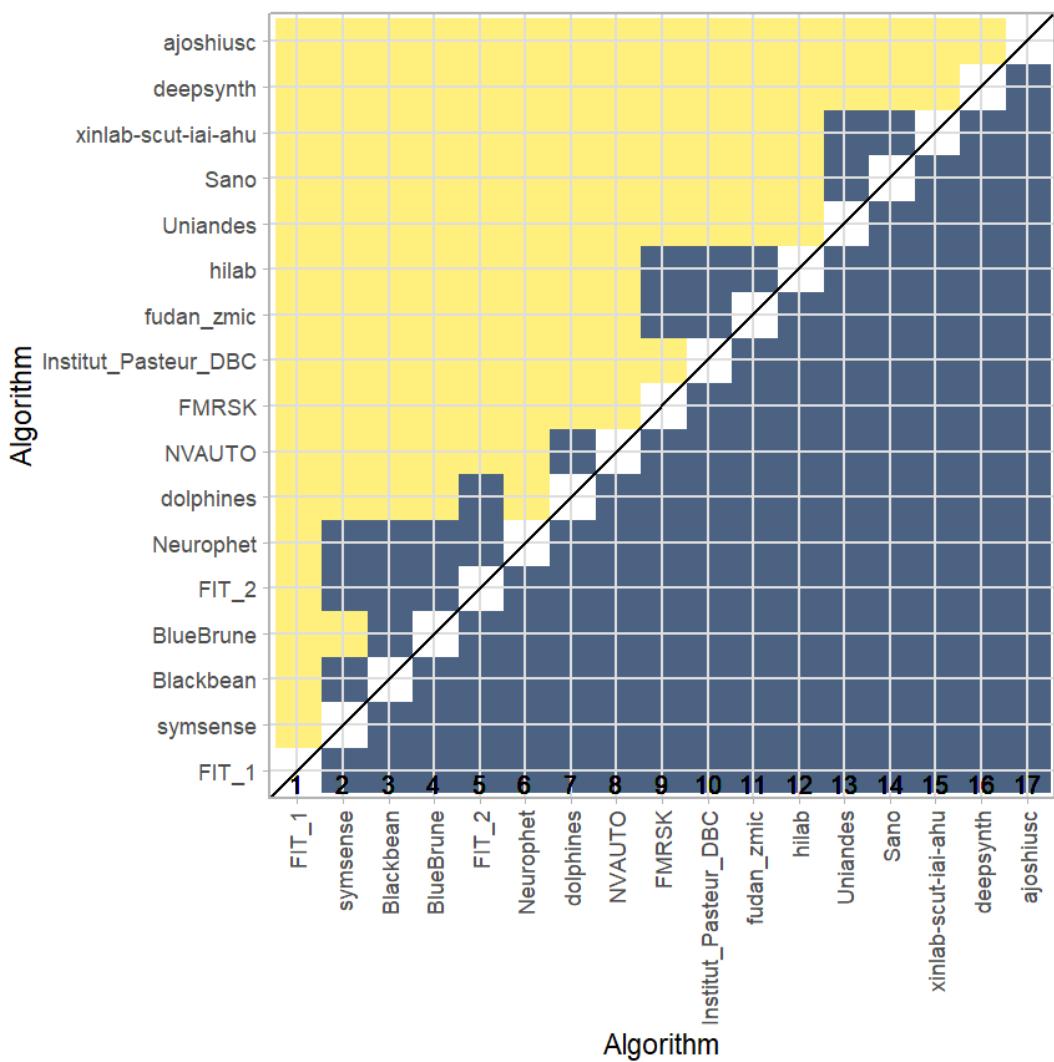
Summary Kendall's tau:

Task	mean	median	q25	q75
dummyTask	0.9621029	0.9705882	0.9411765	0.9705882



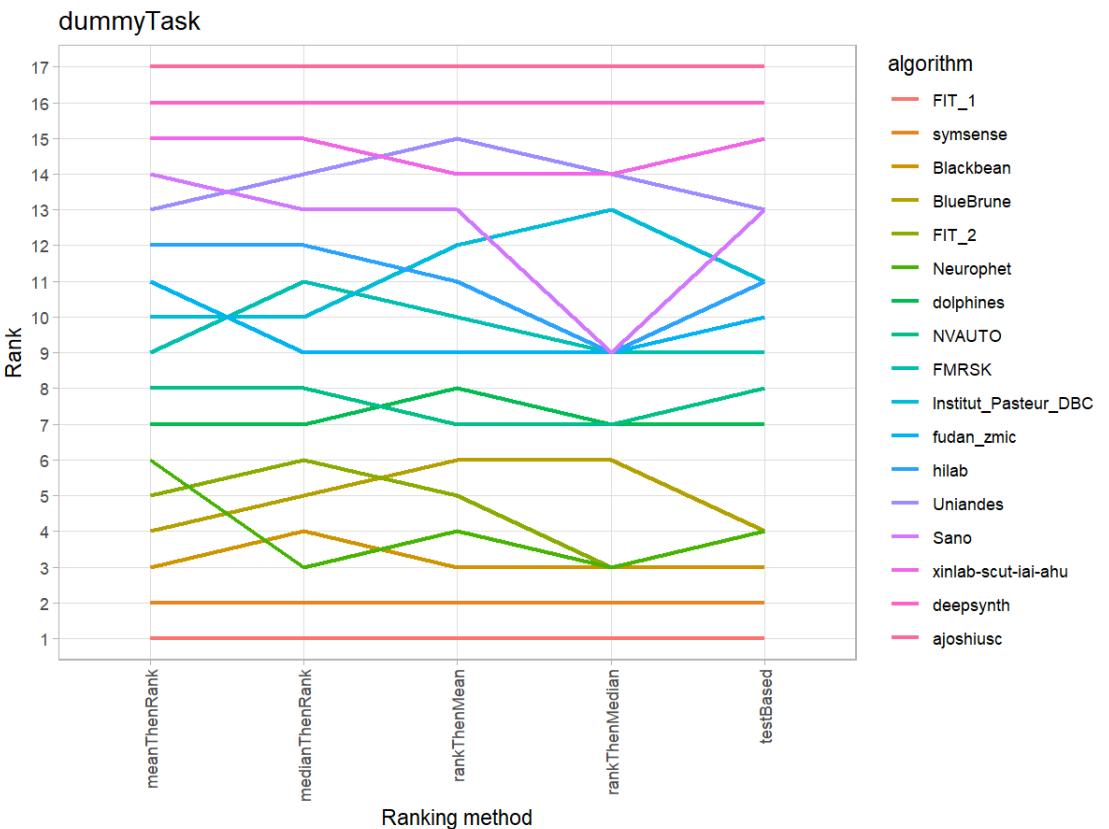
## *Significance maps* for visualizing ranking stability based on statistical significance

*Significance maps* depict incidence matrices of pairwise significant test results for the one-sided Wilcoxon signed rank test at a 5% significance level with adjustment for multiple testing according to Holm. Yellow shading indicates that metric values from the algorithm on the x-axis were significantly superior to those from the algorithm on the y-axis, blue color indicates no significant difference.



### Ranking robustness to ranking methods

*Line plots* for visualizing ranking robustness across different ranking methods. Each algorithm is represented by one colored line. For each ranking method encoded on the x-axis, the height of the line represents the corresponding rank. Horizontal lines indicate identical ranks for all methods.



### 34.3 References

Wiesenfarth, M., Reinke, A., Landman, B.A., Eisenmann, M., Aguilera Saiz, L., Cardoso, M.J., Maier-Hein, L. and Kopp-Schneider, A. Methods and open-source toolkit for analyzing and visualizing challenge results. *Sci Rep* **11**, 2369 (2021). <https://doi.org/10.1038/s41598-021-82017-6>

M. J. A. Eugster, T. Hothorn, and F. Leisch, “Exploratory and inferential analysis of benchmark experiments,” Institut fuer Statistik, Ludwig-Maximilians-Universitaet Muenchen, Germany, Technical Report 30, 2008. [Online]. Available: <http://epub.ub.uni-muenchen.de/4134/>.

## 35 Benchmarking report for Hausdorff Metrics – NiftyMIC Reconstruction Method

created by challengeR v1.0.2  
23 October, 2023

This document presents a systematic report on the benchmark study “Hausdorff Metrics”. Input data comprises raw metric values for all algorithms and cases. Generated plots are:

- Visualization of assessment data: Dot- and boxplot, podium plot and ranking heatmap
- Visualization of ranking stability: Blob plot, violin plot and significance map, line plot

Details can be found in Wiesenfarth et al. (2021).

### 35.1 Ranking

Algorithms within a task are ranked according to the following ranking scheme:  
*aggregate using function (“mean”) then rank*

The analysis is based on 17 algorithms and 343 cases. 0 missing cases have been found in the data set.

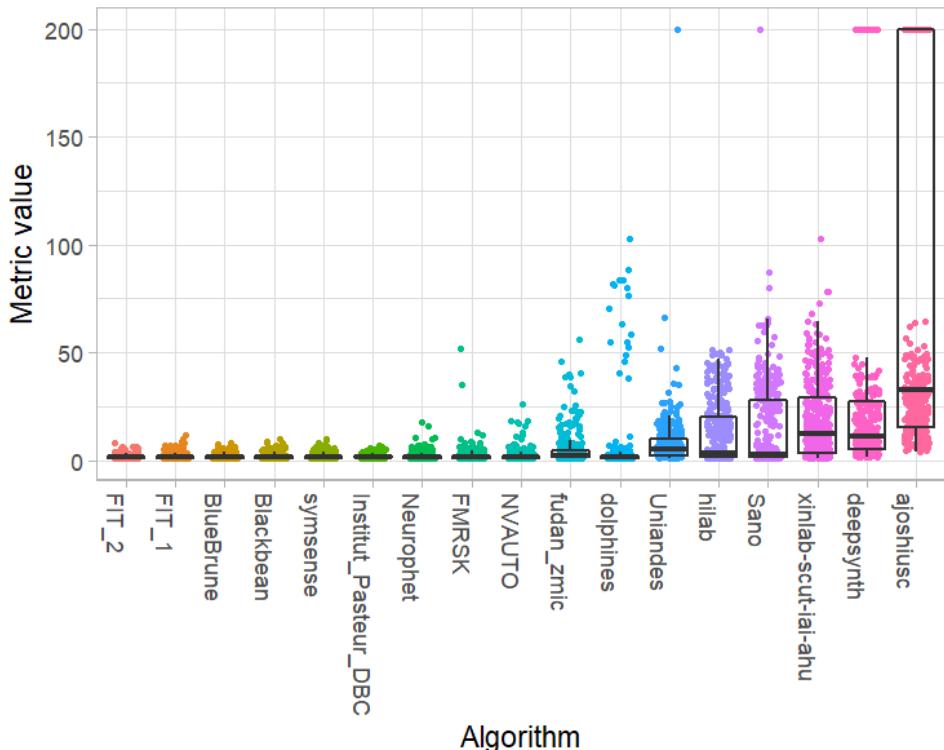
Ranking:

	Hausdorff_mean	rank
FIT_2	1.729183	1
FIT_1	1.892675	2
BlueBrune	1.902297	3
Blackbean	1.964173	4
symsense	1.976255	5
Institut_Pasteur_DBC	1.995587	6
Neurophet	2.009161	7
FMRSK	2.414913	8
NVAUTO	2.422028	9
fudan_zmic	4.905231	10
dolphines	5.304687	11
Uniandes	8.746812	12
hilab	12.423486	13
Sano	15.082354	14
xinlab-scut-iai-ahu	18.263125	15
deepsynth	38.416113	16
ajoshiusc	76.100706	17

### 35.2 Visualization of raw assessment data

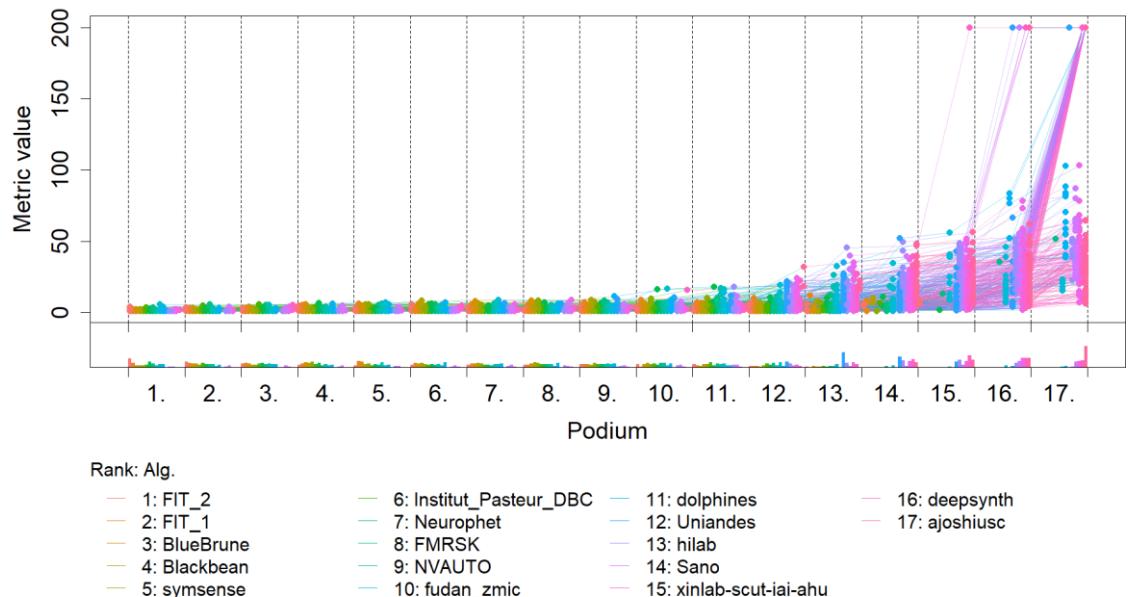
#### Dot- and boxplot

*Dot- and boxplots* for visualizing raw assessment data separately for each algorithm. Boxplots representing descriptive statistics over all cases (median, quartiles and outliers) are combined with horizontally jittered dots representing individual cases.



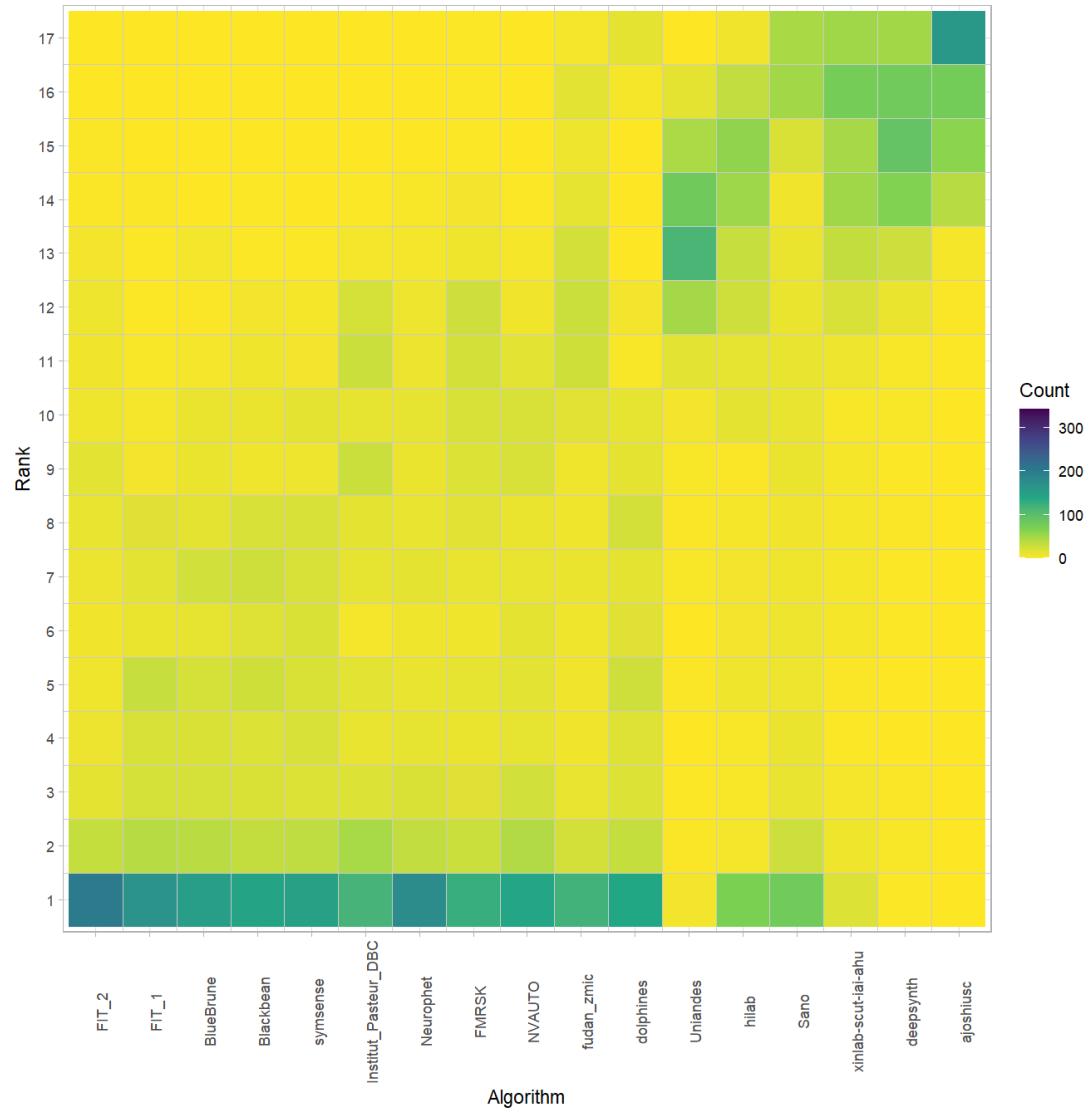
### Podium plot

*Podium plots* (see also Eugster et al., 2008) for visualizing raw assessment data. Upper part (spaghetti plot): Participating algorithms are color-coded, and each colored dot in the plot represents a metric value achieved with the respective algorithm. The actual metric value is encoded by the y-axis. Each podium (here:  $p=17$ ) represents one possible rank, ordered from best (1) to last (here: 17). The assignment of metric values (i.e. colored dots) to one of the podiums is based on the rank that the respective algorithm achieved on the corresponding case. Note that the plot part above each podium place is further subdivided into  $p$  “columns”, where each column represents one participating algorithm (here:  $p = 17$ ). Dots corresponding to identical cases are connected by a line, leading to the shown spaghetti structure. Lower part: Bar charts represent the relative frequency for each algorithm to achieve the rank encoded by the podium place.



## Ranking heatmap

*Ranking heatmaps* for visualizing raw assessment data. Each cell  $(i, A_j)$  shows the absolute frequency of cases in which algorithm  $A_j$  achieved rank  $i$ .

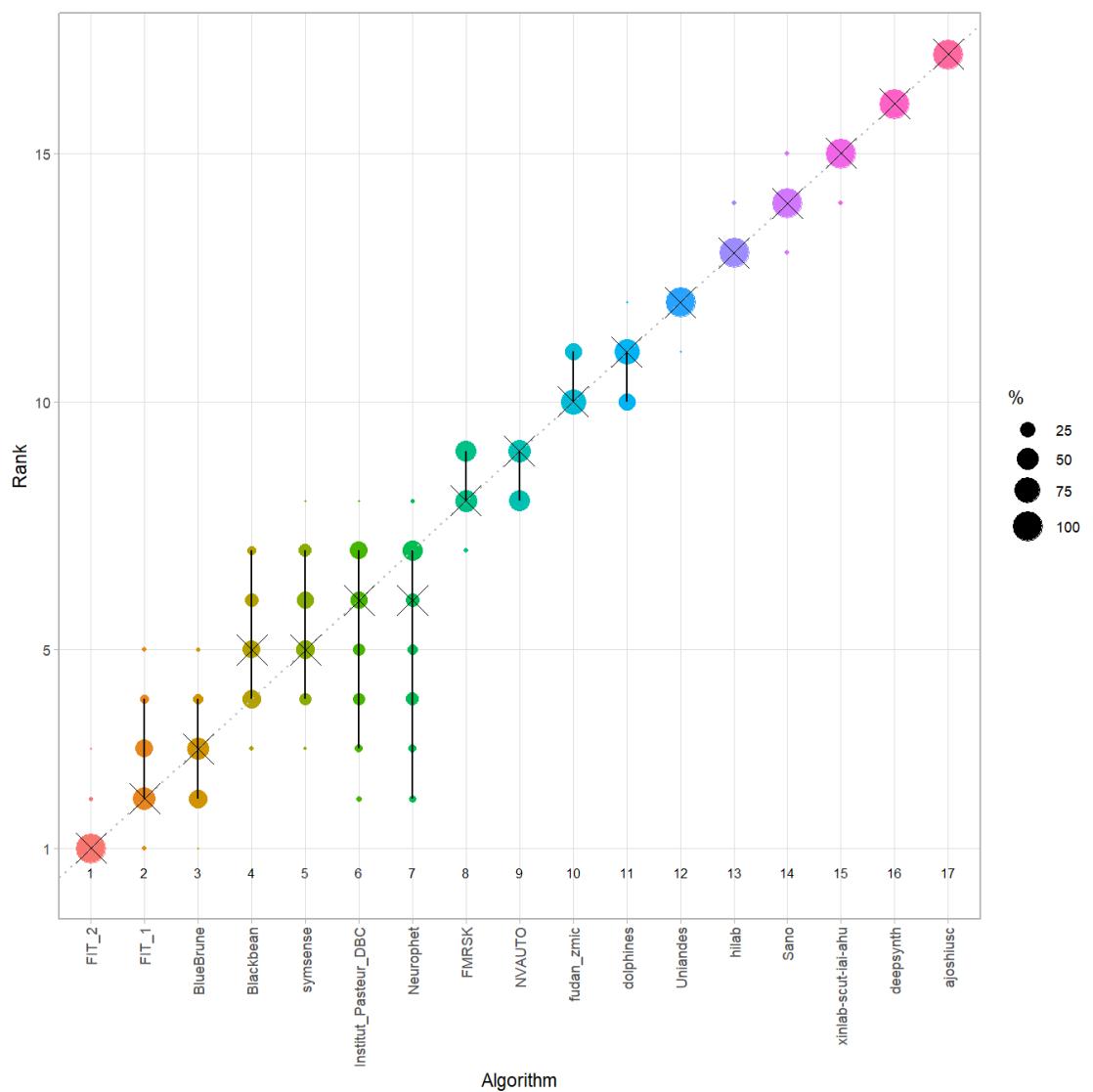


### Visualization of ranking stability

#### *Blob plot for visualizing ranking stability based on bootstrap sampling*

Algorithms are color-coded, and the area of each blob at position ( $A_i, \text{rank } j$ ) is proportional to the relative frequency  $A_i$  achieved rank  $j$  across  $b = 1000$  bootstrap samples. The median rank for each algorithm is indicated by a black cross. 95% bootstrap intervals across bootstrap samples are indicated by black lines.

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> =
## "none")` instead.
```

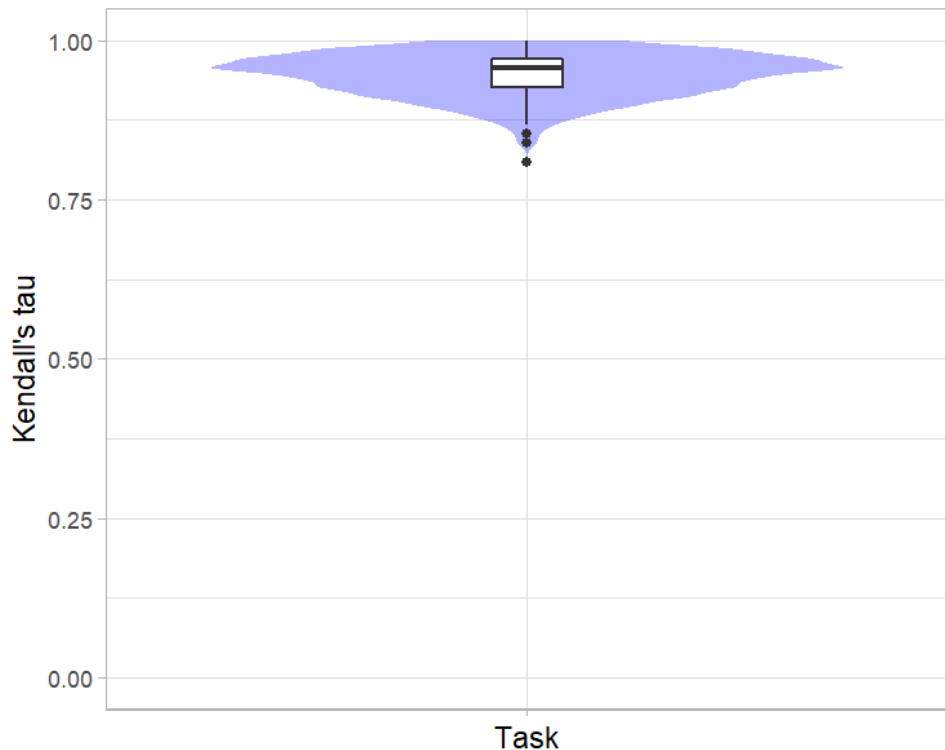


### ***Violin plot for visualizing ranking stability based on bootstrapping***

The ranking list based on the full assessment data is pairwise compared with the ranking lists based on the individual bootstrap samples (here  $b = 1000$  samples). For each pair of rankings, Kendall's  $\tau$  correlation is computed. Kendall's  $\tau$  is a scaled index determining the correlation between the lists. It is computed by evaluating the number of pairwise concordances and discordances between ranking lists and produces values between  $-1$  (for inverted order) and  $1$  (for identical order). A violin plot, which simultaneously depicts a boxplot and a density plot, is generated from the results.

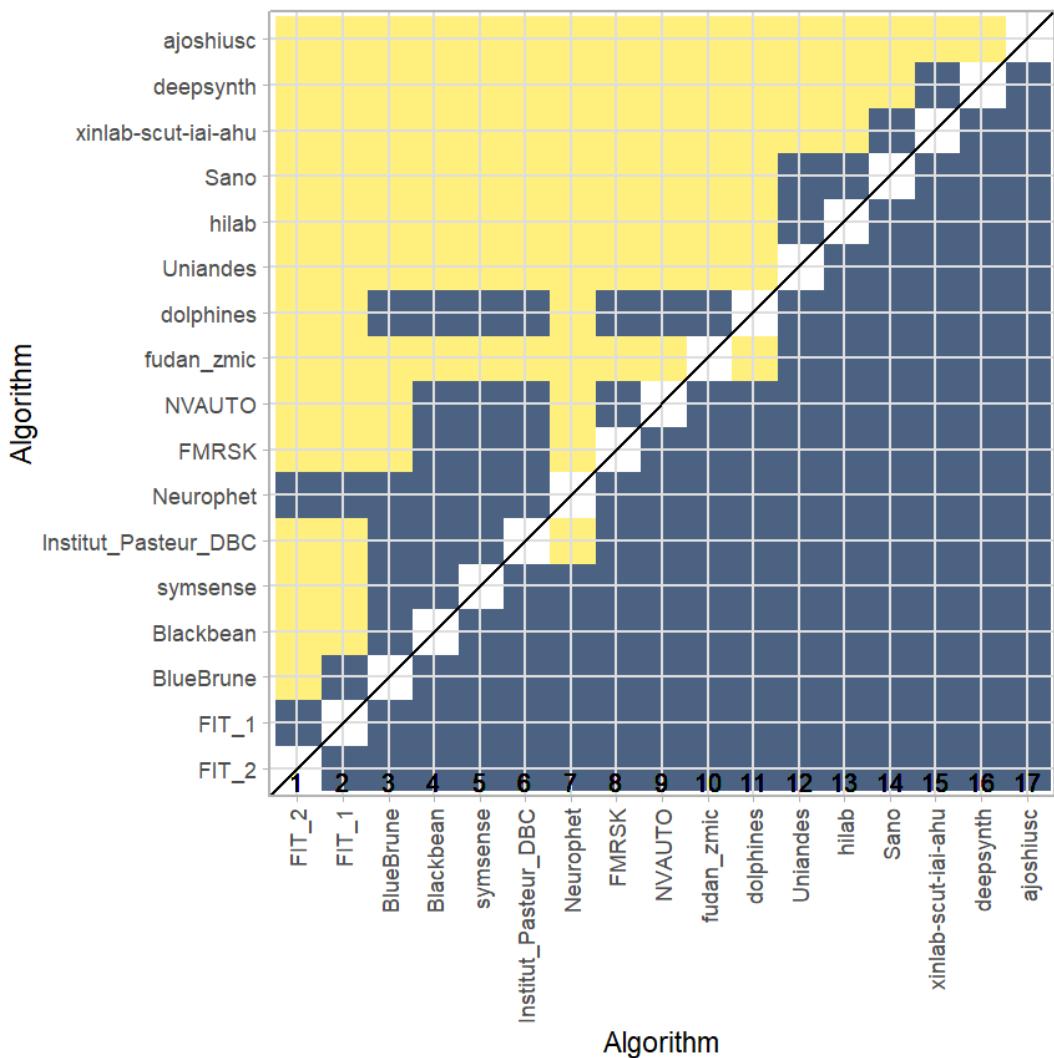
Summary Kendall's tau:

Task	mean	median	q25	q75
dummyTask	0.9453088	0.9558824	0.9264706	0.9705882



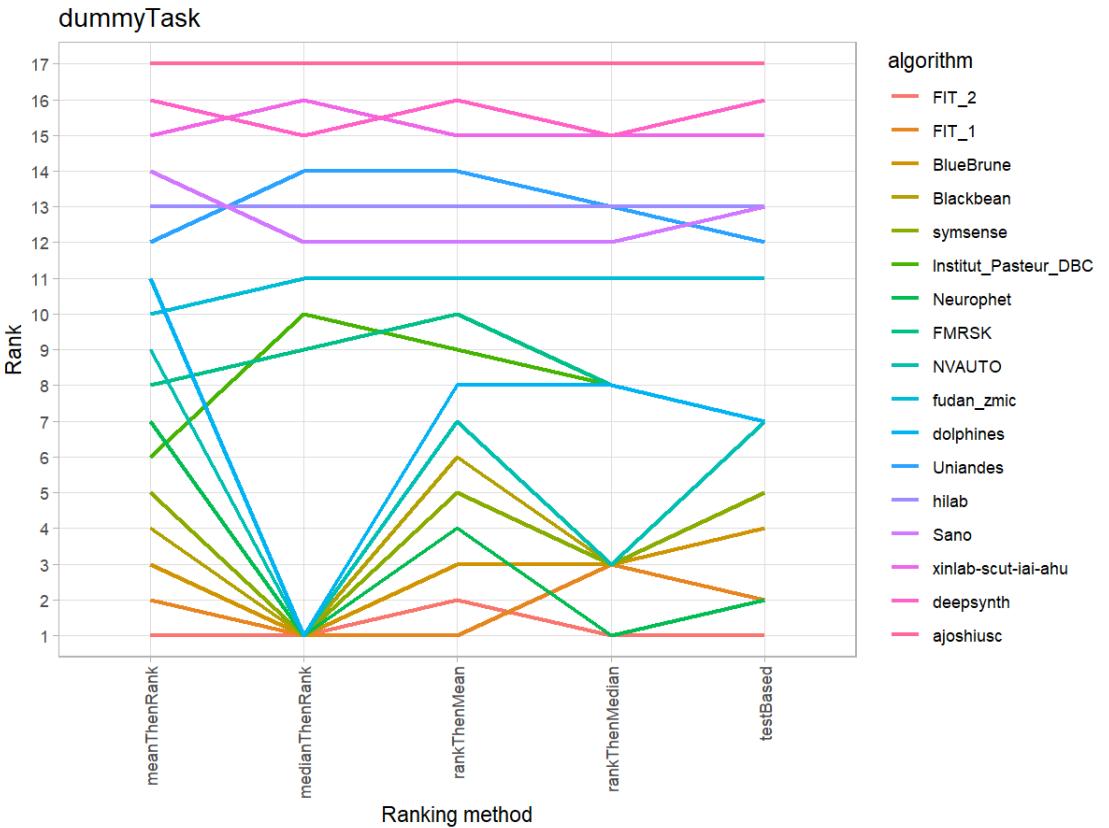
## *Significance maps* for visualizing ranking stability based on statistical significance

*Significance maps* depict incidence matrices of pairwise significant test results for the one-sided Wilcoxon signed rank test at a 5% significance level with adjustment for multiple testing according to Holm. Yellow shading indicates that metric values from the algorithm on the x-axis were significantly superior to those from the algorithm on the y-axis, blue color indicates no significant difference.



### Ranking robustness to ranking methods

*Line plots* for visualizing ranking robustness across different ranking methods. Each algorithm is represented by one colored line. For each ranking method encoded on the x-axis, the height of the line represents the corresponding rank. Horizontal lines indicate identical ranks for all methods.



### 35.3 References

Wiesenfarth, M., Reinke, A., Landman, B.A., Eisenmann, M., Aguilera Saiz, L., Cardoso, M.J., Maier-Hein, L. and Kopp-Schneider, A. Methods and open-source toolkit for analyzing and visualizing challenge results. *Sci Rep* **11**, 2369 (2021). <https://doi.org/10.1038/s41598-021-82017-6>

M. J. A. Eugster, T. Hothorn, and F. Leisch, “Exploratory and inferential analysis of benchmark experiments,” Institut fuer Statistik, Ludwig-Maximilians-Universitaet Muenchen, Germany, Technical Report 30, 2008. [Online]. Available: <http://epub.ub.uni-muenchen.de/4134/>.

## 36 Benchmarking report for Volume Similarity Metrics – NiftyMIC Reconstruction Method

created by challengeR v1.0.2  
23 October, 2023

This document presents a systematic report on the benchmark study “Volume Similarity Metrics”. Input data comprises raw metric values for all algorithms and cases. Generated plots are:

- Visualization of assessment data: Dot- and boxplot, podium plot and ranking heatmap
- Visualization of ranking stability: Blob plot, violin plot and significance map, line plot

Details can be found in Wiesenfarth et al. (2021).

### 36.1 Ranking

Algorithms within a task are ranked according to the following ranking scheme:  
*aggregate using function (“mean”) then rank*

The analysis is based on 17 algorithms and 343 cases. 0 missing cases have been found in the data set.

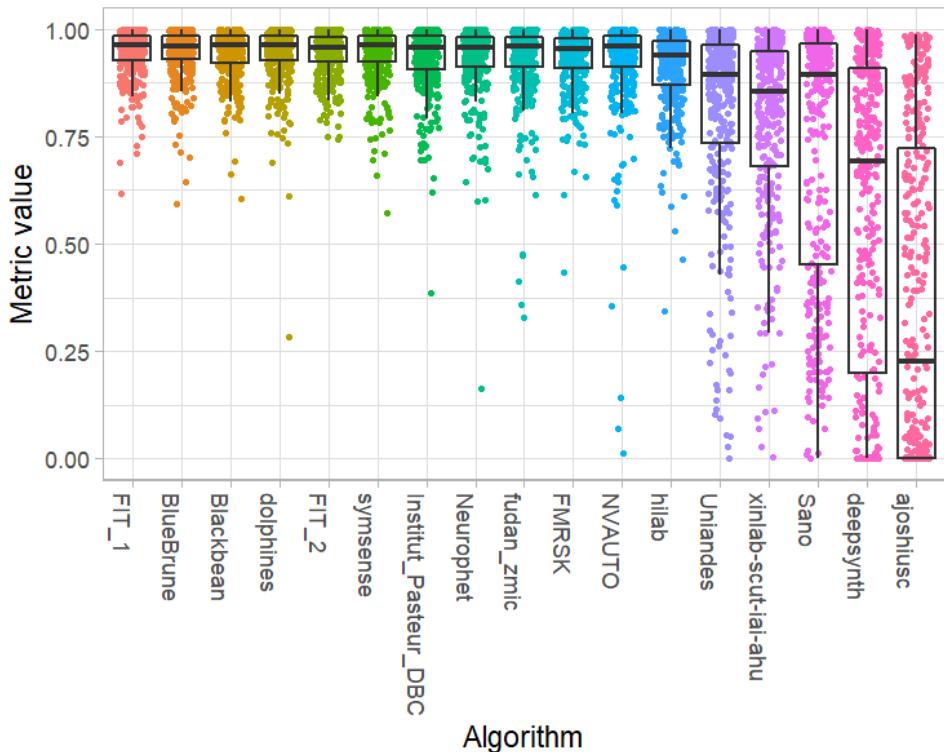
Ranking:

	Volume_Similarity_mean	rank
FIT_1	0.9458558	1
BlueBrune	0.9439420	2
Blackbean	0.9432667	3
dolphines	0.9428963	4
FIT_2	0.9424504	5
symsense	0.9421589	6
Institut_Pasteur_DBC	0.9327780	7
Neurophet	0.9316737	8
fudan_zmic	0.9300533	9
FMRSK	0.9294705	10
NVAUTO	0.9245963	11
hilab	0.9086920	12
Uniandes	0.8082391	13
xinlab-scut-iai-ahu	0.7885668	14
Sano	0.7280521	15
deepsynth	0.5758699	16
ajoshiusc	0.3599621	17

### 36.2 Visualization of raw assessment data

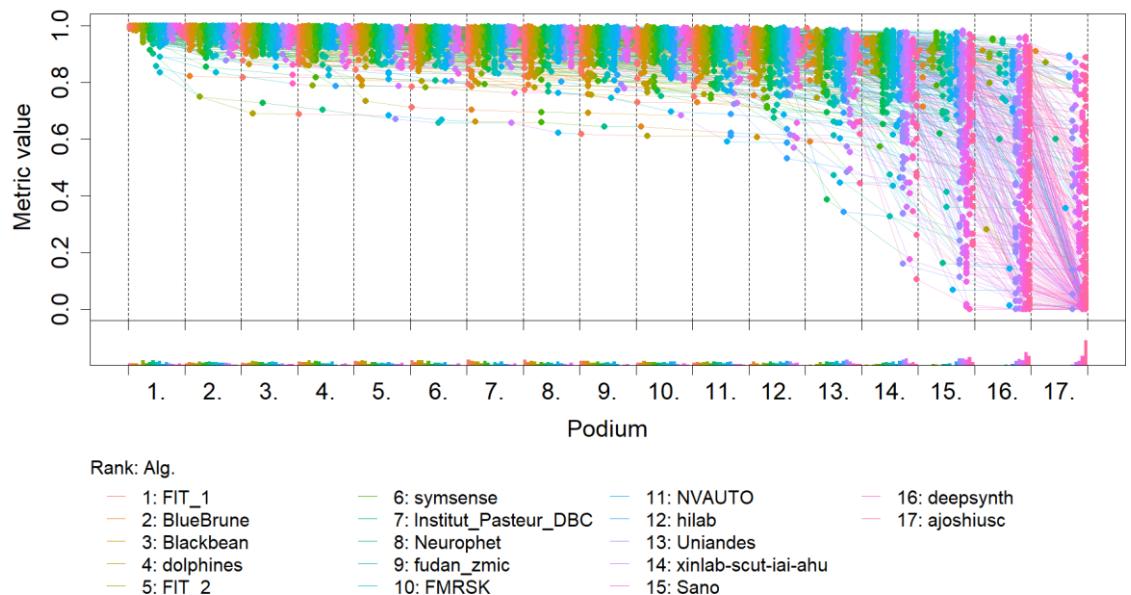
#### Dot- and boxplot

*Dot- and boxplots* for visualizing raw assessment data separately for each algorithm. Boxplots representing descriptive statistics over all cases (median, quartiles and outliers) are combined with horizontally jittered dots representing individual cases.



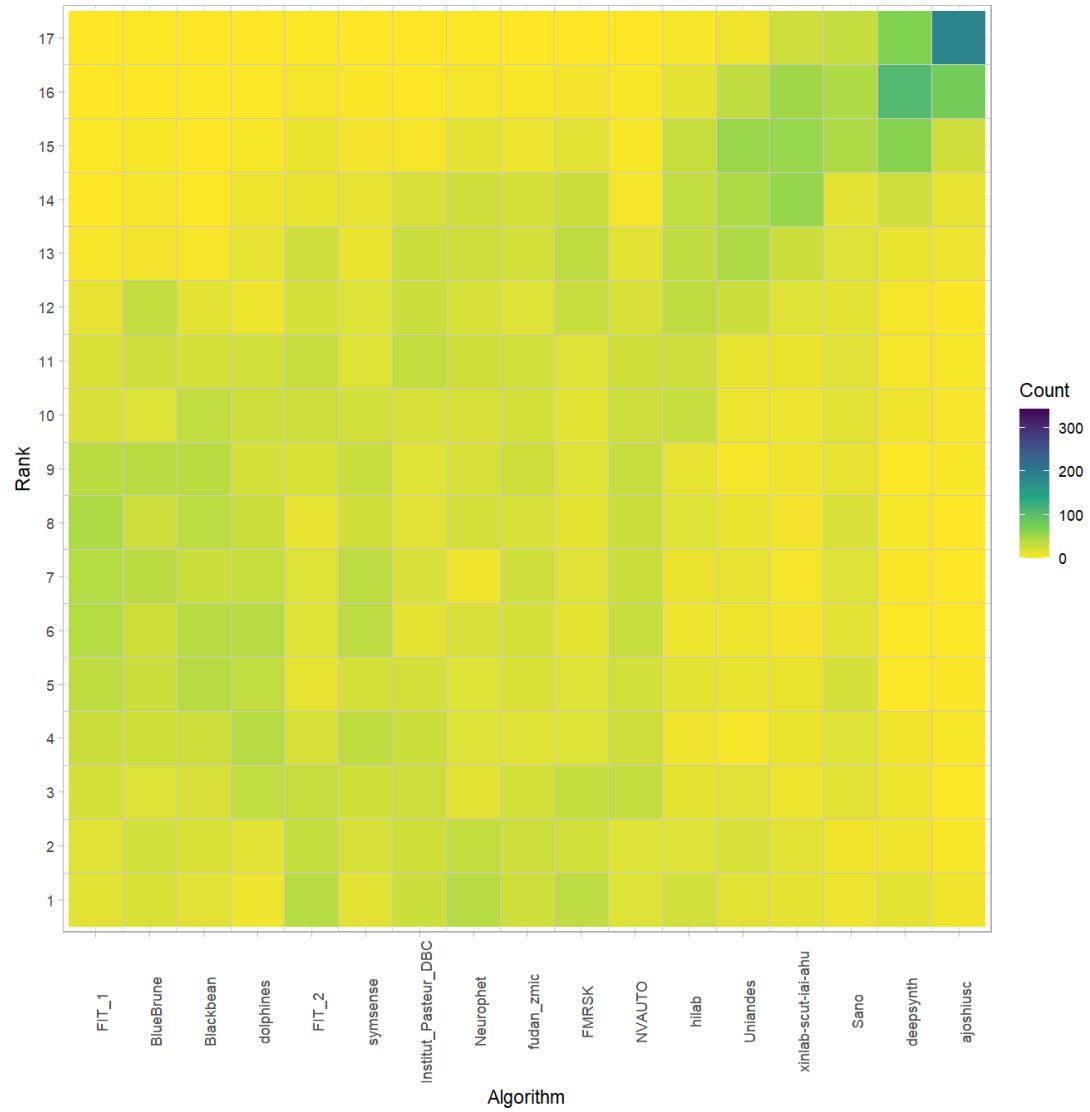
## Podium plot

*Podium plots* (see also Eugster et al., 2008) for visualizing raw assessment data. Upper part (spaghetti plot): Participating algorithms are color-coded, and each colored dot in the plot represents a metric value achieved with the respective algorithm. The actual metric value is encoded by the y-axis. Each podium (here:  $p=17$ ) represents one possible rank, ordered from best (1) to last (here: 17). The assignment of metric values (i.e. colored dots) to one of the podiums is based on the rank that the respective algorithm achieved on the corresponding case. Note that the plot part above each podium place is further subdivided into  $p$  “columns”, where each column represents one participating algorithm (here:  $p = 17$ ). Dots corresponding to identical cases are connected by a line, leading to the shown spaghetti structure. Lower part: Bar charts represent the relative frequency for each algorithm to achieve the rank encoded by the podium place.



## Ranking heatmap

*Ranking heatmaps* for visualizing raw assessment data. Each cell  $(i, A_j)$  shows the absolute frequency of cases in which algorithm  $A_j$  achieved rank  $i$ .

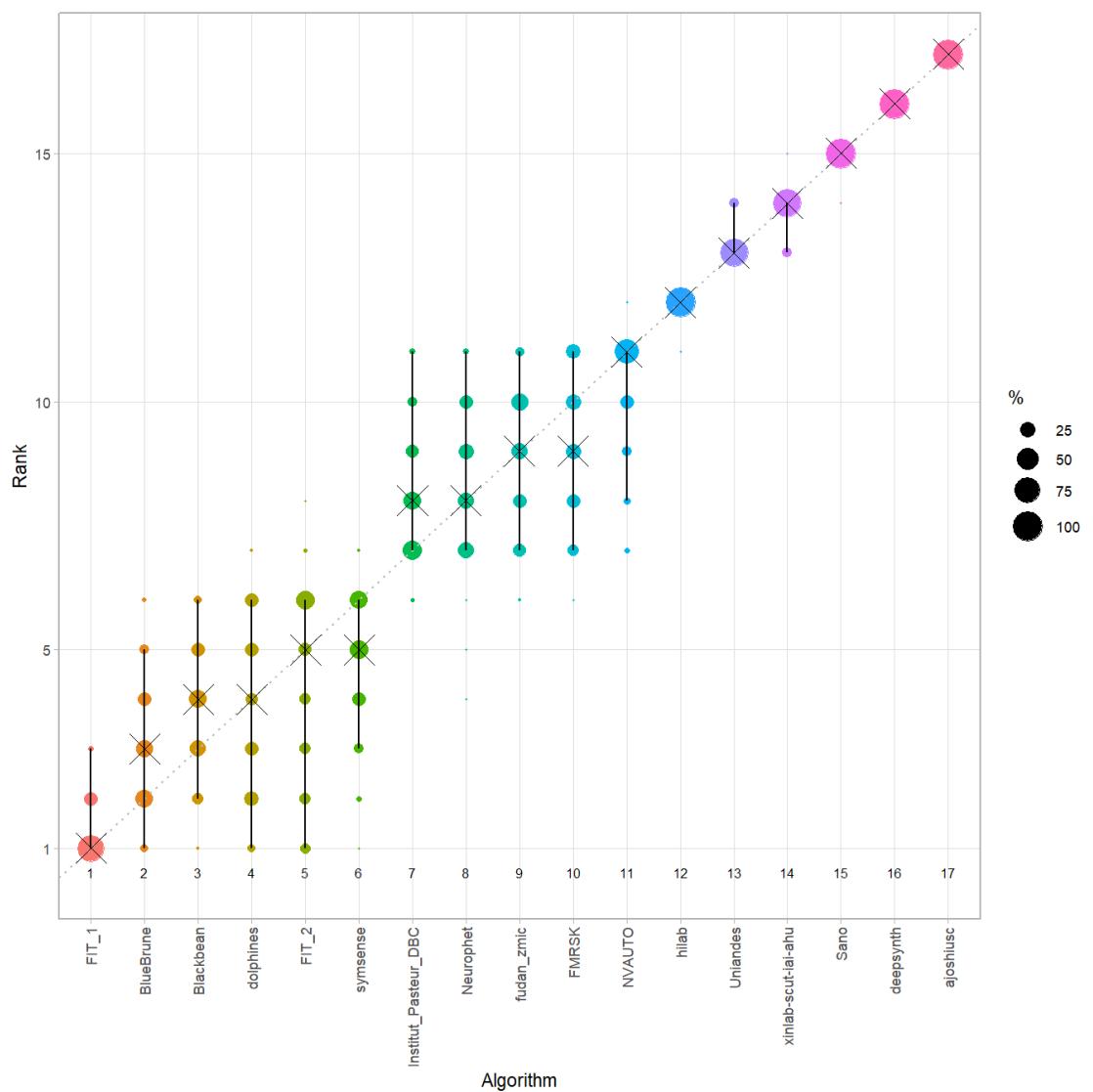


## Visualization of ranking stability

## **Blob plot for visualizing ranking stability based on bootstrap sampling**

Algorithms are color-coded, and the area of each blob at position  $(A_i, \text{rank } j)$  is proportional to the relative frequency  $A_i$  achieved rank  $j$  across  $b = 1000$  bootstrap samples. The median rank for each algorithm is indicated by a black cross. 95% bootstrap intervals across bootstrap samples are indicated by black lines.

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides(<scale> = ## "none")` instead.
```

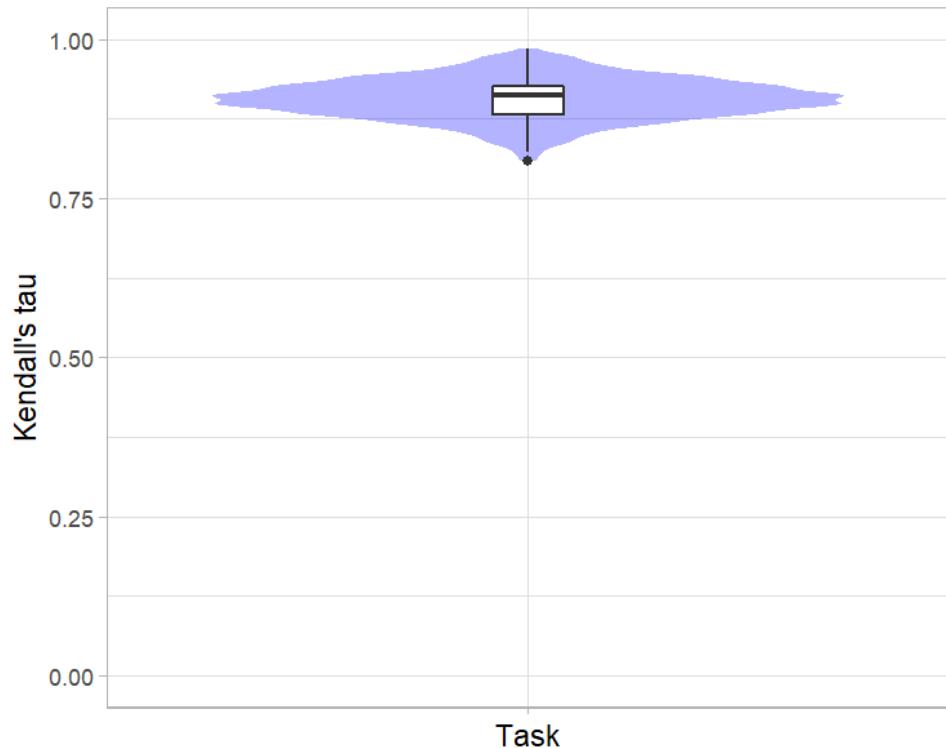


### ***Violin plot for visualizing ranking stability based on bootstrapping***

The ranking list based on the full assessment data is pairwise compared with the ranking lists based on the individual bootstrap samples (here  $b = 1000$  samples). For each pair of rankings, Kendall's  $\tau$  correlation is computed. Kendall's  $\tau$  is a scaled index determining the correlation between the lists. It is computed by evaluating the number of pairwise concordances and discordances between ranking lists and produces values between  $-1$  (for inverted order) and  $1$  (for identical order). A violin plot, which simultaneously depicts a boxplot and a density plot, is generated from the results.

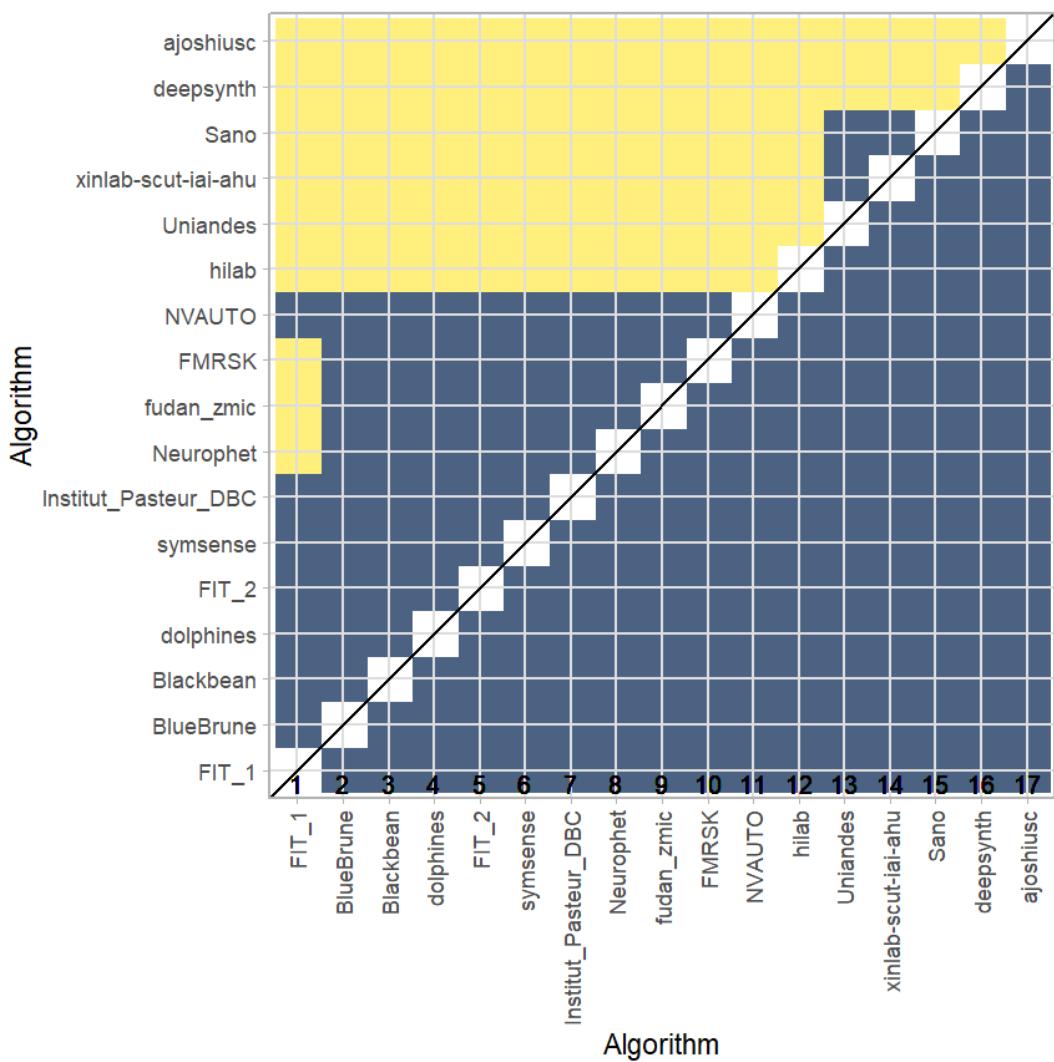
Summary Kendall's tau:

Task	mean	median	q25	q75
dummyTask	0.9056618	0.9117647	0.8823529	0.9264706



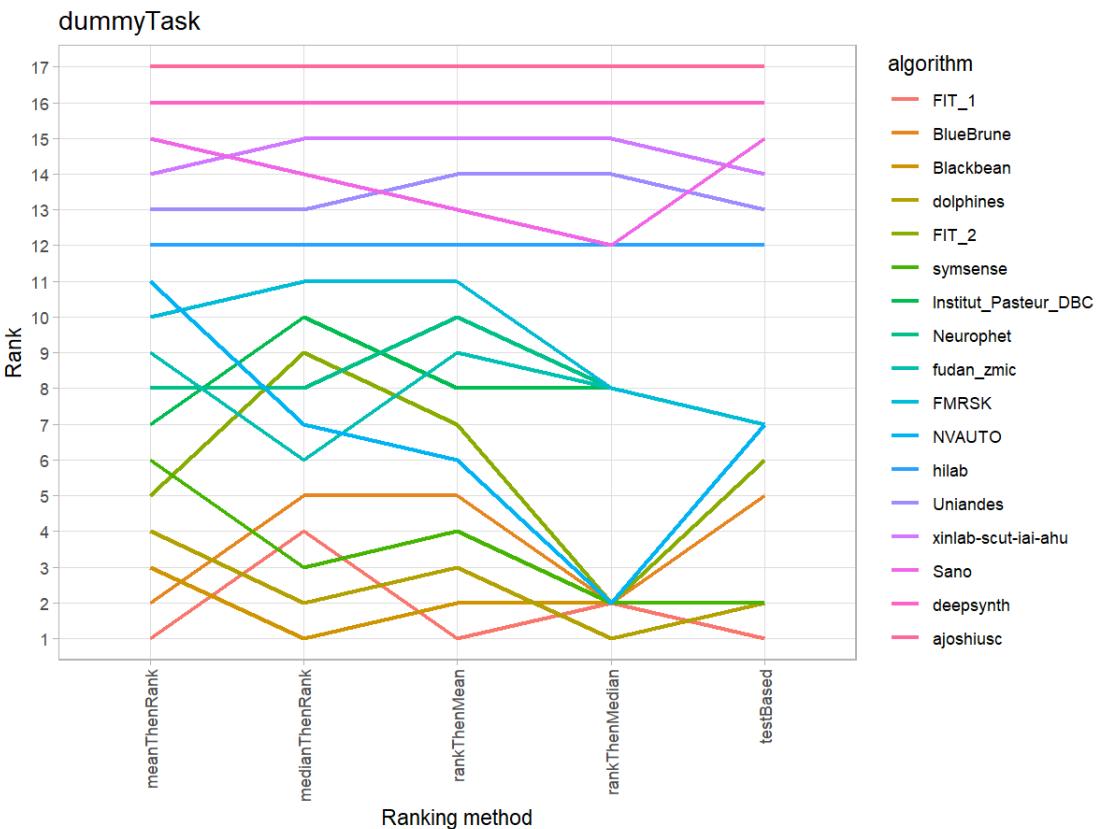
## *Significance maps* for visualizing ranking stability based on statistical significance

*Significance maps* depict incidence matrices of pairwise significant test results for the one-sided Wilcoxon signed rank test at a 5% significance level with adjustment for multiple testing according to Holm. Yellow shading indicates that metric values from the algorithm on the x-axis were significantly superior to those from the algorithm on the y-axis, blue color indicates no significant difference.



### Ranking robustness to ranking methods

*Line plots* for visualizing ranking robustness across different ranking methods. Each algorithm is represented by one colored line. For each ranking method encoded on the x-axis, the height of the line represents the corresponding rank. Horizontal lines indicate identical ranks for all methods.



### 36.3 References

Wiesenfarth, M., Reinke, A., Landman, B.A., Eisenmann, M., Aguilera Saiz, L., Cardoso, M.J., Maier-Hein, L. and Kopp-Schneider, A. Methods and open-source toolkit for analyzing and visualizing challenge results. *Sci Rep* **11**, 2369 (2021). <https://doi.org/10.1038/s41598-021-82017-6>

M. J. A. Eugster, T. Hothorn, and F. Leisch, “Exploratory and inferential analysis of benchmark experiments,” Institut fuer Statistik, Ludwig-Maximilians-Universitaet Muenchen, Germany, Technical Report 30, 2008. [Online]. Available: <http://epub.ub.uni-muenchen.de/4134/>.