The integration of land change modeling framework FUTURES into GRASS GIS 7

Anna Petrasova, Vaclav Petras, Douglas A. Shoemaker, Monica A. Dorning, Ross K. Meentemeyer

Center for Geospatial Analytics, North Carolina State University, Raleigh, NC, USA

Abstract

Many valuable models and tools developed by scientists are often inaccessible to their potential users because of non-existent sharing infrastructure or lack of documentation. Case in point is the FUTure Urban-Regional Environment Simulation (FUTURES), a patch-based land change model for generating scenario-based regional forecasts of urban growth pattern. Despite a high-impact publication, few scientists, planners, or policy makers have adopted FUTURES due to complexity in use and lack of direct access. We seek to address these issues by integrating FUTURES into GRASS GIS, a free and open source GIS and research platform for geospatial domain. This integration will enable us to take advantage of GRASS GIS tools for landscape structure analysis, and thus eliminate the need to use proprietary software for data preprocessing. Moreover, integration into GRASS GIS simplifies the distribution of FUTURES across all main operating systems and ensures maintainability of our project in the future. We will present our use case of integrating this advanced land change model into GRASS GIS platform and discuss the current state of the integration as well as the planned steps to achieve our vision of simple-to-use and fully free and open source FUTURES.

Keywords

geospatial, urbanization, open science, reproducibility, free and open source

1 Introduction

Despite all currently available technologies and tools for collaborative research and software development, many academic projects, although published in high-impact peer-reviewed journals are not adopted by disciplinary communities and thus fail to have broader impact for which they were intended. In the case of sophisticated geospatial models and analyses, acceptance of particular tools requires more than releasing the code with appropriate license online. Detailed and updated documentation, along with sample data of sufficient complexity, is required to demonstrate the model features and evaluate its suitability for use in research. A well defined user and programming interface implemented in a programming language widely used by disciplinary practitioners is crucial for further usage and extensions. Availability of the model across different operating systems is not only convenient for various users but it also simplifies model coupling.

The introduction of the land change modeling framework FUTURES (Meentemeyer et al., 2012) created to map regional projections of urban growth is an example of disconnect between modelers-developers and potential users. The authors of the FUTURES model first published a study of land development dynamics in the rapidly expanding metropolitan region of Charlotte, North Carolina, and later an analysis of the impacts of urbanization on natural resources under different conservation strategies (Dorning et al., 2015). Despite the recognition of high accuracy and novelty of the model, its use has been limited to FUTURES' authors and close collaborators. Limited access to the model together with missing documentation have slowed the progress of updating and adding new features, effectively barring the land change community from adopting the model.
Publishing the model as open source software is a solution that can address many of these disconnects, and create new opportunities for the scientific community to explore, apply and modify FUTURES for their own research. In this work-in-progress we rework the FUTURES urban growth model as an open source geospatial tool, highly relevant for the land change community, by integrating it into GRASS GIS, a free and open source GIS (Neteler et al., 2012). GRASS GIS natively provides many features, including landscape structure analysis, efficient large raster data processing and spatio-temporal visualizations, making it a suitable geospatial research platform for FUTURES.

2 GRASS GIS as a research platform for FUTURES

The FUTURES framework consists of several interconnected submodels (Figure 1). The core component is a stochastic, patch-growing algorithm (PGA) that bridges field-based and object-based representations of landscape change by constructing discrete land conversion events understood as urban growth. The Potential submodel feeds PGA with urban growth suitability based on statistical relationships of historic land change and significant environmental, infrastructural, and socioeconomic factors. The Demand submodel estimates population demand for urban development based on increases in population concurrent with the rate of landscape change. PGA, as the FUTURES engine, was written in C for performance reasons, whereas Potential and Demand were originally informally implemented as R scripts and ArcGIS workflows.
From the perspective of FUTURES architecture, GRASS GIS is a highly qualified geospatial platform for deployment. GRASS GIS functionality is divided into independent modules written in C and Python, which makes GRASS GIS compatible with FUTURES modular structure. This architecture allows for GRASS GIS to run on different operating systems and platforms including high performance computing (HPC) clusters (Metz, Rocchini & Neteler, 2014). Distribution of contributed scientific models is ensured through the GRASS GIS Addons repository, a browsable revision control repository maintained by OSGeo Foundation.
The source code in all GRASS GIS repositories is maintained by the GRASS GIS Development team. GRASS GIS has a long history of integrating and preserving scientific analytical tools (Chemin et al., 2015). Models integrated in the past are still maintained, improved and used with or without contribution from original authors. References to the original work are part of the documentation

and the original authors are cited in the derived works, see for example Di Leo et al. (2013). This long-term maintenance solves the issues occurring with different operating system, compilers, libraries and application programming interface (API) changes. GRASS GIS libraries and API offer standardized ways to effectively read, write, and organize geospatial data including optimizations for reading and processing of large raster and vector files which relieves the burden of implementing these low-level operations from the modeler-programmer. In addition, GRASS GIS allows to create a standardized interface for modules; using this mechanism GRASS GIS automatically generates a GUI, command line interface and HTML documentation for every module.
GRASS GIS users can easily work with R statistical software. Smooth transitioning between GRASS GIS and the R environment is possible thanks to spgrass6 and rgrass7 packages for R. The same applies for Python where GRASS GIS itself provides the API for Python so that Python libraries such as Numpy, SciPy, and Matplotlib can be easily used (Zambelli, Gebbert, S., & Ciolli, 2013).

3 Integrating FUTURES into GRASS GIS

We focused first on the patch-growing algorithm as the core component of FUTURES. Currently, the PGA part of FUTURES is fully integrated into GRASS GIS and available as part of a growing addon meta-module r.futures that will ultimately incorporate all the components. Since it is the most computationally demanding part of FUTURES we are currently exploring optimization options. However, just using efficient GRASS GIS I/O libraries resulted in significant speedup. The implementation of the two statistical submodels will be formalized as GRASS GIS addon modules using GRASS GIS Python API and NumPy to perform geospatial tasks and simple statistics. Potential as the more complex statistical model in R will be wrapped to provide a GRASS GIS interface to simplify chaining the modules. Large amounts of time and effort have been spent in manual preprocessing of input data, therefore we are currently automating the process by adding a new Calibration submodel which evaluates the patch size, shape and distribution needed for the PGA submodel (Figure 1).

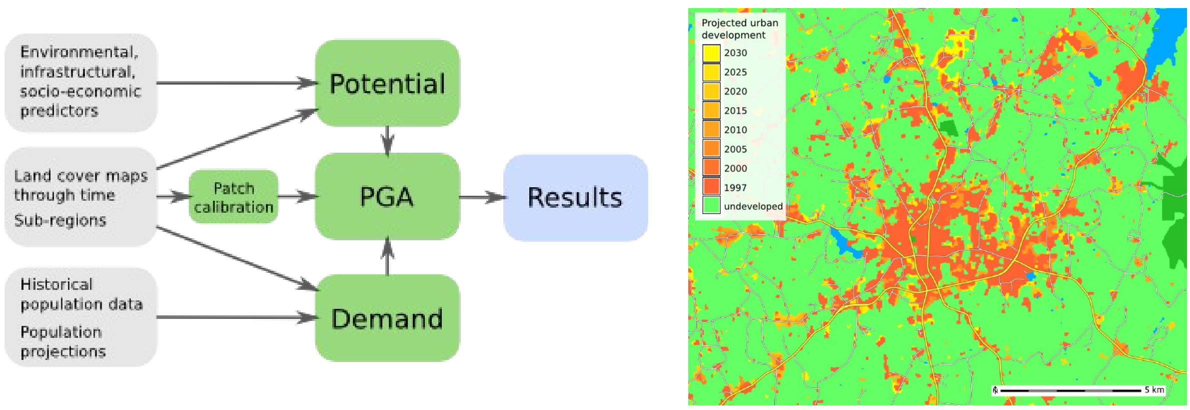

Figure 1: *On the left FUTURES simplified schema with inputs in gray and submodels identified for integration into GRASS GIS as modules in green. On the right the result of one run of FUTURES showing the incremental urban growth in part of Charlotte region, NC in thirty years.*

To support different geographic scales and wider range of applications we plan to design the GRASS GIS interface of the FUTURES addon modules to be flexible enough to allow users to input data at different levels of complexity or even substitute a selected FUTURES submodel with their own implementation. Moreover, the PGA could be reused for different applications than urban growth (e.g., epidemiology), and therefore we aim at generalizing the interface to accommodate such cases. Once all the FUTURES components are integrated into GRASS GIS, we will automate the entire procedure by chaining the FUTURES modules into a meta-module, useful not just for new FUTURES users but also for running FUTURES on HPC systems.
During the GRASS GIS FUTURES implementation we will prepare sample data for testing purposes based on publicly available data. All documentation, including parameter description and examples based on the dataset, will be distributed as manual pages to FUTURES addon modules. Upon completion and testing of each component, new addons will be published in the GRASS GIS repository (grass.osgeo.org/grass70/manuals/addons/r.futures.html).

References

- ✔ Bivand, R. (2007). Using the R–Grass interface. OSGeo Journal, 1, 36-38.
- ✔ Chemin, Y., Petras, V., Petrasova, A., Landa, M., Gebbert, S., Zambelli, P., Neteler, M., Löwe, P., & Di Leo, M. (2015). GRASS GIS: a peer-reviewed scientific platform and future research repository. Poster at General Assembly 2015 of European Geosciences Union, Vienna, Austria.
- ✔ Di Leo, M., de Rigo, D., Rodriguez-Aseretto, D., Bosco, C., Petroliagkis, T., Camia, A., & San-Miguel-Ayanz, J. (2013). Dynamic data driven ensemble for wildfire behaviour assessment: a case study. In Environmental Software Systems. Fostering Information Sharing (pp. 11-22). Springer Berlin Heidelberg.
- ✔ Dorning, M. A., Koch, J., Shoemaker, D. A., & Meentemeyer, R. K. (2015). Simulating urbanization scenarios reveals tradeoffs between conservation planning strategies. Landscape and Urban Planning, 136, 28–39. doi:10.1016/j.landurbplan.2014.11.011
- ✔ Meentemeyer, R. K., Tang, W., Dorning, M. A., Vogler, J. B., Cunniffe, N. J., & Shoemaker, D. A. (2012). FUTURES: Multilevel Simulations of Emerging Urban–Rural Landscape Structure Using a Stochastic Patch-Growing Algorithm. Annals of the Association of American Geographers, (October 2012), 23. doi:10.1080/00045608.2012.707591
- ✔ Metz, M., Rocchini, D., & Neteler, M. (2014). Surface temperatures at the continental scale: tracking changes with remote sensing at unprecedented detail. Remote Sensing, 6(5), 3822-3840.
- ✔ Neteler, M., Bowman, M. H., Landa, M., & Metz, M. (2012). GRASS GIS: A multi-purpose open source GIS. Environmental Modelling & Software, 31, 124-130. doi:10.1016/j.envsoft.2011.11.014
- ✔ Zambelli, P., Gebbert, S., & Ciolli, M. (2013). Pygrass: An object oriented python application programming interface (API) for geographic resources analysis support system (GRASS) geographic information system (GIS). ISPRS International Journal of Geo-Information, 2(1), 201-219.