TABLE VII: Summary of General and Domain-Specific Guidelines

Authors	Title	Ref.	Field	Summary of findings (related to installation methods)
Jimenez et al (2017)	Four simple recommendations to encourage best practices in research software	[33]	General	no reference to installation instructions.
Eva Martin del Pico et al (2024)	FAIRsoft—a practical implementation of FAIR principles for research software	[53]	Bioinformatics	no reference to installation instructions.
Hermann et al (2022)	Documenting research software in engineering science	[25]	General	no reference to installation instructions.
Greg Wilson et al (2014)	Best Practices for Scientific Computing	[73]	General	no reference to installation instructions.
Lamprecht et al (2020)	Towards FAIR principles for research software	[37]	General	no reference to installation instructions.
Barker et al (2022)	Introducing the FAIR Principles for research software	[5]	General	no reference to installation instructions.
Greg Wilson et al (2017)	Good enough practices in scientific computing	[74]	Comp. biology	no reference to installation instructions.
Hastings et al (2014)	Ten recommendations for software engineering in research	[24]	Comp. biology	no reference to installation instructions.
Sandve et al (2013)	Ten Simple Rules for Reproducible Computational Research	[62]	Comp. biology	Rule 3: Archive exact versions of all external programs used to ensure reproducibility; consider storing a virtual machine image if necessary.
Benjamin D. Lee (2018)	Ten simple rules for documenting scientific software	[38]	Software Engineering	Rule 4: Include a ReadMe file with basic information.
Yolanda Gil et al. (2016)	Best practices for documenting and sharing research	[21]	GeoScience	Metadata should describe what is needed to install and run the software, including runtime dependencies (e.g., libraries).
Morhan et al. (2017)	Ten simple rules for making research software robust	[66]	Comp. Biology	Rule 2 and Rule 6: Provide compilation or installation instructions and Use build tools and package managers for installation.
Karimzadeh et al. (2017)	Top considerations for creating bioinformatics software documentation	[34]	Bioinformatics	Focus on platform-specific instructions and dependency management.
Fabien et al. (2018)	Re-run, Repeat, Reproduce, Reuse, Replicate: Transforming Code into Scientific Contributions	[7]	Neuroinformatics	reproducible code is verifiable and fosters the development of derivative works by other researchers.
Kai Zheng et al. (2015)	Ease of adoption of clinical natural language processing software: An evaluation of five systems	[78]	Neuroinformatics	Remedy strategies (1) more operation system specific use instructions; (2) including screen walk-throughs in use instructions so users know what to expect and what might have gone wrong; (4) avoiding jargon and acronyms in materials intended for end users; and (5) packaging prerequisites required within software distributions so that prospective adopters of the software do not have to obtain each of the third-party components on their own.
Connolly et al. (2023)	Software Engineering Practices in Academia: Promoting Readability, Resilience, and Reuse	[11]	Data Science	Structuring software into shareable packages enhances reusability; methods include PyPI, CRAN, Conda, Docker, and cloud VMs.
Patel et al. (2023)	Making Biomedical Research Software FAIR	[52]	Bioinformatics	Guidelines for making biomedical research software FAIR.
Libuit KG et al. (2023)	Proposed Standards for Public Health Bioinformatics Software	[43]	Bioinformatics	Providing clear local installation and/or remote access instructions for public health bioinformatics and Installation instructions should include a step-by-step list
Jean-Quartier et al. (2024)	Sharing practices for reproducible research	[31]	Software Engineering	Key points include downloadability, installability on different platforms, and open-source code with version history.
Daniel S.Katz et al. (2021)	Taking a fresh look at FAIR for research software	[katz]	Software Engineering	Key points include executable files.
Daniel S.Katz et al. (2021)	Fresh Look at FAIR for Research Software	[katz]	Software Engineering	Key points include executable files.
F. da Veiga Leprevost et al. (2014)	On best practices in the development of bioinformatics software	[41]	Bioinformatics	documenting source code targeted to different groups: software developers use to find source code. With easy-to-follow guidelines and instructions for non-programmers, it is possible to improve software usability.
Andreas Prlić et al. (2012)	Ten Simple Rules for the Open Development of Scientific Software	[55]	Computational biology	Rule 5: Keep software simple to increase user adoption by using standard methods and clear documentation.
M-O Gewaltig et al. (2014)	Current Practice in Software Development for Computational Neuroscience and How to Improve It	[20]	Computational neuroscience	Software in interpreted languages is easier to install and runs without warnings or errors.