The Journal of Open Source Software

bca-survival: A Python package for body composition-based survival analysis in medical imaging research

Eric Frodl [1¶], **Benedikt Wichtlhuber**[1], **Matthias Neitzel** [1], **and Andreas Michael Bucher** [1]

1 Goethe University Frankfurt, University Hospital, Department of Radiology and Nuclear Medicine, Germany ¶ Corresponding author

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor:

Submitted: 16 December 2025
Published: unpublished

Summary

Body composition analysis (BCA) derived from computed tomography (CT) imaging has emerged as an important prognostic factor in oncology, geriatrics, and critical care medicine. Parameters such as skeletal muscle mass, visceral adipose tissue, and subcutaneous fat have been associated with treatment outcomes, survival, and quality of life across diverse patient populations. However, integrating CT-derived body composition metrics with clinical survival data requires substantial data processing and statistical expertise, creating barriers for clinical researchers.

`bca-survival` is an open-source Python package that streamlines the analysis pipeline from CT-derived body composition measurements to survival analysis results. The package provides a unified interface for merging clinical data with body composition measurements, performing Cox proportional hazards regression (both univariate and multivariate), generating Kaplan-Meier survival curves, and exporting results in multiple formats. It is specifically designed to integrate with output from automated segmentation algorithms such as the Body and Organ Analysis (BOA) algorithm (Haubold et al., 2024) and TotalSegmentator (Wasserthal et al., 2023), both built on the nnU-Net framework (Isensee et al., 2021).

Statement of need

The relationship between body composition and clinical outcomes is an active area of research in oncology, geriatrics, and critical care medicine (Caan et al., 2018; Keyl et al., 2024). Sarcopenia (low muscle mass) and changes in adipose tissue distribution have been linked to chemotherapy toxicity, surgical complications, and overall survival in various cancer types (Kazemi-Bajestani et al., 2016; Shachar et al., 2016). Modern deep learning algorithms can automatically segment body compartments from CT scans, producing detailed measurements of skeletal muscle, visceral adipose tissue, and subcutaneous adipose tissue at scale (Haubold et al., 2024; Wasserthal et al., 2023).

Despite the availability of automated segmentation tools, translating these measurements into meaningful survival analyses remains challenging. Researchers must address several interconnected tasks: data integration between imaging measurements and clinical databases, time-to-event calculation from diagnosis or treatment dates, statistical modeling with appropriate handling of covariates and confounders, and results dissemination in formats suitable for publications. While general-purpose survival analysis packages exist, such as `lifelines` (Davidson-Pilon, 2019) for Python and the `survival` package for R (Therneau & Grambsch,

2000), these tools are not addressing the domain-specific preprocessing and workflow challenges inherent to body composition research.

`bca-survival` builds upon `lifelines` to provide a higher-level abstraction specifically designed for body composition survival studies. Where `lifelines` requires researchers to manually prepare data, match patients across datasets, calculate time-to-event variables, and implement stratification logic, `bca-survival` encapsulates these steps into a unified interface. The package directly ingests output from segmentation algorithms like BOA, automatically matches imaging measurements to clinical records by patient identifier, computes survival times from date columns, and provides configurable stratification strategies (median, percentile, fixed cutoffs) for Kaplan-Meier analysis without requiring custom code. Additionally, `bca-survival` includes automatic multicollinearity detection via variance inflation factors for multivariate models—a common pitfall when analyzing correlated body composition variables—and correction variable support for univariate analyses to adjust for confounders like age or tumor stage.

By handling these domain-specific complexities, `bca-survival` reduces the technical barrier for clinicians and researchers investigating body composition as a prognostic factor, enabling larger-scale studies and facilitating reproducible research.

Functionality

The core functionality centers around the `BCASurvivalAnalyzer` class, which accepts clinical and measurement dataframes and provides methods for comprehensive survival analysis:

```python
from bca_survival import BCASurvivalAnalyzer
import pandas as pd

# Load clinical and body composition data
clinical_data = pd.read_csv('clinical_data.csv')
bca_measurements = pd.read_csv('bca_measurements.csv')

# Initialize analyzer with patient matching
analyzer = BCASurvivalAnalyzer(
    df_main=clinical_data,
    df_measurements=bca_measurements,
    main_id_col='patient_id',
    measurement_id_col='id',
    start_date_col='diagnosis_date',
    event_date_col='death_date',
    event_col='event_status'
)

# Univariate Cox regression for body composition variables
results = analyzer.univariate_cox_regression([
    'l5::WL::imat::mean_ml',
    'l5::WL::tat::mean_ml',
    'l5::WL::muscle::mean_ml'
])

# Kaplan-Meier curves with configurable stratification
analyzer.kaplan_meier_plot(
    'l5::WL::muscle::mean_ml',
    split_strategy='median'
)

# Multivariate analysis with automatic multicollinearity handling
```

```
model = analyzer.multivariate_cox_regression([
    'l5::WL::muscle::mean_ml',
    'age',
    'tumor_stage'
])
```

Key features of the package include automatic patient matching between clinical and imaging datasets with handling of missing measurements, flexible stratification strategies for survival curves (median, mean, percentile, or fixed cutoffs), variance inflation factor (VIF)-based multicollinearity detection and variable removal, and support for correction variables in univariate analyses to adjust for confounders.

The package also includes command-line utilities that integrate with existing clinical workflows. The `boa-extract` tool extracts measurements from BOA algorithm output directories, converting hierarchical folder structures into analysis-ready tabular format. The `bca-merge` tool combines clinical Excel files with body composition measurements based on patient identifiers. The `survival-result-converter` tool exports analysis results to PDF, CSV, and TXT formats for reporting, and `pdf-report-extractor` provides secure PDF encryption for patient data distribution.

Implementation

`bca-survival` is built on the established scientific Python ecosystem. Survival analysis functionality relies on `lifelines` (Davidson-Pilon, 2019), a well-validated survival analysis library implementing Cox proportional hazards models and Kaplan-Meier estimation. Data manipulation uses `pandas` (McKinney, 2010) and `numpy` (Harris et al., 2020), while visualizations are created with `matplotlib` (Hunter, 2007) and `seaborn` (Waskom, 2021). Statistical modeling is supplemented by `statsmodels` (Seabold & Perktold, 2010) for variance inflation factor calculations and `scikit-learn` (Pedregosa et al., 2011) for data standardization.

The package follows modern Python packaging standards using `pyproject.toml` configuration and `setuptools_scm` for version management from git tags. Comprehensive type hints support static analysis with `mypy`. The test suite uses `pytest` with coverage reporting, and code quality is enforced through `black`, `isort`, and `flake8`. Continuous integration via GitHub Actions runs tests across Python 3.9-3.11 on Linux, macOS, and Windows. Documentation is generated using Sphinx with autodoc support and is automatically deployed to GitHub Pages. The package is distributed via PyPI, enabling installation with `pip install bca-survival`.

Research applications

`bca-survival` is designed to support research investigating the prognostic value of body composition in clinical outcomes. Target applications include studies of sarcopenia as a predictor of survival in cancer patients, longitudinal analysis of body composition changes during chemotherapy or immunotherapy, investigation of adipose tissue distribution as a marker of treatment toxicity, and multi-site studies requiring standardized, reproducible analysis pipelines.

The package has been developed to support body composition research at University Hospital Frankfurt and is intended for use by the broader medical imaging and oncology research communities. By lowering the technical barrier to survival analysis, `bca-survival` aims to enable more researchers to investigate the clinical significance of CT-derived body composition metrics.

The Journal of Open Source Software

Acknowledgements

We acknowledge the developers of the Body and Organ Analysis (BOA) algorithm and TotalSegmentator for making automated body composition segmentation accessible to the research community. This project was funded by "NUM 3.0" (FKZ: 01KX2524)

Author Contributions

Eric Frodl: Conceptualization, software development, documentation, testing, deployment, manuscript writing - original draft, review & editing.

Benedikt Wichtlhuber: Software development, testing, manuscript review .

Matthias Neitzel: Field testing, user feedback, bug fixing, documentation.

Andreas Michael Bucher: Domain expertise, requirements specification, medical/clinical validation, supervision, manuscript review.

Caan, B. J., Feliciano, E. M. C., Prado, C. M., Alexeeff, S., Kroenke, C. H., Bradshaw, P., Quesenberry, C. P., Weltzien, E. K., Castillo, A. L., Olobatuyi, T. A., & others. (2018). Association of muscle and adiposity measured by computed tomography with survival in patients with nonmetastatic breast cancer. *JAMA Oncology*, *4*(6), 798–804.

Davidson-Pilon, C. (2019). lifelines: survival analysis in Python. *Journal of Open Source Software*, *4*(40), 1317. https://doi.org/10.21105/joss.01317

Harris, C. R., Millman, K. J., Walt, S. J. van der, Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., Kerkwijk, M. H. van, Brett, M., Haldane, A., Río, J. F. del, Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Haubold, J., Baldini, G., Parmar, V., Schaarschmidt, B. M., Koitka, S., Kroll, L., Landeghem, N. van, Umutlu, L., Forsting, M., Nensa, F., & Hosch, R. (2024). BOA: a CT-Based Body and Organ Analysis for Radiologists at the Point of Care. *Investigative Radiology*, *59*(6), 433–441. https://doi.org/10.1097/RLI.0000000000001040

Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering*, *9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

Isensee, F., Jaeger, P. F., Kohl, S. A. A., Petersen, J., & Maier-Hein, K. H. (2021). nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation. *Nature Methods*, *18*(2), 203–211. https://doi.org/10.1038/s41592-020-01008-z

Kazemi-Bajestani, S. M. R., Mazurak, V. C., & Baracos, V. (2016). Computed tomography-defined muscle and fat wasting are associated with cancer clinical outcomes. *Seminars in Cell & Developmental Biology*, *54*, 2–10. https://doi.org/10.1016/j.semcdb.2015.09.001

Keyl, J., Bucher, A., Jungmann, F., Hosch, R., Ziller, A., Armbruster, R., Malkomes, P., Reissig, T., Koitka, S., Tzianopoulos, I., & others. (2024). Prognostic value of deep learning-derived body composition in advanced pancreatic cancer—a retrospective multicenter study. *ESMO Open*, *9*(1), 102219.

McKinney, W. (2010). Data Structures for Statistical Computing in Python. *Proceedings of the 9th Python in Science Conference*, 56–61. https://doi.org/10.25080/Majora-92bf1922-00a

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830. http://jmlr.org/papers/

v12/pedregosa11a.html

Seabold, S., & Perktold, J. (2010). statsmodels: Econometric and Statistical Modeling with Python. *Proceedings of the 9th Python in Science Conference*, 92–96. https://doi.org/10.25080/Majora-92bf1922-011

Shachar, S. S., Williams, G. R., Muss, H. B., & Nishijima, T. F. (2016). Prognostic value of sarcopenia in adults with solid tumours: A meta-analysis and systematic review. *European Journal of Cancer*, *57*, 58–67. https://doi.org/10.1016/j.ejca.2015.12.030

Therneau, T. M., & Grambsch, P. M. (2000). *Modeling Survival Data: Extending the Cox Model*. Springer. https://doi.org/10.1007/978-1-4757-3294-8

Waskom, M. L. (2021). seaborn: statistical data visualization. *Journal of Open Source Software*, *6*(60), 3021. https://doi.org/10.21105/joss.03021

Wasserthal, J., Breit, H.-C., Meyer, M. T., Pradella, M., Hinck, D., Sauter, A. W., Heye, T., Boll, D. T., Cyriac, J., Yang, S., Bach, M., & Segeroth, M. (2023). TotalSegmentator: Robust Segmentation of 104 Anatomic Structures in CT Images. *Radiology: Artificial Intelligence*, *5*(5), e230024. https://doi.org/10.1148/ryai.230024