R web service clients for DSCTool

version 1.5

Tome Eftimov, Urban Škvorc, Gašper Petelin, Rok Hribar,
Gorjan Popovski, Peter Korošec

May 28, 2020

The DSCTool was developed and implemented to make all the required knowledge for making different performance evaluations accessible from one place by guiding the user from providing input data (optimisation algorithm results) and selection of desired comparison to the final result of comparison. Further the implementation provides natural progress for all steps of performance evaluation, so no extra knowledge from the user is required.

1 REST web services

A web service is software that supports one or more open protocols and standards defined for exchanging data between applications or systems and makes itself available over the internet. Due to usage of open protocols and standards, software can be written in arbitrary programming language and run on various platforms. This makes web service widely interoperable and decoupled from implementations that uses their functionalities. The protocols most commonly used are the Simple Object Access Protocol (SOAP) and REpresentational State Transfer (REST). Due to its simplicity, flexibility, and lightweightedness, we have decided to use REST. Since REST is based on HTTP, we are able use basic HTTP methods, such as GET, PUT, POST, and DELETE. A REST web service access point is defined by Uniform Resource Identifier (URI), which is a string of characters that unambiguously identifies an individual resource. Data that is being transmitted can use different standard representations, such as XML or JSON.

The web services are accessible from following HTTPS URL `https://ws.ijs.si:8443/dsc-1.5/service/` and are using JSON for input/output data representation. More details about the DSCTool web services are available at `https://ws.ijs.si:8443/dsc-1.5/documentation.pdf`.

2 R web services clients

For easy integration into the IOHProfiler, we are going to present the DSCTool web service clients written in R programming language together with helper function that can be used to create the required input json for each web service.

The DSCTool services are free to use, however a registration is needed to use them. For the integration with IOHProfiler, we set a username and password, which can be used to make the web services calls. This username and password should be hidden from any of the local versions of the IOHProfiler. They can request for their own username and password.

- Username - IOHProfiler;
- Password - LjY3HfhGBxfR;

We should mention that the provided username and password should be used only in the online version of the IOHProfiler, while for local installations each user should request (with registration) their own username and password.

All helper functions to create the input json files work with the output template that can be accessed at `http://cs.ijs.si/dl/dsctool/input-data.csv`.

Additionally two helper functions are also implemented in order to get the data from the template with respect to selected algorithm or benchmark problem.

2.1 Split per algorithm

split_per_algorithm(data_temp,algorithm,DIM)

- *data_temp* - data frame or data matrix that reads the data from the template.
- *algorithm* - algorithm's name for which we want to obtain the data.
- *DIM* - benchmark problem dimension.

The output is a *data frame*, which consists of the optimisation results (y and x_i, $i = 1, \ldots, DIM$) for a given algorithm.

2.2 Split per problem

split_per_problem(data_temp,problem)

- *data_temp* - data frame or data matrix that reads the data from the template.
- *problem* - benchmark problem name for which we want to obtain the data.

The output is a *data frame*, which consists of the data from the data_temp only reduced for a given benchmark problem.

2.3 Registration service

Registration_json(name,affiliation,email,username,password)

- *Helper function to create the registration json.*
- *name* - set your name.
- affiliation - set your affiliation.
- email - set your email.
- username - set your username.
- password - set your password.

The output is a json file that can be used as input for the manage/user service.

Registration_client(postfield)

- *R web service client for the manage/user web service.*
- *postfield* - input json for the rankings web service. It can be created using the *Registration_json(name,affiliation,email,username,password)* helper function.

The output is a message ”Success!” if the registration is successful.

2.4 Ranking service

DSC_rank_service_parser(name,alpha,epsilon,monte_carlo_iterations,data_raw)

- *Helper function for the ranking web service.*
- *name* - can be “KS” or “AD”, which is related to which statistical test is used for comparing distributions, “KS” means two-sample Kolmogorov-Smirnov test, while “AD” is the two-sample Anderson-Darling test.
- *alpha* - significance level for hypothesis testing (e.g 0.05).
- *epsilon* - parameter which should be set if we want to apply practical DSC (pDSC). When epsilon is 0, we have the original DSC.
- *monte_carlo_iterations* - the number of Monte Carlo samples that should be used if we want to apply the Monte Carlo DSC ranking scheme for practical significance. For the original DSC, this number should be set at 0.
- *data_raw* - data frame or data matrix that reads the data from the template.

The output is a json file that can be used as input for the ranking service (see `http://cs.ijs.si/dl/dsctool/dsc.json`).

DSC_client(username,password,postfield)

- *R web service client for the ranking web service.*
- *username* - username used in the registration phase.
- *password* - password set in the registration phase.
- *postfield* - input json for the rankings web service. It can be created using the *DSC_rank_service_parser(name,alpha,epsilon,monte_carlo_iterations,data_raw)* helper function.

The output is a json file, which consists of the ranked matrix and some other parameters including the recommended omnibus statistical tests that can be used for the analysis (see `http://cs.ijs.si/dl/dsctool/dsc.result`).

2.5 Multivariate service

eDSC_rank_service_parser(desired_distribution,alpha,data_raw)

- *Helper function for the multivariate web service.*
- *desired_distribution* - defines the preferred distribution of solutions,with 0 preferring clustered distribution and with 1 preferring sparse distributions.
- *alpha* - significance level for hypothesis testing (e.g 0.05).
- *data_raw* - data frame or data matrix that reads the data from the template.

The output is a json file that can be used as input for the multivariate service (see `http://cs.ijs.si/dl/dsctool/multivariate.json`).

eDSC_client(username,password,postfield)

- *R web service client for the multivariate web service.*
- *username* - username used in the registration phase.
- *password* - password set in the registration phase.
- *postfield* - input json for the multivariate web service. It can be created using the *eDSC_rank_service_parser(desired_distribution,alpha,data_raw)* helper function.

The output is a json file, which consists of the ranked matrix and some other parameters including the recommended omnibus statistical tests that can be used for the analysis (see `http://cs.ijs.si/dl/dsctool/multivariate.result`).

2.6 Ensemble service

Ensemble_create_json(l,names,method)

- *Helper function for the support/create/ensemble service.*
- *l* - a list of results from the DSC rank service obtained for each quality indicator that is used.
- *names* - a vector with quality indicators names.
- *method* - ensemble method name, it can be ”average”, ”hierarchical”, or ”data-driven”.

The output is a json file that can be used as input for the support/create/ensemble (see `http://cs.ijs.si/dl/dsctool/create-ensemble.json`).

Ensemble_support_json_client(username,password,postfield)

- *R web service client for the support/create/ensemble web service. It creates input json for the ensemble web service.*
- *username* - username used in the registration phase.
- *password* - password set in the registration phase.
- *postfield* - input json for the support/create/ensemble web service. It can be created using the *Ensemble_create_json(l,names,method)* helper function.

The output is a json file that can be fruther used fot the ensemble service (see `http://cs.ijs.si/dl/dsctool/ensemble.json`).

Ensemble_client(username,password,postfield)

- *R web service client for the ensemble web service.*
- *username* - username used in the registration phase.
- *password* - password set in the registration phase.
- *postfield* - input json for the ensemble web service. It is used for multi-objective optimization.

The output is a json file, which consists of the ranked matrix and some other parameters including the recommended omnibus statistical tests that can be used for the analysis (see `http://cs.ijs.si/dl/dsctool/ensemble.result`).

2.7 Omnibus service

Omnibus_json(result,alpha,name,parametric_tests)

- *Helper function for the omnibus web service.*
- *result* - output json from rank, multivariate, or ensemble web service. It is used to take the ranked matrix that should be further analyzed.
- *alpha* - significance level for hypothesis testing (e.g 0.05).
- *name* - name of a statistical test that is selected for the analysis. The appropriate statistical tests are listed in the result.
- *parametric_tests* - it can be true or false. Also, this results is already set in the result parameter.

The output is a json file that can be used as input for the omnibus service (see `http://cs.ijs.si/dl/dsctool/omnibus.json`).

Omnibus_client(username,password,postfield)

- *R web service client for the omnibus web service.*

- *username* - username used in the registration phase.
- *password* - password set in the registration phase.
- *postfield* - input json for the omnibus web service. It can be created using the *Omnibus_json(result,alpha,name,parametric_tests)* helper function.

The output is a json file, which consists of the results from the hypothesis testing (see `http://cs.ijs.si/dl/dsctool/omnibus.result`).

2.8 Post-hoc service

Posthoc_json(result,k,n,base_algorithm)

- *Helper function for the omnibus web service.*
- *result* - output json from the omnibus web service. It is used to take the algorithms means that should be further analyzed.
- *k* - the number of compared algorithms.
- *n* - the number of benchmark problems.
- *base_algorithm* - the algorithm's name, which is selected as a control algorithm.

The output is a json file that can be used as input for the posthoc service (see `http://cs.ijs.si/dl/dsctool/posthoc.json`).

Posthoc_client(username,password,postfield)

- *R web service client for the posthoc web service.*
- *username* - username used in the registration phase.
- *password* - password set in the registration phase.
- *postfield* - input json for the posthoc web service. It can be created using the *Posthoc_json(result,k,n,base_algorithm)* helper function.

The output is a json file, which consists of the results of post-hoc service consists of a list of four results for each of the algorithms with respect to selected control algorithm. Z-value (ZValue) which is the result of selected statistic, unadjusted p-value (UnadjustedPValue) that is calculated from Z-value, and two adjusted p-values according to Holm and Hochberg procedures (see `http://cs.ijs.si/dl/dsctool/posthoc.result`).

2.9 Visualize service

performViz_json(result,method)

- *Helper function for the visualize web service.*
- *result* - output json from rank, multivariate, or ensemble web service. It is used to take the ranked matrix that should be further visualized.
- *method* - method used for visualization, currently only *performance2vec* is available.

The output is a json file that can be used as input for the visualize service.

performViz_client(username,passwrod,postfield, dest_file)

- *R web service client for the visualize web service.*
- *username* - username used in the registration phase.
- *password* - password set in the registration phase.
- *postfield* - input json for the visualize web service. It can be created using the *performViz_json(result,method)* helper function.
- *dest_file* - name of the file into which the image will be saved.

The result of visualize service consist of the *PNG* image, which shows performViz type of visualization. Later on different visualizations will be be added.