

VISCERAL Anatomy3 Benchmark

Guidelines for Participation v1.0

Document History

v1.0 - 20141118 - Released version of document

1. Introduction

1.1 Registration

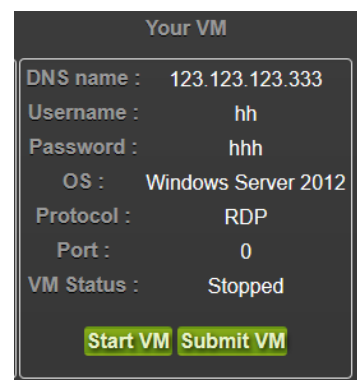
The first step in participation is registration. This is done online on the following page: <http://visceral.eu:8080/register/Registration.xhtml>

During the registration process, participants will be required to sign and upload a participation agreement.

Once the participant is registered, logging into the registration system will reveal the *participant dashboard*.

1.2 Virtual Machine

The medical imaging data is stored on the Microsoft Azure Cloud. When participants register successfully, they will receive a virtual machine (VM) in the Microsoft Azure cloud (Windows or Linux VMs are available), provided and financed by VISCERAL, with the support of Microsoft Research. The information for accessing the VM will appear on the participant dashboard in the registration system (after a delay of up to a week).

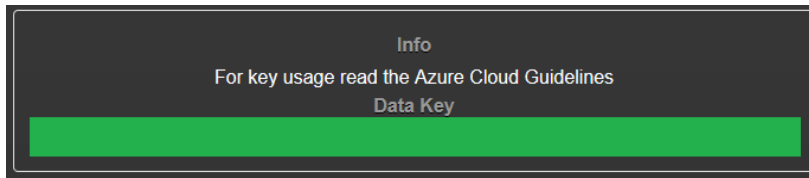


Note that each participating group should only register for the VISCERAL Benchmark once. After successful registration, the person completing the registration will get root access to the assigned VM, and will be able to create logins for colleagues. Please shut down the VM if it will not be used for a longer time (using the "Stop VM" button on the participant dashboard in the registration system). It can again be started using the "Start VM" button.

Documents about the Benchmark, including information on using the VM and a list of volumes can be found by pressing [View Benchmark Files](#). The evaluation software that will be used for calculating the evaluation metrics (see Section 7) is installed on the VM.

1.3 Training Data

The training data can be accessed from the VM and can also be downloaded via ftp (See section 1.3.1). The *Data Key* for accessing the data is provided on the participant dashboard in the registration system once the VM is assigned. Please only access the data on the cloud from within the assigned VM — accessing the data from outside the cloud results in additional costs for the organisers.



A list of all files in the training data set can be downloaded from the participant dashboard in the registration system by clicking on [View Benchmark Files](#).

The image file URLs are constructed as:

`cURL+filename+saKey`

cURL: container URL,

`http://visceralstorage1.blob.core.windows.net/trainingset/`

filename: `PatientID_ModalityCounter_ModalityName_RegionName.nii.gz`

saKey: shared access key, e.g.

`?sr=c&si=readonly&sig=Z6909Vz8TU0RxawtASpmpWZnT%2FhF2OgJOI7iEt60mis%3D`

1.3.1 Training Data downloads outside the cloud

For full training set downloads, an FTP access will be provided upon request. Please send an email for further information to oscar.jimenez@hevs.ch CC: ivan.eggel@hevs.ch

2. Organ Segmentation Task of VISCERAL Anatomy3 Benchmark

Within the organ segmentation task, participants can elect to participate in one or more sub-tasks, summarised below. “Full run” segmentation tasks require that the organs be detected and then segmented, while “Half run” tasks are organ segmentation-only tasks, and do not require organ detection to be done. Participants wishing to participate in the “Half run” task should request access to the location centroid information from the organisers (send an e-mail to Oscar.Jimenez@hevs.ch with ivan.eggel@hevs.ch in cc). This contains the location of the centroid of each of the organs to be segmented. Once access to the centroid information is granted, its shared access key will appear on the participant dashboard in the registration system.

The following sub-tasks are available for standard organ segmentation:

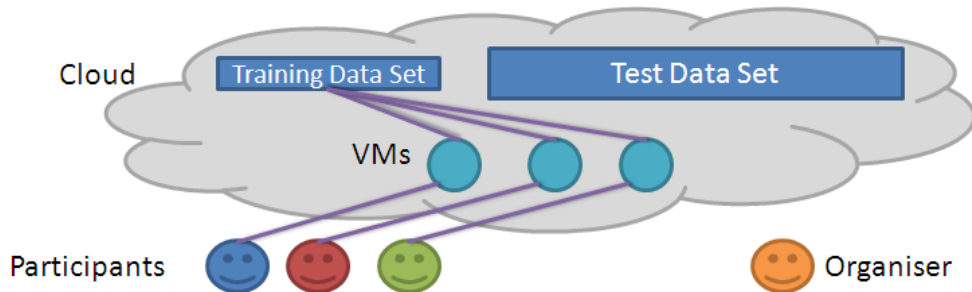
1. Full run segmentation
2. Half run segmentation

Participants can participate in a single sub-task.

3. Benchmark Organisation

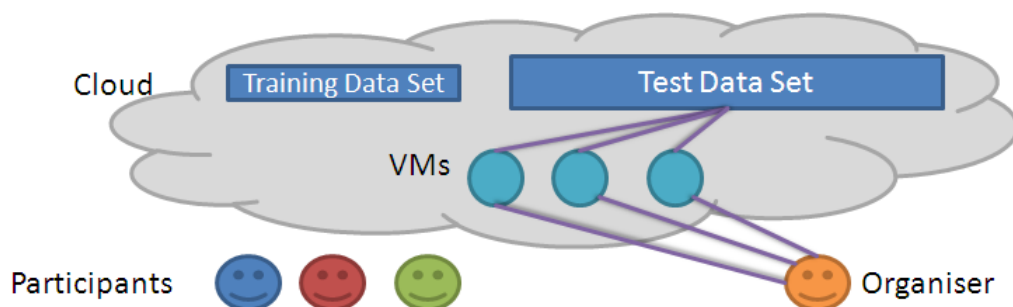
The benchmark runs in two phases.

3.1 Training phase



The participants each have their own VM in the cloud, linked to an annotated training data set of the same structure as the test data set. Software for carrying out the benchmark tasks must be placed into the VMs by the participants by the submission deadline. The software must be at least executable binaries and all libraries and other support required to execute the software. Source code is not required and must be removed from the VM by the submission deadline if the organizers should not see it (although even if source code is there, the organisers will not copy it to anywhere outside of the VM that it is in. Participants requiring additional security with respect to code or binaries can request to sign an NDA with the organisers). The software must satisfy all specifications in this document. The test data set is not accessible to the participants.

3.2 Testing phase



During the training phase of the Anatomy3 benchmark, participants will be able to submit their VMs and have their algorithms evaluated by the organizers on the test data. All software should be installed on the VM as required by the specifications, **participants must press `Submit VM` on the participant dashboard** to evaluate their algorithms. This lets the organizers know that they can take over the VM from the participants, execute the software installed on the VMs on the unseen test data set and evaluate the results. There will be a result leaderboard in the participants dashboard where the already evaluated results will be displayed. This evaluation process can be performed iteratively during the training phase (starting in January 2015).

On the benchmark deadline, the results available in the participant dashboard will be considered as final results in the **VISCERAL Anatomy Grand Challenge**¹. Delays caused by the participants executable (crashes, error messages ...) during the automatic evaluation phase might influence the total number of volumes evaluated. Participants should also make sure that the output segmentations are compatible with the evaluation tool before submitting their VM.

4. Further Information

Detailed information on the file formats used can be found in VISCERAL Deliverable 2.2.1².

Further information on the dataset and the modalities can be found in VISCERAL Deliverable 2.3.1³. This document also contains, in Section 3, the decisions taken regarding the manual annotation of the images.

All participants and organisers are automatically registered to the participants-anatomy3_benchmark@visceral.eu mailing list, and can post on the mailing list. Use this list to communicate only among participants and the organisers, to ask questions, draw attention to problems or share hints and tips.

A LinkedIn group has been set-up for discussion about the Benchmark. Ask questions and make comments on this group:

<http://www.linkedin.com/groups/VISCERAL-Benchmark-Discussion-5089631>

You can also follow VISCERAL on Twitter: @VisceralEU.

5. Evaluation metrics

The following metrics have been selected:

5.1 Segmentation evaluation:

- Dice coefficient
- Average distance
- Adjusted Rand Index
- Interclass Correlation

6. Program and File Format Conventions

After the submission deadline, for those VMs submitted by participants (“Submit VM” button), the organisers will run the participants’ programs installed in the VMs on the

¹ VISCERAL Anatomy Grand Challenge Workshop:

<http://www.visceral.eu/workshops/anatomy-grand-challenge-workshop/>

² VISCERAL Deliverable 2.2.1:

<http://www.visceral.eu/assets/Uploads/Deliverables/VISCERAL-D2-2-1.pdf>

³ VISCERAL Deliverable 2.3.1:

<http://www.visceral.eu/assets/Uploads/Deliverables/VISCERAL-D2.3.1.pdf>

unseen test data. Please ensure that all of the following naming and calling conventions and file format conventions are followed, to ensure that this works smoothly.

6.1 Program naming and Calling

One executable file (can be a script or compiled program) with the name

`execute_standard` or `execute_standard.{extension}`

must be in the home directory of the *azureuser* user (for those using Linux VMs), or on the Windows Desktop of the *azureuser* user (for those using Windows VMs) of the virtual machine. The *azureuser* user will be the only username available when the participant is assigned the virtual machine after registration.

Parameters

This executable file must take the following set of parameters, which are explained below:

```
-i [URL file to segment]
-o [output path]
-c [ConfigurationID]
-l [URL centroids for files to segment]
-r [RadLexIDs to segment]
```

The image file URLs are constructed as:

`cURL+filename+saKey`

cURL: container URL,

`http://visceralstorage1.blob.core.windows.net/trainingset/`

filename: `PatientID_ModalityCounter_ModalityName_RegionName.nii.gz`

saKey: shared access key, e.g.

`?sr=c&si=readonly&sig=Z69O9Vz8TU0RxawtASpmpWZnT%2FhF2OgJOI7iEt60mis%3D`

A full list of URLs of all training set files can be downloaded from the registration system.

Standard organ segmentation	
Parameter	Explanation
-i [URL file to segment] REQUIRED	The URL of a file to segment + shared access key. The file will be in NIFTI.GZ format. The filename contains info on: PatientID, ModalityID, ModalityAbbreviation <code>cURL+PatientID_ModalityCounter_ModalityName_RegionName.nii.gz+saKey</code>
-o [output path] REQUIRED	The full path in which output files are to be stored in: - Windows VM: temporary drive (D:) - Linux VM: /mnt/resource
-r [RadLexIDs to segment] OPTIONAL	The program should segment the organs given by the sequence of one or more RadLexID taken from the following table: - Kidneys (two, most often)

	<ul style="list-style-type: none"> <i>Left Kidney (RadLexID 29663)</i> <i>Right Kidney (RadLexID 29662)</i> - Spleen (<i>RadLexID 86</i>) - Liver (<i>RadLexID 58</i>) - Lungs (two) <ul style="list-style-type: none"> <i>Left Lung (RadLexID 1326)</i> <i>Right Lung (RadLexID 1302)</i> - Urinary bladder (<i>RadLexID 237</i>) - Rectus muscle (two) <ul style="list-style-type: none"> <i>Muscle body of left rectus abdominis (RadLexID 40358)</i> <i>Muscle body of right rectus abdominis (RadLexID 40357)</i> - Lumbar Vertebra 1 (<i>RadLexID 29193</i>) - Thyroid (<i>RadLexID 7578</i>) - Pancreas (<i>RadLexID 170</i>) - Psoas muscle (two) <ul style="list-style-type: none"> <i>Left psoas major muscle (RadLexID 32249)</i> <i>Right psoas major muscle (RadLexID 32249)</i> - Gallbladder (without ductus) (<i>RadLexID 187</i>) - Sternum (<i>RadLexID 2473</i>) - Aorta (<i>RadLexID 480</i>) - Trachea (<i>RadLexID 1247</i>) - Adrenal glands (two) <ul style="list-style-type: none"> <i>Left Adrenal Gland (RadLexID 30325)</i> <i>Right Adrenal Gland (RadLexID 30324)</i>
<p>-l [URL location initializer centroids for files to segment] OPTIONAL</p>	<p>Upon request only! This depends on the evaluation sub-task the participant is part of:</p> <ol style="list-style-type: none"> 1. fully automatic (sub-tasks 1 and 2), i.e., only image as input, the participant algorithm has to localize and segment from there 2. initialization is provided (sub-task 3), i.e., we provide the centroids of organ segmentations as initializations <p>The participant will have to indicate the choice at latest by the submission deadline. Participants participating in sub-task 3 will have access to a directory that holds the centroids of the organs that should be segmented.</p> <p><code>cURL+PatientID_ModalityCounter_ModalityName_RegionName_StructureInitializer.fcsv+saKey</code></p>
<p>-c [ConfigurationID] REQUIRED</p>	<p>The program can have up to 5 configurations, indicated by the integers 1 to 5. A configuration can be for example different sets of segmentation parameters. The participants are free to choose what constitutes a configuration.</p>

Organs to segment

It is not necessary that a program be able to segment all of the organs in the list. The program should accept all of the organ IDs in the above table, and provide no output for organ IDs that are not supported. Evaluation will only be done on the organ types that a program segments.

It is permitted to use different algorithms for different organs, but the executable file should take care of calling the correct algorithm for a given organ ID.

Configuration IDs

The program should accept all five configuration IDs, and provide no output if a particular configuration ID is not implemented. During the evaluation phase, the organisers will begin

with executing configuration ID 1, followed by the remaining configuration IDs in increasing order, until all configurations have been executed, or the period of weeks available for the evaluation calculations runs out. In the latter case, only the results for the configurations for which the execution completed will be provided.

Output Files

Output	Explanation
segmentation files	<p>Segmentation file is one file per segmented organ. The output file must be written to the [output path]. The file is a NIFTI.GZ file that contains binary values: 0 ... background 1 ... organ of interest. [0,1] ... fuzzy answers are possible, too, i.e. one value in the range 0 to 1 per voxel.</p> <p>The file name has to follow the convention:</p> <p>PatientID_ModalityCounter_ModalityName_RegionName_RadLexID_ParticipantID_ConfigurationID.nii.gz</p> <p>Note that the first part of the filename is identical to the volume file name the segmentation is performed on: PatientID_ModalityCounter_ModalityName_RegionName</p> <p>RadLexID ... identifies the anatomical structure ParticipantID ... is a unique identifier for every participant (assigned by the registration system, visible on the participant information page after login). ConfigurationID ... the same as the input parameter [configurationID]</p>

Example

1. Segmentation output: During the evaluation phase, the organisers will call the program of the participant with participantID **5xd9t** (who did not request to use location initialisers) as follows:

```
execute_standard -i
http://visceralstorage1.blob.core.windows.net/testset/CT1.nii.gz?sr=c&si=readonly&sig=Z6909Vz8TU0RxawtASpmpWZnT%2FhF2OgJOI7iEt60mis%3D
-o /mnt/resource/output_p1/ -c 1 -r 1247 58
```

meaning that it should segment the trachea and liver in the file CT1.nii.gz using configuration 1 and write the files:

```
/mnt/resource/output_p1/CT1_1247_5xd9t_1.nii.gz
```

and

```
/mnt/resource/output_p1/CT1_58_5xd9t_1.nii.gz
```

2. Segmentation output making use of initialization centroids: During the evaluation phase, the organisers will call the program of the participant with participantID **h23d8r** (who requested to use location initialisers) as follows:

```
execute_standard -i
http://visceralstorage1.blob.core.windows.net/testset/001_1_CT_wb.nii.gz?sr=c&si=readonly&sig=Z6909Vz8TU0RxawtASpmpWZnT%2FhF2OgJOI7iEt60mis%3D -o
/mnt/resource/output_p1/ -c 1 -r 1247 58 -l
001_1_CT_wb_StructureInitializer.fcsv
```

meaning that it should segment the trachea and liver in image 001_1_CT_wb.nii.gz using configuration 1, and write the files:

```
/mnt/resource/output_p1/001_1_CT_wb_1247_h23d8r_1.nii.gz
and
/mnt/resource/output_p1/001_1_CT_wb_58_h23d8r_1.nii.gz
```

6.2 Specification: segmentation centroid file

```
PatientID_ModalityCounter_ModalityName_RegionName_StructureInitializer.fcsv
```

```
#
# Each row entry in this file contains a landmark specification for the
# respective volume
# A entry is structured as follows <radlexID>, x-coordinate, y-coordinate,
# z-coordinate
# The coordinates are provided in world (scanner) coordinates
#
RADLEXID,x,y,z
RADLEXID,x,y,z
RADLEXID,x,y,z
RADLEXID,x,y,z
RADLEXID,x,y,z
RADLEXID,x,y,z
RADLEXID,x,y,z
RADLEXID,x,y,z
RADLEXID,x,y,z
```

7. Evaluation Software

7.1 Segmentation Evaluation

To evaluate segmentations against the ground truth, the program `EvaluateSegmentation` is provided. This software is available on the the virtual machine assigned to each participant. Participants should make sure that the output segmentations and landmark files are compatible with the tool before submitting their VM.

Description

`EvaluateSegmentation` is a command that compares two volumes (a test segmentation and a ground truth segmentation) using 22 different metrics that were selected as a result of comprehensive research into the metrics used in the medical volume segmentations. `EvaluateSegmentation` provides the following measures:

Similarity

1. Dice Coefficient
2. Jaccard Coefficient
3. Area under ROC Curve (one system state)

4. Cohen Kappa
5. Rand Index
6. Adjusted Rand Index
7. Interclass Correlation
8. Volumetric Similarity Coefficient
9. Mutual Information

Distance

10. Hausdorff Distance
11. Average Distance
12. Mahanabolis Distance
13. Variation of Information
14. Global Consistency Error
15. Coefficient of Variation
16. Probabilistic Distance

Classic measures

17. Sensitivity (Recall, true positive rate)
18. Specificity (true negative rate)
19. Precision
20. F-Measure
21. Accuracy
22. Fallout (false positive rate)

Supported Images

`EvaluateSegmentation` supports all 3D file formats that are supported by ITK, e.g. `.nii`, `.mha`, etc. The two Images should however have the same dimensions (the same number of voxels). There should be only one label in an image, where a voxel value can be either zero (background) or a value between zero and one $[0,1]$ that denotes the fuzzy membership or the probability that the corresponding voxel belongs to the label.

Syntax

USAGE: `EvaluateSegmentation truthURL segmentPath [-thd threshold] [-xml xmlpath] [-use all|DICE,JACRD, ...]`

where:

`truthURL`: `cURL+path to truth image+saKey`

`segmentPath`: `path to image being evaluated`

`-th threshold`: `before evaluation convert fuzzy images to binary using the given threshold`

`-xml xmlpath`: `path to xml file where results should be saved`

`-help`: `more information`

`-use metriclist`: `this option can be used to specify which metrics should be used. Metriclist consists of the codes of the desired metrics separated by commas.`

For those metrics that accept parameters, it is possible to pass these parameters by writing them between two `@` characters, e.g. `-use MUTINF,FMEASR@0.5@`. This option tells the tool to calculate the mutual information and the F-Measure at $\beta=0.5$.

Possible codes for metriclist are:

all: calculate all available metrics (default)
DICE: calculate Dice Coefficient
JACRD: calculate Jaccard Coefficient
GCOERR: calculate Global Consistency Error
VOLSMTY: calculate Volumetric Similarity Coefficient
KAPPA: calculate Cohen Kappa
AUC: calculate Area under ROC Curve (one system state)
RNDIND: calculate Rand Index
ADJRIND: calculate Adjusted Rand Index
ICCORR: calculate Interclass Correlation
MUTINF: calculate Mutual Information
FALLOUT: calculate Fallout (false positive rate)
COEFVAR: calculate Coefficient of Variation
AVGDIST: calculate Average Distance
HDRFDST: calculate Hausdorff Distance HDRFDST@0.95@ means use 0.95

quantile to

avoid outliers. Default is quantile of 1 which means exact Hausdorff distance

VARINFO: calculate Variation of Information
PROBDST: calculate Probabilistic Distance
MAHLNBS: calculate Mahanabolis Distance
SNSVTY: calculate Sensitivity (Recall, true positive rate)
SPCFTY: calculate Specificity (true negative rate)
PRCISON: calculate Precision
FMEASR: calculate F-Measure FMEASR@0.5@ means use 0.5 as a value for beta

in

the F-Measure

ACURCY: calculate Accuracy

Examples

Example 1: EvaluateSegmentation truth.nii segment.nii -use RNDIND,HDRFDST@0.96@,FMEASR@0.5@ -xml result.xml

This example shows how to compare two NIFTI images providing the Rand Index, Hausdorff distance and the F-Measure and save the results in result.xml.

The values between two @ symbols are parameters to the specific measures in this case the quantile value used with Hausdorff distance to avoid outliers. The second value (0.5) is the beta value used with the F-Measure.

Example 2: EvaluateSegmentation truth.nii segment.nii -use all -th 0.5

This example compares two images using all available metrics. Before comparing the images, they are converted to binary images using a threshold of 0.5,

that is voxels with values in $[0,0.5)$ are considered as background and those with values in $[0.5,1]$ are assigned the label with a membership of 1.