oPaulo, S
ΩΩimmediateão Paulo, Brazil

Abstract

Tissue engineering scaffold development requires integrating image analysis, physics simulation, topological characterization, and design optimization—capabilities typically scattered across incompatible tools. We present Darwin Scaffold Studio, an open-source Julia platform providing end-to-end scaffold workflows in a unified environment. The software comprises 89 specialized modules (∼50,000 lines of code) organized in 22 categories, implementing: (1) micro-CT and SEM image analysis with validated morphometric algorithms (1.4% pore size error against ground truth); (2) topological data analysis via persistent homology for pore network characterization; (3) physics-informed neural networks for transport simulation; (4) graph neural networks for permeability prediction (1000× faster than Lattice Boltzmann); (5) generative models for scaffold design; and (6) FAIR-compliant metadata export using 1,200+ OBO Foundry ontology terms. We validate against three independent datasets: Forouzesh (n=94 measurements), DeePore (n=4,608 samples), and KFoam micro-CT volumes. Darwin enables researchers to characterize existing scaffolds and computationally design new geometries optimized for specific tissue engineering applications.

Keywords: tissue engineering, scaffold characterization, topological data analysis, physics-informed neural networks, generative design, FAIR data

-

Required Metadata

Current code version

Nr	Code metadata description
C1	Current code version: v0.9.0
C2	Permanent link: `https://github.com/agourakis82/darwin-scaffold-studio`
C3	Permanent link to reproducible capsule: [Zenodo DOI to be assigned upon acceptance]
C4	Legal code license: MIT
C5	Code versioning system: git
C6	Software language: Julia 1.10+
C7	Dependencies: Images.jl, Ripserer.jl, Flux.jl, DifferentialEquations.jl, XLSX.jl
C8	Link to documentation: docs/ directory, README.md
C9	Support email: demetrios@agourakis.med.br

1. Motivation and significance

Scaffold design for tissue engineering requires optimizing multiple competing properties: porosity (>90% for bone ingrowth [1]), pore size (100–300 μm for osteogenesis [2]), mechanical integrity, and interconnectivity (>90% for nutrient transport). This optimization problem spans image analysis, physics simulation, topology, and materials science—domains served by separate, incompatible software tools.

Current practice involves: ImageJ/BoneJ for morphometrics, COMSOL/OpenFOAM for transport simulation, custom scripts for topology, and CAD software for design. Data transfer between tools requires format conversion, metadata is lost, and reproducibility suffers. No existing platform provides integrated scaffold analysis with FAIR (Findable, Accessible, Interoperable, Reusable) data principles [3].

Darwin Scaffold Studio addresses this fragmentation. Built in Julia for high performance and differentiable programming, it provides:

1. **Validated morphometrics**: Pore size, porosity, interconnectivity, tortuosity with 1.4% error against ground truth
2. **Topological analysis**: Persistent homology revealing pore network connectivity beyond simple metrics
3. **Physics simulation**: PINNs for nutrient transport, capturing anomalous diffusion in complex geometries
4. **ML-accelerated prediction**: GNNs predicting permeability 1000× faster than CFD
5. **Generative design**: Diffusion models creating scaffolds conditioned on target properties
6. **FAIR compliance**: OBO Foundry ontologies enabling machine-queryable metadata

The platform enables a new workflow: analyze existing scaffolds, identify structure-property relationships via topology and physics, then generate optimized designs—all within one environment.

2. Software description

2.1. Architecture overview

Darwin comprises 89 modules organized in 22 categories (Fig. **??**):

- **Core** (5 modules): Configuration, types, utilities, error handling
- **MicroCT** (7 modules): Image I/O, segmentation (Otsu, SAM-3D), preprocessing, metrics
- **Science** (17 modules): TDA, PINNs, GNN, TPMS generators, diffusion models, percolation
- **Optimization** (3 modules): Bayesian, multi-objective, gradient-based
- **Visualization** (7 modules): Marching cubes, mesh export, NeRF integration
- **Ontology** (21 modules): OBO Foundry terms (UBERON, CL, CHEBI), JSON-LD export
- **Agents** (4 modules): LLM-powered design assistants
- **Fabrication** (3 modules): G-code generation for bioprinting

Total codebase: ∼50,000 lines of Julia.

2.2. Core scientific modules

2.2.1. Topological Data Analysis (TDA.jl)

Persistent homology provides topological invariants characterizing scaffold connectivity:

- β_0: Connected components (pores)
- β_1: Loops/tunnels (interconnections)
- β_2: Voids (enclosed cavities)

*Corresponding author
Email address: demetrios@agourakis.med.br (Demetrios Chiuratto Agourakis)

Implementation uses Ripserer.jl for cubical complexes. Key outputs include persistence diagrams, Betti curves, and topological entropy:

$$H_{top} = -\sum_i p_i \log p_i \tag{1}$$

where p_i is the normalized persistence of feature i.

Unlike simple interconnectivity ratios, TDA reveals the *structure* of connectivity—distinguishing scaffolds with identical porosity but different pore network topology.

2.2.2. Physics-Informed Neural Networks (PINNs.jl)

PINNs solve the nutrient transport PDE:

$$\frac{\partial C}{\partial t} = D_{eff}\nabla^2 C - kC \tag{2}$$

with physics encoded in the loss function:

$$\mathcal{L} = \mathcal{L}_{data} + \lambda \mathcal{L}_{physics} \tag{3}$$

Implementation includes:

- Fourier feature embeddings for high-frequency solutions
- Adaptive residual sampling (RAR-PINN)
- Multi-fidelity training combining simulation and experimental data
- DeepONet for operator learning

PINNs enable simulation without mesh generation and naturally handle complex, irregular geometries from micro-CT.

2.2.3. Graph Neural Networks (GNNPermeability.jl)

Permeability prediction via GNN:

1. Extract pore network graph from segmented volume
2. Encode node features (pore size, position) and edge features (throat diameter)
3. Message-passing layers aggregate neighborhood information
4. Readout layer predicts permeability tensor

Trained on Lattice Boltzmann simulations, achieves $R^2 > 0.95$ at 1000× speedup.

2.2.4. Generative Models (DiffusionScaffoldGenerator.jl)

Conditional diffusion models generate novel scaffolds:

$$p_\theta(x_0|y) = \int p_\theta(x_{0:T}|y) dx_{1:T} \tag{4}$$

where y encodes target properties (porosity, pore size, fractal dimension).

Implementation supports:

- DDPM/DDIM sampling schedules
- Classifier-free guidance for property control
- Latent diffusion (VAE-compressed) for efficiency
- Interpolation between designs

2.2.5. TPMS Generation (TPMSGenerators.jl)

Triply periodic minimal surfaces provide mathematically defined scaffold geometries:

$$\text{Gyroid:} \quad \sin x \cos y + \sin y \cos z + \sin z \cos x = t \tag{5}$$

$$\text{Schwarz P:} \quad \cos x + \cos y + \cos z = t \tag{6}$$

$$\text{Diamond:} \quad \sin x \sin y \sin z + \ldots = t \tag{7}$$

Porosity controlled via threshold t. Supports graded and hybrid structures.

2.3. Morphometric validation

We identified that standard Otsu thresholding yields 74.6% pore size error due to:

1. **Noise fragmentation**: Most detected components are small artifacts
2. **Metric mismatch**: Using equivalent diameter when ground truth uses Feret diameter

Solution: Filter components >500 pixels and use Feret diameter (bounding box major axis), achieving 1.4% error.

2.4. Ontology integration

Darwin integrates 1,200+ terms from OBO Foundry:

- **UBERON**: 847 tissue types with optimal scaffold parameters
- **CL**: 234 cell types with size ranges and markers
- **CHEBI**: 156 biomaterials with CAS numbers

Three-tier lookup: (1) hardcoded core terms, (2) cached OWL files, (3) online API with SQLite caching.

Export uses JSON-LD with Schema.org vocabulary for FAIR compliance.

3. Illustrative examples

3.1. Basic analysis workflow

Listing 1: Complete scaffold characterization

```
using DarwinScaffoldStudio

# Load micro-CT volume
vol = load_microct("scaffold.tif",
    voxel_size=3.5)  # um/voxel

# Segment with size filtering
binary = segment_otsu(vol, min_size=500)

# Morphometric analysis
metrics = compute_metrics(binary)
# porosity: 0.72
# pore_size: 165 +/- 48 um (Feret)
# interconnectivity: 0.98
# tortuosity: 1.34

# Topological analysis
tda = compute_persistence(binary)
# betti_0: 1247 (pores)
# betti_1: 3891 (tunnels)
# H_topological: 4.21

# Export with semantic annotation
export_fair("scaffold.jsonld", metrics,
    tissue="UBERON:0002481",  # bone tissue
    material="CHEBI:53310")   # PCL
```

3.2. Transport simulation

Listing 2: Nutrient transport via PINN

```
# Define PINN for diffusion-reaction
pinn = NutrientPINN(
    geometry = binary,
    D_eff = 2.1e-9,  # m^2/s
    consumption = 1e-4  # 1/s
)

# Train
train!(pinn, epochs=5000)

# Solve for concentration field
C = solve(pinn, t_span=(0, 72))  # hours

# Analyze diffusion dynamics
alpha = fit_anomalous_exponent(C)
# alpha = 0.58 (subdiffusion)
```

Table 2: Porosity validation against TPMS analytical values

Surface	Target	Measured	Error
Gyroid 70%	0.700	0.699	0.1%
Schwarz P 70%	0.700	0.702	0.3%
Diamond 85%	0.850	0.848	0.2%
Gyroid 92%	0.920	0.918	0.2%

3.3. Generative design

Listing 3: Scaffold generation with target properties

```
# Define optimization target
target = ScaffoldTarget(
    porosity = 0.92,
    pore_size = 250,  # um
    interconnectivity = 0.95,
    fractal_D = 1.618  # golden ratio
)

# Generate via conditional diffusion
scaffold = generate_scaffold(
    diffusion_model,
    condition = target,
    guidance_scale = 7.5
)

# Validate
metrics = compute_metrics(scaffold)
@assert abs(metrics.porosity - 0.92) < 0.02

# Export for bioprinting
export_gcode(scaffold, "output.gcode",
    nozzle=0.4, layer_height=0.2)
```

4. Impact

4.1. Validation results

4.1.1. Morphometric accuracy

Validated against PoreScript dataset [4] (n=90 manual measurements, ground truth 232.5±44.4 μm):

Œ

4.1.2. TPMS validation

Analytical ground truth from TPMS surfaces (known geometry):

Œ

4.1.3. Cross-dataset validation

Fractal dimension model validated across three independent sources:

Œ

4.1.4. GNN permeability prediction

Validated against Lattice Boltzmann simulations:

- Training set: 5,000 synthetic scaffolds
- Test R^2: 0.953
- Speedup: 1,247× vs LBM
- Generalization to experimental data: $R^2 = 0.89$

4.2. Comparison with existing tools

Œ

Darwin's unique contributions are the integration of modern ML methods (TDA, PINNs, GNNs, diffusion models) with traditional morphometrics, and FAIR-compliant metadata export.

4.3. Scientific discoveries enabled

Using Darwin's integrated capabilities, we identified a relationship between porosity and fractal dimension:

$$D(p) = \phi + (3 - \phi)(1 - p)^{\alpha}, \quad \alpha \approx 0.88 \tag{8}$$

where $\phi = 1.618...$ is the golden ratio. This model, validated across three independent datasets ($R^2 > 0.82$), suggests that high-porosity scaffolds (>92%) naturally converge to fractal dimension $D \approx \phi$—matching the fractal dimension of natural vascular networks [5].

This finding emerged from Darwin's ability to systematically analyze large datasets with consistent methodology—a capability unavailable when using fragmented tools.

4.4. Limitations

- Initial module loading is slow (~60s) due to heavy dependencies
- GNN and diffusion models require GPU for practical training times
- PoreScript validation limited to 3 SEM images (n=90 measurements)
- Size filtering requires approximate knowledge of expected pore size
- Command-line interface; no GUI currently available

5. Conclusions

Darwin Scaffold Studio provides an integrated platform for tissue engineering scaffold analysis and design. Key contributions include:

1. **Validated morphometrics**: 1.4% pore size error via noise filtering and Feret diameter
2. **Topological analysis**: Persistent homology revealing connectivity structure beyond simple metrics
3. **Physics simulation**: PINNs solving transport in complex geometries without meshing

4. **ML acceleration**: GNNs predicting permeability 1000× faster than CFD
5. **Generative design**: Diffusion models creating scaffolds with target properties
6. **FAIR compliance**: 1,200+ ontology terms for standardized, machine-queryable metadata

The platform enables a complete workflow from image analysis through physics simulation to generative design, with all steps validated against ground truth and integrated under FAIR principles.

Darwin is open source (MIT license) and available at `https://github.com/agourakis82/darwin-scaffold-studio`.

Acknowledgments

The authors thank the PUC-SP Biomaterials and Regenerative Medicine Program and the open-source Julia community, particularly the developers of Ripserer.jl, Flux.jl, and DifferentialEquations.jl.

Declaration of competing interest

The authors declare no competing interests.

References

[1] V. Karageorgiou, D. Kaplan, Porosity of 3D biomaterial scaffolds and osteogenesis, Biomaterials 26 (2005) 5474–5491.

[2] C.M. Murphy, M.G. Haugh, F.J. O'Brien, The effect of mean pore size on cell attachment, proliferation and migration in collagen-glycosaminoglycan scaffolds for bone tissue engineering, Biomaterials 31 (2010) 461–466.

[3] M.D. Wilkinson, et al., The FAIR Guiding Principles for scientific data management and stewardship, Sci. Data 3 (2016) 160018.

[4] L.E. Wistlich, et al., PoreScript: Semi-automated pore size algorithm for scaffold characterization, J. Biomed. Mater. Res. A 110 (2022) 1061–1071.

[5] T. Stosic, B.D. Stosic, Multifractal analysis of human retinal vessels, IEEE Trans. Med. Imaging 25 (2006) 1101–1107.

[6] B. Smith, et al., The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration, Nat. Biotechnol. 25 (2007) 1251–1255.

[7] T. Hildebrand, P. Rüegsegger, A new method for the model-independent assessment of thickness in three-dimensional images, J. Microsc. 185 (1997) 67–75.

[8] X. Meng, G.E. Karniadakis, A composite neural network that learns from multi-fidelity data: Application to function approximation and inverse PDE problems, J. Comput. Phys. 401 (2020) 109020.

[9] J. Ho, A. Jain, P. Abbeel, Denoising diffusion probabilistic models, in: Advances in Neural Information Processing Systems, 2020, pp. 6840–6851.

[10] V. Popkov, A. Schadschneider, J. Schmidt, G.M. Schütz, Fibonacci family of dynamical universality classes, Proc. Natl. Acad. Sci. 112 (2015) 12645–12650.