Contents

Introduction

To provide a comprehensive background for the following chapters that focus on the interaction of human mesenchymal stromal cells (hMSCs) with multiple myeloma (MM) cells, this

Semi-Big Data is Causing a Reproducibility Bottleneck in Modern Biosciences

The advent of advanced technologies in biosciences has ushered in an era of Big Data, characterized by unprecedented volumes and complexities of data (Yang et al., 2017; Ekmekci et al., 2016). This rise has been paralleled by significant challenges in data analysis, particularly impacting the reproducibility of scientific research. Studies such as Baker's 2016 survey in Nature revealed that more than 70% of researchers have tried and failed to reproduce another scientist's experiments, highlighting a reproducibility crisis that questions the reliability of scientific findings (Baker, 2016; Begley & Ioannidis, 2015; Ioannidis, 2005).

Reproducibility is foundational to scientific research, ensuring that findings are reliable and verifiable (Goodman et al., 2016). Reproducibility was compromised by data nontransparency, as researchers lacked the infrastructure or habits[1] to provide raw data for independent verification, both for small and big data (Peng, 2011; Mesirov, 2010). Data nontransparency also includes cryptic or complete lack of explanations regarding processing pipelines, which are essential for understanding how data was processed and analyzed (Gosselin, 2021). Re-analysis of data often times did not support the original conclusions (Witwer, 2013).

In big data, recent advances have been made to standardize both raw data availability and processing pipelines, such as in RNAseq analysis

have shown how automation and sophisticated software can aid in making data analysis more reproducible by providing standardized and repeatable analysis protocols (Mei & Zhang, 2018). However, a gap exists for datasets that do not require or can not utilize these high-powered tools.

The awareness for complex datasets is rising in biosciences, as tutorials for *multidimensional data* are available (Krzywinski & Savig, 2013; Dunn et al., 2017).

In this thesis, the author introduces the concept of *semi-big data* to describe datasets with a size that is at the edge of manageability: Semi-big data is large and/or complex enough to make manual analysis extremely time-consuming, yet don't require or are incompatible with traditional Big Data tools. These datasets are commonly produced by methods such as automated microscopy or multiplex qPCR. Researchers are often forced to revert to basic tools like *Microsoft Excel* for analyzing semi-big data (Incerti et al., 2019). This could be due to the data's complexity requiring a high manual intervention to ensure accuracy of every processing step. However, manual analysis not only is laborious and prone to errors, but also represents a significant barrier to reproducibility, as most reviewers lack the time to repeat the same in-depth

[1] Peng (2011): *"[...] old habits die hard, and many will be unwilling to discard the hours spent learning existing systems."*

processing to verify the results.

A remedy to this bottleneck are either writing custom programming scripts, or tools that allow batch-processing, for example to execute the same processing steps on multiple samples. `Fiji/ImageJ`, a standard tool in microscopy, allows both macros and scripting for batch processing of multiple images and automating multiple processing steps into a pipeline. Another example is `PyMOL`, a standard tool in protein structural biology, which also has a Python command interface. Intriguingly, Microsoft has recently integrated a Python interpreter into *Excel*, allowing researchers to automate tasks and analyze data more efficiently (Excel, 2023).

However, custom scripts and tools of batch-processing can still pose a challenge to reproducibility: They are often highly specialized on a single use-case their adoption rate is very low. Also, batch-processing scripts often time don't outlay intermediate results comprehensively, decreasing the confidence in the final results. Besides, software scripts often do not follow standards that ensure functionality in different software environments, or lack necessary documentation to execute them properly, assuming the scripts are working as intended (Sandve et al., 2013; Peng, 2011).

Given the unique challenges posed by semi-big data, there is a critical need for new analytical tools specifically tailored to this data category. These tools must combine the ease of use found in basic software with the robust, analytical capabilities of more sophisticated systems. By automating the analysis process and standardizing outputs, such tools could significantly enhance reproducibility, allowing researchers to focus on scientific inquiry rather than data management intricacies.

In conclusion, as bioscience continues to generate increasingly complex datasets, the distinction between biologists and bioinformaticians blurs, emphasizing the need for all researchers to adopt computational tools. The development of new software to handle semi-big data effectively is not just an enhancement but a necessity to ensure the future reliability and efficiency of scientific research. This thesis proposes a framework for understanding and addressing the semi-big data challenges, setting the stage for a discussion on innovative solutions like the software tools described in subsequent chapters.

In conclusion, the integration of coding in bioscience research is not just a trend but a necessity. As the field continues to evolve, the demarcation between biologists and computational scientists blurs, underscoring the importance of coding skills for the next generation of researchers. The ability to code is fast becoming an indispensable asset, as integral to bioscience as traditional laboratory skills.

This thesis aims to address this call by providing a software tool that automates the analysis of semi-big data, ensuring that the results are

A main reason to write software is to define re-usable instructions for task automation (Narzt et al., 1998).

Modern Standards of Software Development

The complexity of software code makes it prone to errors, which can prevent its usage by persons other than the author himself. This is a problem for the general scientific community, as the software is often essential for reproduction (Sandve et al., 2013). Hence, modern journals aim to enforce standards to software development, including software written and used by biological researchers (Smith et al., 2018). Here, we provide a brief overview of the standards utilized by `plotastic` that to ensure its reliability and reproducibility by the scientific community (Peng, 2011).

Modern software development is a long-term commitment of maintaining and improving code after initial release (Boswell & Foucher, 2011). Hence, it is good practice to write the software such that it is scalable, maintainable and usable. Scalability or, to be precise, structural scalability means that the software can easily be expanded with new features without major modifications to its architecture (Bondi, 2000). This is achieved by writing the software in a modular fashion, where each module is responsible for a single function. Maintainability means that the software can easily be fixed from bugs and adapted to new requirements (Kazman et al., 2020). This is achieved by writing the code in a clear and readable manner, and by writing tests that ensure that the code works as expected (Boswell & Foucher, 2011). Usability is hard to define (Brooke, 1996), yet one can consider a software as usable if the commands have intuitive names and if the software's manual, termed "documentation", is up-to-date and easy to understand for new users with minimal coding experience. A software package that has not received an update for a long time (approx. one year) could be considered abandoned. Abandoned software is unlikely to be fully functional, since it relies on other software (dependencies) that has changed in functionality or introduce bugs that were not expected by the developers of all dependencies. Together, software that's scalable, maintainable and usable requires continuous changes to its codebase. There are best practices that standardize the continuous change of the codebase, including version control, continuous integration (often referred to as CI), and software testing.

Version control is a system that records changes to the codebase line by line, allowing the documentation of the history of the codebase, including who made which changes and when. This is required to isolate new and experimental features into newer versions and away from the stable version that's known to work. The most popular version control system is Git, which is considered the industry standard for software development (Chacon & Straub, 2024). Git can use GitHub.com as a platform to store and host codebases in the form of software repositories.

GitHub's most famous feature is called "pull request". A pull request is a request from anyone registered on GitHub to include their changes to the codebase (as in "please pull this into your main code"). One could see pull requests as the identifying feature of the open source community, since it exposes the codebase to potentially thousands of independent developers, reaching a workforce that is impossible to achieve with closed source models used by paid software companies.

Continuous integration (CI) is a software development practice in which developers integrate code changes into a shared repository several times a day (Duvall et al., 2007). Each integration triggers the test suite, aiming to detect errors as soon as possible. The test suite includes building the software, setting up an environment for the software to run and then executing the programmed tests, ensuring that the software runs as a whole. Continuous integration is often used together with software branches. Branches are independent copies of the codebase that are meant to be merged back into the original code once the changes are finished. Since branches accumulate multiple changes over time, this can lead to minor incompatibilities between the branches of all developers (integration conflicts), which is something that CI helps to prevent.

Continuous integration especially relies on a thorough software testing suite. Software testing is the practice of writing code that checks if the codebase works as expected (Myers et al., 2011). The main type of software testing is unit testing, which tests the smallest units of the codebase (functions and classes) in isolation (Listing 1).

Listing 1: Example of an arbitrary Python function and its respective unit test function. The first function simply returns the number 5. The second function tests if the first function indeed returns the number 5. The test function is named with the prefix "`test_`" and is placed in a file that ends with the suffix "`_test.py`". The test function is executed by the testing framework `pytest`. Note that code after "#" is considered a comment and won't be executed.

```
# Define a function called "give_me_five" that returns the number 5
def give_me_five():
    return 5
# Define a test function asserting that "give_me_five" returns 5
def test_give_me_five():
    assert give_me_five() == 5
```

The quality of the software testing suite is measured by the code coverage, the precision of the tests, and the number of test-cases that are checked. The code coverage is the percentage of the codebase that is called by the testing functions, which should be as close to 100% as possible, although it does not measure how well the code is tested. The precision of the test is not a measurable quantity, but it represents if the tests truly checks if the code works as expected. The number of test-cases is the number of different scenarios that are checked by the testing functions, for example testing every possible option or combinations of options for functions that have multiple options. The most popular software testing framework for Python is `pytest`, which is utilized by `plotastic` (Krekel et al., 2004).

Together, the standards of software development, including version control, continuous integration, and software testing, ensure that the software is scalable, maintainable, and usable. This is especially important for software that is used by the scientific community, as it ensures that the software is working as expected at defined versions years after publishing scientific results.

Python as a Programming Language

Here, we provide a general overview of the Python programming language, explaining terms like *"type"*, *"method"*, etc., in order to prepare readers without prior programming experience for the following chapters. We also describe the design principles of Python to lay out the key concepts that differentiate Python compared to other programming languages. A more detailed tutorial on Python that's specialized for bioscientists is found in Ekmekci et al. 2016

Listing 2: Example of readable Python code. This one-line code returns the words (string) `'Hello, World!'` when executed. The command is straightforward and easy to understand.

```python
print("Hello, World!")
# Output: Hello, World!
```

Languages such as Python are considered *"high-level"*, which means that it is designed to be easy to read and write, but also independent of hardware by hiding (*"abstracting"*) underlying details (*The Python Language Reference*, 2024). A key principle of Python is the emphasis on implementing a syntax that is concise and close to human language (Listing 2, Listing 3).

Listing 3: Example of less readable code written in the low-level programming language C. This code is doing exactly the same as the Python code in Listing 2. The command is harder to understand because more steps are needed to access the same functionality, including the definition of a function

```c
#include <stdio.h>
int main() {
    printf("Hello, World!");
    return 0;
}
// Output: Hello, World!
```

Furthermore, Python is an *interpreted* language, which means that the code is executed line by line. This makes coding easier because the programmer can see the results of the code immediately after writing it, and error messages point to the exact line where the error occurred. This is in contrast to *compiled* languages, where the code has to be compiled into machine code before it can be executed. The advantage of compiled languages is that the code runs faster, because the machine code is optimized for the hardware.

Python automates tasks that would otherwise require an advanced understanding of computer hardware, like the need for manual allocation of memory space. This is achieved by

using a technique called *"garbage collection"*, which automatically frees memory space that is no longer needed by the program. This is a feature that is not present in low-level programming languages like C or C++, that were designed to maximize control over hardware.

Another hallmark of Python is its *dynamic typing system*. In Python the type is inferred automatically during code execution (Listing 4). This is in contrast to *statically* typed languages like C, where the type of a variable has to be declared explicitly and cannot be changed during code execution (Listing 5) (**?**).

Listing 4: Example of dynamic typing in Python. The variable "a" is assigned the value 5, which is of type integer. The variable "a" is then assigned the value "Hello, World!", which is of type string. Python allows dynamic re-assignment of variables with different types. Note that code after "#" is considered a comment and won't be executed.

```
a = 5  # Type integer
a = 5.0  # Type float
a = 'Hello, World!'  # Type string
a = True  # Type boolean
a = False  # Type boolean
a = [1, 2, 3]  # Type list of integers
a = {'name': 'Regina'}  # Type dictionary
```

Listing 5: Example of static typing in C. The variable "a" is declared as an integer (`int`), and can only store integers. The variable "a" is then assigned the value 5, which is an integer. The variable "a" is then assigned the value `'Hello, World!'`, which is a string. This results in a compilation error, because the variable "a" can only store integers. Note that code after "//" is considered a comment and won't be executed.

```
int a;  // Declare type as integer
a = 5;
a = 'Hello, World!';  // Compilation error!
```

Dynamic typing makes Python a very beginner-friendly language, since one does not have to keep track of the type of each variable. However, this also makes Python a slower language, because the interpreter has to check the type of each variable during code execution. Also, developing code with dynamic typing systems is prone to introducing bugs ("type errors"), because it allows unexperienced developers to convert variables from one type to another without noticing, leading to unexpected behavior. Hence, larger Python projects require disciplined adherence to programming conventions. One such convention is *type hinting*, which is a way to explicitly note the type of a variable. Type hinting does not have an effect on the code, but it makes the code more readable and understandable for other developers, and allows for development environments to detect type errors before execution (Listing 6) (van Rossum et al., 2014).

Python supports both functional and object-oriented programming paradigms. In functional programming, the code is written in a way that the program is a sequence of function calls,

Listing 6: Example of type hints used in Python. Explicitly stating the type of the variable is optional and does not change the behavior of the code as shown in Listing 4.

```python
a: int = 5
a: str = 'Hello, World!'
```

where each function call returns a value that is used in the next function call (Listing 7). This approach is useful when multiple actions have to be performed on the same data and the structure of the data is relatively simple, for example a string of a gene sequence.

Listing 7: Example of functional programming in Python. The code defines a function called "`find_restriction_site`" that finds the position of a restriction site in a gene. The function "`cut`" uses the function "`find_restriction_site`" to cut the gene at the restriction site.

```python
def find_restriction_site(gene: str):
    return gene.find('GCGC')

def cut(gene: str):
    position = find_restriction_site(gene)
    return gene[position:]

gene1 = 'TGAGCTGAGCTGATGCGCTATATTTAGGCG'
gene1_cut = cut(gene1)
print(gene1_cut)
# Output: GCGCTATATTTAGGCG
```

When the data itself gains in complexity, for example when storing not just the gene sequence, but also the promotor sequence, an object-oriented approach is more suitable (Listing 8). Object-oriented programming is a programming paradigm that uses objects and classes. An object is a collection of both data and functions, and a class is a blueprint for creating objects. The data of an object is stored as attributes. Functions that are associated with an object are called methods.

Listing 8: Example of object oriented programming in Python. The class is called "`Gene`" and has four methods, "`__init__`", "`find_promotor`", "`find_restriction_site`" and "`cut`". The method "`__init__`" is called when creating ("initializing") an object, which fills the object with user-defined data. The parameter "`self`" is used to reference the object itself internally. "`find_promotor`" is a method that finds the position of the promotor in the gene and is called during object initialization.

```python
class Gene:
    def __init__(self, sequence: str):
        self.sequence: str = sequence # Save sequence as attribute
        self.promotor: str = self.find_promotor()
    def find_promotor(self):
        return self.sequence.find('TATA')
    def find_restriction_site(self):
        return self.sequence.find('GCGC')
    def cut(self):
        position = self.find_restriction_site()
        return self.sequence[position:]
```

```
12
13 gene1 = Gene(sequence='TGAGCTGAGCTGATGCGCTATATTTAGGCG') # Create object
14 gene1_cut = gene1.cut() # Call the method cut
15 print(gene1_cut) # Show result
16 # Output: GCGCTATATTTAGGCG
```

A major benefit of using an object oriented versus a functional approach is that the data itself programmable, enabling the programmer to define the behavior of the data itself through methods. This is achieved by using the keyword "self" to reference the object itself inside the class. For example, one could extend the class "Gene" with a method that finds the promotor of the gene and stores it as an attribute (Listing 8).

When designing software, both functional and object oriented programming can be used together, where object oriented programming is often used to design the program's overall architecture, and functional programming is used to implement the algorithms of the program's features. This allows for scalability of the software, as every single class is extended through the addition of new methods. Furthermore, classes can be expanded in their functionalities through inheritance (Listing 9).

Listing 9: Example of inheritance in Python. The class "mRNA" inherits from the class "Gene". The class "mRNA" has two methods, "__init__" and "find_stopcodon". The method "find_stopcodon" finds the position of stop codons.

```
1  # Define a class called mRNA inheriting from the class Gene
2  class mRNA(Gene):
3      def __init__(self, sequence: str):
4          super().__init__(sequence)  # Get attributes from parent class
5          self.sequence.replace('T', 'U')  # Replace thymine with uracil
6      def find_stopcodons(self):
7          return self.sequence.find('UGA')
8
9  mrna1 = mRNA(sequence='TGAGCTGAGCTGATGCGCTATATTTAGGCG') # Create object
10 print(mrna1.find_stopcodons())  # Call the method translate
11 # Output: [0, 5, 10]
```

Inheritance is a feature of object-oriented programming that allows a class to access every attribute and method of a parent class. For example, one could extend the class "Gene" with a class "mRNA", by writing a class "mRNA" that inherits from the class "Gene".

Together, Python is not just beginner-friendly, but also well respected for its ease in development, which is why it is widely used in professional settings for web development, data analysis, machine learning, biosciences and more (Ekmekci et al., 2016; Rayhan & Gross, 2023).

The Potential of Python Data Science Packages for Biomedicine

Python includes a vast number of built-in packages used for basic data-types, software development, simple math operations, etc., (*The Python Language Reference*, 2024). Still, Python relies on packages developed by its users to provide specialized tools for data analysis. A Python package consists of multiple Python *modules*, where each module is a text-file with a `.py` ending containing Python code. Famous examples of such packages are `pytorch` and `tensorflow`, that are used to build models of artificial intelligence, including *ChatGPT* (Paszke et al., 2019; Abadi et al., 2016; Radford et al., 2019). Here, we outlay the most important packages used for `plotastic` in Chapter 2 and present examples how these packages are utilized in modern biomedical research.

`Interactive Python`: The standard Python interface is insufficient for data science, because it lacks the tools to quickly and conveniently visualize and explore data. `IPython` can be understood as an enhanced version of the standard Python interpreter, designed to improve the interactivity of Python code execution (Perez & Granger, 2007). `IPython` introduces features like rich media support to display graphics, but also helps users to use correct python data types through dynamic type introspection, detecting errors in the code. This functionality is akin to what *MATLAB* and *RStudio* provide through their advanced graphical user interfaces and extensive debugging tools. `IPython` is most often utilized in the form of *Jupyter Notebooks*.

`Jupyter`: `Jupyter` is an evolution of `IPython`, introducing the *Jupyter notebook* format, which has the file-ending `.ipynb` (Kluyver et al., 2016). Jupyter Notebooks are documents that combine both code and text structured as *code cells* and *markdown cells*, respectively. Markdown cells allow the author to provide additional information with text formatting, for example structuring the document with headings and subheadings, adding hyperlinks, images and mathematical formulas. Code cells can be executed individually, displaying the output directly below the cell. This allows for an interactive exploration of data, but also makes Jupyter Notebooks a very human-readable format that outlays data analysis in a clear manner with precise and reproducible documentation of all data processing steps. A major benefit of Jupyter Notebooks are interchangable *Kernels*, allowing the execution of code in different programming languages, such as R, Julia, and C++ (Giorgi et al., 2022). Today, Jupyter Notebooks have become a standard format compatible with collaborative platforms like *Google Colab* and *JupyterLab*, but also professional software development tools like *VS Code*, and *PyCharm*. For biomedical research, Jupyter Notebooks hold great potential to improve reproducibility, as they provide a standardized format to present data analyses, and are found in the supplemental of modern publications of both bioinformatics and wet-lab research (Taskiran et al., 2024; Bosch-Queralt et al., 2022; Howe & Chain, 2015).

NumPy: Central processing units (CPU) usually execute one instruction on one data point at a time. For manipulating tabular data, this is inefficient as the same instruction must be repeatedly loaded for every data point. `NumPy` accelerates the mathematical capabilities of Python by enabling large-scale operations on multi-dimensional arrays and matrices with high efficiency (Harris et al., 2020). One key feature of `NumPy` is the implementation of "vectorization" or SIMD (Single Instruction, Multiple Data) instructions. SIMD allows multiple data points to be processed simultaneously, significantly speeding up operations that are inherently parallelizable, such as matrix addition or multiplication. `NumPy`'s syntax and functional approach to array manipulation have set a standard for matrix computation, influencing the design of advanced AI frameworks such as `PyTorch` and `mlx`, which mirrors several of `NumPy's` functionalities to facilitate ease of use for those familiar with `NumPy` (Paszke et al., 2019; *Ml-Explore/Mlx*, 2024). This standardization has made `NumPy` an attractive tool not only in genomics (Ding et al., 2023), but also for modern clinical applications like imaging technologies and augmented-reality in surgery (Thompson et al., 2020).

Pandas: Tables are the most common way to store experimental results. `Pandas` extends Python with a tabular datatype, called `DataFrame`, which allows for easy data manipulation with integrated indexing (Mckinney, 2011). The intuitive interface of Pandas can be likened to *Microsoft Excel*; however, it is vastly more powerful due to its speed, functionality, and ability to handle larger datasets, e.g. by running efficient `numpy` vectorization in the background. Unlike *Excel*, Pandas enables automation by summarizing processing commands into scripts, documenting each step, and ensuring reproducibility. `Pandas` is used in biomedicine for data wrangling, data cleaning, and data analysis, as it allows for the integration of multiple data sources into a single table (Santos et al., 2020).

matplotlib: `matplotlib` is a plotting library that provides a wide range of static, animated, and interactive plots and graphs (Hunter, 2007). It serves as the foundation for many visualization tools and is particularly valued for its flexibility and customization options. For example, `Pandas` uses `matplotlib` to plot column datapoints directly from a `DataFrame` object, creating histograms or scatter plots, which is useful for preliminary data analysis and checking data distributions. However, `matplotlib` uses a low-level syntax, hence plots generated by `matplotlib` can be cumbersome to format and customize.

seaborn: While the low-level syntax of `matplotlib` is valued for its flexibility, formatting publication grade plots can be laborious, and its inconsistent syntax can make it difficult to remember the correct commands for different plot types. `seaborn` is a high-level interface on top of `matplotlib` that offers a more intuitive and highly standardized syntax across a wide array of plot types (Waskom, 2021). `seaborn` also integrates closely with `Pandas` data structures: It automatically groups datapoints, calculates measures of both central tendency (e.g. mean,

median) and variance (e.g. standard deviation), and displays them into the plot (e.g. error bars). This completely replaces manual calculation of descriptive statistics. `seaborn` also offers intuitive grouping (*facetting*) of data points, which simplifies the creation of complex visualizations involving multidimensional data, making it easier to reveal patterns and relationships via color encoding, faceting, and automated statistical fits. This is particularly useful in biomedical research for visualizing and understanding complex datasets, such as large quantities of protein data (Krzywinski & Savig, 2013; Weiss, 2022). `seaborn` could indirectly contribute to improving reproducibility in biomedical research by making visualizations of complex data very accessible through an easy and standardized syntax.

`Pingouin`: Integrating both data visualization and statistical analysis is beneficial for researchers who wish to conduct advanced statistical analysis without switching between different software environments. `Pingouin` is designed to be a user-friendly statistical tool that offers a straightforward syntax for performing statistical tests, which are commonly implemented in R (Vallat, 2018). Unlike R, `Pingouin` integrates seamlessly within the Python ecosystem, which allows combining data manipulation, analysis, and visualization all in one platform. This improves reproducibility by reducing the number of software tools required to analyze data. Despite its potential to streamline the data analysis process, `Pingouin` has not been widely adopted by biomedical research, yet. One example of a study that utilized `Pingouin` is the work of Kelly et al. (2023) in the field of Patient Public Involvement (PPI), producing an ethical matrix that allows for the inclusion of stakeholder opinion in medical research design. This lack of `Pingouin`'s adoption in biomedicine could be due to recent development and the dominance of R in the field. However, since Python offers multiple benefits over R in syntax, software development, runtime performance and integration with other tools (like including performant C++ code), `Pingouin` is an attractive standard for future statistical analyses in biomedicine (Gorelick & Ozsvald, 2020).

Together, these python packages form the backbone of modern data analysis in Python, often times combining software from different languages to accelerate certain features, while retaining the ease of use and readability that Python is known for. This is particularly advantageous in the field of biomedicine, where the requirements of modern data analysis are often complex and require a high degree of flexibility and customization.

Aims

This project defines these aims:

- Characterize the interaction between myeloma cells and mesenchymal stromal cells
- Develop methods
- Face the challenge of time-dependent cell adhesion through
- Provide tools to analyze multidimensional cell adhesion data

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., … Zheng, X. (2016, March). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems* (No. arXiv:1603.04467). arXiv. Retrieved 2024-03-07, from http://arxiv.org/abs/1603.04467 doi: 10.48550/arXiv.1603.04467

Aggarwal, R., Ghobrial, I. M., & Roodman, G. D. (2006, October). Chemokines in multiple myeloma. *Experimental hematology*, *34*(10), 1289–1295. Retrieved 2023-04-02, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3134145/ doi: 10.1016/j.exphem.2006.06.017

Akhmetzyanova, I., McCarron, M. J., Parekh, S., Chesi, M., Bergsagel, P. L., & Fooksman, D. R. (2020). Dynamic CD138 surface expression regulates switch between myeloma growth and dissemination. *Leukemia*, *34*(1), 245–256. Retrieved 2023-04-04, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6923614/ doi: 10.1038/s41375-019-0519-4

Alcorta-Sevillano, N., Macías, I., Infante, A., & Rodríguez, C. I. (2020, December). Deciphering the Relevance of Bone ECM Signaling. *Cells*, *9*(12), 2630. Retrieved 2023-12-20, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7762413/ doi: 10.3390/cells9122630

Alsayed, Y., Ngo, H., Runnels, J., Leleu, X., Singha, U. K., Pitsillides, C. M., … Ghobrial, I. M. (2007, April). Mechanisms of regulation of CXCR4/SDF-1 (CXCL12)-dependent migration and homing in multiple myeloma. *Blood*, *109*(7), 2708–2717. doi: 10.1182/blood-2006-07-035857

Anders, S., Pyl, P. T., & Huber, W. (2015, January). HTSeq–a Python framework to work with high-throughput sequencing data. *Bioinformatics (Oxford, England)*, *31*(2), 166–169. doi: 10.1093/bioinformatics/btu638

Andrews, S. (2010). *FASTQC. A quality control tool for high throughput sequence data.*

Armstrong, R. A. (2014, September). When to use the Bonferroni correction. *Ophthalmic & Physiological Optics: The Journal of the British College of Ophthalmic Opticians (Optometrists)*, *34*(5), 502–508. doi: 10.1111/opo.12131

Baker, M. (2016, May). 1,500 scientists lift the lid on reproducibility. *Nature*, *533*(7604), 452–454. Retrieved 2024-04-22, from https://www.nature.com/articles/533452a doi: 10.1038/533452a

Bao, L., Lai, Y., Liu, Y., Qin, Y., Zhao, X., Lu, X., … Huang, X. (2013, September). CXCR4 is a good survival prognostic indicator in multiple myeloma patients. *Leukemia Research*, *37*(9), 1083–1088. doi: 10.1016/j.leukres.2013.06.002

Barzilay, R., Ben-Zur, T., Bulvik, S., Melamed, E., & Offen, D. (2009, May). Lentiviral delivery of LMX1a enhances dopaminergic phenotype in differentiated human bone marrow mesenchymal stem cells. *Stem cells and development*, *18*(4), 591–601. doi: 10.1089/scd.2008.0138

Begley, C. G., & Ioannidis, J. P. A. (2015, January). Reproducibility in science: Improving the standard for basic and preclinical research. *Circulation Research*, *116*(1), 116–126. doi: 10.1161/CIRCRESAHA.114.303819

Bianco, P. (2014). "Mesenchymal" stem cells. *Annual review of cell and developmental biology*, *30*, 677–704. doi: 10.1146/annurev-cellbio-100913-013132

Bladé, J., Beksac, M., Caers, J., Jurczyszyn, A., von Lilienfeld-Toal, M., Moreau, P., … Richardson, P. (2022, March). Extramedullary disease in multiple myeloma: A systematic literature review. *Blood Cancer Journal*, *12*(3), 1–10. Retrieved 2023-03-24, from https://www.nature.com/articles/s41408-022-00643-3 doi: 10.1038/s41408-022-00643-3

Blonska, M., Zhu, Y., Chuang, H. H., You, M. J., Kunkalla, K., Vega, F., & Lin, X. (2015, February). Jun-regulated genes promote interaction of diffuse large B-cell lymphoma with the microenvironment. *Blood*, *125*(6), 981–991. Retrieved 2023-03-01, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4319238/ doi: 10.1182/blood-2014-04-568188

Bondi, A. B. (2000, September). Characteristics of scalability and their impact on performance. In *Proceedings of the 2nd international workshop on Software and performance* (pp. 195–203). New York, NY, USA: Association for Computing Machinery. Retrieved 2024-03-07, from https://dl.acm.org/doi/10.1145/350391.350432 doi: 10.1145/350391.350432

Bosch-Queralt, M., Tiwari, V., Damkou, A., Vaculčiaková, L., Alexopoulos, I., & Simons, M. (2022, March). A fluorescence microscopy-based protocol for volumetric measurement of lysolecithin lesion-associated de- and re-myelination in mouse brain. *STAR protocols*, *3*(1), 101141. doi: 10.1016/j.xpro.2022.101141

Boswell, D., & Foucher, T. (2011). *The Art of Readable Code: Simple and Practical Techniques for Writing Better Code*. "O'Reilly Media, Inc.".

Bou Zerdan, M., Nasr, L., Kassab, J., Saba, L., Ghossein, M., Yaghi, M., … Chaulagain, C. P. (n.d.). Adhesion molecules in multiple myeloma oncogenesis and targeted therapy. *International Journal of Hematologic Oncology*, *11*(2), IJH39. Retrieved 2023-02-01, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9136637/ doi: 10.2217/ijh-2021-0017

Brandl, A., Solimando, A. G., Mokhtari, Z., Tabares, P., Medler, J., Manz, H., … Beilhack, A. (2022, March). Junctional adhesion molecule C expression specifies a CD138low/neg multiple myeloma cell population in mice and humans. *Blood Advances*, *6*(7), 2195–2206. Retrieved 2023-04-04, from https://doi.org/10.1182/bloodadvances.2021004354 doi: 10.1182/bloodadvances.2021004354

Brooke, J. (1996, January). SUS – a quick and dirty usability scale. In (pp. 189–194).

Burger, R., Guenther, A., Bakker, F., Schmalzing, M., Bernand, S., Baum, W., … Gramatzki, M. (2001). Gp130 and ras mediated signaling in human plasma cell line INA-6: A cytokine-regulated tumor model for plasmacytoma. *The Hematology Journal: The Official Journal of the European Haematology Association*, *2*(1), 42–53. doi: 10.1038/sj.thj.6200075

Burger, R., Günther, A., Bakker, F., Schmalzing, M., Bernand, S., Baum, W., … Gramatzki, M. (2001, January). Gp130 and ras mediated signaling in human plasma cell line INA6: A cytokine-regulated tumor model for plasmacytoma. *Hematology Journal - HEMATOL J*, *2*, 42–53. doi: 10.1038/sj.thj.6200075

Bustin, S. A. (2014, December). The reproducibility of biomedical research: Sleepers awake! *Biomolecular Detection and Quantification*, *2*, 35–42. Retrieved 2024-03-18, from https://www.sciencedirect.com/science/article/pii/S2214753515000030 doi: 10.1016/j.bdq.2015.01.002

Caplan, A. (1991). Mesenchymal stem cells. *Journal of orthopaedic research : official publication of the Orthopaedic Research Society*, *9*(5), 641–50. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/1870029 doi: 10.1002/jor.1100090504

Caplan, A. I. (1994, July). The mesengenic process. *Clinics in plastic surgery*, *21*(3), 429–435.

Carlson, M. (2016). Org.Hs.eg.db. *Bioconductor*. Retrieved 2023-06-09, from http://bioconductor.org/packages/org.Hs.eg.db/ doi: 10.18129/B9.bioc.org.Hs.eg.db

Chacon, S., & Straub, B. (2024, March). *Git - Book*. Retrieved 2024-03-07, from https://git-scm.com/book/de/v2

Charlier, F., Weber, M., Izak, D., Harkin, E., Magnus, M., Lalli, J., … Repplinger, S. (2022, October). *Trevismd/statannotations: V0.5*. Zenodo. Retrieved 2023-11-16, from https://zenodo.org/record/7213391 doi: 10.5281/ZENODO.7213391

Chatterjee, M., Hönemann, D., Lentzsch, S., Bommert, K., Sers, C., Herrmann, P., … Bargou, R. C. (2002, November). In the presence of bone marrow stromal cells human multiple myeloma cells become independent of the IL-6/gp130/STAT3 pathway. *Blood*, *100*(9), 3311–3318. doi: 10.1182/blood-2002-01-0102

Cooper, G. M. (2000). The Cell: A Molecular Approach. 2nd Edition. *Sinauer Associates*, Proliferation in Development and Differentiation. Retrieved from https://www.ncbi.nlm.nih.gov/books/NBK9906/

da Silva Meirelles, L., Chagastelles, P. C., & Nardi, N. B. (2006, June). Mesenchymal stem cells reside in virtually

all post-natal organs and tissues. *Journal of cell science*, *119*(Pt 11), 2204–2213. doi: 10.1242/jcs.02932

Ding, W., Goldberg, D., & Zhou, W. (2023, August). PyComplexHeatmap: A Python package to visualize multimodal genomics data. *iMeta*, *2*(3), e115. doi: 10.1002/imt2.115

Dobin, A., Davis, C. A., Schlesinger, F., Drenkow, J., Zaleski, C., Jha, S., … Gingeras, T. R. (2013, January). STAR: Ultrafast universal RNA-seq aligner. *Bioinformatics*, *29*(1), 15–21. Retrieved 2023-05-27, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3530905/ doi: 10.1093/bioinformatics/bts635

Dominici, M., Le Blanc, K., Mueller, I., Slaper-Cortenbach, I., Marini, F., Krause, D., … Horwitz, E. (2006). Minimal criteria for defining multipotent mesenchymal stromal cells. The International Society for Cellular Therapy position statement. *Cytotherapy*, *8*(4), 315–317. doi: 10.1080/14653240600855905

Dotterweich, J., Schlegelmilch, K., Keller, A., Geyer, B., Schneider, D., Zeck, S., … Schütze, N. (2016, December). Contact of myeloma cells induces a characteristic transcriptome signature in skeletal precursor cells -Implications for myeloma bone disease. *Bone*, *93*, 155–166. doi: 10.1016/j.bone.2016.08.006

Dunn, W., Burgun, A., Krebs, M.-O., & Rance, B. (2017, November). Exploring and visualizing multidimensional data in translational research platforms. *Briefings in Bioinformatics*, *18*(6), 1044–1056. Retrieved 2024-04-23, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5862238/ doi: 10.1093/bib/bbw080

Duvall, P., Matyas, S., & Glover, A. (2007). *Continuous integration: Improving software quality and reducing risk*. Pearson Education. Retrieved from https://books.google.de/books?id=PV9qfEdv9L0C

Dziadowicz, S. A., Wang, L., Akhter, H., Aesoph, D., Sharma, T., Adjeroh, D. A., … Hu, G. (2022, January). Bone Marrow Stroma-Induced Transcriptome and Regulome Signatures of Multiple Myeloma. *Cancers*, *14*(4), 927. Retrieved 2022-10-25, from https://www.mdpi.com/2072-6694/14/4/927 doi: 10.3390/cancers14040927

Ekmekci, B., McAnany, C. E., & Mura, C. (2016, July). An Introduction to Programming for Bioscientists: A Python-Based Primer. *PLOS Computational Biology*, *12*(6), e1004867. Retrieved 2024-03-10, from https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1004867 doi: 10.1371/journal.pcbi.1004867

Evers, M., Schreder, M., Stühmer, T., Jundt, F., Ebert, R., Hartmann, T. N., … Leich, E. (2023, March). Prognostic value of extracellular matrix gene mutations and expression in multiple myeloma. *Blood Cancer Journal*, *13*(1), 43. doi: 10.1038/s41408-023-00817-7

Ewels, P., Magnusson, M., Lundin, S., & Käller, M. (2016, October). MultiQC: Summarize analysis results for multiple tools and samples in a single report. *Bioinformatics*, *32*(19), 3047–3048. Retrieved 2023-06-09, from https://doi.org/10.1093/bioinformatics/btw354 doi: 10.1093/bioinformatics/btw354

Excel, M. (2023, August). *Announcing Python in Excel: Combining the power of Python and the flexibility of Excel*. Retrieved 2024-03-11, from https://techcommunity.microsoft.com/t5/excel-blog/announcing-python-in-excel-combining-the-power-of-python-and-the/ba-p/3893439

Fazeli, P. K., Horowitz, M. C., MacDougald, O. A., Scheller, E. L., Rodeheffer, M. S., Rosen, C. J., & Klibanski, A. (2013, March). Marrow Fat and Bone—New Perspectives. *The Journal of Clinical Endocrinology and Metabolism*, *98*(3), 935–945. Retrieved 2023-12-20, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3590487/ doi: 10.1210/jc.2012-3634

Fernandez-Rebollo, E., Mentrup, B., Ebert, R., Franzen, J., Abagnale, G., Sieben, T., … Wagner, W. (2017, July). Human Platelet Lysate versus Fetal Calf Serum: These Supplements Do Not Select for Different Mesenchymal Stromal Cells. *Scientific Reports*, *7*, 5132. Retrieved 2023-05-02, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5506010/ doi: 10.1038/s41598-017-05207-1

Frassanito, M. A., Cusmai, A., Iodice, G., & Dammacco, F. (2001, January). Autocrine interleukin-6 production and highly malignant multiple myeloma: Relation with resistance to drug-induced apoptosis. *Blood*, *97*(2), 483–489. doi: 10.1182/blood.v97.2.483

Friedenstein, A., & Kuralesova, A. I. (1971, August). Osteogenic precursor cells of bone marrow in radiation

chimeras. *Transplantation*, *12*(2), 99–108.

Friedenstein, A. J., Piatetzky-Shapiro, I. I., & Petrakova, K. V. (1966, December). Osteogenesis in transplants of bone marrow cells. *Journal of embryology and experimental morphology*, *16*(3), 381–390.

Gabr, M. M., Zakaria, M. M., Refaie, A. F., Ismail, A. M., Abou-El-Mahasen, M. A., Ashamallah, S. A., … Ghoneim, M. A. (2013). Insulin-producing cells from adult human bone marrow mesenchymal stem cells control streptozotocin-induced diabetes in nude mice. *Cell transplantation*, *22*(1), 133–145. doi: 10.3727/096368912X647162

Garcés, J.-J., Simicek, M., Vicari, M., Brozova, L., Burgos, L., Bezdekova, R., … Paiva, B. (2020, February). Transcriptional profiling of circulating tumor cells in multiple myeloma: A new model to understand disease dissemination. *Leukemia*, *34*(2), 589–603. doi: 10.1038/s41375-019-0588-4

García-Ortiz, A., Rodríguez-García, Y., Encinas, J., Maroto-Martín, E., Castellano, E., Teixidó, J., & Martínez-López, J. (2021, January). The Role of Tumor Microenvironment in Multiple Myeloma Development and Progression. *Cancers*, *13*(2). Retrieved 2021-02-02, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7827690/ doi: 10.3390/cancers13020217

Gentleman. (n.d.). *Bioconductor - BiocViews.* Retrieved 2023-06-09, from https://bioconductor.org/packages/3.17/BiocViews.html

Ghobrial, I. M. (2012, July). Myeloma as a model for the process of metastasis: Implications for therapy. *Blood*, *120*(1), 20–30. Retrieved 2022-10-15, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3390959/ doi: 10.1182/blood-2012-01-379024

Giorgi, F. M., Ceraolo, C., & Mercatelli, D. (2022, April). The R Language: An Engine for Bioinformatics and Data Science. *Life*, *12*(5), 648. Retrieved 2024-04-21, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9148156/ doi: 10.3390/life12050648

Glavey, S. V., Naba, A., Manier, S., Clauser, K., Tahri, S., Park, J., … Ghobrial, I. M. (2017, November). Proteomic characterization of human multiple myeloma bone marrow extracellular matrix. *Leukemia*, *31*(11), 2426–2434. Retrieved 2023-09-05, from https://www.nature.com/articles/leu2017102 doi: 10.1038/leu.2017.102

Gomez-Cabrero, D., Abugessaisa, I., Maier, D., Teschendorff, A., Merkenschlager, M., Gisel, A., … Tegnér, J. (2014, March). Data integration in the era of omics: Current and future challenges. *BMC Systems Biology*, *8*(2), I1. Retrieved 2024-03-18, from https://doi.org/10.1186/1752-0509-8-S2-I1 doi: 10.1186/1752-0509-8-S2-I1

Gómez-López, G., Dopazo, J., Cigudosa, J. C., Valencia, A., & Al-Shahrour, F. (2019, May). Precision medicine needs pioneering clinical bioinformaticians. *Briefings in Bioinformatics*, *20*(3), 752–766. doi: 10.1093/bib/bbx144

Goodman, S. N., Fanelli, D., & Ioannidis, J. P. A. (2016, June). What does research reproducibility mean? *Science Translational Medicine*, *8*(341), 341ps12-341ps12. Retrieved 2024-03-18, from https://www.science.org/doi/10.1126/scitranslmed.aaf5027 doi: 10.1126/scitranslmed.aaf5027

Gorelick, M., & Ozsvald, I. (2020). *High Performance Python: Practical Performant Programming for Humans.* "O'Reilly Media, Inc.".

Gosselin, R.-D. (2021, February). Insufficient transparency of statistical reporting in preclinical research: A scoping review. *Scientific Reports*, *11*(1), 3335. Retrieved 2024-03-11, from https://www.nature.com/articles/s41598-021-83006-5 doi: 10.1038/s41598-021-83006-5

Gramatzki, M., Burger, R., Trautman, U., Marschalek, R., Lorenz, H., Hansen-Hagge, T., … Kalden, J. (1994). Two new interleukin-6 dependent plasma cell lines carrying a chromosomal abnormality involving the IL-6 gene locus. , *84 Suppl. 1*, 173a-173a. Retrieved 2023-03-24, from https://www.cellosaurus.org/cellopub/CLPUB00060

Greenstein, S., Krett, N. L., Kurosawa, Y., Ma, C., Chauhan, D., Hideshima, T., … Rosen, S. T. (2003, April). Characterization of the MM.1 human multiple myeloma (MM) cell lines: A model system to elucidate the characteristics, behavior, and signaling of steroid-sensitive and -resistant MM cells. *Experimental Hematology*, *31*(4), 271–282. doi: 10.1016/s0301-472x(03)00023-7

Gronthos, S., Graves, S. E., Ohta, S., & Simmons, P. J. (1994, December). The STRO-1+ fraction of adult human bone marrow contains the osteogenic precursors. *Blood*, *84*(12), 4164–4173.

Harrington, D. P., & Fleming, T. R. (1982). A Class of Rank Test Procedures for Censored Survival Data. *Biometrika*, *69*(3), 553–566. Retrieved 2023-08-07, from https://www.jstor.org/stable/2335991 doi: 10.2307/2335991

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., … Oliphant, T. E. (2020, September). Array programming with NumPy. *Nature*, *585*(7825), 357–362. Retrieved 2023-08-09, from https://www.nature.com/articles/s41586-020-2649-2 doi: 10.1038/s41586-020-2649-2

Hideshima, T., Mitsiades, C., Tonon, G., Richardson, P. G., & Anderson, K. C. (2007, August). Understanding multiple myeloma pathogenesis in the bone marrow to identify new therapeutic targets. *Nature Reviews Cancer*, *7*(8), 585–598. Retrieved 2023-02-07, from https://www.nature.com/articles/nrc2189 doi: 10.1038/nrc2189

Hose, D., Rème, T., Hielscher, T., Moreaux, J., Messner, T., Seckinger, A., … Goldschmidt, H. (2011, January). Proliferation is a central independent prognostic factor and target for personalized and risk-adapted treatment in multiple myeloma. *Haematologica*, *96*(1), 87–95. doi: 10.3324/haematol.2010.030296

Hothorn, T., & Lausen, B. (n.d.). *Maximally Selected Rank Statistics in R.* Retrieved from http://cran.r-project.org/web/packages/maxstat/index.html.

Howe, A., & Chain, P. S. G. (2015). Challenges and opportunities in understanding microbial communities with metagenome assembly (accompanied by IPython Notebook tutorial). *Frontiers in Microbiology*, *6*, 678. doi: 10.3389/fmicb.2015.00678

Hu, X., Villodre, E. S., Larson, R., Rahal, O. M., Wang, X., Gong, Y., … Debeb, B. G. (2021, January). Decorin-mediated suppression of tumorigenesis, invasion, and metastasis in inflammatory breast cancer. *Communications Biology*, *4*(1), 72. doi: 10.1038/s42003-020-01590-0

Huang, S.-Y., Lin, H.-H., Yao, M., Tang, J.-L., Wu, S.-J., Hou, H.-A., … Tien, H.-F. (2015). Higher Decorin Levels in Bone Marrow Plasma Are Associated with Superior Treatment Response to Novel Agent-Based Induction in Patients with Newly Diagnosed Myeloma - A Retrospective Study. *PloS One*, *10*(9), e0137552. doi: 10.1371/journal.pone.0137552

Hunter, J. D. (2007, May). Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering*, *9*(3), 90–95. Retrieved 2023-11-15, from https://ieeexplore.ieee.org/document/4160265 doi: 10.1109/MCSE.2007.55

Incerti, D., Thom, H., Baio, G., & Jansen, J. P. (2019, May). R You Still Using Excel? The Advantages of Modern Software Tools for Health Technology Assessment. *Value in Health*, *22*(5), 575–579. Retrieved 2024-03-11, from https://www.sciencedirect.com/science/article/pii/S1098301519300506 doi: 10.1016/j.jval.2019.01.003

Ioannidis, J. P. A. (2005, August). Why Most Published Research Findings Are False. *PLOS Medicine*, *2*(8), e124. Retrieved 2024-04-22, from https://journals.plos.org/plosmedicine/article?id=10.1371/journal.pmed.0020124 doi: 10.1371/journal.pmed.0020124

Jansen, B. J. H., Gilissen, C., Roelofs, H., Schaap-Oziemlak, A., Veltman, J. A., Raymakers, R. A. P., … Adema, G. J. (2010, April). Functional differences between mesenchymal stem cell populations are reflected by their transcriptome. *Stem cells and development*, *19*(4), 481–490. doi: 10.1089/scd.2009.0288

Kaplan, E. L., & Meier, P. (1958, June). Nonparametric Estimation from Incomplete Observations. *Journal of*

the American Statistical Association, *53*(282), 457–481. Retrieved 2023-08-07, from http://www.tandfonline.com/doi/abs/10.1080/01621459.1958.10501452 doi: 10.1080/01621459.1958.10501452

Katz, B.-Z. (2010, June). Adhesion molecules—The lifelines of multiple myeloma cells. *Seminars in Cancer Biology*, *20*(3), 186–195. Retrieved 2021-07-04, from https://linkinghub.elsevier.com/retrieve/pii/S1044579X10000246 doi: 10.1016/j.semcancer.2010.04.003

Kawano, M. M., Huang, N., Tanaka, H., Ishikawa, H., Sakai, A., Tanabe, O., … Kuramoto, A. (1991, December). Homotypic cell aggregations of human myeloma cells with ICAM-1 and LFA-1 molecules. *British Journal of Haematology*, *79*(4), 583–588. doi: 10.1111/j.1365-2141.1991.tb08085.x

Kazman, R., Bianco, P., Ivers, J., & Klein, J. (2020, December). *Maintainability* (Report). Carnegie Mellon University. Retrieved 2024-03-07, from https://kilthub.cmu.edu/articles/report/Maintainability/12954908/1 doi: 10.1184/R1/12954908.v1

Kelly, B. S., Kirwan, A., Quinn, M. S., Kelly, A. M., Mathur, P., Lawlor, A., & Killeen, R. P. (2023, May). The ethical matrix as a method for involving people living with disease and the wider public (PPI) in near-term artificial intelligence research. *Radiography (London, England: 1995)*, *29 Suppl 1*, S103-S111. doi: 10.1016/j.radi.2023.03.009

Kibler, C., Schermutzki, F., Waller, H. D., Timpl, R., Müller, C. A., & Klein, G. (1998, June). Adhesive interactions of human multiple myeloma cell lines with different extracellular matrix molecules. *Cell Adhesion and Communication*, *5*(4), 307–323. doi: 10.3109/15419069809040300

Kluyver, T., Ragan-Kelley, B., Pérez, F., Granger, B., Bussonnier, M., Frederic, J., … Jupyter Development Team (2016). *Jupyter Notebooks—a publishing format for reproducible computational workflows*. Retrieved 2024-04-20, from https://ui.adsabs.harvard.edu/abs/2016ppap.book...87K doi: 10.3233/978-1-61499-649-1-87

Krekel, H., Oliveira, B., Pfannschmidt, R., Bruynooghe, F., Laugher, B., & Bruhin, F. (2004). *Pytest*. Retrieved from https://github.com/pytest-dev/pytest

Krzywinski, M., & Savig, E. (2013, July). Multidimensional data. *Nature methods*, *10*(7), 595. Retrieved 2024-04-22, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6092027/

Kuric, M., & Ebert, R. (2024, March). Plotastic: Bridging Plotting and Statistics in Python. *Journal of Open Source Software*, *9*(95), 6304. Retrieved 2024-03-11, from https://joss.theoj.org/papers/10.21105/joss.06304 doi: 10.21105/joss.06304

Lai, T.-Y., Cao, J., Ou-Yang, P., Tsai, C.-Y., Lin, C.-W., Chen, C.-C., … Lee, C.-Y. (2022, April). Different methods of detaching adherent cells and their effects on the cell surface expression of Fas receptor and Fas ligand. *Scientific Reports*, *12*(1), 5713. Retrieved 2023-06-01, from https://www.nature.com/articles/s41598-022-09605-y doi: 10.1038/s41598-022-09605-y

Leek, J. T., & Peng, R. D. (2015, April). Statistics: P values are just the tip of the iceberg. *Nature*, *520*(7549), 612–612. Retrieved 2024-04-22, from https://www.nature.com/articles/520612a doi: 10.1038/520612a

Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., … 1000 Genome Project Data Processing Subgroup (2009, August). The Sequence Alignment/Map format and SAMtools. *Bioinformatics*, *25*(16), 2078–2079. Retrieved 2023-06-09, from https://doi.org/10.1093/bioinformatics/btp352 doi: 10.1093/bioinformatics/btp352

Maichl, D. S., Kirner, J. A., Beck, S., Cheng, W.-H., Krug, M., Kuric, M., … Jundt, F. (2023, September). Identification of NOTCH-driven matrisome-associated genes as prognostic indicators of multiple myeloma patient survival. *Blood Cancer Journal*, *13*(1), 1–6. Retrieved 2023-09-05, from https://www.nature.com/articles/s41408-023-00907-6 doi: 10.1038/s41408-023-00907-6

McKinney, W. (2010, January). Data Structures for Statistical Computing in Python. In (pp. 56–61). doi: 10.25080/Majora-92bf1922-00a

Mckinney, W. (2011, January). Pandas: A Foundational Python Library for Data Analysis and Statistics.

Python High Performance Science Computer.

Mei, H., & Zhang, L. (2018, May). Can big data bring a breakthrough for software automation? *Science China Information Sciences*, *61*, 056101. doi: 10.1007/s11432-017-9355-3

Mesirov, J. P. (2010, January). Accessible Reproducible Research. *Science*, *327*(5964), 415–416. Retrieved 2024-04-22, from https://www.science.org/doi/10.1126/science.1179653 doi: 10.1126/science.1179653

Ml-explore/mlx. (2024, April). ml-explore. Retrieved 2024-04-21, from https://github.com/ml-explore/mlx

Moreno-Indias, I., Lahti, L., Nedyalkova, M., Elbere, I., Roshchupkin, G., Adilovic, M., … Zomer, A. L. (2021, February). Statistical and Machine Learning Techniques in Human Microbiome Studies: Contemporary Challenges and Solutions. *Frontiers in Microbiology*, *12*. Retrieved 2024-03-18, from https://www.frontiersin.org/journals/microbiology/articles/10.3389/fmicb.2021.635781/full doi: 10.3389/fmicb.2021.635781

Muruganandan, S., Roman, A. A., & Sinal, C. J. (2009, January). Adipocyte differentiation of bone marrow-derived mesenchymal stem cells: Cross talk with the osteoblastogenic program. *Cellular and molecular life sciences : CMLS*, *66*(2), 236–253. doi: 10.1007/s00018-008-8429-z

Myers, G. J., Sandler, C., & Badgett, T. (2011). *The art of software testing* (3rd ed.). Wiley Publishing. Retrieved from https://malenezi.github.io/malenezi/SE401/Books/114-the-art-of-software-testing-3-edition.pdf

Narzt, W., Pichler, J., Pirklbauer, K., & Zwinz, M. (1998, January). A Reusability Concept for Process Automation Software..

Newville, M., Stensitzki, T., Allen, D. B., & Ingargiola, A. (2014, September). *LMFIT: Non-Linear Least-Square Minimization and Curve-Fitting for Python.* Zenodo. Retrieved 2023-05-30, from https://zenodo.org/record/11813 doi: 10.5281/zenodo.11813

Nilsson, K., Bennich, H., Johansson, S. G., & Pontén, J. (1970, October). Established immunoglobulin producing myeloma (IgE) and lymphoblastoid (IgG) cell lines from an IgE myeloma patient. *Clinical and Experimental Immunology*, *7*(4), 477–489.

Nowotschin, S., & Hadjantonakis, A.-K. (2010, August). Cellular dynamics in the early mouse embryo: From axis formation to gastrulation. *Current opinion in genetics & development*, *20*(4), 420–427. doi: 10.1016/j.gde.2010.05.008

Okuno, Y., Takahashi, T., Suzuki, A., Ichiba, S., Nakamura, K., Okada, T., … Imura, H. (1991, February). In vitro growth pattern of myeloma cells in liquid suspension or semi-solid culture containing interleukin-6. *International Journal of Hematology*, *54*(1), 41–47.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., … Chintala, S. (2019, December). *PyTorch: An Imperative Style, High-Performance Deep Learning Library* (No. arXiv:1912.01703). arXiv. Retrieved 2024-03-07, from http://arxiv.org/abs/1912.01703 doi: 10.48550/arXiv.1912.01703

Peng, R. D. (2011, December). Reproducible Research in Computational Science. *Science*, *334*(6060), 1226–1227. Retrieved 2024-03-18, from https://www.science.org/doi/10.1126/science.1213847 doi: 10.1126/science.1213847

Perez, F., & Granger, B. E. (2007, May). IPython: A System for Interactive Scientific Computing. *Computing in Science & Engineering*, *9*(3), 21–29. Retrieved 2024-04-20, from https://ieeexplore.ieee.org/document/4160251 doi: 10.1109/MCSE.2007.53

Perneger, T. V. (1998, April). What's wrong with Bonferroni adjustments. *BMJ : British Medical Journal*, *316*(7139), 1236–1238. Retrieved 2021-11-24, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1112991/

Pittenger, M. F., Mackay, A. M., Beck, S. C., Jaiswal, R. K., Douglas, R., Mosca, J. D., … Marshak, D. R. (1999). Multilineage Potential of Adult Human Mesenchymal Stem Cells. , *284*(April), 143–148. doi: 10.1126/

science.284.5411.143

Polager, S., & Ginsberg, D. (2009, October). P53 and E2f: Partners in life and death. *Nature Reviews Cancer*, *9*(10), 738–748. Retrieved 2023-02-14, from https://www.nature.com/articles/nrc2718 doi: 10.1038/nrc2718

The Python Language Reference. (2024). Retrieved 2024-03-07, from https://docs.python.org/3/reference/index.html

R Core Team. (2018). *R: A language and environment for statistical computing* [Manual]. Vienna, Austria. Retrieved from https://www.R-project.org/

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language Models are Unsupervised Multitask Learners.. Retrieved 2024-03-07, from https://www.semanticscholar.org/paper/Language-Models-are-Unsupervised-Multitask-Learners-Radford-Wu/9405cc0d6169988371b2755e573cc28650d14dfe

Rajkumar, S. V., Dimopoulos, M. A., Palumbo, A., Blade, J., Merlini, G., Mateos, M.-V., … Miguel, J. F. S. (2014, November). International Myeloma Working Group updated criteria for the diagnosis of multiple myeloma. *The Lancet. Oncology*, *15*(12), e538-548. doi: 10.1016/S1470-2045(14)70442-5

Rajkumar, S. V., & Kumar, S. (2020, September). Multiple myeloma current treatment algorithms. *Blood Cancer Journal*, *10*(9), 94. Retrieved 2023-07-03, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7523011/ doi: 10.1038/s41408-020-00359-2

Ramakers, C., Ruijter, J. M., Deprez, R. H., & Moorman, A. F. (2003, March). Assumption-free analysis of quantitative real-time polymerase chain reaction (PCR) data. *Neuroscience Letters*, *339*(1), 62–66. Retrieved 2022-11-27, from https://linkinghub.elsevier.com/retrieve/pii/S0304394002014234 doi: 10.1016/S0304-3940(02)01423-4

Rayhan, A., & Gross, D. (2023). *The Rise of Python: A Survey of Recent Research.* doi: 10.13140/RG.2.2.27388.92809

Robinson, M. D., McCarthy, D. J., & Smyth, G. K. (2010, January). edgeR: A Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics (Oxford, England)*, *26*(1), 139–140. Retrieved from https://www.ncbi.nlm.nih.gov/pubmed/19910308 doi: 10.1093/bioinformatics/btp616

Sacchetti, B., Funari, A., Remoli, C., Giannicola, G., Kogler, G., Liedtke, S., … Bianco, P. (2016). No identical "mesenchymal stem cells" at different times and sites: Human committed progenitors of distinct origin and differentiation potential are incorporated as adventitial cells in microvessels. *Stem Cell Reports*, *6*(6), 897–913. Retrieved from http://dx.doi.org/10.1016/j.stemcr.2016.05.011 doi: 10.1016/j.stemcr.2016.05.011

Sandve, G. K., Nekrutenko, A., Taylor, J., & Hovig, E. (2013, October). Ten Simple Rules for Reproducible Computational Research. *PLoS Computational Biology*, *9*(10), e1003285. Retrieved 2024-03-07, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3812051/ doi: 10.1371/journal.pcbi.1003285

Santos, B. S., Silva, I., Ribeiro-Dantas, M. d. C., Alves, G., Endo, P. T., & Lima, L. (2020, October). COVID-19: A scholarly production dataset report for research analysis. *Data in Brief*, *32*, 106178. doi: 10.1016/j.dib.2020.106178

Sanz-Rodríguez, F., Ruiz-Velasco, N., Pascual-Salcedo, D., & Teixidó, J. (1999, December). Characterization of VLA-4-dependent myeloma cell adhesion to fibronectin and VCAM-1: VLA-4-dependent Myeloma Cell Adhesion. *British Journal of Haematology*, *107*(4), 825–834. Retrieved 2023-04-02, from http://doi.wiley.com/10.1046/j.1365-2141.1999.01762.x doi: 10.1046/j.1365-2141.1999.01762.x

Sarin, V., Yu, K., Ferguson, I. D., Gugliemini, O., Nix, M. A., Hann, B., … Wiita, A. P. (2020, October). Evaluating the efficacy of multiple myeloma cell lines as models for patient tumors via transcriptomic correlation analysis. *Leukemia*, *34*(10), 2754–2765. doi: 10.1038/s41375-020-0785-1

Seabold, S., & Perktold, J. (2010). Statsmodels: Econometric and Statistical Modeling with Python. In *Python in Science Conference* (pp. 92–96). Austin, Texas. Retrieved 2023-05-29, from https://conference.scipy.org/

proceedings/scipy2010/seabold.html doi: 10.25080/Majora-92bf1922-011

Seckinger, A., Delgado, J. A., Moser, S., Moreno, L., Neuber, B., Grab, A., … Vu, M. D. (2017, March). Target Expression, Generation, Preclinical Activity, and Pharmacokinetics of the BCMA-T Cell Bispecific Antibody EM801 for Multiple Myeloma Treatment. *Cancer Cell*, *31*(3), 396–410. Retrieved 2023-07-21, from https://www.cell.com/cancer-cell/abstract/S1535-6108(17)30016-8 doi: 10.1016/j.ccell.2017.02.002

Seckinger, A., Hillengass, J., Emde, M., Beck, S., Kimmich, C., Dittrich, T., … Hose, D. (2018). CD38 as Immunotherapeutic Target in Light Chain Amyloidosis and Multiple Myeloma-Association With Molecular Entities, Risk, Survival, and Mechanisms of Upfront Resistance. *Frontiers in Immunology*, *9*, 1676. doi: 10.3389/fimmu.2018.01676

Shenghui, H., Nakada, D., & Morrison, S. J. (2009). Mechanisms of Stem Cell Self-Renewal. *Annual Review of Cell and Developmental Biology*, *25*(1), 377–406. Retrieved from https://doi.org/10.1146/annurev.cellbio.042308.113248 doi: 10.1146/annurev.cellbio.042308.113248

Smith, A. M., Niemeyer, K. E., Katz, D. S., Barba, L. A., Githinji, G., Gymrek, M., … Vanderplas, J. T. (2018). Journal of Open Source Software (JOSS): Design and first-year review. *PeerJ Preprints*, *4*, e147. doi: 10.7717/peerj-cs.147

Solimando, A. G., Malerba, E., Leone, P., Prete, M., Terragna, C., Cavo, M., & Racanelli, V. (2022, September). Drug resistance in multiple myeloma: Soldiers and weapons in the bone marrow niche. *Frontiers in Oncology*, *12*, 973836. Retrieved 2022-10-23, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9533079/ doi: 10.3389/fonc.2022.973836

Sprynski, A. C., Hose, D., Caillot, L., Rème, T., Shaughnessy, J. D., Barlogie, B., … Klein, B. (2009, May). The role of IGF-1 as a major growth factor for myeloma cell lines and the prognostic relevance of the expression of its receptor. *Blood*, *113*(19), 4614–4626. Retrieved 2023-06-29, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2691749/ doi: 10.1182/blood-2008-07-170464

Standal, T., Seidel, C., Plesner, T., Sanderson, R., Waage, A., Børset, M., & Sundan, A. (2002, November). Osteoprotegerin is bound, internalized, and degraded by multiple myeloma cells. *Blood*, *100*, 3002–7. doi: 10.1182/blood-2002-04-1190

Stock, P., Bruckner, S., Winkler, S., Dollinger, M. M., & Christ, B. (2014, April). Human bone marrow mesenchymal stem cell-derived hepatocytes improve the mouse liver after acute acetaminophen intoxication by preventing progress of injury. *International journal of molecular sciences*, *15*(4), 7004–7028. doi: 10.3390/ijms15047004

Tabolacci, C., De Martino, A., Mischiati, C., Feriotto, G., & Beninati, S. (2019, January). The Role of Tissue Transglutaminase in Cancer Cell Initiation, Survival and Progression. *Medical Sciences*, *7*(2), 19. Retrieved 2023-03-17, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6409630/ doi: 10.3390/medsci7020019

Tai, Y.-T., Li, X.-F., Breitkreutz, I., Song, W., Neri, P., Catley, L., … Anderson, K. C. (2006, July). Role of B-cell-activating factor in adhesion and growth of human multiple myeloma cells in the bone marrow microenvironment. *Cancer Research*, *66*(13), 6675–6682. doi: 10.1158/0008-5472.CAN-06-0190

Tam, P. P., & Beddington, R. S. (1987, January). The formation of mesodermal tissues in the mouse embryo during gastrulation and early organogenesis. *Development (Cambridge, England)*, *99*(1), 109–126.

Taskiran, I. I., Spanier, K. I., Dickmänken, H., Kempynck, N., Pančíková, A., Ekşi, E. C., … Aerts, S. (2024, February). Cell-type-directed design of synthetic enhancers. *Nature*, *626*(7997), 212–220. Retrieved 2024-04-21, from https://www.nature.com/articles/s41586-023-06936-2 doi: 10.1038/s41586-023-06936-2

Team, T. P. D. (2020, February). *Pandas-dev/pandas: Pandas.* Zenodo. Retrieved from https://doi.org/10.5281/zenodo.3509134 doi: 10.5281/zenodo.3509134

Terpos, E., Ntanasis-Stathopoulos, I., Gavriatopoulou, M., & Dimopoulos, M. A. (2018, January). Pathogenesis of bone disease in multiple myeloma: From bench to bedside. *Blood Cancer Journal*, *8*(1), 7. doi: 10.1038/

s41408-017-0037-4

Thompson, S., Dowrick, T., Ahmad, M., Xiao, G., Koo, B., Bonmati, E., … Clarkson, M. J. (2020, July). SciKit-Surgery: Compact libraries for surgical navigation. *International Journal of Computer Assisted Radiology and Surgery*, *15*(7), 1075–1084. doi: 10.1007/s11548-020-02180-5

Ullah, I., Subbarao, R. B., & Rho, G. J. (2015). Human mesenchymal stem cells - current trends and future prospective Bioscience Reports.
doi: 10.1042/BSR20150025

Urashima, M., Chauhan, D., Uchiyama, H., Freeman, G., & Anderson, K. (1995, April). CD40 ligand triggered interleukin-6 secretion in multiple myeloma. *Blood*, *85*(7), 1903–1912. Retrieved 2021-02-01, from https://ashpublications.org/blood/article/85/7/1903/123565/CD40-ligand-triggered-interleukin6-secretion-in doi: 10.1182/blood.V85.7.1903.bloodjournal8571903

Vallat, R. (2018, November). Pingouin: Statistics in Python. *Journal of Open Source Software*, *3*(31), 1026. Retrieved 2023-05-29, from https://joss.theoj.org/papers/10.21105/joss.01026 doi: 10.21105/joss.01026

van Rossum, G., Lehtosalo, J., & Langa, L. (2014). *PEP 484 – Type Hints | peps.python.org.* Retrieved 2024-03-08, from https://peps.python.org/pep-0484/

Van Valckenborgh, E., Croucher, P. I., De Raeve, H., Carron, C., De Leenheer, E., Blacher, S., … Vanderkerken, K. (2004, September). Multifunctional role of matrix metalloproteinases in multiple myeloma: A study in the 5T2MM mouse model. *The American Journal of Pathology*, *165*(3), 869–878. doi: 10.1016/S0002-9440(10)63349-4

Viguet-Carrin, S., Garnero, P., & Delmas, P. D. (2006, March). The role of collagen in bone strength. *Osteoporosis International*, *17*(3), 319–336. Retrieved 2023-12-20, from https://doi.org/10.1007/s00198-005-2035-9 doi: 10.1007/s00198-005-2035-9

Wadgaonkar, R., Phelps, K. M., Haque, Z., Williams, A. J., Silverman, E. S., & Collins, T. (1999, January). CREB-binding protein is a nuclear integrator of nuclear factor-kappaB and p53 signaling. *The Journal of Biological Chemistry*, *274*(4), 1879–1882. doi: 10.1074/jbc.274.4.1879

Waskom, M. L. (2021, April). Seaborn: Statistical data visualization. *Journal of Open Source Software*, *6*(60), 3021. Retrieved 2023-03-26, from https://joss.theoj.org/papers/10.21105/joss.03021 doi: 10.21105/joss.03021

Webster, G. A., & Perkins, N. D. (1999, May). Transcriptional Cross Talk between NF-κB and p53. *Molecular and Cellular Biology*, *19*(5), 3485–3495. Retrieved 2023-07-04, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC84141/

Weetall, M., Hugo, R., Maida, S., West, S., Wattanasin, S., Bouhel, R., … Friedman, C. (2001, June). A Homogeneous Fluorometric Assay for Measuring Cell Adhesion to Immobilized Ligand Using V-Well Microtiter Plates. *Analytical Biochemistry*, *293*(2), 277–287. Retrieved 2022-09-25, from https://linkinghub.elsevier.com/retrieve/pii/S0003269701951401 doi: 10.1006/abio.2001.5140

Weiss, C. J. (2022, September). Visualizing protein big data using Python and Jupyter notebooks. *Biochemistry and Molecular Biology Education: A Bimonthly Publication of the International Union of Biochemistry and Molecular Biology*, *50*(5), 431–436. doi: 10.1002/bmb.21621

Wickham, H. (2014, September). Tidy Data. *Journal of Statistical Software*, *59*, 1–23. Retrieved 2023-11-15, from https://doi.org/10.18637/jss.v059.i10 doi: 10.18637/jss.v059.i10

Wilkins, A., Kemp, K., Ginty, M., Hares, K., Mallam, E., & Scolding, N. (2009, July). Human bone marrow-derived mesenchymal stem cells secrete brain-derived neurotrophic factor which promotes neuronal survival in vitro. *Stem cell research*, *3*(1), 63–70. doi: 10.1016/j.scr.2009.02.006

Wilkinson, M. D., Dumontier, M., Aalbersberg, I. J., Appleton, G., Axton, M., Baak, A., … Mons, B. (2016, March). The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data*,

3(1), 160018. Retrieved 2024-03-18, from https://www.nature.com/articles/sdata201618 doi: 10.1038/sdata.2016.18

Witwer, K. W. (2013, February). Data submission and quality in microarray-based microRNA profiling. *Clinical chemistry*, *59*(2), 392–400. Retrieved 2024-04-22, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4037921/ doi: 10.1373/clinchem.2012.193813

Wong, A. D., & Searson, P. C. (2017, November). Mitosis-mediated intravasation in a tissue-engineered tumor-microvessel platform. *Cancer research*, *77*(22), 6453–6461. Retrieved 2023-07-14, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5690825/ doi: 10.1158/0008-5472.CAN-16-3279

Xu, W., Zhang, X., Qian, H., Zhu, W., Sun, X., Hu, J., … Chen, Y. (2004, July). Mesenchymal stem cells from adult human bone marrow differentiate into a cardiomyocyte phenotype in vitro. *Experimental biology and medicine (Maywood, N.J.)*, *229*(7), 623–631.

Yang, A., Troup, M., & Ho, J. W. (2017, July). Scalability and Validation of Big Data Bioinformatics Software. *Computational and Structural Biotechnology Journal*, *15*, 379–386. Retrieved 2024-03-07, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5537105/ doi: 10.1016/j.csbj.2017.07.002

Zeissig, M. N., Zannettino, A. C. W., & Vandyke, K. (2020, December). Tumour Dissemination in Multiple Myeloma Disease Progression and Relapse: A Potential Therapeutic Target in High-Risk Myeloma. *Cancers*, *12*(12). Retrieved 2021-02-03, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7761917/ doi: 10.3390/cancers12123643

Zerbino, D. R., Achuthan, P., Akanni, W., Amode, M. R., Barrell, D., Bhai, J., … Flicek, P. (2018, January). Ensembl 2018. *Nucleic Acids Research*, *46*(D1), D754-D761. Retrieved 2023-05-27, from https://doi.org/10.1093/nar/gkx1098 doi: 10.1093/nar/gkx1098

Zhou, F., Meng, S., Song, H., & Claret, F. X. (2013, November). Dickkopf-1 is a key regulator of myeloma bone disease: Opportunities and challenges for therapeutic intervention. *Blood reviews*, *27*(6), 261–267. Retrieved 2023-02-18, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4133945/ doi: 10.1016/j.blre.2013.08.002

Zhou, Y., Zhou, B., Pache, L., Chang, M., Khodabakhshi, A. H., Tanaseichuk, O., … Chanda, S. K. (2019, April). Metascape provides a biologist-oriented resource for the analysis of systems-level datasets. *Nature Communications*, *10*(1), 1523. Retrieved 2023-02-09, from https://www.nature.com/articles/s41467-019-09234-6 doi: 10.1038/s41467-019-09234-6

Appendices

A Supplementary Data & Methods

A.1 Figures

A.2 Tables

A.3 Materials & Methods

B Documentation of plotastic