

MIGREW database: typical use cases

Kazantsev F.V.^{1,2*}, Skolotneva E.S.¹, Salina E.A.^{1,2}, Lashin S.A.^{1,2}

¹*Institute of Cytology and Genetics, SB RAS, Novosibirsk, Russia*

²*Novosibirsk State University, Novosibirsk, Russia*

* *e-mail: kazfdr@bionet.nsc.ru*

Recently we have released the MIGREW database (<https://migrew.sysbio.cytogen.ru/migrew>). It stores information on wheat immunity to rusts and powdery mildew that shared between the following objects: diseases, plants, chromosomes, genes, markers, protocols, pathogens and papers. The data focuses on effectiveness of wheat resistance genes in different regions of the Russian Federation. For each object of the database, one can obtain all levels of objects it is related with. Some typical use cases with the database were implemented in the MIGREW web application. On the other hand, these use cases are hardcoded through the application web forms. It is a challenge to get information that is not presented in the predefined author's logic. To take into account possible users requirements we have also developed REST API for the MIGREW database (https://migrew.sysbio.cytogen.ru/migrew_api). It is the direct way to the data acquisition that could be used by any programming/modeling tool that support REST service calls (Python, R or Matlab, for instance). If one uses Python in a daily work, he/she is familiar with the wide set of available tools and libraries for data analysis and data representation. Using these tools, one can write a script for data processing and results visualization. Here we demonstrate several use cases of the MIGREW data access and their visualization on global map with Python through the REST API: first – visualization of effectiveness of the wheat resistance gene or group of co-segregated resistance genes in the regions of the Russian Federation; second – occurrence rate of the virulence gene in the pathogen populations of the Russian Federation. Using these scenarios as a basis, one can develop his/her own scripts for MIGREW data accessing/processing and visualization in Python. *Acknowledgements:* The study has been supported by the Budget project 0324-2019-0040 and RFBR grant No. 17-29-08018.