

# Toward Programming 3D Shape Formation in Mammalian Cells

Jesse Tordoff

Massachusetts Institute of Technology  
tordoff@mit.edu

Jacob Beal

Raytheon BBN Technologies  
jakebeal@ieee.org

Ron Weiss

Massachusetts Institute of Technology  
rweiss@mit.edu

## 1 MOTIVATION

Biological cells are remarkably effective at predictable and resilient formation of complex three-dimensional shapes, as aptly demonstrated by most multicellular life on this planet. Not only can intricate shapes be formed with high reliability, but organisms also maintain functional integration of the entire system throughout development, as well as adapting form in response to environmental conditions, damage, and other disruptions. Moreover, these feats of manufacturing are accomplished entirely with reprocessed locally harvested materials.

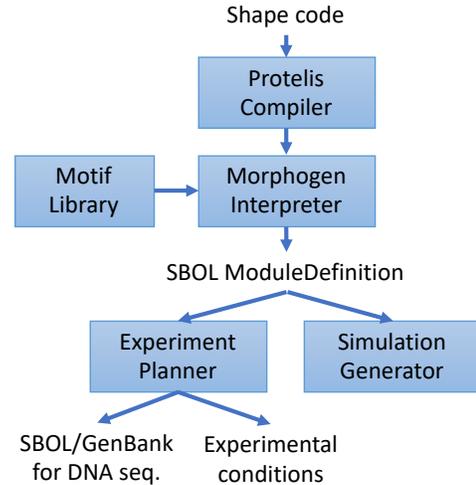
Our goal is to make these sorts of capabilities available for human engineering as well, through the reprogramming of living cells. We have selected mammalian cells as an initial platform for investigation, as these cell lines are well-studied, tractable for engineering, physically large and robust, and natively host many tools useful for shape formation. Furthermore, study of natural morphogenesis and relationships between evolution and development [2] suggests a set of natural “building blocks”—such as cell-sorting, gradient-based coordinates, and differential growth—that might be combined modularly to program shape formation. We are thus pursuing a research program of shape formation through isolation of biological shape formation “building blocks” and development of a system of genetic circuits for combining such building blocks into programs for the formation of complex three-dimensional shapes.

Toward this end, we have developed prototype motif-based compilation software for mapping from high-level three-dimensional shape specifications to genetic construct and sample designs. This compiler is supported by characterization experiments and an analytical workflow for converting experimental results into formulae for setting design parameters. Preliminary results from this work are promising, and we are now working to extend these into a full proof of concept for three-dimensional shape formation in mammalian cells.

## 2 APPROACH: MOTIF-BASED COMPILATION

The core idea behind our approach is motif-based compilation, building on our previous work with the Proto BioCompiler [1]. Under this approach, each operation in a high-level programming language is associated with a biological motif—a “template” design comprising a partial system specification, with variables for inputs and outputs. These motifs are then stitched together by instantiating a motif for each operation and connecting each motif instance input to its corresponding motif instance output (as specified by high-level program structure) to form a complete biological system specification. This specification may then be further refined by optimization, mapping of abstract parts to specific instances, etc.

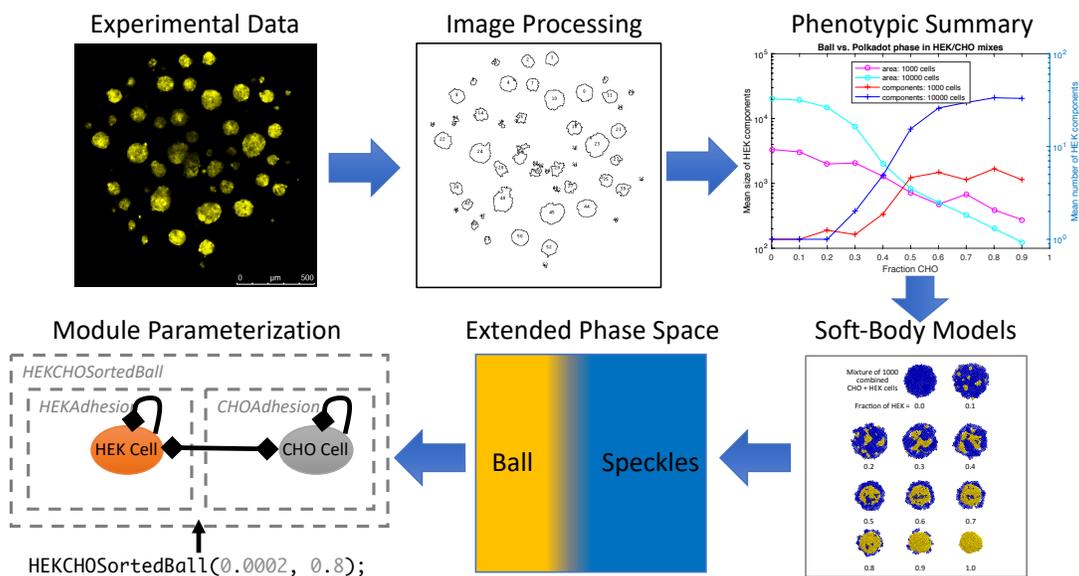
For our current implementation (Figure 1), we have updated our motif-based compiler to be based on the Protelis programming language [3], a Java-hosted aggregate programming language



**Figure 1: Programmed shape-formation architecture: a high-level shape specification is compiled by Protelis and interpreted by Morphogen (using a motif library) to produce an SBOL specification of the complete system. This may then be sent to a simulator for validation and/or exported for experiment as DNA sequence and sample specifications.**

with a more accessible syntax and better suited for adaptation and integration. We then swap the standard Protelis interpreter for a biological interpreter implementation that we call Morphogen, which transforms the program into a biological systems specification. In particular, Morphogen uses a Java-based library of motifs, each of which maps a Protelis operation to an SBOL ModuleDefinition [4] specifying a set of biological parts and interactions. A complete system specification is also a ModuleDefinition, constructed by the Morphogen interpreter: each time an Protelis operation is invoked, its associated ModuleDefinition is instantiated as a Module in the overall system ModuleDefinition, and its ports are linked to other Modules based on the Protelis program.

Our system thus takes in a high-level shape specification in Protelis and transforms it into an SBOL ModuleDefinition that specifies a biological system that is expected to produce the specified shape. From there, the specification may either be sent to a simulator for verification or else may be exported for realization in the laboratory as a set of DNA sequences and specifications of how sequences, strains, and other reagents should be combined to form experimental samples. We have implemented a preliminary version of this architecture, and are in the process of developing a number of composable motifs, including cell-sorting, cell-to-cell communication, symmetry breaking, cell type differentiation, and phase synchronization.



**Figure 2: Motif development workflow: microscopy images are processed to produce an initial phenotypic analysis of shape formation behavior. This analysis is used to parameterize models to refine characterization of the space of realizable behaviors, from which functions are extracted to constrain and parameterize applications of motifs in Protelis programs.**

### 3 DEVELOPMENT OF MOTIFS

Motif development is key to realizing our approach. For each shape formation “building block,” we need to not only demonstrate the capability of interest, such as cell-sorting, but also need to evaluate the effective range over which that capability can be realized and need to establish the numerical relationship between the values of inputs and experimental parameters and the properties of realized shapes. To this end, we have also developed a workflow for analysis of experimental data and its refinement into functions for validity testing and parameterization of motif applications in Morphogen interpretation of Protelis programs.

Figure 2 illustrates this workflow, along with examples of preliminary results taken from our use of this workflow in development of cell-sorting motifs. Shape formation experiments with a prospective motif produce microscopy data, which is processed through an image analysis pipeline to produce detailed statistics of the patterns formed by cells—in this example, cell-sorting producing a “polka dot” pattern. From these statistics, we produce an initial phenotypic analysis (in this example, the transition between “sorted ball” and “polka dot” behaviors), which is then refined with the aid of soft-body models to produce a predicted phase space of behaviors interpolated and extrapolated beyond experimental results. From this, we then extract functions that both constrain and parameterize applications of the motifs in Protelis programs.

We have applied this workflow to develop motifs based on cell-sorting: a “sorted ball” with differentiated interior and exterior, and “polka-dot” patterns. Furthermore, we have demonstrated that with such motifs we can reconstruct the plans that produced experimental results from appropriate Protelis specifications and that we can produce compiler errors for shapes that cannot be reliably realized.

### 4 CONTRIBUTIONS & FUTURE DIRECTIONS

Thus far, we have developed an architecture for programming three-dimensional shape formation in mammalian cells, comprising a motif-based compiler and a supporting experimental and analytical workflow for parameterization of designs being produced by the compiler. Preliminary results from this architecture are promising, showing that the compiler can reconstruct experimental designs and reject specifications that cannot be achieved. The next steps in development will be to extend and enhance the collection of experiments and models to expand the set of motifs available to the compiler, then deploy these to predict and demonstrate programmed formation of more complex three-dimensional shapes.

### ACKNOWLEDGMENTS

Additional authors (alphabetically): Bryan Bartley, Gizem Gumuskaya, Katherine Kiwimagi, Matej Krajnc, Kevin Lebo, Stanislav Shvartsman, Allen Tseng, Nicholas Walczak. This work has been supported by the Defense Advanced Research Projects Agency under Contract No. W911NF-17-2-0098. This document does not contain technology or technical data controlled under either U.S. International Traffic in Arms Regulation or U.S. Export Administration Regulations.

### REFERENCES

- [1] J. Beal, T. Lu, and R. Weiss. Automatic compilation from high-level biologically-oriented programming language to genetic regulatory networks. *PLoS ONE*, 6(8):e22490, August 2011.
- [2] S. B. Carroll. *Endless Forms Most Beautiful: The New Science of Evo Devo and the Making of the Animal Kingdom*. W. W. Norton & Company, 2005.
- [3] D. Pianini, M. Viroli, and J. Beal. Protelis: practical aggregate programming. In *Proceedings of the 30th Annual ACM Symposium on Applied Computing*, pages 1846–1853. ACM, 2015.
- [4] N. Roehner et al. Sharing structure and function in biological design with SBOL 2.0. *ACS Synthetic Biology*, 5(6):498–506, 2016.