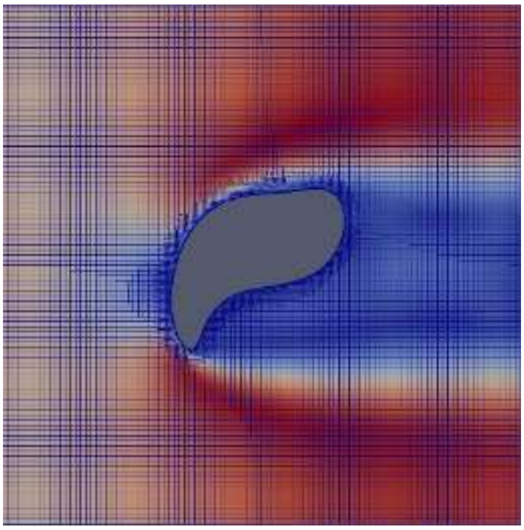
Deep Learning for Steady-State Fluid Flow Prediction in the Advania Data Centers Cloud

An UberCloud Experiment *)



renumics

KIT Karlsruher Institut für Technologie

With Support From

advania data centers

Uber Cloud

Hewlett Packard Enterprise

intel

*) Based on the master these of Jannik Zürn, Renumics GmbH

UberCloud Case Study 211

http://www.TheUberCloud.com

September 30, 2018

Welcome!

The UberCloud* Experiment started in July 2012, with a discussion about cloud adoption in technical computing and a list of technical and cloud computing challenges and potential solutions. We decided to explore these challenges further, hands-on, and the idea of the UberCloud Experiment was born, also due to the excellent support from INTEL generously sponsoring these experiments!

We found that especially small and medium enterprises in digital manufacturing would strongly benefit from technical computing in HPC centers and in the cloud. By gaining access on demand from their desktop workstations to additional compute resources, their major benefits are: the agility gained by shortening product design cycles through shorter simulation times; the superior quality achieved by simulating more sophisticated geometries and physics and by running many more iterations to look for the best product design; and the cost benefit by only paying for what is really used. These are benefits that increase a company's innovation and competitiveness.

Tangible benefits like these make technical computing - and more specifically technical computing as a service in the cloud - very attractive. But how far away are we from an ideal cloud model for engineers and scientists? In the beginning, we didn't know. We were just facing challenges like security, privacy, and trust; conservative software licensing models; slow data transfer; uncertain cost & ROI; availability of best suited resources; and lack of standardization, transparency, and cloud expertise. However, in the course of this experiment, as we followed each of the 205 teams closely and monitored their challenges and progress, we've got an excellent insight into these roadblocks, how our teams have tackled them, and how we are now able to reduce or even fully resolve them.

Deep learning has emerged as a new promising application for cloud-based HPC computing. An important aspect in this context is the generation and management of training data. The Renumics platform supports these processes in particular within engineering application. This project between Renumics GmbH (Karlsruhe) and UberCloud was established to explore the benefits of additional cloud computing resources that can be used to create a large amount of data samples in a fraction of the time a desktop computer would need to create them. More specifically, the goal of this case study was to apply Artificial Neural Networks to solve fluid flow problems to significantly decrease time-to-solution while preserving much of the accuracy of a traditional CFD solver. Creating a large number of simulation samples is paramount to let the neural network learn the dependencies between simulated design and flow field around it. In this project, we wanted to explore whether the overall accuracy of the neural network can be improved when more samples are being created in UberCloud's OpenFOAM container on Advania Data Centers HPCFLOW Cloud and then used during the training of the neural network.

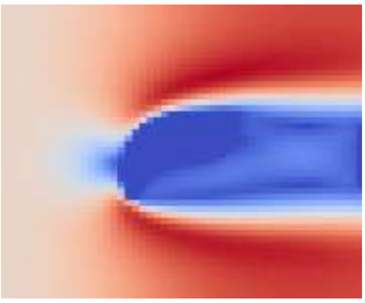
We want to thank our main UberCloud Experiment **sponsors INTEL and HPE** for generously supporting all 211 UberCloud Experiments.

Now, enjoy reading!
Wolfgang Gentzsch and Burak Yenier

**) UberCloud is the online community and marketplace where engineers and scientists discover, try, and buy Computing Power as a Service, on demand. Engineers and scientists can explore and discuss how to use this computing power to solve their demanding problems, and to identify the roadblocks and solutions, with a crowd-sourcing approach, jointly with our engineering and scientific community. Learn more about the UberCloud at: http://www.TheUberCloud.com.*

Team 211

Deep Learning for Steady-State Fluid Flow Prediction in the Advania Data Centers Cloud



"The overhead of creating high volumes of samples can be effectively compensated by the high-performance containerized computing environment provided by UberCloud and Advania."

1 MEET THE TEAM

End-User: Jannik Zuern, Renumics GmbH, Karlsruhe, Germany
Software Provider: OpenFOAM open source CFD software
Resource Provider: Advania Data Centers Cloud, Iceland
HPC and AI Experts: Stefan Suwelack, Markus Stoll, and Jannik Zuern, Renumics; Joseph Pareti, AI Consultant; and Ender Guler, UberCloud Inc.

2 USE CASE

Solving fluid flow problems using Computational Fluid Dynamics (CFD) is demanding both in terms of computer processing power and in terms of simulation duration. Artificial neural networks (ANN) can learn complex dependencies between high-dimensional variables. This ability is exploited in a data-driven approach to CFD that is presented in this case study. An ANN is applied in predicting the fluid flow given only the shape of the object that is to be simulated. The goal of the approach is to apply an ANN to solve fluid flow problems to significantly decrease time-to-solution while preserving much of the accuracy of a traditional CFD solver. Creating a large number of simulation samples is paramount to let the neural network learn the dependencies between simulated design and flow field around it.

This project between Renumics GmbH (Karlsruhe) and UberCloud Inc. was therefore established to explore the benefits of additional cloud computing resources that can be used to create a large amount of simulation samples in a fraction of the time a desktop computer would need to create them. In this project, we want to explore whether the overall accuracy of the neural network can be improved when more samples are being created in the UberCloud Container und then used during the training of the neural network. UberCloud kindly provided the cloud infrastructure, a CentOS Docker container with an OpenFOAM installation, and additional tech support during the project development.

3 WORKFLOW OVERVIEW

In order to create the simulation samples automatically, a comprehensive workflow was established.

As a **first step**, random two-dimensional shapes are created. These shapes have to be diverse enough to let the neural network learn the dependencies between different kinds of shapes and their respective surrounding flow fields.

In the **second step**, these shapes are meshed and added to an OpenFOAM simulation case template (Fig. 1). This template is simulated using the steady-state solver OpenFOAM solver simpleFOAM.

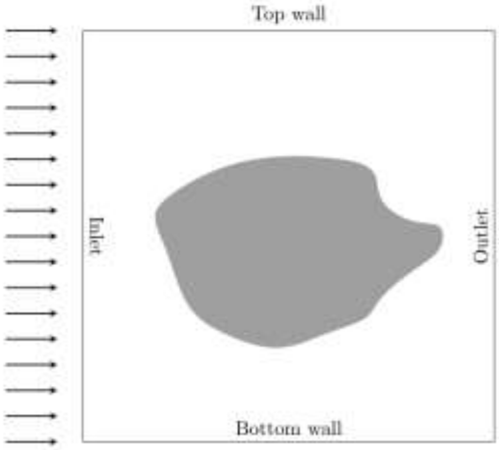


Figure 1: Simulation case setup. The flow enters the simulation domain through the inlet, flows around the arbitrarily shaped obstacle (dark grey shade) and leaves the simulation domain through the outlet.

In the **third step**, the simulation results are Post-Processed using the open-source visualization tool ParaView. The flow-fields are resampled on a rectangular regular grid to simplify the information processing by the neural net.

In the **fourth and final step**, both the simulated design and the flow fields are fed into the input queue of the neural network. After training, the neural network is able to infer a flow field merely from seeing the to-be-simulated design.

In Figure 2, a visualization of the four-step Deep Learning workflow is shown.

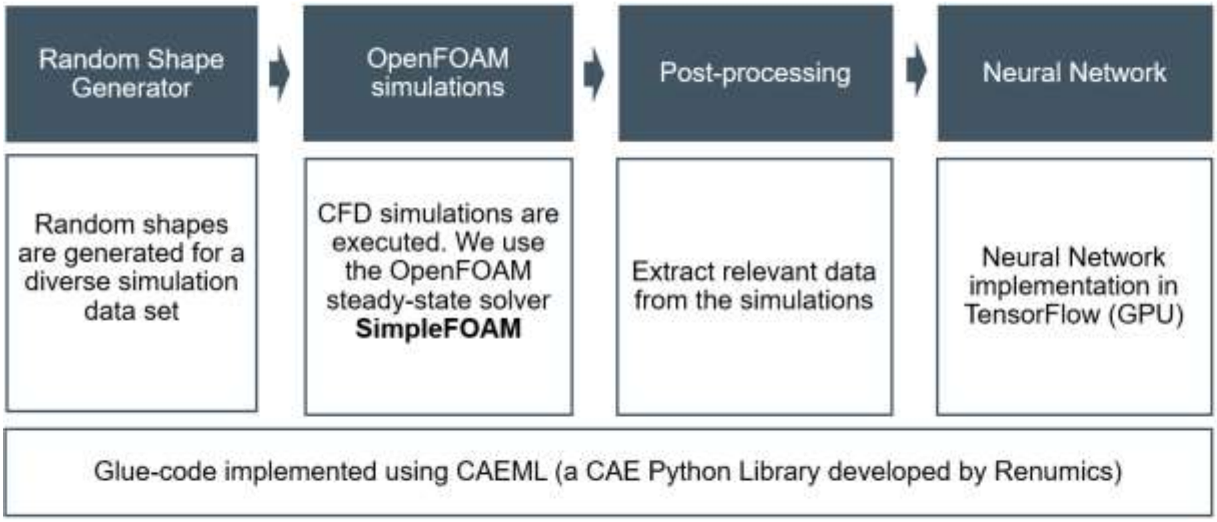


Figure 2: Deep Learning workflow.

Hardware specs

The hardware specs of the Advania Data Centers compute node hosting the UberCloud container are as follows:

- 2 x 16 core Intel Xeon CPU E5-2683 v4 @ 2.10 GHz
- GPU: none
- Memory: 251 GB

The hardware specs of the previously used desktop workstation are as follows:
- 2 x 6 core Intel i7-5820K CPU @ 3.30 GHz
- GPU: GeForce GTX 1080 (8GB GDDR5X memory)
- Memory: 32 GB

4 RESULTS

Time needed to create samples

As a first step, we compared the time it takes to create samples on the desktop workstation computer with the time it takes to create the same number of samples on the UberCloud container. Figure 3 illustrates the difference in time it took to create 10,000 samples

On the desktop computer it took 13h 10min to create these 10,000 samples. In the UberCloud OpenFOAM container in the Advania Data Centers Cloud, it took about 2h 4min to create 10,000 samples, which means that a speedup of 6.37 could be achieved using the UberCloud container.

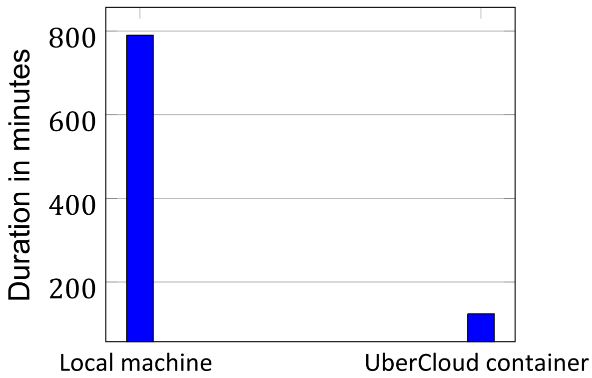


Figure 3: Comparison between Local machine and UberCloud container.

Neural Network performance evaluation

A total of 70,000 samples were created. We compare the losses and accuracies of the neural network for different training set sizes. In order to determine the loss and the accuracy of the neural network, we first must define, what these terms actually mean.

- Performance for generating the flow field data set and Tensorflow training

Setup	2-D external flow	3-D internal flow
Time for 10.000 simulations	13.2 h	152.5 h
Time for training	23.7 h	48.5 h

- Neural network prediction of flow field

Setup	2-D external flow	3-D internal flow
Time for CFD solver	4.7 s	55.0 s
Time of neural network prediction	3 ms	120 ms
Speedup factor with deep leaning	**1566**	**458**

Figure 4: Performance and speedup of flow simulations with neural network prediction.

Definitions

Loss: The loss of the neural network prediction describes how wrong the prediction of the neural network was. The output, or prediction, of the neural network in our project is a $N \times M \times 2$ tensor since the network tries to predict a fluid flow field with N elements in x-direction, M elements in y-direction, and two flow velocity components (velocity in x-direction and velocity in y-direction). A mean-squared-error metric was used to calculate the loss l:

$$l = \frac{1}{2} \sum_{d=0}^{1} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} (v_{dij} - \overline{v}_{dij})^2 \qquad (1)$$

where v_{dij} denotes the ground-truth velocity component in dimension d at the grid coordinates (i,j), $\overline{v}_{dij}$ denotes the predicted velocity component at the same position and in the same dimension. The goal of every machine learning algorithm is to minimize the loss of the neural network using numerical optimization schemes such as Stochastic Gradient Descent. Thus, a loss of 0.0 for all samples would mean that every flow velocity field in the dataset is predicted perfectly.

Accuracy: In order to be able to make sensible statements about the validity of the prediction of the neural network, metrics have to be defined that describe the level of accuracy that the neural network achieves. In general, the accuracy of a neural network describes how accurate the prediction of the neural network was. While the loss of a neural network is the metric that is being minimized during training, a small prediction loss does not necessarily mean that the corresponding prediction is physically meaningful. In general, however, a small prediction loss usually corresponds with a high accuracy. Different measurements of how accurate the outputs of the neural network are needed to express the validity of the predictions. A highly accurate prediction should have high values for all formulated accuracy measurements and a low loss at the same time. These accuracies can have values between 0.0 and 1.0, where an accuracy of 0.0 indicates that the prediction of the neural network does not at all coincide with the ground truth flow metric that is examined, and an accuracy of 1.0 means that the prediction coincides perfectly with the ground truth flow metric. Bear in mind that a low loss does not necessary cause high accuracy and vice versa. However, the two measurements are typically correlated.

In this study, two different accuracies were evaluated: Divergence accuracy and Drag accuracy:

- **Divergence accuracy:** Numerical CFD solvers aim to find a solution to the continuity equation and the momentum equation. For an incompressible fluid, the continuity equation dictates that the divergence of the velocity vector field is zero for every point in the simulation domain. This follows the intuition that at no point in the simulation domain fluid is generated (divergence would be greater than zero) or ceases to exist (divergence would be smaller than zero). By design, the Finite Volume Method preserves this property of the fluid even in a discretized form. A data-driven approach should as well obey this rule.
- **Cell accuracy:** The number of correctly predicted grid cells in the two- or three-dimensional grid yields an intuitive metric for how well the neural network predicts fluid flow behavior. As the network will never be able to predict the fluid flow velocity down to the last digit of a floating-point number, the following approach is proposed: If the relative error between the network prediction and the actual flow velocity is smaller than 5%, the respective grid cell is declared as predicted correctly. The cell accuracy can be calculated by counting the number of correctly predicted grid cells and dividing the results by the total number of grid cells.

5 TRAINING RESULTS

The generated samples are divided into the training and validation datasets. The training- and validation loss for different numbers of training samples was evaluated. Concretely, the neural net was trained three times from scratch with 1,000, 10,000, and 70,000 training samples respectively.

The following training parameters were used for all neural network training runs:

- Batch size: 32
- Dropout rate: 0.5
- Learning rate: 5×10^{-4}
-

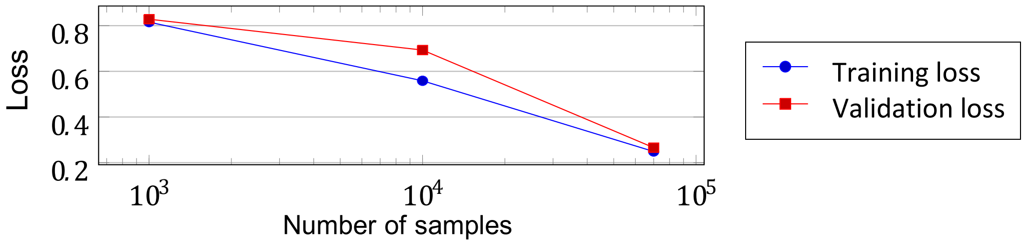


Figure 5: Loss after 50,000 training steps.

It can be observed that both training- and validation losses are lowest for the 70k samples training and are highest for the 1k training samples. The more different samples the neural network processes during the training process the better faster it is able to infer a flow velocity field from the shape of the simulated object suspended in the fluid. The validation loss tends to be higher than the training loss for all tested numbers of samples, which is a typical property of machine learning algorithms. Figure 6 shows the loss after 300,000 training steps:

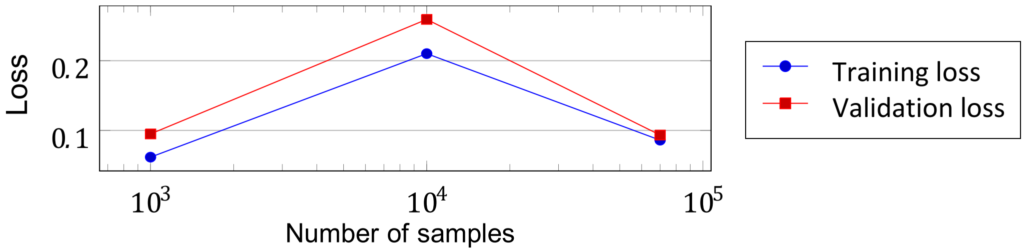


Figure 6: Loss after 300,000 training steps.

Surprisingly, the final training- and validation losses for the 70k samples training session are as low as the losses for the 1k samples training session. Generally speaking, no clear tendency towards lower losses when increasing the set of the training samples could be observed. This result is somewhat surprising since we expected the final losses at the end of the training process to show a similar tendency towards lower losses for higher numbers of samples. We assume that the number of samples does not heavily influence the final loss for extensive training sessions with many hundreds of thousand training steps. Finally, in Figure 7 the divergence and grid accuracies are visualized.

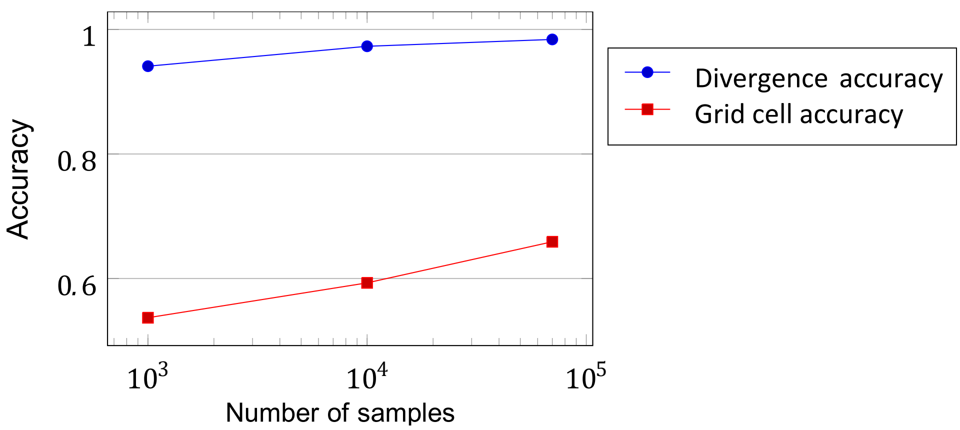


Figure 7: Validation accuracies after training.

Both the divergence accuracy and the grid cell accuracy show higher values for larger numbers of samples. While the divergence accuracy shows overall high values going from 0.94 for 1,000 samples to 0.98 for 70,000 samples, the grid cell accuracy also increases from a value of 0.53 for 1,000 samples to a value 0.66 for 70,000 samples. To recap: a grid accuracy of 0.66 means that approximately two thirds of all velocity grid cells were predicted correctly within 5% relative error to the correct value.

Figure 8 illustrates the difference between the ground truth flow field (left image) and the predicted flow field (right image) for one exemplary simulation sample after 300,000 training steps. The arrow direction indicates the flow direction and the arrow color indicates the flow velocity. Visually, no difference between the two flow fields can be made out.

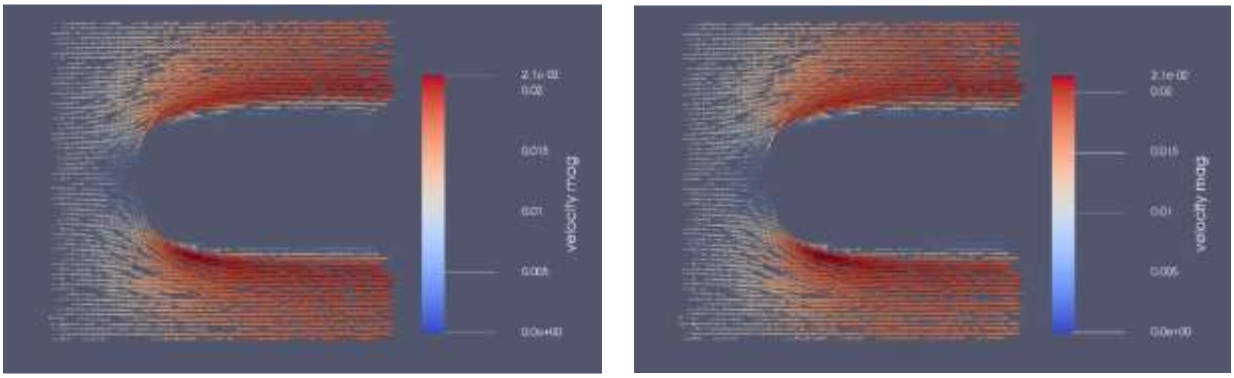


Figure 8: Exemplary simulated flow field (left image) and predicted flow field (right image).

CONCLUSION

We were able to prove a mantra amongst machine learning engineers: *The more data the better*. We showed that the training of the neural network is substantially faster using a large dataset of samples compared to smaller datasets of samples. Additionally, the proposed metrics for measuring the accuracies of the neural network predictions exhibited higher values for the larger numbers of samples. The overhead of creating high volumes of additional samples can be effectively compensated by the high-performance containerized (based on Docker) computing node provided by UberCloud on the Advania Data Centers Cloud. A speed-up of more than 6 compared to a state-of-the-art desktop workstation allows creating the tens of thousands of samples needed for the neural network training process in a matter of hours instead of days.

In order to train more complex models (e.g. for transient 3D flow models) much more training data will be required. Thus, software platforms for training data generation and management as well as flexible compute infrastructure will become increasingly important.

Case Study Author – Jannik Zuern, Renumics GmbH

Thank you for your interest in our free and voluntary UberCloud Experiment!

If you, as an end-user, would like to participate in an UberCloud Experiment to explore hands-on the end-to-end process of on-demand Technical Computing as a Service, in the Cloud, for your business then please register at: http://www.theubercloud.com/hpc-experiment/.

If you, as a service provider, are interested in a SaaS solution and promoting your services on the UberCloud Marketplace then please send us a message at https://www.theubercloud.com/help/.

2013 Compendium of case studies: https://www.theubercloud.com/ubercloud-compendium-2013/
2014 Compendium of case studies: https://www.theubercloud.com/ubercloud-compendium-2014/
2015 Compendium of case studies: https://www.theubercloud.com/ubercloud-compendium-2015/
2016 Compendium of case studies: https://www.theubercloud.com/ubercloud-compendium-2016/
2018 Compendium of case studies: https://www.theubercloud.com/ubercloud-compendium-2018/

The UberCloud Experiments & Teams received several prestigious international Awards, among other:

- **HPCwire Readers Choice Award 2013:** http://www.hpcwire.com/off-the-wire/ubercloud-receives-top-honors-2013-hpcwire-readers-choice-awards/
- **HPCwire Readers Choice Award 2014:** https://www.theubercloud.com/ubercloud-receives-top-honors-2014-hpcwire-readers-choice-award/
- **Gartner Cool Vendor Award 2015:** http://www.digitaleng.news/de/ubercloud-names-cool-vendor-for-oil-gas-industries/
- **HPCwire Editors Award 2017:** https://www.hpcwire.com/2017-hpcwire-awards-readers-editors-choice/
- **IDC/Hyperion Research Innovation Excellence Award 2017:** https://www.hpcwire.com/off-the-wire/hyperion-research-announces-hpc-innovation-excellence-award-winners-2/

If you wish to be informed about the latest developments in technical computing in the cloud, then please **register at http://www.theubercloud.com/** and you will get our free monthly newsletter.

To learn how deep learning can automate your engineering workflows, please contact the Renumics team (info@renumics.com) and schedule a free intro call.

With support from