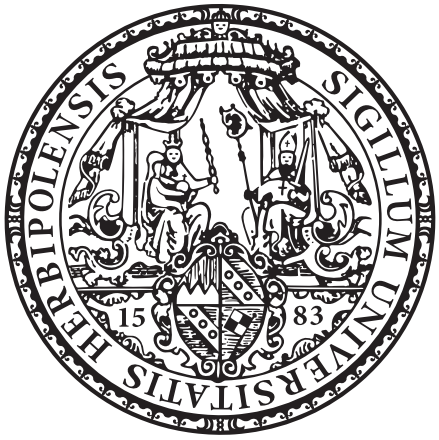


Development and Semi-Automated Analysis of an *in vitro* Dissemination Model for Myeloma Cells Interacting with Mesenchymal Stromal Cells

Entwicklung und semi-automatisierte Analyse eines *in vitro* Modells für die Disseminierung von Myelomzellen in Interaktion mit mesenchymalen Stromazellen

---

Doctoral Thesis for a Doctoral Degree

at the

GRADUATE SCHOOL OF LIFE SCIENCES,
JULIUS-MAXIMILIANS-UNIVERSITÄT WÜRZBURG,
SECTION BIOMEDICINE

submitted by

Martin Kuric

from

Bad Neustadt a.d. Saale

Würzburg, 2024

Submitted on: .....
Office stamp

Members of the *Promotionskomitee*:

Chairperson:	Prof. Dr. Uwe Gbureck
Primary Supervisor:	Prof. Dr. rer. nat. Regina Ebert
Supervisor (Second):	Prof. Dr. med. Franziska Jundt
Supervisor (Third):	Prof. Dr. rer. nat. Torsten Blunk

Date of Public Defense: .....

Date of Receipt of Certificates: .....

This work was conducted at the Department of Musculoskeletal Tissue Regeneration (Bernhard-Heine-Centre for Locomotive Research), University of Würzburg from 08.04.2018 to 31.03.2024 under the supervision of Prof Dr. rer. nat. Regina Ebert.

Acknowledgements

Lorem Ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum

Summary

This PhD thesis integrates biomedical research and data science, focusing on an *in vitro* model for studying myeloma cell dissemination and a Python-based tool, `plotastic`, for semi-automated analysis of multidimensional datasets. Two major challenges are adressed: (1) understanding the early steps of myeloma dissemination and (2) improving data analysis efficiency to address the complexity- and reproducibility bottlenecks currently present in biomedical research.

In the experimental component, primary human mesenchymal stromal cells (hMSCs) are co-cultured with INA-6 myeloma cells to study cell proliferation, attachment, and detachment via time-lapse microscopy. Key findings reveal that detachment often follows cell division, predominantly driven by daughter cells. Novel separation techniques were developed to isolate myeloma subpopulations for further characterization by RNAseq, cell viability, and apoptosis assays. Differential expression of adhesion and retention factors upregulated by INA-6 cells correlates with patient survival. Overall, this work provides insights into myeloma dissemination mechanisms and identifies genes that potentially counteract dissemination through adhesion, which could be relevant for the design of new therapeutics.

To manage complex data, a Python-based software named `plotastic` was developed that streamlines analysis and visualization of multidimensional datasets. `plotastic` is built on the idea that statistical analyses are performed based on how the data is visualized. This approach not only simplifies data analysis, but semi-automates analysis in a standardized statistical protocol. The thesis becomes a case study as it reflects on the application of `plotastic` to the *in vitro* model, demonstrating how the software facilitates rapid adjustments and refinements in data analysis and presentation. Such efficiency is crucial for handling semi-big data transparently, which — despite being managable — is complex enough to complicate analysis and reproducibility.

Together, this thesis illustrates the synergy between experimental methodologies and advanced data analysis tools. The *in vitro* model provides a robust platform for studying myeloma dissemination, while `plotastic` addresses the need for efficient data analysis. Combined, they offer a comprehensive approach to handling complex experiments, advancing both cancer biology and research practices, in support of exploratory and transparent analysis of challenging phenomena.

Zusammenfassung

Diese Doktorarbeit integriert biomedizinische Forschung und Datenwissenschaften und konzentriert sich auf ein *in vitro*-Modell zur Untersuchung der Dissemination von Myelomzellen sowie ein Python-basiertes Werkzeug, `plotastic`, zur semi-automatisierten Analyse multidimensionaler Datensätze. Zwei Hauptprobleme werden bearbeitet: (1) das Verständnis der frühen Schritte der Myelomdissemination und (2) die Verbesserung der Effizienz der Datenanalyse, um die derzeit in der biomedizinischen Forschung vorhandenen Engpässe bezüglich Komplexität und Reproduzierbarkeit zu adressieren.

Im experimentellen Teil werden primäre menschliche mesenchymale Stromazellen (hMSCs) mit INA-6-Myelomzellen kokultiviert, um Zellproliferation, Anhaftung und Ablösung mittels Zeitraffer-Mikroskopie zu untersuchen. Zentrale Erkenntnisse zeigen, dass die Ablösung oft auf die Zellteilung folgt und vorwiegend von Tochterzellen angetrieben wird. Neue Trennungstechniken wurden entwickelt, um Myelom-Subpopulationen für weitere Charakterisierungen durch RNAseq, Zellviabilität und Apoptose-Assays zu isolieren. Die differentielle Expression von Adhäsions- und Retentionsfaktoren, die durch INA-6 Zellen hochreguliert werden, korreliert mit dem Überleben der Patienten. Insgesamt liefert diese Arbeit Einblicke in die Mechanismen der Myelomdissemination und identifiziert Gene, die potenziell die Dissemination durch Adhäsion konterkarieren könnten, was für die Entwicklung neuer Therapeutika relevant sein könnte.

Zur Verwaltung komplexer Daten wurde eine Python-basierte Software namens `plotastic` entwickelt, welche die Analyse und Visualisierung multidimensionaler Datensätze optimiert. `plotastic` basiert auf der Idee, dass statistische Analysen basierend darauf durchgeführt werden, wie die Daten visualisiert werden. Dieser Ansatz vereinfacht nicht nur die Datenanalyse, sondern automatisiert sie auch teilweise in einem standardisierten statistischen Protokoll. Die Arbeit wird zu einer Fallstudie, da sie die Anwendung von `plotastic` auf das *in vitro*-Modell reflektiert und zeigt, wie die Software schnelle Anpassungen und Verfeinerungen in der Datenanalyse und -präsentation erleichtert. Eine solche Effizienz ist entscheidend für den transparenten Umgang mit Semi-Big-Data, die trotz ihrer Handhabbarkeit komplex genug ist, um die Analyse und Reproduzierbarkeit zu erschweren.

Zusammengefasst veranschaulicht diese Dissertation die Synergie zwischen experimentellen Methoden und fortgeschrittenen Werkzeugen der Datenanalyse. Das *in vitro*-Modell bietet eine robuste Plattform für die Untersuchung der Myelomdissemination, während `plotastic` den Bedarf an effizienter Datenanalyse adressiert. Zusammen bieten sie einen umfassenden Ansatz für die Bearbeitung komplexer Experimente, fördern sowohl die Krebsbiologie als auch die Forschungspraktiken und unterstützen die explorative und transparente Analyse herausfordernder Phänomene.

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

Human Mesenchymal Stem/Stromal Cells

Explaining what a mesenchymal stromal cell (MSC) is, is not such an easy task as one might expect. MSCs are derived from multiple MSCs different sources, serve a wide array of functions and are always isolated as a heterogenous group of cells. This makes it particularly challenging to find a consensus on their exact definition, nomenclature, exact function and *in vivo* differentiation potential. Therefore, the most effective approach to describe hMSCs is to present their historical context.

hMSCs first gained popularity as a stem cell. Stem cells lay the foundation of multicellular organisms. Embryonic stem cells orchestrate the growth and patterning during embryonic development, while adult stem cells are responsible for regeneration during adulthood. The classical definition of a stem cell is that of a relatively undifferentiated cell that divides asymmetrically, producing another stem cell and a differentiated cell (Cooper, 2000; Shenghui et al., 2009). Because of their significance in biology and regenerative medicine, stem cells have become a prominent subject in modern research. Especially human mesenchymal stromal cells (hMSCs) have proven to be a promising candidate in this context (Ullah et al., 2015).

Mesenchyme first appears in embryonic development during gastrulation. There, cells that are committed to a mesodermal fate, lose their cell junctions and exit the epithelial layer in order to migrate freely. This process is called epithelial-mesenchymal transition (Tam & Beddington, 1987; Nowotschin & Hadjantonakis, 2010). Hence, the term mesenchyme describes non-epithelial embryonic tissue differentiating into mesodermal lineages such as bone, muscles and blood. Interestingly, it was shown nearly twenty years earlier that cells within adult bone marrow seemed to have mesenchymal properties as they were able to differentiate into bone tissue (A. J. Friedenstein et al., 1966; A. Friedenstein & Kuralesova, 1971; Bianco, 2014). This was the origin of the "mesengenic process"-hypothesis: This concept states that mesenchymal stem cells serve as progenitors for multiple mesodermal tissues (bone, cartilage, muscle, marrow stroma, tendon, fat, dermis and connective tissue) during both adulthood and embryonic development (A. Caplan, 1991; A. I. Caplan, 1994). The mesenchymal nature of these cells (termed bone marrow stromal cells: BMSCs) was confirmed later when they were shown to differentiate into adipocytic (fat) and chondrocytic (cartilage) lineages (Pittenger et al., 1999). Since then, the term "mesenchymal stem cell" (MSC) has grown popular as an adult multipotent precursor to a couple of mesodermal tissues. hMSCs derived from bone marrow (hMSCs) were shown to differentiate into osteocytes, chondrocytes, adipocytes and cardiomyocytes (Gronthos et al., 1994; Muruganandan et al., 2009; Xu et al., 2004) Most impressively, these cells also exhibited ectodermal and endodermal differentiation potential, as they produced neuronal cells, pancreatic cells and hepatocytes (Barzilay et al., 2009; Wilkins et al., 2009; Gabr et al., 2013; Stock et al., 2014).

Furthermore, cultures with MSC properties can be established from "virtually every postnatal organs and tissues", and not just bone marrow (da Silva Meirelles et al., 2006). However, it has to be noted that hMSCs can differ greatly in their transcription profile and *in vivo* differentiation potential depending on which tissue they originated from (Jansen et al., 2010; Sacchetti et al., 2016).

Since "hMSCs' are a heterogenous group of cells, they were defined by their *in vitro* characteristics. A minimal set of criteria are the following (Dominici et al., 2006): First, hMSCs must be plastic adherent. Second, they must express or lack a set of specific surface antigens (positive for CD73, CD90, CD105; negative for CD45, CD34, CD11b, CD19). Third, hMSCs must differentiate to osteoblasts, adipocytes and chondroblasts *in vitro*. Together, hMSCs exhibit diverse differentiation potentials and can be isolated from multiple sources of the body. This offers great opportunity for regenerative medicine, if the particular hMSC-subtype is properly characterized.

Multiple Myeloma

Multiple myeloma arises from clonal expansion of malignant plasma cells in the bone marrow (BM). At diagnosis, myeloma cells have disseminated to multiple sites in the skeleton and, in some cases, to virtually any tissue (Rajkumar & Kumar, 2020; Bladé et al., 2022).

Myeloma-hMSC Interactions

Since plasma cells can not survive outside the bone marrow, MM cells also require survival signals for growth and disease progression. These signals are produced by the bone marrow microenvironment, including ECM, MSCs and ACs (Kibler et al., 1998; García-Ortiz et al., 2021).

Myeloma Bone Disease

Bone is a two-phase system in which the mineral phase provides the stiffness and the collagen fibers provide the ductility and ability to absorb energy (Viguet-Carrin et al., 2006). On a molecular level, bone tissue is composed of extracellular matrix (ECM) proteins that are calcified by hydroxyapatite crystals. This ECM consists mostly of collagen type I, but also components with major regulatory activity, such as fibronectin and proteoglycans that are essential for healthy bone physiology (Alcorta-Sevillano et al., 2020). Bone tissue is actively remodeled by bone-forming osteoblasts and bone-degrading osteoclasts. Osteoblasts are derived from mesenchymal

stromal cells (MSCs) that reside in the bone marrow (A. J. Friedenstein et al., 1966; Pittenger et al., 1999). MSCs also give rise to adipocytes (ACs) to form Bone Marrow Adipose Tissue (BMAT), which can account for up to 70% of bone marrow volume (Fazeli et al., 2013).

MM indirectly degrades bone tissue by stimulating osteoclasts and inhibiting osteoblast differentiation, which leads to MM-related bone disease (MBD) (Glavey et al., 2017). MBD is present in 80% of patients at diagnosis and is characterized by osteolytic lesions, osteopenia and pathological fractures (Terpos et al., 2018).

Dissemination of Myeloma Cells

dissemination is still widely unclear - multistep process - invasion, intravasation, intravascular arrest, extravasation, colonization - overcome adhesion, retention, and dependency on the BM microenvironment - loss of adhesion factors such as CD138

Multidimensional Data in Biomedical Research

As modern biosciences advance, researchers increasingly encounter datasets that are influenced by a variety of independent variables, such as time, dosage, and environmental conditions. These variables introduce multidimensional complexity into datasets, challenging traditional analysis methods. For instance, cell adhesion studies, which are crucial for understanding cellular interactions and cancer metastasis, often require analyses across multiple time points and varying adhesion molecule concentrations, demonstrating a time-dependent variability that significantly impacts biological interpretations (Rebl et al., 2010; McKay et al., 1997; Bolado-Carrancio et al., 2020).

Multidimensional data encompass datasets where multiple *independent variables* (here reffered to as *factors*) can influence one *dependent variables* (*outcomes*) (Krzywinski & Savig, 2013). In biomedicine, dependent variables are often continuous (intervals or ratios), whereas independent variables are often categorical (ordinal or nominal), respectively. Categorical variables comprise discrete values called categories or *levels*, which are assigned to experimental conditions or measurement modalities, for example the factor *'time'* could comprise three levels: *'0 h'*, *'24 h'*, and *'48 h'*. Such setups are attractive, because they are compatible with common hypothesis tests, such as ANOVA etc. (Motulsky, 2018): If the levels of one factor are associated with a different outcome, that factor is considered to have an influence on the dependent variable. Multiple factors address multiple hypotheses, including the influence from each individual factor, but also potential interactions between factors. This makes it crucial to design analysis strategies that can reveal the true structure and value of the data (Krzywinski & Savig, 2013).

A primary example of multidimensional data is multiplex RT-qPCR, where the expression levels of various genes are measured across different samples under varying conditions (Bustin, 2014). Here, the dependent variable is typically the fold change expression values derived from $\Delta\Delta$Ct calculations (Brankatschk et al., 2012). The independent variables include the genes being measured and the experimental conditions under which the samples are processed.

Microscopy data further illustrate the complexity of multidimensional datasets (Rueden et al., 2017). In this context, the dependent variable might be a quantifiable feature, such as cell count or morphological metrics extracted from image analyses. The independent variables can expand immensely to include factors such as well-plate coordinates in a 96-well plate, Z-positions in confocal microscopy, and time points in time-lapse studies.

Lastly, big-data aggregation tools like `Metascape` provide a rich source of multidimensional data by integrating various dependent variables, such as gene expression fold changes and associated p-values, with independent variables spanning gene identifiers, gene ontology terms, and

ontology classes derived from multiple databases (Y. Zhou et al., 2019). Despite the provision of summarized graphical outputs, the raw data often remain in complex, nested formats within Excel sheets, posing significant challenges for hypothesis-driven research.

This extensive integration of multiple dimensions requires sophisticated visualization and analysis techniques. While basic statistical visualizations suffice for one- or two-dimensional data, more complex data sets necessitate advanced techniques, which allow researchers to visualize and interact with data in ways that elucidate the underlying patterns and relationships (Dunn et al., 2017). However, the gap between available visualization tools and the needs of clinicians or biologists without extensive bioinformatics training remains wide, emphasizing the need for intuitive, user-friendly tools that bridge this knowledge gap and enhance the accessibility of complex data analyses (Dunn et al., 2017).

Nontransparencies in Biomedical Data Analyses

The advent of advanced technologies in biosciences has ushered in an era of *big data*, characterized by unprecedented volumes and complexities of data (Bubendorf, 2001; 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 the Baker (2016) survey 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 (Begley & Ioannidis, 2015; Ioannidis, 2005).

Reproducibility is considered foundational to scientific research, ensuring that findings are reliable and verifiable. Still, its meaning requires precise definition (Goodman et al., 2016). The common understanding of scientific reproduction implies not only that detailed information is provided to enable independent repetition (*transparency*), but also that time and effort is invested into repeating the experiments (*corroboration*). However, since modern biomedical journals are demanding novelty research, and since experiments have become highly specialized and time-intensive, repeating someone else's work is considered neither interesting[1] nor possible for most publications (Flier, 2022; Peng, 2011). Hence, the meaning of reproducibility is confined to *transparency*, a concept that has been applied to many fields, including clinical trials (Goodman et al., 2016; Committee on Strategies for Responsible Sharing of Clinical Trial Data et al., 2015).

Nevertheless, there is a surprising amount of evidence for nontransparencies in biomedical data analyses: For Microarray-based miRNA profiling, raw data was not reported in more than

[1] Flier (2022): *"There are no scientists with the interest, resources, or incentives to "repeat" or confirm this vast sea of published work, so whether the findings they report are reproducible will simply never be assessed."*

40 % of 127 articles, making independent verification impossible (Witwer, 2013). The same study also found that re-analysis of data often times did not support the original conclusions. Furthermore, 44 % of 233 preclinical articles describe statistical tests insufficiently, while few don't describe them at all (Gosselin, 2021). Another study reviewed 147 papers in the field of optometrics and found that 91 % did not discuss their rationale of correcting p-values for multiple comparisons (e.g. Bonferroni correction) (Armstrong, 2014). However, given that the exact use of multiple comparisons corrections has been under debate for decades, it is reasonable to assume that researchers lack the confidence to report their technique in detail (Perneger, 1998; Moran, 2003; Sullivan & Feinn, 2021). In general, P-values are target of extreme scrutiny and also the cause of many arguments, which themselves are of questionable statistical reasoning[2] (Leek & Peng, 2015). Additionally, statistical illiteracy is a well-known problem among clinicians (Lakhlifi et al., 2023). Among biomedical researchers, 77 % state that they have not received formal training in data literacy, including visualization and public deposition of data, although they understand its high relevance (Federer et al., 2016). Correspondingly, it has been communicated that there is a lack of intuitive tools to embed computational work into publications, but also a lack of bioinformaticians to translate computation into clinics (Mesirov, 2010; Smith et al., 2018; Gómez-López et al., 2019). Therefore, nontransparencies in biomedical analyses are not only caused by a habit[3] of insufficient reporting, but could be exacerbated by the confusions caused by currently available methodologies and the lack of proper training.

Semi-Big Data: Big Enough to Cause Problems

Recent advances in big data analysis have significantly improved the standardization of both raw data availability and processing pipelines (Gomez-Cabrero et al., 2014). Particularly in RNAseq analysis, automation and the use of sophisticated software have established standards that enhance reproducibility across studies. For example, tools such as `STAR` and `HISAT` for sequence alignment, and `Cufflinks` and `DESeq` for differential expression analysis, rely on scripts that standardize processing steps to produce repeatable and verifiable results (Dobin et al., 2013; Kim et al., 2015; Trapnell et al., 2012; Love et al., 2014). These frameworks not only automate data handling but also ensure that data analysis protocols are followed consistently, reducing human error and variability between different users or laboratories.

However, this level of standardization and automation has not been mirrored in the analysis of *semi-big data*. Semi-big data, as introduced in this thesis, describes datasets that are on the cusp of manageability: substantial enough to overwhelm manual analysis methods yet not

[2]Leek & Peng (2015): *"Arguing about the P value is like focusing on a single misspelling, rather than on the faulty logic of a sentence"*

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

sufficiently large or uniform to justify the heavy computational frameworks developed for big data. Such data are frequently generated in experiments like automated microscopy or multiplex qPCR, where the scale and complexity of the data can vary significantly depending on the experimental design and objectives (Krzywinski & Savig, 2013).

Researchers often revert to basic tools such as *Microsoft Excel* for analyzing these semi-big datasets (Incerti et al., 2019). While Excel provides familiarity and immediate accessibility, it lacks the sophisticated data handling capabilities necessary for efficient and error-free processing of complex (multidimensional) datasets. This reliance on manual methods not only makes the analysis laborious and prone to mistakes but also significantly impedes the reproducibility of research findings. The time and effort required to replicate analyses done manually mean that validating findings from semi-big data can be prohibitively challenging for peer reviewers and other researchers in the field.

Given these challenges, there is a critical need for developing new tools and frameworks specifically tailored for semi-big data. These tools should bridge the gap between the simplicity of user-friendly software like Excel and the robust, script-based automation seen in big data frameworks. By providing standardized, repeatable, and easy-to-use methods for handling complex datasets, such tools could significantly enhance the reliability and efficiency of research involving semi-big data, ultimately supporting broader scientific inquiry and verification.

The Shortcomings of Common Biomedical Analysis Tools

Interactive software systems commonly used for exploratory data analysis in biomedical research often lack mechanisms to track and reproduce the researcher's actions systematically. Even when analysis is performed using scripting languages, the integration of results from multiple packages without a coherent record of the commands and code used undermines reproducibility. This practice can obscure analysis, making it difficult, if not impossible, for other researchers to replicate the results (Leek & Peng, 2015; Peng, 2011; Mesirov, 2010; Localio et al., 2018).

A particularly illustrative example is *GraphPad Prism*, a tool ubiquitously employed across biomedical disciplines for statistical analysis. Despite its widespread use, it does contribute to data analysis nontransparencies due to *Prism*'s closed-source nature and the common journal practice of not requiring detailed methodological transparency in its usage, a practice that is common in biostatistics literature (Gosselin, 2021; Localio et al., 2018). Furthermore, *GraphPad Prism* still requires manual data entry and lacks the robustness and automation necessary for handling multidimensional or semi-big data. Although, *GraphPad Prism* is compatible *"multiple variable tables"* —similar to long-form tables known from Wickham (2014)—, but does not automatically graph these kinds of tables, but only user specified subsets (*GraphPad Prism 10*

User Guide, 2024).

Moreover, *Microsoft Excel*, another staple in data processing in biomedicine, is notoriously inadequate for handling multidimensional data and complex statistical analyses. Its limitations include poor error tracking, absence of change documentation (audit trails), and a propensity for introducing errors that often go unnoticed, such as converting gene names to dates (Ziemann et al., 2016). To compensate for these shortcomings, *Microsoft* has recently integrated a Python interpreter into *Excel*, allowing researchers to automate tasks and analyze data efficiently and correctly (Excel, 2023).

Indeed, many common tools in biomedicine allow for scripting or automation to handle semi-big data more effectively. For example, `Fiji/ImageJ`, a popular image processing platform, supports extensive macro and scripting capabilities (Rueden et al., 2017). These features enable researchers to automate batch processing of image data, streamlining tasks that would otherwise require laborious manual input. Similarly, `PyMOL`, a leading tool in protein structural biology, utilizes Python scripting to automate complex tasks, allowing for detailed molecular modeling and visualization that are reproducible and scalable across datasets (*PyMOL*, 2024; Rigsby & Parker, 2016).

Although automation scripts used in tools like `Fiji/ImageJ` and `PyMOL` improve transparency for publishing singular data analysis pipelines, they still face challenges that can impede their reproducibility (Peng, 2011; Sandve et al., 2013): These scripts sometimes require specialized software environments, where setting up dependencies and configurations can be complex enough to discourage replication efforts. Additionally, these scripts do not always provide comprehensive outputs of intermediate steps, which is crucial for verifying and understanding the progression of data analysis (Sandve et al., 2013).

On the other hand, when scripts are designed to be more generalized and distributed—for instance, as a `Fiji/ImageJ` plugin or a standalone application—they can make substantial contributions to scientific research by enabling other researchers to apply these tools to their own data sets (Narzt et al., 1998; Wilkinson et al., 2016). However, this approach also comes with its own set of challenges (Sandve et al., 2013). Often, these generalized tools often lack comprehensive user-manuals (*documentation*) are not thoroughly tested across different platforms or data sets, which can lead to unexpected errors that can not be fixed by the user. Moreover, even when these tools are available, they frequently suffer from low adoption rates, meaning that few people are familiar with the details of such tools, further decreasing the confidence and reproducibility in the final results.

Given these complexities, there is a pressing need for new analytical tools specifically designed for semi-big data. These tools must strike a balance between the ease of use found in basic software and the robust, analytical capabilities of more sophisticated systems. By providing

standardized workflows, comprehensive documentation, and ensuring cross-platform compatibility, these tools can significantly enhance reproducibility. They not only allow researchers to perform analyses more efficiently but also ensure that these analyses are robust, transparent, and easily verifiable by the broader scientific community.

This thesis presents a software environment developed in Python, designed to bridge this gap. It demonstrates that even minimal coding skills can be leveraged to create powerful tools that standardize and accelerate the analysis of semi-big data, ultimately fostering more reproducible and trustworthy scientific research.

Modern Standards of Software Development

A main reason to write software is to define re-usable instructions for task automation (Narzt et al., 1998). 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, documenting of the detailed history of the codebase, including the person and timepoint of every change. 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 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 offer 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.

What makes Python an "Easy" 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

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 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!
```

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, but is harder to understand because more steps are needed, including the import of a library `stdio.h` and the definition of a function called `main`.

```c
#include <stdio.h>          // Import functions for standard input & output
int main() {                // Define a function called '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) (*The Python Language Reference*, 2024).

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.

```python
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.

```c
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).

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, but behaves exactly as shown in Listing 4.

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

To make Python as easy as possible, python packages aim to reduce the amount of code that has to be written by the user. For example, the package `matplotlib` is a plotting library where every command is written such that the user immediately understands its purpose, like plotting a line or labeling an axis (Listing 7). Hence `matplotlib` code is a sequence of simple function calls, where the state of the plot is modified and saved in the background line by line.

Listing 7: Example of using pre-written functions of a Python package. The functions of the package `matplotlib.pyplot` become accessable by importing the package as `plt`, where plt serves as an alias (or rather shortcut) to access the functions of the package. Then, two arbitrary lists are defined, `x` and `y`. These datapoints are plotted (scatterplot) using the function `plot` The plots x- and y-axes are then labeled and saved as an image. The code is written in a sequence of function calls, where the state of the plot is saved in the background. The plot is then displayed using the function `show`.

```
import matplotlib.pyplot as plt  # Make functions accessible via plt
x = [1, 2, 3, 4, 5]              # Define arbitrary x values
y = [1, 4, 9, 16, 25]            # Define arbitrary y values
plt.plot(x, y)                   # Plot x against y
plt.xlabel('Timepoint')          # Add a label to the x axis
plt.ylabel('Foldchange')         # Add a label to the y axis
plt.title('Gene Expression')     # Add a title above the plot
plt.savefig('plot.png')          # Save plot as image onto harddrive
plt.show()                       # Show the plot preview
```

However, when no pre-written functions or packages are available, Python offers the tools of a general purpose programming language to write and deploy custom code easily. Programming styles can be classified into two main paradigms: *functional* and *object-oriented* programming, which can be understood as different ways to structure code. Python supports both paradigms. In *functional* programming, the code is written in a way that the program is a sequence of function calls, where each function call returns a value that is used in the next function call (Listing 8). 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.

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 9). 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.

Listing 8: 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):      # Define a function
    return gene.find('GCGC')               # Find the position of 'GCGC'

def cut(gene: str):                        # Define another function
    position = find_restriction_site(gene) # Use the function above
    return gene[position:]                 # Cut the gene at the position

gene1 = 'TGAGCTGAGCTGATGCGCTATATTTAGGCG'   # Define an arbitrary gene
gene1_cut = cut(gene1)                     # Cut the gene
print(gene1_cut)                           # Show the result
# Output: GCGCTATATTTAGGCG
```

The data of an object is stored as *attributes*. Functions that are associated with an object are called *methods*.

Listing 9: 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:                                # Define a Gene class
    def __init__(self, sequence: str):     # Define how a Gene object is created
        self.sequence: str = sequence      # Save sequence as attribute
        self.promotor: str = self.find_promotor() # Automatically find promotor
    def find_promotor(self):               # Define how to find the promotor
        return self.sequence.find('TATA')
    def find_restriction_site(self):       # Define how to find restriction site
        return self.sequence.find('GCGC') # Find the position of 'GCGC'
    def cut(self):                         # Define how to cut the gene
        position = self.find_restriction_site() # Call the method above
        return self.sequence[position:]    # Cut the gene at the position

gene1 = Gene(sequence='TGAGCTGAGCTGATGCGCTATATTTAGGCG') # Create Gene object
gene1_cut = gene1.cut()                    # Call the method cut
print(gene1_cut)                           # Show result
# 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 9).

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 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".

Listing 10: 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.

```python
class mRNA(Gene):          # Define the mRNA class, inheriting from Gene class
    def __init__(self, sequence: str):   # Define how an mRNA object is created
        super().__init__(sequence)       # Get attributes from parent class
        self.sequence.replace('T', 'U')  # Replace thymine with uracil
    def find_stopcodons(self):           # Define how to find stop codons
        return self.sequence.find('UGA') # Find the position of 'UGA'

mrna1 = mRNA(sequence='TGAGCTGAGCTGATGCGCTATATTTAGGCG') # Create mRNA object
print(mrna1.find_stopcodons())              # Show the position of stop codons
# Output: [0, 5, 10]
```

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. Another 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 are a powerful solution for improving reproducibility: They elegantly combine both documentation and code execution into a concise presentation of the data analysis process, hence being an intuitive tool to both capture and embed computational work directly into papers, a requirement postulated by Mesirov (2010). Jupyter notebooks are increasingly 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; Hannun et al., 2023). 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 Listing 7 (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 PhD thesis is designed to bridge significant gaps in the understanding and analysis of myeloma cell behavior and the handling of complex biomedical datasets. The specific aims are as follows:

- Develop an *in vitro* model to elucidate the mechanisms of myeloma cell dissemination in interaction with mesenchymal stromal cells (hMSCs), focusing particularly on:
  - Observing and quantifying cell proliferation, attachment, and detachment dynamics using time-lapse microscopy.
  - Isolating and characterizing distinct myeloma subpopulations interacting with hMSCs to understand differential gene expression related to cell adhesion and patient survival.
- Design and implement a Python-based software tool, `plotastic`, to facilitate the analysis of multidimensional datasets generated in biomedical research. This tool will aim to:
  - Streamline the data analysis process, making it more efficient and reproducible.
  - Integrate visualization and statistical analysis capabilities to ensure that data analysis protocols are aligned with the ways in which data is visualized.
  - Provide a case study demonstrating the application of `plotastic` in the analysis of *in vitro* dissemination experiments, emphasizing the tool's ability to handle semi-big data and enhance reproducibility.
- Synthesize the findings from the experimental and software development components to advance the understanding of myeloma dissemination and improve research practices in biomedical data analysis.

These aims are crafted to address both the biological and technical challenges in current cancer research methodologies and data science applications in biomedicine, fostering advancements that could lead to novel therapeutic strategies and more robust scientific inquiries.

Chapter 1: Modelling Myeloma Dissemination *in vitro*

Abstract

Multiple myeloma involves early dissemination of malignant plasma cells across the bone marrow; however, the initial steps of dissemination remain unclear. Human bone marrow- derived mesenchymal stromal cells (hMSCs) stimulate myeloma cell expansion (e.g., IL-6) and simultaneously retain myeloma cells via chemokines (e.g., CXCL12) and adhesion factors. Hence, we hypothesized that the imbalance between cell division and retention drives dissemination. We present an *in vitro* model using primary hMSCs co-cultured with INA-6 myeloma cells. Time-lapse microscopy revealed proliferation and attachment/detachment dynamics. Separation techniques (V-well adhesion assay and well plate sandwich centrifugation) were established to isolate MSC-interacting myeloma subpopulations that were characterized by RNAseq, cell viability and apoptosis. Results were correlated with gene expression data ($n = 837$) and survival of myeloma patients ($n = 536$). On dispersed hMSCs, INA-6 saturate hMSC-surface before proliferating into large homotypic aggregates, from which single cells detached completely. On confluent hMSCs, aggregates were replaced by strong heterotypic hMSC-INA-6 interactions, which modulated apoptosis time-dependently. Only INA-6 daughter cells (nMA-INA6) detached from hMSCs by cell division but sustained adherence to hMSC-adhering mother cells (MA-INA6). Isolated nMA-INA6 indicated hMSC-autonomy through superior viability after IL-6 withdrawal and upregulation of proliferation-related genes. MA-INA6 upregulated adhesion and retention factors (CXCL12), that, intriguingly, were highly expressed in myeloma samples from patients with longer overall and progression-free survival, but their expression decreased in relapsed myeloma samples. Altogether, *in vitro* dissemination of INA-6 is driven by detaching daughter cells after a cycle of hMSC-(re)attachment and proliferation, involving adhesion factors that represent a bone marrow-retentive phenotype with potential clinical relevance.

Statement of Significance

Novel methods describe *in vitro* dissemination of myeloma cells as detachment of daughter cells after cell division. Myeloma adhesion genes were identified that counteract *in vitro* detachment with potential clinical relevance.

Introduction

Multiple myeloma arises from clonal expansion of malignant plasma cells in the bone marrow (BM). At diagnosis, myeloma cells have disseminated to multiple sites in the skeleton and, in some cases, to "virtually any tissue" (Bladé et al., 2022; Rajkumar et al., 2014). However, the mechanism through which myeloma cells initially disseminate remains unclear. Dissemination is a multistep process involving invasion, intravasation, intravascular arrest, extravasation, and colonization (Zeissig et al., 2020). To initiate dissemination, myeloma cells overcome adhesion, retention, and dependency on the BM microenvironment, which could involve the loss of adhesion factors such as CD138 (Akhmetzyanova et al., 2020; García-Ortiz et al., 2021). BM retention is mediated by multiple factors: First, chemokines (CXCL12 and CXCL8) produced by mesenchymal stromal cells (MSCs), which attract plasma cells and prime their cytoskeleton and integrins for adhesion (Aggarwal et al., 2006; Alsayed et al., 2007). Second, myeloma cells must overcome the anchorage and physical boundaries of the extracellular matrix (ECM), consisting of e.g. fibronectin, collagens, and proteoglycans such as decorin (Hu et al., 2021; Huang et al., 2015; Katz, 2010; Kibler et al., 1998). Simultaneously, ECM provides signals inducing myeloma cell cycle arrest or progression the cell cycle (Hu et al., 2021; Katz, 2010). ECM is also prone to degradation, which is common in several osteotropic cancers, and is the cause of osteolytic bone disease. This is driven by a 'vicious cycle' that maximizes bone destruction by extracting growth factors (EGF and TGF-β) that are stored in calcified tissues (Glavey et al., 2017). Third, direct contact with MSCs physically anchors myeloma cells to the BM (Zeissig et al., 2020; Sanz-Rodríguez et al., 1999). Fourth, to disseminate to distant sites, myeloma cells require, at least partially, independence from essential growth and survival signals provided by MSCs in the form of soluble factors or cell adhesion signaling (García-Ortiz et al., 2021; Chatterjee et al., 2002; Hideshima et al., 2007). For example, the VLA4 (Myeloma)–VCAM1 (MSC)-interface activates NF- B in both myeloma and MSCs, inducing IL-6 expression in MSCs. The independence from MSCs is then acquired through autocrine survival signaling (Frassanito et al., 2001; Urashima et al., 1995). In short, anchorage of myeloma cells to MSCs or ECM is a 'double-edged sword': adhesion counteracts dissemination, but also presents signaling cues for growth, survival, and drug resistance (Solimando et al., 2022).

To address this ambiguity, we developed an *in vitro* co-culture system modeling diverse adhesion modalities to study dissemination, growth, and survival of myeloma cells and hMSCs. Co-cultures of hMSCs and the myeloma cell line INA-6 replicated tight interactions and aggregate growth, akin to "microtumors" in Ghobrial's metastasis concept (Ghobrial, 2012). We characterized the growth conformations of hMSCs and INA-6 as homotypic aggregation *vs.* heterotypic hMSC adherence and their effects on myeloma cell survival. We tracked INA-6 detachments from aggregates and hMSCs, thereby identifying a potential "disseminated" sub-

population lacking strong adhesion. We developed innovative techniques (V-well adhesion assay and well plate sandwich centrifugation) to separate weakly and strongly adherent subpopulations for the subsequent analysis of differential gene expression and cell survival. Notably, our strategy resolves the differences in gene expression and growth behavior between cells of one cell population in "direct" contact with MSCs. In contrast, previous methods differentiated between "direct" and "indirect" cell-cell contact using transwell inserts (Dziadowicz et al., 2022). To evaluate whether genes mediating adhesion and growth characteristics of INA-6 were associated with patient survival, we analyzed publicly available datasets (Seckinger et al., 2017, 2018).

Materials and Methods

See Appendix A.3 for a complete method list and description.

Ethics Statement

Primary human MSCs were collected with the written informed consent of all patients. The procedure was conducted in accordance with recognized ethical guidelines (Helsinki Declaration) and approved by the local Ethics Committee of the University of Würzburg (186/18).

Cultivation and Co-Culturing of primary hMSCs and INA-6

Primary human MSCs were obtained from the femoral head of 34 non-myeloma patients (Appendix A: Tab. 1: 21 male and 13 female, mean age 68.9 ± 10.6) undergoing elective hip arthroplasty. The INA-6 cell line (*DSMZ Cat# ACC-862*, *RRID:CVCL_5209*, link) was initially isolated from a pleural effusion sample obtained from an 80-year-old male with multiple myeloma (Burger, Günther, et al., 2001; Gramatzki et al., 1994). hMSCs were not tested for mycoplasma, whereas stocks of INA-6 were tested in this study (Appendix A: Tab. 1) using the *Venor GeM OneStep* kit (Minerva Biolabs, Berlin, Germany). For each co-culture, hMSCs were seeded 24 h before INA-6 addition to generate the MSC-conditioned medium (CM). INA-6 cells were washed with PBS, resuspended in MSC medium, and added to hMSCs so that the co-culture comprised 33 % (v/v) of CM gathered directly from the respective hMSC donor. The co-cultures were not substituted for IL-6 (Chatterjee et al., 2002).

Cell Viability and Apoptosis Assay

Cell viability and apoptosis rates were measured using *CellTiter-Glo Luminescent Cell Viability Assay* and *Caspase-Glo 3/7 Assay*, respectively (Promega GmbH, Mannheim, Germany).

Automated Fluorescence Microscopy

Microscopic images were acquired using an Axio Observer 7 (Zeiss) with a COLIBRI LED light source and motorized stage top using 5x and 10x magnification. The tiled images had an automatic 8–10 % overlap and were not stitched.

Live Cell Imaging

hMSCs (stained with PKH26) were placed into an ibidi Stage Top Incubation System and equilibrated to 80 % humidity and 5 % CO_2. INA-6 (2×10^3 cells/cm^2) were added directly before the start of acquisition. Brightfield and fluorescence images of up to 13 mm^2 of the co-culture area were acquired every 15 min for 63 h. Each event of interest was manually analyzed and categorized into defined event parameters.

V-well Adhesion Assay

INA-6 cells were arrested during mitosis by two treatments with thymidine, followed by nocodazole. Arrested INA-6 were released and added to 96 V-well plates (10^4 cells/cm^2) on top of confluent hMSCs and adhered for 1–3 h. The co-culture was stained with calcein-AM (Thermo Fisher Scientific, Darmstadt, Germany) before non-adherent INA-6 were pelleted into the tip of the V-well (555 g, 5–10 min). MSC-adhering INA-6 cells were manually detached by rapid pipetting. The pellet brightness was measured microscopically and the pellet was isolated by pipetting.

Cell Cycle Profiling by Image Cytometry

Isolated INA-6 cells were fixed in 70 % ice-cold ethanol, washed, resuspended in PBS, distributed in 96-well plates, and stained with Hoechst 33342. The plates were scanned at 5x magnification. A pre-trained convolutional neural network (Intellesis, Zeiss) was fine-tuned to segment the scans into single nuclei and exclude fragmented nuclei. Nuclei were filtered to exclude extremes of size and roundness. The G0/G1 frequency was determined by Gaussian curve fitting.

Well Plate Sandwich Centrifugation (WPSC)

hMSCs were grown to confluence in 96-well plates coated with collagen I (rat tail; Corning, NY, USA). INA-6 cells were added and the cells were allowed to adhere for 24 h. A second plate ("catching plate") was attached upside down to the top of the co-culture plate. That "well plate sandwich" was turned around and the content of the co-culture plate was centrifuged into the catching plate three times (40 s at 110 g) while gently adding 30 µL of medium in between centrifugation steps. Non-MSC-adhering INA-6 cells were collected from the catching plate, whereas MSC-adhering INA-6 cells were isolated by digesting the co- culture with accutase. For RNA sequencing (RNAseq), all samples were purified using anti-CD45 magnetic-assisted cell sorting (Miltenyi Biotec B.V. & Co. KG, Bergisch Gladbach).

RNA Isolation

RNA was isolated using the *NucleoSpin RNA II Purification Kit* (Macherey-Nagel) according to the manufacturer's instructions. RNA was isolated from INA-6 cells co-cultured with a unique hMSC donor ($n = 5$ for RNA sequencing, $n = 11$ for qPCR).

RNA sequencing, Differential Expression, and Functional Enrichment Analysis

RNA sequencing (RNAseq) was performed at the Core Unit Systems Medicine, University of Würzburg. mRNA was enriched with polyA beads. Fastq files were aligned to the GRCh38 reference genome using `STAR` (*RRID:SCR_004463*, link) and raw read counts were generated using `HTseq` (*RRID:SCR_005514*, link) (Anders et al., 2015; Dobin et al., 2013; Zerbino et al., 2018). Differential gene expression was analyzed using `edgeR` in R (version 3.6.3) (*RRID:SCR_012802*, link). Functional enrichment analysis was performed using `Metascape` (*RRID:SCR_016620*, link) (Y. Zhou et al., 2019).

RT-qPCR

RNA (1 µg) was reverse transcribed using *SuperScript IV reverse transcriptase* (Thermo Fisher Scientific). qPCR was performed using 10 µL *GoTaq qPCR Master Mix* (Promega), 1 :10 diluted cDNA, and 5 pmol of primers obtained from Biomers.net or Qiagen (Appendix A: Tab. 3).

Statistical Analysis

Inferential statistics were performed using Python (IPython, *RRID:SCR_001658*, link) (3.10) packages *pingouin* (0.5.1) and `statsmodels` (0.14.0) (Vallat, 2018; Seabold & Perktold, 2010). The figures were plotted using `plotastic` (0.0.1) (Kuric & Ebert, 2024). Normality (for $n \geq 4$) and sphericity were ensured using Mauchly's and Shapiro-Wilk tests, respectively. Data points were Log_{10} transformed to convert the scale from multiplicative to additive or to fulfill sphericity requirements. $p = 0.05 > * > 0.01 > ** > 10^{-3} > *** > 10^{-4} > ****$. p-values were either adjusted (p-adj) or not adjusted (p-unc) for family wise error rate. Power calculations were not performed to determine the sample size.

Patient Cohort, Analysis of Survival and Expression

Survival and gene expression data were obtained as previously described (Seckinger et al., 2017, 2018) and are available at the European Nucleotide Archive (ENA) under accession numbers

PRJEB36223 and PRJEB37100. The expression level was categorized into "high" and "low" using `maxstat` (Maximally selected Rank Statistics) thresholds (Hothorn & Lausen, n.d.).

Data Availability Statement

A detailed description of the methods is provided in the Supplementary Material section. Raw tabular data and examples of analyses and videos are available in the github repository, link. Raw RNAseq data are available from the NCBI Gene Expression Omnibus (GEO) (*RRID:SCR_005012*, link) (GSE261423). Microscopy data are available at BioStudies (EMBL-EBI) (*RRID:SCR_004727*, link) (S-BIAD1092).

Results

INA-6 Cells Saturate hMSC-Interaction to Proliferate into Aggregates

hMSCs are isolated as a heterogeneous cell population. To analyze whether INA-6 cells could adhere to every hMSC, we saturated hMSCs with INA-6. A seeding ratio of 1:4 (hMSC:INA-6) resulted in the occupation of $93 \pm 6\,\%$ of single hMSCs by one or more INA-6 cells within 24 hours after INA-6 addition, escalating to 6 % after 48 hours (Fig. 1A, B). Therefore, most hMSCs provide an interaction surface for INA-6 cells.

INA-6 exhibits homotypic aggregation when cultured alone, a phenomenon observed in some freshly isolated myeloma samples (up to 100 cells after 6 hours) (Kawano et al., 1991; Okuno et al., 1991). Adding hMSCs at a 1:1 ratio led to smaller aggregates after 24 hours (size 1–5 cells), all of which were distributed over $52 \pm 2\,\%$ of all hMSCs (Fig. 1A, B). Intriguingly, INA-6 aggregation was notably absent when grown on confluent hMSCs, and occurred only when heterotypic interactions were limited to 0.2 hMSCs per INA-6 cell (Fig. 1C). We concluded that INA-6 cells prioritize heterotypic over homotypic interactions.

To monitor the formation of such aggregates, we conducted live-cell imaging of hMSC/INA-6 co-cultures for 63 hours. We observed that INA-6 cells adhered long after cytokinesis, constituting $55 \pm 12\,\%$ of all homotypic interactions between 13 hours and 26 hours, increasing to more than 75 % for the remainder of the co-culture (Fig. 1D). Therefore, homotypic INA-6 aggregates were mostly formed by cell division.

Apoptosis of INA-6 Depends on Ratio Between Heterotypic and Homotypic Interaction

Although direct interaction with hMSCs has been shown to enhance myeloma cell survival through NF-κB signaling (Hideshima et al., 2007), the impact of aggregation on myeloma cell viability during hMSC interaction remains unclear. To address this, we measured the cell viability (ATP) and apoptosis rates of INA-6 cells growing as homotypic aggregates compared to those in heterotypic interactions with hMSCs by modulating hMSC density (Fig. 1E). To equalize the background signaling caused by soluble MSC-derived factors, all cultures were incubated in hMSC-conditioned medium and the results were normalized to INA-6 cells cultured without direct hMSC contact (Fig. 1E, left).

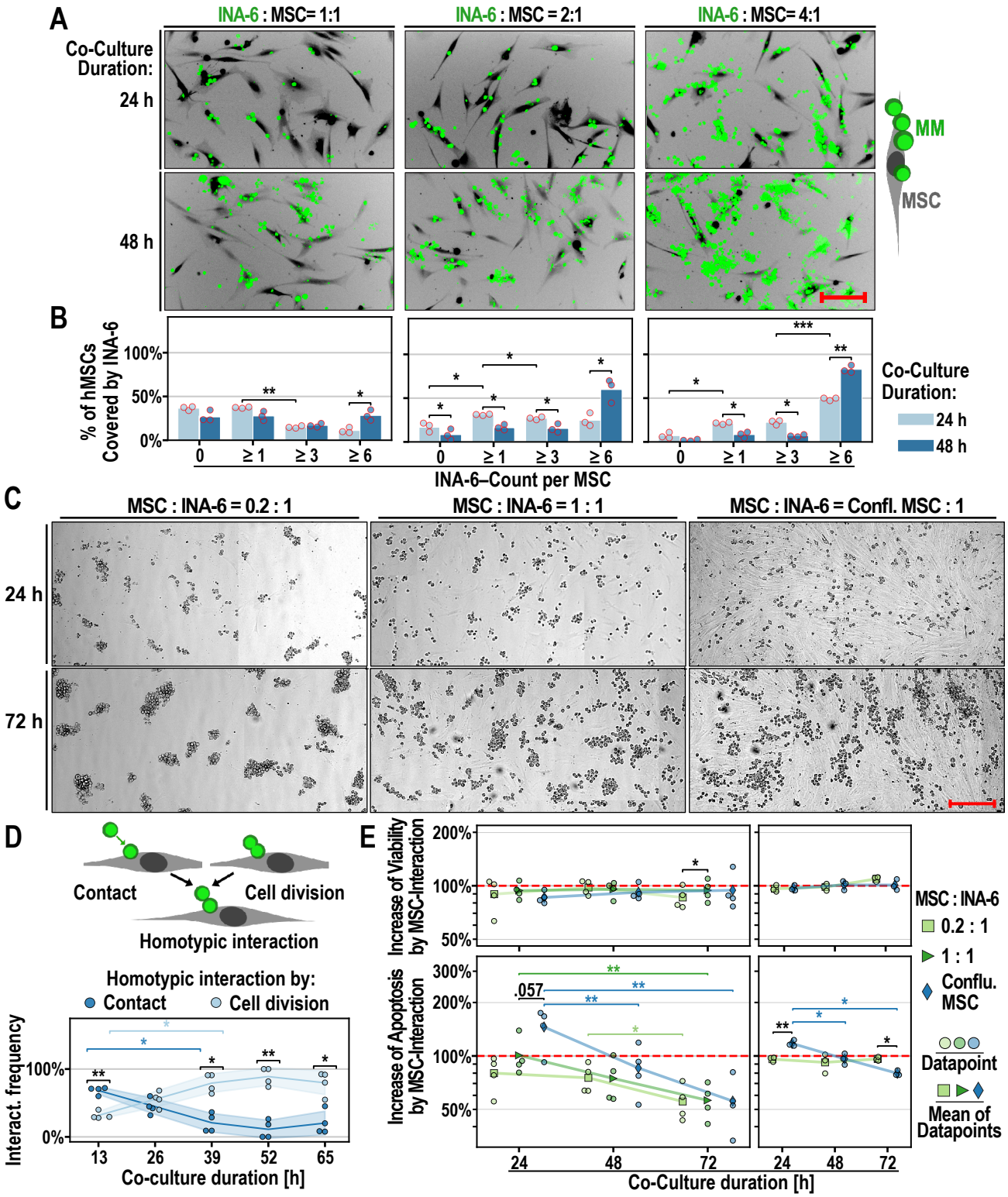


Figure 1: INA-6 growth conformations and survival on hMSCs. **A:** Interaction of INA-6 (green) with hMSCs (black, negative staining) at different INA-6 densities (constant hMSC densities). Scale bar = 200 µm. **B:** Frequency of single hMSCs (same as A) that are covered by INA-6 of varying group sizes. Technical replicates = three per datapoint; 100 single hMSCs were evaluated per technical replicate. **C:** Interaction of INA-6 – *continued on next page*

Figure 1: *continued from previous page* – with hMSCs at different hMSC densities (constant INA-6 densities). Scale bar = 300 µm. **D:** Two types of homotypic interaction: Attachment after cell contact and sustained attachment of daughter cells after cell division. Datapoints represent one of four independent time-lapse recordings, each evaluating 116 interaction events. **E:** Effects of hMSC-density on the viability (ATP, top) and apoptosis (Caspase3/7 activity, bottom). INA-6:MSC ratio = 4:1; Technical replicates = four per datapoint; **E left:** Signals were measured in INA-6 washed off from hMSCs and normalized by INA-6 cultured in MSC-conditioned medium (= red line) ($n = 4$). **E right:** Signals were measured in co-cultures and normalized by the sum of the signals measured in hMSC and INA-6 cultured separately (= red line) ($n = 3$). **Statistics:** Paired t-test, two-factor RM- ANOVA. Datapoints represent independent co-cultures with hMSCs from three (A, B, D, E right), four (E left) unique donors. Confl. = Confluent.

INA-6 viability (ATP) was not affected by the direct adhesion of hMSCs at any density. However, apoptosis rates decreased over time [$F(2,6) = 23.29$, p-unc $= 1.49\times10^{-3}$] (Two-factor RM-ANOVA), interacting significantly with MSC density [$F(4,12) = 6.98$, p-unc $= 3.83\times10^{-3}$] For example, 24 hours of adhesion to confluent MSCs increased apoptosis rates by 1.46 ± 0.37 fold, while culturing INA-6 cells on dispersed hMSCs (ratio 1:1) did not change the apoptosis rate (1.01 ± 0.26 fold).

We presumed that sensitive apoptotic cells might have been lost when harvesting INA-6 cells from hMSCs. Hence, we measured survival parameters in the co-culture and in hMSC and INA-6 cells cultured separately (Fig. 1E, right). We defined MSC interaction effects when the survival measured in the co-culture differed from the sum of the signals measured from INA-6 and hMSCs alone. RM-ANOVA confirmed that adherence to confluent MSCs increased apoptosis rates of INA-6 cells 24 hours after adhesion and decreased after 72 hours [$F(2,4) = 26.86$, p-unc $= 4.80\times10^{-3}$] (interaction between MSC density and time, Two-factor RM-ANOVA), whereas INA-6 cells were unaffected when grown on dispersed hMSCs. In summary, the growth conformation of INA-6 cells, measured as the ratio between homotypic aggregation and heterotypic MSC interactions, affected apoptosis rates of INA-6 cells.

Single INA-6 Cells Detach Spontaneously from Aggregates of Critical Size

Using time-lapse microscopy, we observed that 26 ± 8 % of INA-6 aggregates growing on single hMSCs spontaneously shed INA-6 cells (Fig. 2A, B; Supplementary Video 1). Notably, all detached cells exhibited similar directional movements, suggesting entrainment in convective streams generated by temperature gradients within the incubation chamber. INA-6 predominantly detached from other INA-6 cells or aggregates (Fig. 2C), indicating weaker adhesive forces in homotypic interactions than in heterotypic interactions. The detachment frequency increased after 52 hours, when most aggregates that shed INA-6 cells were categorized as large (greater than 30 cells) (Fig. 2D). Since approximately 10-20 INA-6 cells already fully covered a single hMSC, we suggest that myeloma cell detachment depended not only on hMSC saturation but also required a minimum aggregate size. Interestingly, INA-6 detached mostly as

single cells, independent of aggregate size categories [$F(2,6)=4.68$, p-unc$=0.059$] (Two-factor RM-ANOVA) (Fig. 2E), showing that aggregates remained mostly stable despite losing cells.

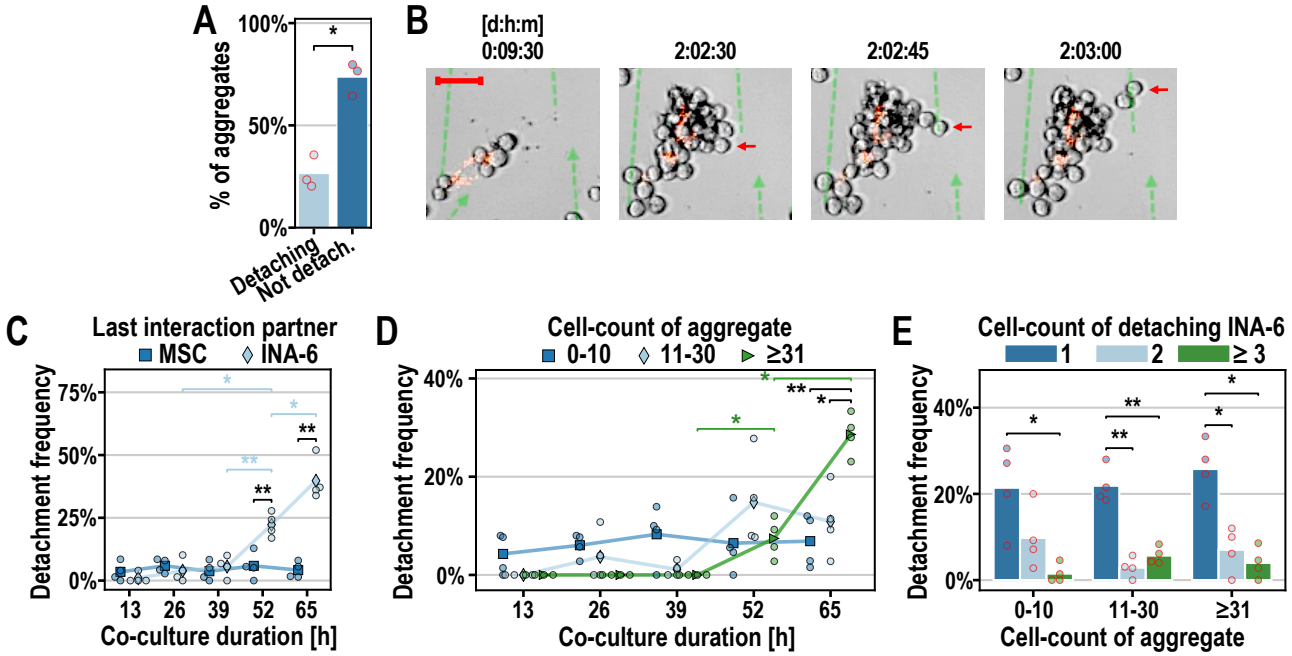


Figure 2: Time-lapse analysis of INA-6 detachment from INA-6 aggregates and hMSCs. **A:** Frequency of observed INA-6 aggregates that did or did not lose INA-6 cell(s). 87 aggregates were evaluated per datapoint. **B:** Example of a "disseminating" INA-6 aggregate growing on fluorescently (PKH26) stained hMSC (from A-D). Dashed green lines are trajectories of detached INA-6 cells. Scale bar = 50 µm. **C-E:** Quantitative assessment of INA-6 detachments. 45 detachment events were evaluated per datapoint. Seeding ratio INA-6:MSC = 4:1. **C:** Most INA-6 cells dissociated from another INA-6 cell and not from an hMSC [$F(1,3)=298$, p-unc$=4.2\text{»}10^{-4}$]. **D:** Detachment frequency of aggregate size categories. **E:** Detachment frequency of INA-6 cells detaching as single, pairs or more than three cells. **Statistics:** (A): Paired-t-test; (C-E): Paired-t-test, Two-factor RM-ANOVA; Datapoints represent three (A) or four (C-E) independent time-lapse recordings of co-cultures with hMSCs from two (A) or three (C-E) unique donors.

Cell Division Generates a Daughter Cell Detached from hMSC

We suspected that cell division drives detachment because we observed that MSC-adhering INA-6 cells could generate daughter cells that "roll over" the mother cell (Fig. 3A; Supplementary Video 2). We recorded and categorized the movement of INA-6 daughter cells in confluent hMSCs after cell division. Half of all INA-6 divisions yielded two daughter cells that remained stationary, indicating hMSC adherence (Fig. 3B, C; Supplementary Video 3). The other half of division events generated one hMSC-adhering (MA-INA6) cell and one non-hMSC-adhering (nMA-INA6) cell, which rolled around the MA-INA6 cell for a median time of 2.5 hours post division (Q1=1.00 hour, Q3=6.25 hours) until it stopped and re-adhered to the hMSC monolayer (Fig. 3D; Supplementary Video 2, Supplementary Video 4). Thus, cell division establishes a time window in which one daughter cell can detach.

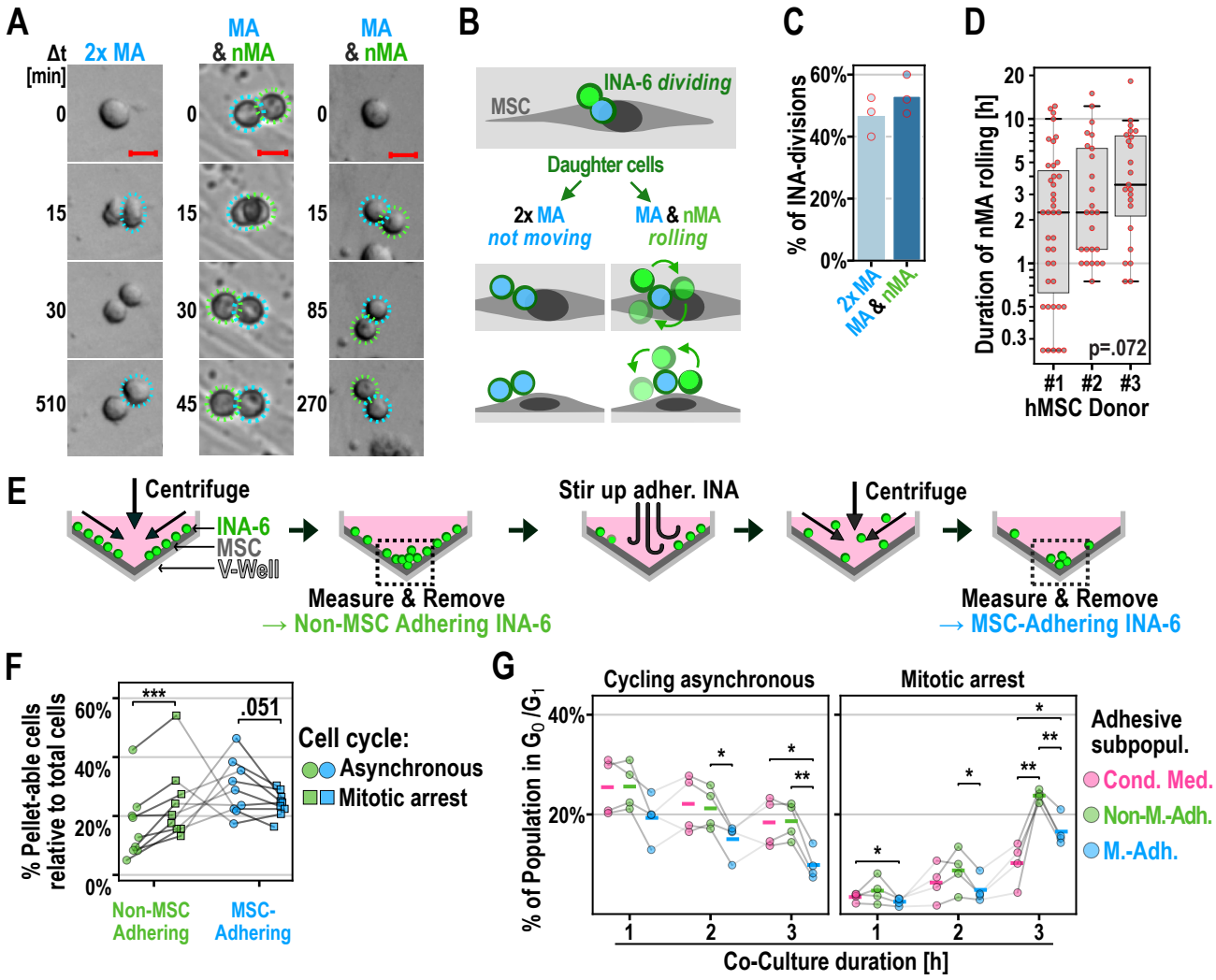


Figure 3: Detachment of INA-6 daughter cells after Cell Division. **A-D:** INA-6 divisions in interaction with confluent hMSCs. Seeding ratio INA-6:MSC = 4:20. **A:** Three examples of dividing INA-6 cells generating either two MA, or one MA and one nMA daughter cells as described in (G). Dashed circles mark mother cells (white), MA cell (blue), and first position of nMA cell (green). Scale bar: 20 µm. **B:** Cell division of MSC-adhering (MA) mother cell can yield one mobile non-MSC-adhering (nMA) daughter cell. **C:** Frequencies of INA-6 pairs defined in (A, B) per observed cell division. 65 divisions were evaluated for each of three independent time-lapse recordings. **D:** Rolling duration of nMA cells after division did not depend on hMSC donor [$H(2) = 5.250$, p-unc $= .072$]. Datapoints represent single nMA-cells after division. **E-G:** Adhesive and cell cycle assessment of MSC-interacting INA-6 subpopulations using the V-Well assay. **E:** Schematic of V-Well Assay (see Appendix A: Fig. 1 for detailed analysis). MSC-interacting subpopulations were separated by subsequent centrifugation and removal of the pellet. The pellet size was quantified by its total fluorescence brightness. Adhering subpopulations were resuspended by rough pipetting. **F:** Relative cell pellet sizes of adhesive INA-6 subpopulations that cycle either asynchronously or were synchronized at mitosis. Gray lines in-between points connect dependent measurements of co-cultures ($n = 9$) that shared the same hMSC-donor and INA-6 culture. Co-cultures were incubated for three different durations (1 h, 2 h and 3 h after INA-6 addition). Time points were pooled, since time did not show an effect on cell adhesion [$F(2, 4) = 1.414$, p-unc $= 0.343$] Factorial RM-ANOVA shows an interaction between cell cycle and the kind of adhesive subpopulation [$F(1, 8) = 42.67$, p-unc $= 1.82 \cdot 10^{-4}$]. Technical replicates = 4 per datapoint. **G:** Cell cycles were profiled in cells gathered from the pellets of four independent co-cultures ($n = 4$) and the frequency of G0/G1 cells are displayed depending on co-culture duration (see Appendix A: Fig. 3 for cell cycle profiles). Four technical replicates were pooled after pelleting. **Statistics:** (D): Kruskal-Wallis H-test. (F): Paired t-test, (G): Paired t-test, two-factor RM-ANOVA. Datapoints represent INA-6 from independent co- cultures with hMSCs from three unique donors.

To validate that cell division reduced adhesion, we measured both the size and cell cycle profile of the nMA-INA6 and MA-INA6 populations using an enhanced V-well assay (method described in Fig. 3E, Appendix A: Fig. 1, 2). For comparison, we fully synchronized and arrested INA-6 cells at mitosis and released their cell cycle immediately before addition to the hMSC monolayer, rendering them more likely to divide while adhering. Mitotic arrest significantly increased the number of nMA-INA6 cells and decreased the number of MA-INA6 cells (Fig. 3F). Furthermore, the nMA-INA6 population contained significantly more cells cycling in the G0/G1 phase than the MA-INA6 population, both in synchronously and asynchronously cycling INA-6 (Fig. 3G, Appendix A: Fig. 3, 4). The number of nMA-INA6 INA-6 cells increased due to a higher cell division frequency. Taken together, we showed that INA-6 detach from aggregates by generating one temporarily detached daughter cell after cell division, a process that potentially contributes to the initiation of dissemination.

RNAseq of Non-MSC-Adhering and MSC-Adhering Subpopulations

To characterize the subpopulations separated by WPSC, we conducted RNAseq, revealing 1291 differentially expressed genes between nMA-INA6 *vs.* CM-INA6, 484 between MA-INA6 *vs.* CM-INA6, and 195 between MA-INA6 *vs.* nMA-INA6. We validated RNAseq and found that the differential expression of 18 genes correlated with those measured with qPCR for each pairwise comparison (Fig. 4C–E, Appendix A: Fig. 5): nMA-INA6 *vs.* CM-INA6 $[\rho(16) = .803,\ p = 6.09\times10^{-5}]$, MA-INA6 *vs.* CM-INA6 $[\rho(16) = .827, p = 2.30\times10^{-5}]$, and MA-INA6 *vs.* nMA-INA6 $[\rho(16) = .746, p = 3.74\times10^{-4}]$ (Spearman's rank correlation). One of the 18 genes (*MUC1*) measured by qPCR showed a mean expression opposite to that obtained by RNAseq (nMA-INA6 *vs.* CM-INA6), although the difference was insignificant (Fig. 4C). For nMA-INA6 *vs.* CM-INA6, the difference in expression measured by qPCR was significant for only two of the 11 genes (*DKK1*, *OPG*), whereas the other genes (*DKK1*, *OPG*, *BCL6*, *BMP4*, *BTG2*, *IL10RB*, *IL24*, *NOTCH2*, *TNFRSF1A*, *TRAF5*) only confirmed the tendency measured by RNAseq (Fig. 4C–E). For MA-INA6 *vs.* CM-INA6, qPCR validated the significant upregulation of seven genes (*DKK1*, *OPG*, *BCL6*, *BMP4*, *BTG2*, *IL10RB*, *IL24*, *NOTCH2*, *TNFRSF1A*, *TRAF5*, *TGM2*, *DCN*, *LOX*, *MMP14*, *MMP2*, *CXCL12*, *CXCL8*), whereas the downregulation of *BMP4* was insignificant.

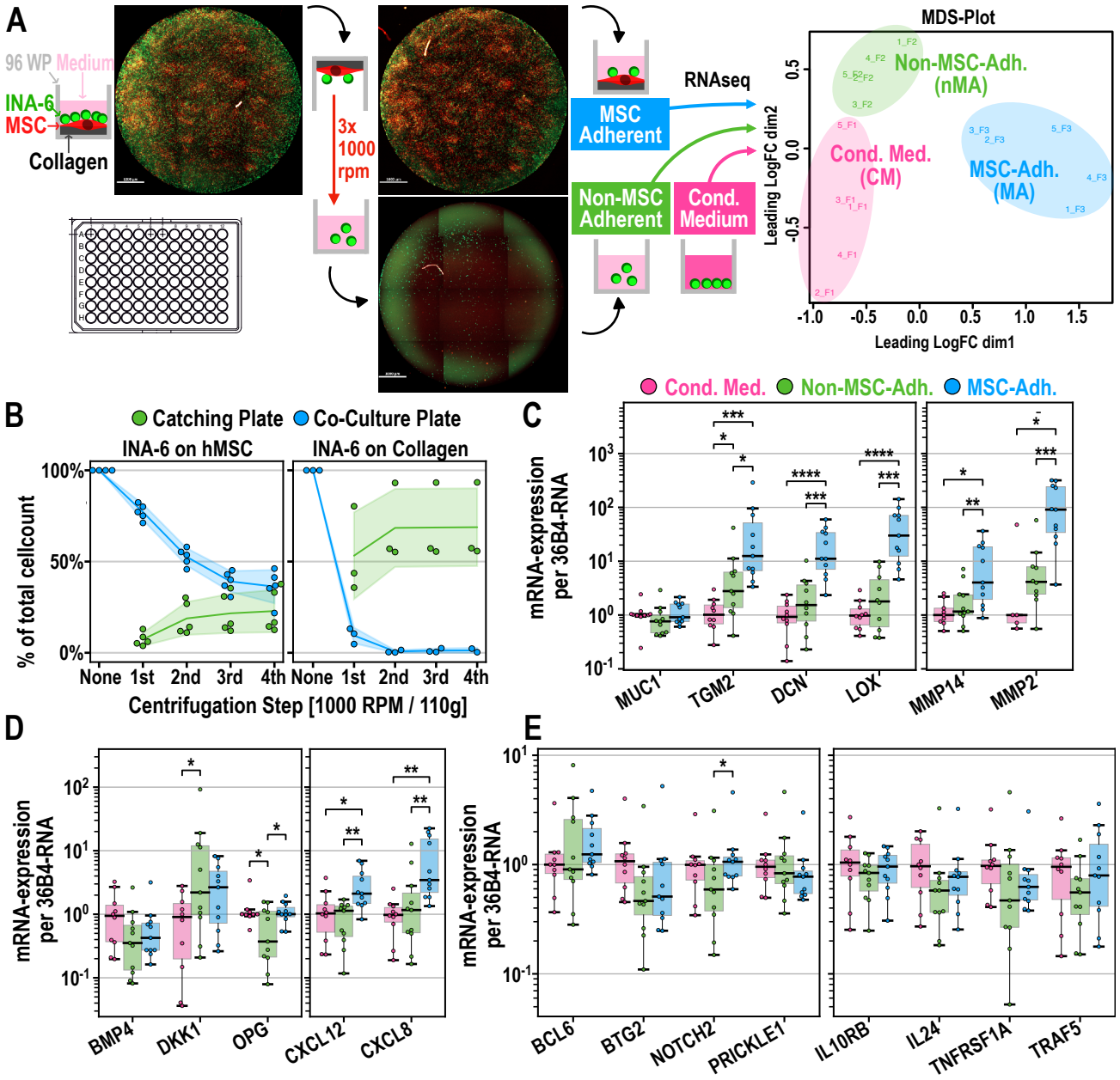


Figure 4: Separation and gene expression of INA-6 subpopulations. **A:** Schematic of "Well-Plate Sandwich Centrifugation" (WPSC) separating nMA- from MA-INA6. A co-culture 96-well plate is turned upside down and attached on top of a "catching plate", forming a "well-plate sandwich". nMA-INA6 cells are collected in the catching plate by subsequent rounds of centrifugation and gentle washing. MA-INA6 are enzymatically dissociated from hMSCs or by rough pipetting. Subsequent RNAseq of MSC-interacting subpopulations reveals distinct expression clusters [right, multidimensional scaling plot (MDS) ($n = 5$)]. **B:** Separation was microscopically tracked after each centrifugation step. **C-E:** RT-qPCR of genes derived from RNAseq results. Expression was normalized to the median of CM-INA6. Samples include those used for RNAseq and six further co-cultures ($n = 11$; non-detects were discarded). **C:** Adhesion factors, ECM proteins, and matrix metalloproteinases. **D:** Factors involved in bone remodeling and bone homing chemokines. **E:** Factors involved in (immune) signaling. **Statistics:** (C-E): Paired t-test. Datapoints represent the mean of three (B-E) technical replicates. INA-6 were isolated from independent co-cultures with hMSCs from five (A, B), nine (C-E) unique donors.

Non-MSC-Adhering INA-6 and MSC-Adhering INA-6 Have Distinct Expression Patterns of Proliferation or Adhesion, Respectively

To functionally characterize the unique transcriptional patterns in nMA-INA6 and MA-INA6, we generated lists of genes that were differentially expressed *vs.* the other two subpopulations [termed nMA *vs.* (MA & CM) and MA *vs.* (nMA & CM)]. Functional enrichment analysis was performed, and the enriched terms were displayed as ontology clusters (Fig. 5A). nMA-INA6 upregulated genes enriched with loosely connected term clusters associated with proliferation (e.g., "positive regulation of cell cycle"). MA-INA6 upregulated genes enriched with tightly connected term clusters related to cell adhesion and the production of ECM factors (e.g., "cell-substrate adhesion"). Similar ontology terms were enriched in the gene lists obtained from pairwise comparisons nMA *vs.* CM, MA *vs.* CM, and MA *vs.* nMA (Fig. 5B). In particular, nMA *vs.* CM (but not MA *vs.* CM) upregulated genes that were enriched with "G1/S transition", showing that WPSC isolated nMA-INA6 daughter cells after cell division.

To check for similarities between lists of differentially expressed genes from hMSC-interacting subpopulations, we performed enrichment analysis on gene lists from the overlaps ("$\cap$") between all pairwise comparisons (Fig. 5B, Appendix A: Fig. 6), and showed the extent of these overlaps in circos plots (Fig. 5C). The overlap between MA *vs.* CM and nMA *vs.* CM showed neither enrichment with proliferation- nor adhesion-related terms but with apoptosis-related terms. A direct comparison of MSC-interacting subpopulations (MA *vs.* nMA) showed a major overlap with MA *vs.* CM (Fig. 5C, middle). This overlap was enriched with terms related to adhesion but not proliferation. Hence, MA-INA6 and nMA-INA6 mostly differed in their expression of adhesion genes.

To assess whether nMA-INA6 and MA-INA6 were regulated by separate transcription factors, we examined the enrichment of curated regulatory networks from the TRRUST database (Fig. 5B, bottom). All the lists were enriched for p53 regulation. E2F1 regulation was observed only in genes upregulated in nMA *vs.* CM and downregulated in MA *vs.* nMA. Genelists involving MA-INA6 were enriched in regulation by subunits of NF-κB (NFKB1/p105 and RELA/p65) and factors of immediate early response (SRF, JUN). Correspondingly, NF-κB and JUN are known to regulate the expression of adhesion factors in multiple myeloma and B-cell lymphoma, respectively (Blonska et al., 2015; Tai et al., 2006).

Taken together, MSC-interacting subpopulations showed unique regulatory patterns, focusing on either proliferation or adhesion.

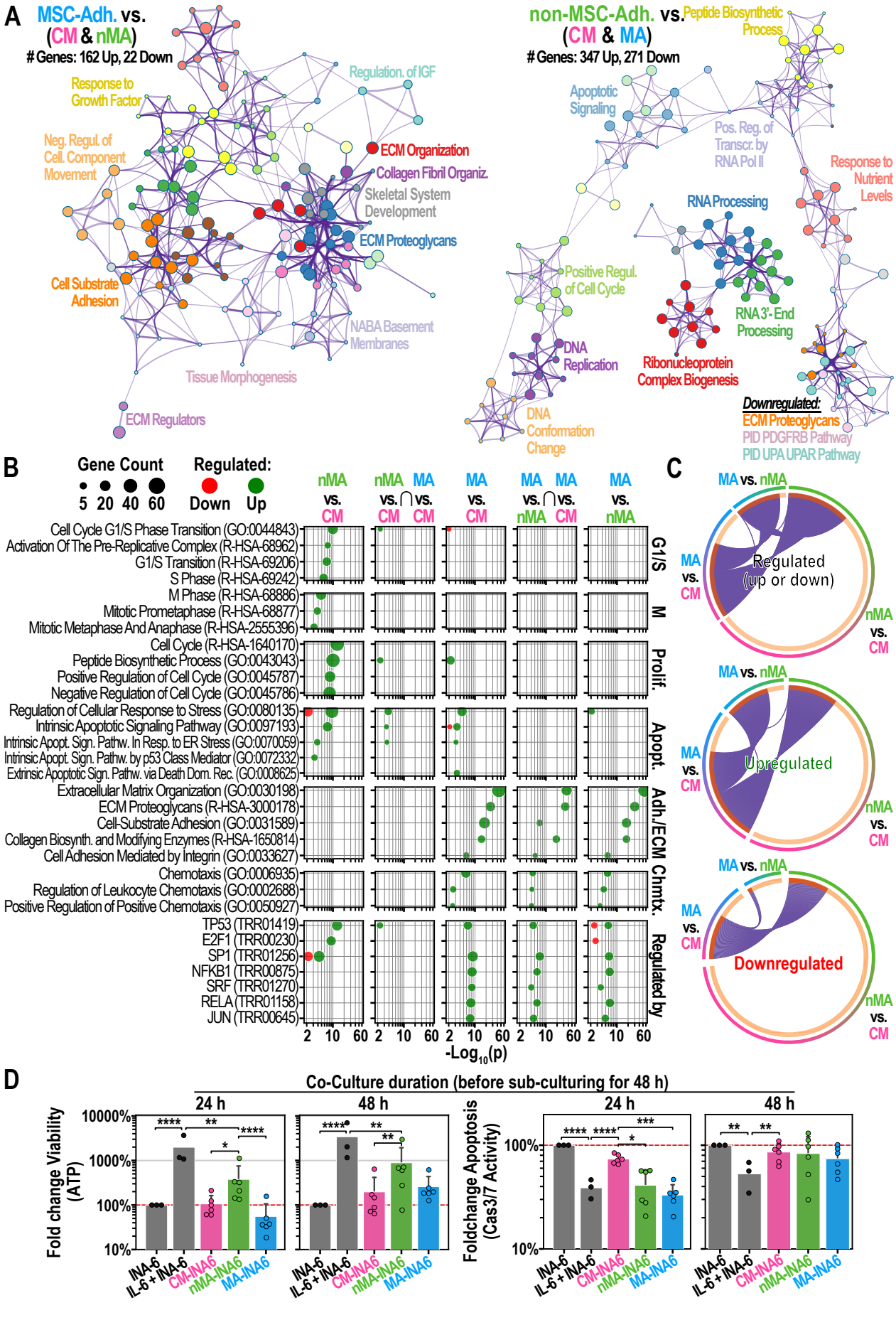

A
MSC-Adh. vs. (CM & nMA)
Genes: 162 Up, 22 Down
Response to Growth Factor
Regulation. of IGF
Neg. Regul. of Cell. Component Movement
ECM Organization
Collagen Fibril Organiz.
Skeletal System Development
ECM Proteoglycans
Cell Substrate Adhesion
NABA Basement Membranes
Tissue Morphogenesis
ECM Regulators
non-MSC-Adh. vs. (CM & MA)
Genes: 347 Up, 271 Down
Peptide Biosynthetic Process
Apoptotic Signaling
Pos. Reg. of Transcr. by RNA Pol II
Response to Nutrient Levels
RNA Processing
Positive Regul. of Cell Cycle
RNA 3'-End Processing
DNA Replication
Ribonucleoprotein Complex Biogenesis
DNA Conformation Change
Downregulated:
ECM Proteoglycans
PID PDGFRB Pathway
PID UPA UPAR Pathway
B
Gene Count
5 20 40 60
Regulated:
Down Up
nMA vs. CM
nMA vs. ∩ vs. CM CM
MA vs. CM
MA vs. ∩ vs. nMA CM
MA vs. nMA
Cell Cycle G1/S Phase Transition (GO:0044843)
Activation Of The Pre-Replicative Complex (R-HSA-68962)
G1/S Transition (R-HSA-69206)
S Phase (R-HSA-69242)
M Phase (R-HSA-68886)
Mitotic Prometaphase (R-HSA-68877)
Mitotic Metaphase And Anaphase (R-HSA-2555396)
Cell Cycle (R-HSA-1640170)
Peptide Biosynthetic Process (GO:0043043)
Positive Regulation of Cell Cycle (GO:0045787)
Negative Regulation of Cell Cycle (GO:0045786)
Regulation of Cellular Response to Stress (GO:0080135)
Intrinsic Apoptotic Signaling Pathway (GO:0097193)
Intrinsic Apopt. Sign. Pathw. In Resp. to ER Stress (GO:0070059)
Intrinsic Apopt. Sign. Pathw. by p53 Class Mediator (GO:0072332)
Extrinsic Apoptotic Sign. Pathw. via Death Dom. Rec. (GO:0008625)
Extracellular Matrix Organization (GO:0030198)
ECM Proteoglycans (R-HSA-3000178)
Cell-Substrate Adhesion (GO:0031589)
Collagen Biosynth. and Modifying Enzymes (R-HSA-1650814)
Cell Adhesion Mediated by Integrin (GO:0033627)
Chemotaxis (GO:0006935)
Regulation of Leukocyte Chemotaxis (GO:0002688)
Positive Regulation of Positive Chemotaxis (GO:0050927)
TP53 (TRR01419)
E2F1 (TRR00230)
SP1 (TRR01256)
NFKB1 (TRR00875)
SRF (TRR01270)
RELA (TRR01158)
JUN (TRR00645)
G1/S
M
Prolif.
Apopt.
Adh./ECM
Chmtx.
Regulated by
2 10 60
-Log10(p)
C
MA vs. nMA
MA vs. CM
nMA vs. CM
Regulated (up or down)
Upregulated
Downregulated
D
Co-Culture duration (before sub-culturing for 48 h)
24 h
48 h
Fold change Viability (ATP)
Foldchange Apoptosis (Cas3/7 Activity)
10000%
1000%
100%
10%
INA-6
IL-6 + INA-6
CM-INA6
nMA-INA6
MA-INA6

Figure 5: Functional analysis of MSC-interacting subpopulations **(A-C):** Functional enrichment analysis of differentially expressed genes (from RNAseq) using Metascape. **A:** Gene ontology (GO) cluster analysis of gene lists that are unique for MA (left) or nMA (right) INA-6. Circle nodes represent subsets of input genes falling into similar GO-term. Node size grows with the number of input genes. Node color defines a shared parent GO-term. Two nodes with a similarity score > 0.3 are linked. **B:** Enrichment analysis of pairwise comparisons between MA subpopulations and their overlaps (arranged in columns). GO terms were manually picked and categorized (arranged in rows). Raw Metascape results are shown in Appendix A: Fig. 6. For each GO-term, the p-values (x-axis) and the counts of matching input genes (circle size) were plotted. The lowest row shows enrichment of gene lists from the TRRUST-database. **C:** Circos plots by Metascape. Sections of a circle represent lists of differentially expressed genes. Purple lines connect same genes appearing in two gene lists. ∩: Overlapping groups, MA: MSC-adhering, nMA: non-MSC-adhering, CM: MSC-Conditioned Medium. **D:** INA-6 were co-cultured on confluent hMSC for 24 h or 48 h, separated by WPSC and sub-cultured for 48 h under IL-6 withdrawal ($n = 6$), except the control (IL-6 + INA-6) ($n = 3$). Signals were normalized (red line) to INA-6 cells grown without hMSCs and IL-6 ($n = 3$). **Statistics:** (D): Paired t-test, two-factor RM-ANOVA. Datapoints represent the mean of four technical replicates. INA-6 were isolated from independent co-cultures with hMSCs from six unique donors.

nMA-INA6 and MA-INA6 Show Increased Apoptosis Signaling Mediated by ER-Stress, p53 and Death Domain Receptors

As previously stated, apoptosis rates increased in INA-6 cells grown on confluent hMSCs compared to CM-INA6 cells after 24 hours of co-culture (Fig. 1D). Since this setup was similar to that used to separate hMSC-interacting subpopulations using WPSC, we looked for enrichment of apoptosis-related terms (Fig. 5B). "Regulation of cellular response to stress" and "intrinsic apoptotic signaling pathway (in response to ER-stress)" are terms that were enriched in nMA *vs.* CM, MA *vs.* CM and their overlap. We also found specific stressors for either nMA-INA6 ("intrinsic apoptotic signaling pathway by p53 class mediator") or MA-INA6 ("extrinsic apoptotic signaling pathway via death domain receptor"). Therefore, apoptosis may be driven by ER stress in both nMA-INA6 and MA-INA6, but also by individual pathways such as p53 and death domain receptors, respectively.

nMA-INA6 and MA-INA6 Regulate Genes Associated with Bone Loss

Myeloma cells cause bone loss by degradation and dysregulation of bone turnover via *DKK1* and *OPG* (Standal et al., 2002; Van Valckenborgh et al., 2004; F. Zhou et al., 2013). RNAseq of hMSC-interacting subpopulations showed enrichment with functional terms "skeletal system development" and "ossification" (Fig. 5A, Appendix A: Fig. 6), as well as the regulation of *MMP2*, *MMP14*, *DKK1*, and *OPG*. Validation by qPCR (Fig. 4C, D) showed that MA-INA6 significantly upregulated both *MMP14* and *MMP2* compared with either nMA-INA6 or CM-INA6. The expression of *DKK1*, however, was upregulated significantly in nMA-INA6 (and not significantly upregulated in MA-INA6), while *OPG* was significantly downregulated only in nMA-INA6.

Together, hMSC-interacting subpopulations might contribute to bone loss through different mechanisms: MA-INA6 expression of matrix metalloproteinases and nMA-INA6 cells via paracrine signaling.

MA-INA6 Upregulate Collagen and Chemokines Associated with Bone Marrow Retention

Retention of myeloma cells within the bone marrow is mediated by adhesion to the ECM (e.g., collagen VI) and the secretion of chemokines (*CXCL8* and *CXCL12*), potentially counteracting dissemination (Alsayed et al., 2007; Katz, 2010). RNAseq of hMSC-interacting subpopulations showed that genes upregulated in MA-INA6 were enriched with collagen biosynthesis and modifying enzymes, as well as chemotaxis and chemotaxis-related terms (Fig. 5B). Using qPCR, we validated the upregulation of collagen crosslinkers (*LOX* and *TGM2*), collagen-binding *DCN*, and chemokines (*CXCL8* and *CXCL12*) in MA-INA6 compared with both nMA-INA6 and CM-INA6 (Fig. 4D). Therefore, MA-INA6 can provide both an adhesive surface and soluble signals for the retention of malignant plasma cells in the bone marrow.

nMA-INA6 Show Highest Viability During IL-6 Withdrawal

Although RNAseq did not reveal IL-6 induction in any WPSC-isolated subpopulation, nMA-INA6 upregulated *IGF-1* 1.35-fold [RNAseq, nMA *vs.* (MA & CM)], which was shown to stimulate growth in CD45+ and IL-6 dependent myeloma cell lines such as INA-6, implying increased autonomy for nMA-INA6 (40). To test the autonomy of hMSC-interacting INA-6 subpopulations, we isolated them using WPSC after 24 hours and 48 hours of co-culture, sub-cultured them for 48 hours under IL-6 withdrawal, and measured both viability and apoptosis (Fig. 5D). Among the subpopulations, nMA-INA6 was the most viable. Compared to MA-INA6, nMA-INA6 increased cell viability by 8 or 4 fold when co-cultured for 24 hours or 48 hours, respectively [Hedges g of Log_{10}(Fold Change) = 2.31 or 0.82]. However, the difference was no longer significant after 48 hours of co-culture, probably because nMA-INA6 adhered to the hMSC layer (turning into MA-INA6) during prolonged co-culture, which could also explain why the viability of MA-INA6 cell subcultures increased with prolonged co-culture. Nevertheless, nMA-INA6 did not achieve the same viability as that of INA-6 cells cultured with IL-6. Despite the differences in viability, subcultures of hMSC-interacting subpopulations did not show any differences in caspase 3/7 activity when co-cultured for 48 hours (Fig. 5D, right).

Overall, among the hMSC-interacting subpopulations, nMA-INA6 had the highest chance of surviving IL-6 withdrawal.

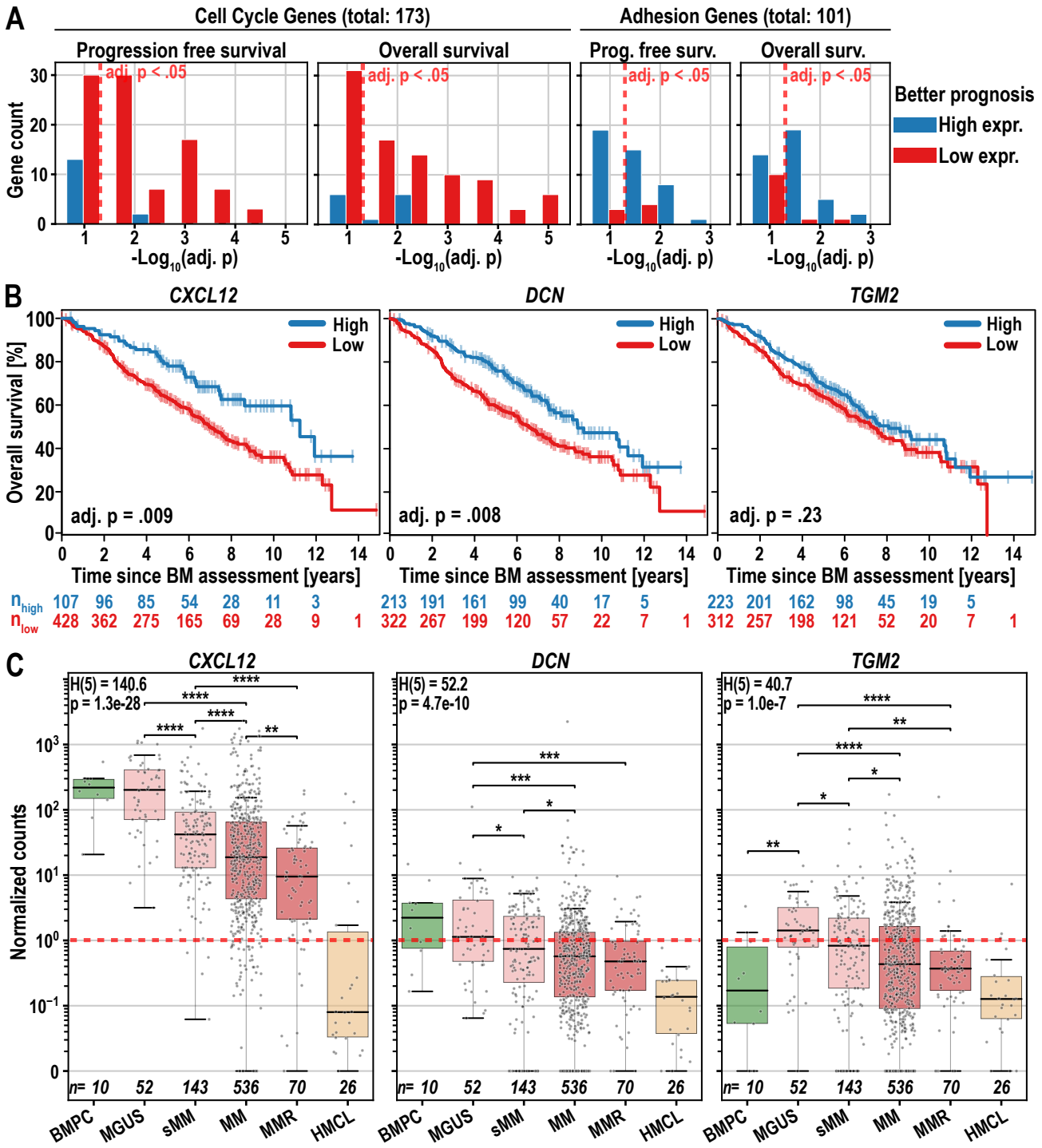


Figure 6: Survival of patients with multiple myeloma regarding the expression levels of adhesion and bone retention genes. **A:** p-value distribution of genes associated with patient survival ($n = 535$) depending on high or low expression levels. Red dashed line marks the significance threshold of p-adj $= 0.05$. Histogram of p-values was plotted using a bin width of $-\log_{10}(0.05)/2$. Patients with high and low gene expression were delineated using maximally selected rank statistics (maxstat). **B:** Survival curves for three genes taken from the list of adhesion genes shown in (A), maxstat thresholds defining high and low expression were: *CXCL12*: 81.08; *DCN*: 0.75; *TGM2*: 0.66 normalized counts. **C:** Gene expression (RNAseq, $n = 873$) measured in normalized counts (edgeR) of *CXCL12*, *DCN* in Bone Marrow Plasma Cell (BMPC), Monoclonal Gammopathy of Undetermined Significance (MGUS), smoldering Multiple Myeloma (sMM), Multiple – *continued on next page*

Figure 6: *continued from previous page* – Myeloma Relapse (MMR), Human Myeloma Cell Lines (HMCL). The red dashed line marks one normalized read count. **Statistics:** (A, B): Log-rank test; (C): Kruskal-Wallis, Mann–Whitney U Test. All *p*-values were corrected using the Benjamini-Hochberg procedure.

Genes Upregulated by MA-INA6 are Associated with an Improved Disease Prognosis

To relate the adhesion of MA-INA6 observed *in vitro* to the progression of multiple myeloma, we assessed patient survival [$n = 535$, Seckinger et al. (2017, 2018)] depending on the expression level of 101 genes, which were upregulated in MA *vs.* (nMA & CM) and are part of the ontology terms "Extracellular matrix organization," "ECM proteoglycans," "cell-substrate adhesion," and "negative regulation of cell-substrate adhesion" (Fig. 6A, Appendix A: Tab. 2). As a reference, we generated a list of 173 cell cycle-related genes that were upregulated by nMA *vs.* (MA & CM).

As expected, longer patient survival was associated with low expression of the majority of cell cycle genes [71 or 68 genes for progression-free survival (PFS) or overall survival (OS)]. Only a few cell cycle genes (two for PFS and seven for OS) were associated with survival when highly expressed. Intriguingly, adhesion genes showed an inverse pattern: a large group of adhesion genes (24 for PFS and 26 for OS) was significantly associated with improved survival when highly expressed, whereas only a few genes (two for PFS and four for OS) improved survival when expressed at low levels (Tab. 1). We concluded that the myeloma-dependent expression of adhesion factors determined in our *in vitro* study correlates with improved patient survival.

Expression of Adhesion- or Retention-related Genes (*CXCL12*, *DCN* and *TGM2*) is Decreased During Progression of Multiple Myeloma

To examine how the disease stage affects the adhesion and bone marrow retention of myeloma cells *in vitro*, we analyzed the expression of *CXCL12* in healthy plasma cell (BMPC) cohorts of patients at different disease stages and in myeloma cell lines (HMCL) [described in Seckinger et al. (2018)] (Fig. 6C). We also included *DCN* and *TGM2* since both are suggested to inhibit metastasis in different cancers by promoting cell-matrix interactions (Hu et al., 2021; Tabolacci et al., 2019). In accordance with independent reports (Huang et al., 2015; Bao et al., 2013), high expression of *CXCL12* and *DCN* by myeloma cells was associated with improved overall survival (adj. $p = .009$ and .008, respectively) (Fig. 6B).

Table 1: Adhesion and ECM genes (shown in Fig. 6A) were filtered by their association with patient survival (p-adj. < 0.01) and were categorized as continuously downregulated during disease progression. The complete list is presented in Appendix A: Tab. 2. Bone Marrow Plasma Cells (BMPC), Monoclonal Gammopathy of Undetermined Significance (MGUS), smoldering Multiple Myeloma (sMM), Multiple Myeloma (MM), and Multiple Myeloma Relapse (MMR). p-unc: unadjusted p-values; p-adj: p-values adjusted using the Benjamini-Hochberg method with 101 genes.

Regulation during disease progression	**Gene**	**Ensemble ID**	**Progression Free / Overall Survival**	**Better Prognosis with high/low expression**	**Association of expression with survival**	
					[p-unc]	**[p-adj]**
Not Downregulated (or overall low expression)	**CCNE2**	ENSG00000175305	Overall	low	5.34E-04	8.64E-03
	MMP2	ENSG00000087245	Prog. Free	high	2.29E-05	2.32E-03
	OSMR	ENSG00000145623	Prog. Free	high	5.67E-04	7.15E-03
Continuously Downregulated (BMPC > MGUS > sMM > MM > MMR)	**AXL**	ENSG00000167601	Overall	high	3.64E-05	1.84E-03
	COL1A1	ENSG00000108821	Prog. Free	high	3.03E-04	4.37E-03
			Overall	high	5.93E-04	8.64E-03
	CXCL12	ENSG00000107562	Prog. Free	high	1.16E-04	2.93E-03
			Overall	high	6.48E-04	8.64E-03
	CYP1B1	ENSG00000138061	Overall	high	6.84E-04	8.64E-03
	DCN	ENSG00000011465	Overall	high	2.47E-04	8.33E-03
	LRP1	ENSG00000123384	Overall	high	4.34E-04	8.64E-03
	LTBP2	ENSG00000119681	Prog. Free	high	9.03E-05	2.93E-03
	CYP1B1	ENSG00000138061	Overall	high	6.84E-04	8.64E-03
	DCN	ENSG00000011465	Overall	high	2.47E-04	8.33E-03
	LRP1	ENSG00000123384	Overall	high	4.34E-04	8.64E-03
	LTBP2	ENSG00000119681	Prog. Free	high	9.03E-05	2.93E-03
	MFAP5	ENSG00000197614	Prog. Free	high	2.43E-04	4.09E-03
	MMP14	ENSG00000157227	Prog. Free	high	6.93E-05	2.93E-03
	MYL9	ENSG00000101335	Prog. Free	high	1.46E-04	2.95E-03
			Overall	high	1.56E-05	1.57E-03

CXCL12 is expressed by BMPCs (median = 219 normalized counts), but its expression levels are significantly lower from MGUS to relapsed multiple myeloma (MMR) (median = 9 normalized counts in MMR and absent expression in most HMCL). *DCN* (but not *TGM2*) was weakly expressed in BMPCs ($Q_1 = 0.7$, $Q_3 = 3.7$, normalized counts), whereas *TGM2* was weakly expressed only in patients with monoclonal gammopathy of undetermined significance (MGUS) ($Q_1 = 0.4$, $Q_3 = 4.1$ normalized counts). The median and upper quartiles of both *DCN*- and *TGM2* decreased continuously after each stage, ending at $Q_3 = 0.9$ and $Q_3 = 0.6$,

respectively, in MMR. 49 of the 101 adhesion genes (Fig. 6A) followed a similar pattern of continuous downregulation in the advanced stages of multiple myeloma (Appendix A: Fig. 7 and 8), of which 19 genes were associated with longer PFS when they were highly expressed. The other 52 (out of 101) adhesion genes that were not downregulated across disease progression (or were expressed at a level too low to make that categorization) contained only five genes that were associated with longer PFS at high expression (Tab. 1, Appendix A: Tab. 2).

Together, the expression of adhesion or bone marrow retention-related markers (*CXCL12*, *DCN*, and *TGM2*) is reduced or lost at advanced stages of multiple myeloma, which could enhance dissemination and reduce retention in the BM microenvironment.

Discussion

In this study, we developed an *in vitro* model to investigate the attachment/detachment dynamics of INA-6 cells to/from hMSCs and established methods to isolate the attached and detached intermediates nMA-INA6 and MA-INA6. Secondly, we characterized a cycle of (re)attachment, division, and detachment, linking cell division to the switch that causes myeloma cells to detach from hMSC adhesion (Fig. 7). Thirdly, we identified clinically relevant genes associated with patient survival, where better or worse survival was based on the adherence status of INA-6 to hMSCs.

INA-6 cells emerged as a robust choice for studying myeloma dissemination *in vitro*, showing rapid and strong adherence, as well as aggregation exceeding MSC saturation. The IL-6 dependency of INA-6 enhanced the resemblance of myeloma cell lines to patient samples, with INA-6 ranking 13th among 66 cell lines (Sarin et al., 2020). Despite variations in bone marrow MSCs between multiple myeloma and healthy states, we anticipated the robustness of our results, given the persistent strong adherence and growth signaling from MSCs to INA-6 during co-cultures (Dotterweich et al., 2016).

We acknowledge that INA-6 cells alone cannot fully represent the complexity of myeloma aggregation and detachment dynamics. However, the diverse adhesive properties of myeloma cell lines pose a challenge. We reasoned that attempting to capture this complexity within a single publication would not be possible. Our focus on INA-6 interactions with hMSCs allowed for a detailed exploration of the observed phenomena, such as the unique aggregation capabilities that facilitate the easy detection of detaching cells *in vitro*. The validity of our data was demonstrated by matching the *in vitro* findings with the gene expression and survival data of the patients (e.g., *CXCL12*, *DCN*, and *TGM2* expression, $n = 873$), ensuring biological consistency and generalizability regardless of the cell line used. The protocols presented in this study offer a cost-efficient and convenient solution, making them potentially valuable for a broader study of cell interactions. We encourage optimizations to meet the varied adhesive properties of the samples, such as decreasing the number of washing steps if the adhesive strength is low. We caution against strategies that average over multiple cell lines without prior understanding their diverse attachment/detachment dynamics, such as homotypic aggregation. Such detailed insights may prove instrumental when considering the diversity of myeloma patient samples across different disease stages (Kawano et al., 1991; Okuno et al., 1991).

The intermediates, nMA-INA6 and MA-INA6, were distinct but shared similarities in response to cell stress, intrinsic apoptosis, and regulation by p53. Unique regulatory patterns were related to central transcription factors: E2F1 for nMA-INA6; and NF-κB, SRF, and JUN for MA-INA6. This distinction may have been established through antagonism between p53

and the NF-κB subunit RELA/p65 (Wadgaonkar et al., 1999; Webster & Perkins, 1999). Similar regulatory patterns were found in transwell experiments with RPMI1-8226 myeloma cells, where direct contact with the MSC cell line HS5 led to NF-κB signaling and soluble factors to E2F signaling (Dziadowicz et al., 2022).

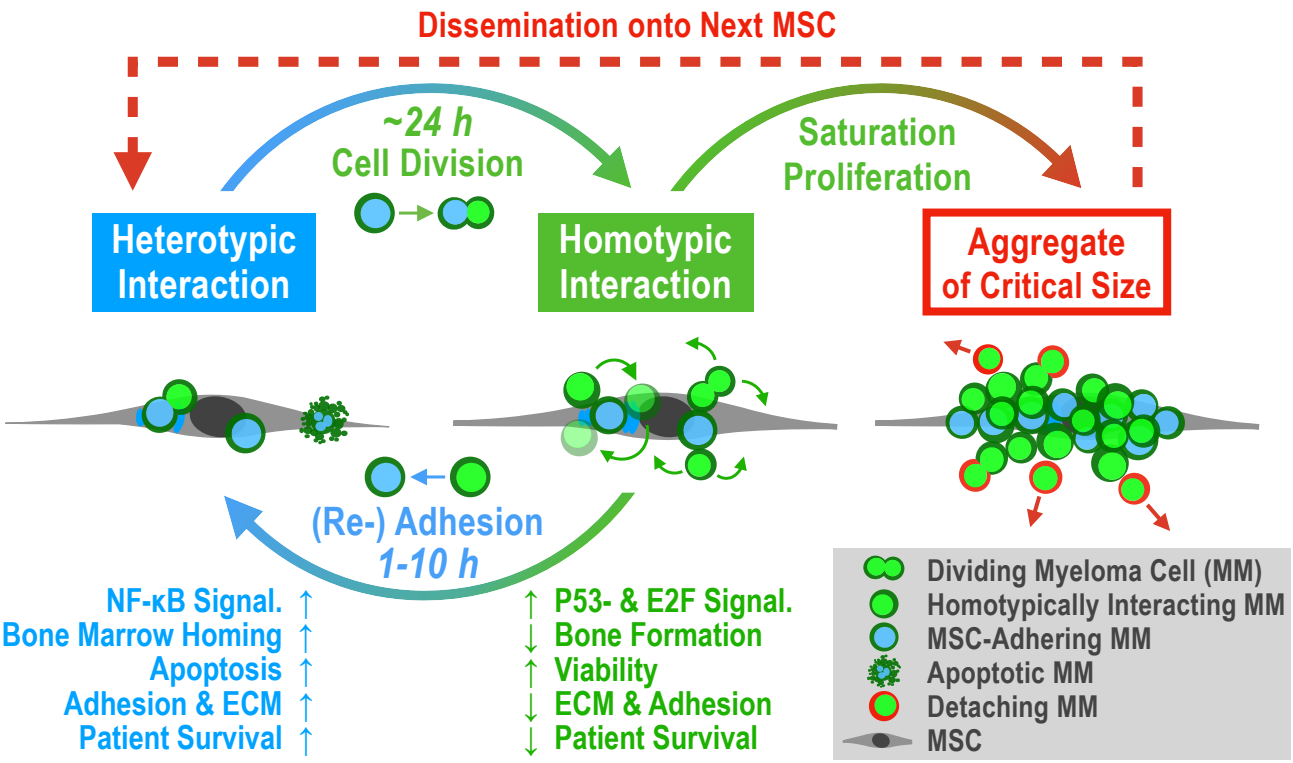


Figure 7: Proposed model of "Detached Daughter Driven Dissemination" (DDDD) in aggregating multiple myeloma. **Heterotypic Interaction:** Malignant plasma cells colonize the bone marrow microenvironment by adhering to an MSC (or osteoblast, ECM, etc.) to maximize growth and survival through paracrine and adhesion mediated signaling, even if contact may trigger initial apoptosis. Gene expression will focus on establishing a strong anchor within the bone marrow, but also on attracting other myeloma cells (via secretion of ECM factors and CXCL12/CXCL8, respectively). **Cell Division:** Cell fission can generate one daughter cell that no longer adheres to the MSC (nMA). **Homotypic Interaction:** If myeloma cells have the capacity to grow as aggregates, the daughter cell stays attached to their MSC-adhering mother cell (MA). **Re-Adhesion:** The daughter cell "rolls around" the mother cell until it re-adheres to the MSC. Our model estimates the rolling duration to be 1–10 h long. **Proliferation & Saturation:** We estimate that a single myeloma cell covers one MSC completely after roughly four population doublings. When heterotypic adhesion is saturated, subsequent daughter cells benefit from a homotypic interaction, since they stay close to growth-factor secreting MSCs and focus gene expression on proliferation (e.g. driven by E2F) and not adhesion (driven by NF- B). **Critical Size:** Homotypic interaction is weaker than heterotypic interaction, and each cell fission destabilizes the aggregate. Hence, detachment of myeloma cells may depend mostly on aggregate size. **Dissemination:** After myeloma cells have detached, they gained a viability advantage through IL-6-independence (with unknown duration), which enhances their survival outside of the bone marrow and allows them to spread throughout the body.

The first subpopulation, nMA-INA6, represented proliferative and disseminative cells; They drove detachment through cell division, which was regulated by E2F, p53, and likely their

crosstalk (Polager & Ginsberg, 2009). nMA-INA6 upregulate cell cycle progression genes associated with worse prognosis, because proliferation is a general risk factor for an aggressive disease course (Hose et al., 2011). Additionally, nMA-INA6 survived IL-6 withdrawal better than CM-INA6 and MA-INA6, implying their ability to proliferate independently of the bone marrow (Bladé et al., 2022). Indeed, xenografted INA-6 cells developed autocrine IL-6 signaling but remained IL-6-dependent after explantation (Burger, Günther, et al., 2001). The increased autonomy of nMA-INA6 cells can be explained by the upregulation of *IGF-1*, being the major growth factor for myeloma cell lines (Sprynski et al., 2009). Other reports characterized disseminating cells differently: Unlike nMA-INA6, circulating myeloma tumor cells were reported to be non-proliferative and bone marrow retentive (Garcés et al., 2020). In contrast to circulating myeloma tumor cells, nMA-INA6 were isolated shortly after detachment and therefore these cells are not representative of further steps of dissemination, such as intravasation, circulation or intravascular arrest (Zeissig et al., 2020). Furthermore, Brandl et al. described proliferative and disseminative myeloma cells as separate entities, depending on the surface expression of CD138 or JAM-C (Akhmetzyanova et al., 2020; Brandl et al., 2022). Although CD138 was not differentially regulated in nMA-INA6 or MA-INA6, both subpopulations upregulated JAM-C, indicating disease progression (Brandl et al., 2022).

Furthermore, nMA-INA6 showed that cell division directly contributed to dissemination. This was because INA-6 daughter cells emerged from the mother cell with distance to the hMSC plane in the 2D setup. A similar mechanism was described in an intravasation model in which tumor cells disrupt the vessel endothelium through cell division and detach into blood circulation (Wong & Searson, 2017). Overall, cell division offers key mechanistic insights into dissemination and metastasis.

The other subpopulation, MA-INA6, represented cells retained in the bone marrow; MA-INA6 strongly adhered to MSCs, showed NF-κB signaling, and upregulated several retention, adhesion, and ECM factors. The production of ECM-associated factors has recently been described in MM.1S and RPMI-8226 myeloma cells (Maichl et al., 2023). Another report did not identify the upregulation of such factors after direct contact with the MSC cell line HS5; hence, primary hMSCs may be crucial for studying myeloma-MSC interactions (Dziadowicz et al., 2022). Moreover, MA-INA6 upregulated adhesion genes associated with prolonged patient survival and showed decreased expression in relapsed myeloma. As myeloma progression implies the independence of myeloma cells from the bone marrow (Bladé et al., 2022; Sarin et al., 2020), we interpreted these adhesion genes as mediators of bone marrow retention, decreasing the risk for dissemination and thereby potentially prolonging patient survival. However, the overall impact of cell adhesion and ECM on patient survival remains unclear. Several adhesion factors have been proposed as potential therapeutic targets (Brandl et al., 2022; Bou Zerdan et al., n.d.). Recent studies have described the prognostic value of multiple ECM genes, such as

those driven by NOTCH (Maichl et al., 2023). Another study focused on ECM gene families, of which only six of the 26 genes overlapped with our gene set (Appendix A: Tab. 2) (Evers et al., 2023). The expression of only one gene (*COL4A1*) showed a different association with overall survival than that in our cohort. The lack of overlap and differences can be explained by dissimilar definitions of gene sets (homology *vs.* gene ontology), methodological discrepancies, and cohort composition.

In summary, our *in vitro* model provides a starting point for understanding the initiation of dissemination and its implications for patient survival, providing innovative methods, mechanistic insights into attachment/detachment, and a set of clinically relevant genes that play a role in bone marrow retention. These results and methods might prove useful when facing the heterogeneity of disseminative behaviors among myeloma cell lines and primary materials.

Chapter 2: Semi-Automating Data Analysis with plotastic

Abstract

plotastic addresses the challenges of transitioning from exploratory data analysis to hypothesis testing in Python's data science ecosystem. Bridging the gap between seaborn and pingouin, this library offers a unified environment for plotting and statistical analysis. It simplifies the workflow with user-friendly syntax and seamless integration with familiar seaborn parameters (y, x, hue, row, col). Inspired by seaborn's consistency, plotastic utilizes a DataAnalysis object to intelligently pass parameters to pingouin statistical functions. Hence, statistics and plotting are performed on the same set of parameters, so that the strength of seaborn in visualizing multidimensional data is extended onto statistical analysis. In essence, plotastic translates seaborn parameters into statistical terms, configures statistical protocols based on intuitive plotting syntax and returns a matplotlib figure with known customization options and more. This approach streamlines data analysis, allowing researchers to focus on correct statistical testing and less about specific syntax and implementations.

Introduction

The reproducibility crisis in research highlights a significant challenge in contemporary biosciences, where a substantial portion of studies faces reproducibility issues (Baker, 2016; Begley & Ioannidis, 2015; Gosselin, 2021). One critical yet often overlooked aspect contributing to this crisis is data management. The literature most often refers to *big data* as the main challenge (Gomez-Cabrero et al., 2014). However, these challenges are also present in smaller datasets, which the author refers to as *semi-big data*. This term describes datasets that – while not extensive enough to necessitate advanced computational tools typically reserved for *big data* – are sufficiently large to render manual analysis very time-intensive. Semi-big data is often generated by methods like automated microscopy or multiplex qPCR, which produce volumes of data that are manageable on a surface level, but pose substantial barriers for in-depth, manual reproducibility (Bustin, 2014; Incerti et al., 2019). This is further complicated by the complexity inherent in multidimensional datasets (Krzywinski & Savig, 2013): Modern biosciences describe processes (e.g. cell-adhesion) that are highly dependent on multiple experimental parameters (factors), like *'time'* or *'kinds of treatments'* (Rebl et al., 2010; McKay et al., 1997). Manually grouping the data by multiple factors (facetting) is challenging and error-prone, especially when the data is not structured in a way that is immediately compatible with statistical tests. Without a clearly documented data analysis protocol and standardized data formats, analysis of multidimensional data becomes nontransparent and too overwhelming for reproduction (Bustin, 2014).

The evolving standards in data analysis advocate for the standardization of analytical pipelines, rationalization of sample sizes, and enhanced infrastructure for data storage, addressing some of these challenges (Goodman et al., 2016; Wilkinson et al., 2016). However, these advancements can place undue pressure on researchers, particularly those with limited training in statistics, underscoring the need for intuitive, user-friendly analytical tools (Federer et al., 2016; Lakhlifi et al., 2023; Armstrong, 2014; Gómez-López et al., 2019; Leek & Peng, 2015).

In this context, `plotastic` emerges as a tool designed to democratize access to sophisticated statistical analysis, offering a user-centric interface that caters to researchers across varying levels of statistical proficiency. `plotastic` simplifies inferential statistics building on the idea that statistical analyses are often performed based on how the data is visualized. This principle is not only intuitive but also statistically sound, because the parameters that structure the figure (e.g. facetting) are often times re-used for statistical testing (e.g. independent variables or factors). By integrating robust statistical methodologies within an accessible framework, `plotastic` could to contribute to enhancing the reproducibility of research in the biosciences (Gomez-Cabrero et al., 2014).

plotastic key design feature is centralizing the facetting parameters utilized by seaborn into a DataAnalysis object. seaborn uses the parameters x, hue, col, row as arguments for many plotting functions. These parameters lay out the structure of the plot, such as which data points are shown on the x-axis, what categories are highlighted by color (hue), and how data is grouped into separate plots (by columns and/or rows). By centralizing these parameters, plotastic ensures that all subsequent analysis steps do not require the user to re-specify these parameters, automating not only statistical analysis, but also all edits applied to the plot, such as p-value annotations.

The user-centric approach of plotastic distinguishes itself from the fully automated pipelines used for big data, which are designed to handle extensive computational tasks. Instead, plotastic focuses on ease-of-use and structures its commands to enable an interactive review of intermediate outputs, a concept the author refers to as *semi-automation* (Tab. 1).

Table 1: Key Principles of Semi-Automation and their Implementation in Plotastic

No.	Principle	Implementation in plotastic
1	**Standardized input:** The data to-be-analyzed follows a strict standardized format. The user should be able to convert their data into that format.	Long-format pandas DataFrames are used as input
2	**Automation over flexibility:** If there is an obvious way to do things, automate it and minimize user input. User options should be added with good reason. Avoid situations where the user is asked to pass the same parameter twice. This reduces the risk of human error, confusion and time spent on configuration.	E.g. passing the parameter "subject" once makes the rest of the pipeline switch automatically to the paired versions of statistical tests.
3	***Out of the box* functionality:** The software's default configuration should provide acceptable (but potentially sub-optimal) results. Beginners should be invited to experiment without the need to learn custom configurations. Options are still available to allow feature-rich adaptions according to the needs of both data and user.	Default tests are standard unpaired t-tests and ANOVA
4	**Focus on intermediate outputs:** The user composes the analysis pipeline using smaller commands that are each designed to provide human-readable output of an intermediate result. Each step is a stage to control quality, allowing quick error detection and troubleshooting.	Processing steps are separated into main steps: assumption tests, factor analysis, post-hoc analysis and plotting
5	**Highly useful error messages:** Never leave the user hanging. Tell him what went wrong *and* what the software was expecting.	E.g.: `ValueError: User passed 'subect' as subject, please choose one of ['subject', 'event', 'region']`

The need for plotastic in this specific project arose from two main challenges (for further details, see summarizing discussion). The first is the author's need for a tool that could handle the complex, multidimensional data generated by e.g. qPCR experiments. These experiments involved the analysis of multiple outcomes across multiple genes, timepoints, method variations, cell-types, biological replicates, technical replicates etc., resulting in datasets that are challenging to analyse manually. Such complexity was necessary, since establishing new

methods required extensive controls and creative variation of the experimental setup. Data analysis had to be automated somehow, since the lab-work itself was already time-intensive. The second challenge was to accept the potential of plot-configured statistical analyses. The author believes that the way data is visualized is often the way it should be analyzed. This vision is not limited to biomedical application, but a general principle that could benefit the scientific community overall. Making plotastic a generalized tool was a conscious decision to maximize its adoption rate and ensure its long-term relevance and quality, of which biomedical research will also benefit.

Statement of Need

Python's data science ecosystem provides powerful tools for both visualization and statistical testing. However, the transition from exploratory data analysis to hypothesis testing can be cumbersome, requiring users to switch between libraries and adapt to different syntaxes. seaborn has become a popular choice for plotting in Python, offering an intuitive interface. Its statistical functionality focuses on descriptive plots and bootstrapped confidence intervals (Waskom, 2021). The library pingouin offers an extensive set of statistical tests, but it lacks integration with common plotting capabilities (Vallat, 2018). statannotations integrates statistical testing with plot annotations, but uses a complex interface and is limited to pairwise comparisons (Charlier et al., 2022).

plotastic addresses this gap by offering a unified environment for plotting and statistical analysis. With an emphasis on user-friendly syntax and integration of familiar seaborn parameters, it simplifies the process for users already comfortable with seaborn. The library ensures a smooth workflow, from data import to hypothesis testing and visualization.

Example

The following code demonstrates how plotastic analyzes the example dataset "fmri", similar to Waskom (2021) (Fig. 1).

```
### IMPORT PLOTASTIC
import plotastic as plst

# IMPORT EXAMPLE DATA
DF, _dims = plst.load_dataset("fmri", verbose = False)

# EXPLICITLY DEFINE DIMENSIONS TO FACET BY
dims = dict(
      y = "signal",    # y-axis, dependent variable
      x = "timepoint", # x-axis, independent variable (within-subject factor)
      hue = "event",   # color,  independent variable (within-subject factor)
      col = "region"   # axes,   grouping variable
)
# INITIALIZE DATAANALYSIS OBJECT
DA = plst.DataAnalysis(
      data=DF,           # Dataframe, long format
      dims=dims,         # Dictionary with y, x, hue, col, row
      subject="subject", # Datapoints are paired by subject (optional)
      verbose=False,     # Print out info about the Data (optional)
)
# STATISTICAL TESTS
DA.check_normality()   # Check Normality
DA.check_sphericity()  # Check Sphericity
DA.omnibus_rm_anova()  # Perform RM-ANOVA
DA.test_pairwise()     # Perform Posthoc Analysis

# PLOTTING
(DA
.plot_box_strip()     # Pre-built plotting function initializes plot
.annotate_pairwise(   # Annotate results from DA.test_pairwise()
      include="__HUE" # Use only significant pairs across each hue
      )
)
```

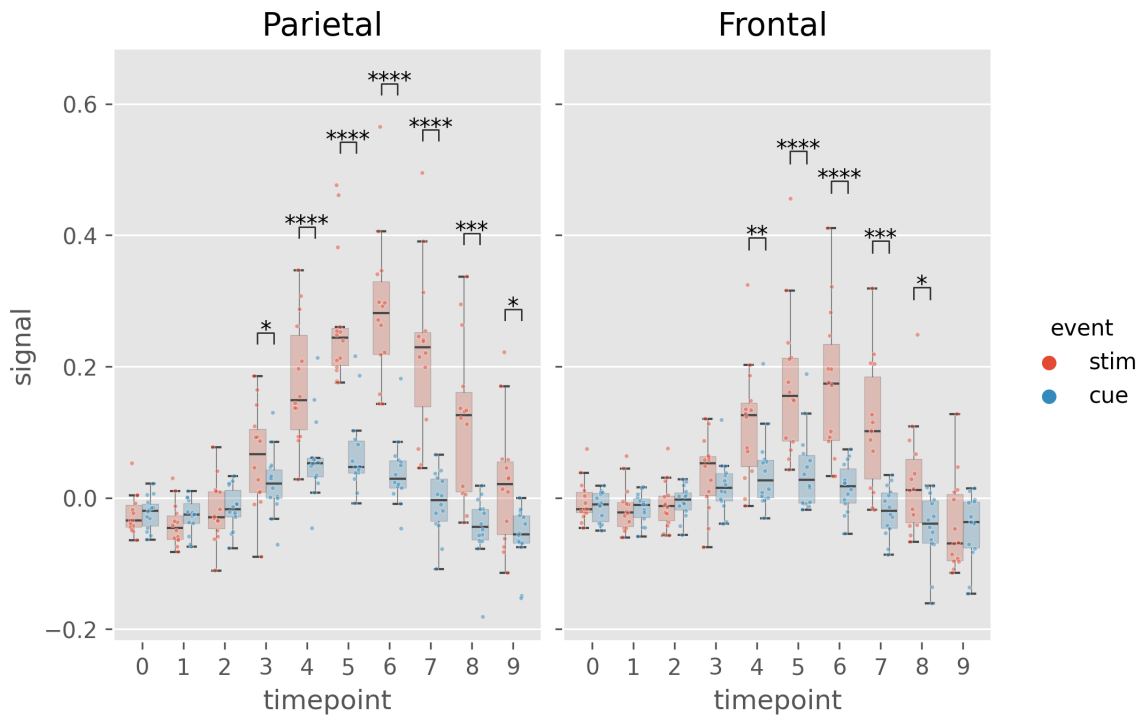


Figure 1: Example figure of plotastic (version 0.1). Image style was set by plt.style.use('ggplot')

Table 2: Results from DA.check_sphericity(). plotastic assesses sphericity after grouping the data by all grouping dimensions (hue, row, col). For example, DA.check_sphericity() grouped the 'fmri' dataset by "region" (col) and "event" (hue), performing four subsequent sphericity tests for four datasets.

'region', 'event'	spher	W	chi2	dof	pval	group count	n per group
'frontal', 'cue'	True	3.26e+20	-462.7	44	1	10	[14]
'frontal', 'stim'	True	2.45e+17	-392.2	44	1	10	[14]
'parietal', 'cue'	True	1.20e+20	-452.9	44	1	10	[14]
'parietal', 'stim'	True	2.44e+13	-301.9	44	1	10	[14]

Table 3: Results of DA.omnibus_rm_anova(). plotastic performs one two-factor RM-ANOVA per axis (grouping the data by row and col dimensions) using x and hue as the within-factors. For this example, DA.omnibus_rm_anova() grouped the 'fmri' dataset by "region" (col), performing two subsequent two-factor RM-ANOVAs. Within-factors are "timepoint" (x) and "event" (hue). For conciseness, GG-Correction and effect sizes are not shown.

'region'	Source	SS	ddof1	ddof2	MS	F	p-unc	stars
'parietal'	timepoint	1.583	9	117	0.175	26.20	3.40e-24	****
'parietal'	event	0.770	1	13	0.770	85.31	4.48e-07	****
'parietal'	timepoint * event	0.623	9	117	0.069	29.54	3.26e-26	****
'frontal'	timepoint	0.686	9	117	0.076	15.98	8.28e-17	****
'frontal'	event	0.240	1	13	0.240	23.44	3.21e-4	***
'frontal'	timepoint * event	0.242	9	117	0.026	13.031	3.23e-14	****

Overview

The functionality of plotastic revolves around a seamless integration of statistical analysis and plotting, leveraging the capabilities of pingouin, seaborn, matplotlib and statannotations (Vallat, 2018; Waskom, 2021; Hunter, 2007; Charlier et al., 2022). It utilizes long-format pandas DataFrames as its primary input, aligning with the conventions of seaborn and ensuring compatibility with existing data structures (**?**Team, 2020; McKinney, 2010).

plotastic was inspired by seaborn using the same set of intuitive and consistent parameters (y, x, hue, row, col) found in each of its plotting functions (Waskom, 2021). These parameters intuitively delineate the data dimensions plotted, yielding 'facetted' subplots, each presenting y against x. This allows for rapid and insightful exploration of multidimensional relationships. plotastic extends this principle to statistical analysis by storing these seaborn parameters (referred to as dimensions) in a DataAnalysis object and intelligently passing them to statistical functions of the pingouin library. This approach is based on the impression that most decisions during statistical analysis can be derived from how the user decides to arrange the data in a plot. This approach also prevents code repetition and streamlines statistical analysis. For example, the subject keyword is specified only once during DataAnalysis initialisation, and plotastic selects the appropriate paired or unpaired version of the test. Using pingouin alone requires the user to manually pick the correct test and to repeatedly specify the subject keyword in each testing function.

In essence, plotastic translates plotting parameters into their statistical counterparts. This translation minimizes user input and also ensures a coherent and logical connection between plotting and statistical analysis. The goal is to allow the user to focus on choosing the correct statistical test (e.g. parametric vs. non-parametric) and worry less about specific implementations.

At its core, plotastic employs iterators to systematically group data based on various dimensions, aligning the analysis with the distinct requirements of tests and plots. Normality testing is performed on each individual sample, which is achieved by splitting the data by all grouping dimensions and also the x-axis (hue, row, col, x). Sphericity and homoscedasticity testing is performed on a complete sampleset listed on the x-axis, which is achieved by splitting the data by all grouping dimensions (hue, row, col) (Tab. 2). For omnibus and posthoc analyses, data is grouped by the row and col dimensions in parallel to the matplotlib axes, before performing one two-factor analysis per axis using x and hue as the within/between-factors. (Tab. 3).

DataAnalysis visualizes data through predefined plotting functions designed for drawing multi-layered plots. A notable emphasis within plotastic is placed on showcasing individual

datapoints alongside aggregated means or medians. In detail, each plotting function initializes a matplotlib figure and axes using plt.subplots() while returning a DataAnalysis object for method chaining. Axes are populated by seaborn plotting functions (e.g., sns.boxplot()), leveraging automated aggregation and error bar displays. Keyword arguments are passed to these seaborn functions, ensuring the same degree of customization. Users can further customize plots by chaining DataAnalysis methods or by applying common matplotlib code to override plotastic settings. Figures are exported using plt.savefig().

plotastic also focuses on annotating statistical information within plots, seamlessly incorporating p-values from pairwise comparisons using statannotations (Charlier et al., 2022). This integration simplifies the interface and enables options for pair selection in multidimensional plots, enhancing both user experience and interpretability.

For statistics, plotastic integrates with the pingouin library to support classical assumption and hypothesis testing, covering parametric/non-parametric and paired/non-paired variants. Assumptions such as normality, homoscedasticity, and sphericity are tested. Omnibus tests include two-factor RM-ANOVA, ANOVA, Friedman, and Kruskal-Wallis. Posthoc tests are implemented through pingouin.pairwise_tests(), offering (paired) t-tests, Wilcoxon, and Mann-Whitney-U.

To sum up, plotastic stands as a unified and user-friendly solution catering to the needs of researchers and data scientists, seamlessly integrating statistical analysis with the power of plotting in Python. It streamlines the workflow, translates seaborn parameters into statistical terms, and supports extensive customization options for both analysis and visualization.

Discussion

As awareness of the complexities associated with multidimensional data in biomedical research increases, there is a growing demand for tools that not only simplify analysis but also enhance its intuitiveness and effectiveness (Dunn et al., 2017). plotastic is designed to meet this demand by seamlessly integrating data visualization with inferential statistics, making sophisticated statistical methods accessible to researchers of all expertise levels. This integration could be pivotal as it allows the visualization of data —how it is grouped and presented— to directly guide the statistical analysis, reducing the need for in-depth statistical knowledge and ensuring that the analyses are intuitively aligned with the visual aspects of the data. This approach could not only simplify the analytical process but also enhance the transparency and reproducibility of research findings.

Statistical Features: A detailed list of implemented and planned features is provided on the GitHub page of the project (Kuric, 2024). plotastic is comprehensive in its current scope, incorporating a robust suite of statistical tests that cater to a wide range of research needs. It includes assumption tests for normality, homoscedasticity, and sphericity, alongside classical statistical tests such as ANOVA and t-tests, available in both parametric and non-parametric forms, as well as paired and unpaired variants. However, its reliance on the pingouin library means that plotastic is subject to the same limitations as pingouin itself. For instance, it does not yet support survival analysis tools like log-rank tests and Kaplan-Meier plots, which are critical for certain biomedical applications. While there are external packages that offer these capabilities, integrating them into plotastic could significantly expand its utility and provide a more unified user experience (Davidson-Pilon, 2019).

One known issue in plotastic is its handling of multiple testing corrections. Currently, plotastic might not correctly apply these corrections when the data is split across different facets with their own y-axes (facetted by row and col keywords), which can lead to potentially incorrect statistical inferences. This is a fixable issue, and plans are in place to address it in upcoming versions to ensure that corrections for multiple testing are appropriately applied across the complete dataset. Additionally, bivariate analysis tools like correlation and regression are not yet implemented, since plotastic focused on data with a categorical x-axis, which is more common in biomedical research.

Plotting Features: The plotting capabilities of plotastic employ all of seaborn's non-facetgrid plotting functions (e.g. sns.boxplot()), which include a wide range of plot types but may not cover all possible visualizations (Waskom, 2021). Future versions could expand the range of specialized plots, for example QQ-plots. plotastic focuses on offering both high- and low-level plotting configuration: Multiplots automate overlaying multiple plot types, which

is extremely useful for displaying raw data points alongside aggregated statistics (barplots, boxplots, etc.), a feature that can be cumbersome to implement manually. Low-level plotting configuration is supported just like in seaborn, since plotastic uses matplotlib as its backend. This level of flexibility is unique to plotastic, serving both beginners and advanced users.

Plot Annotation: Annotating statistical results into plots (e.g. *** above barplots) is a key requirement in modern biomedical journals and could be key feature why researchers choose proprietary software like *GraphPad Prism* over other solutions. plotastic automates this process as well, making it a strong competitor to other statistical software. This is especially useful for re-arranging plots, since the statistical annotations are automatically updated when the plot is re-drawn. This feature is unique to plotastic and could be a key selling point for the software.

Software Testing: The development of plotastic adheres to modern software engineering principles to ensure reliability and maintainability. The project utilizes continuous integration practices, which means that with every change to the codebase, a comprehensive test suite is automatically run to identify potential bugs and ensure that new contributions do not disrupt existing functionalities. This test suite covers approximately 79 % of the testable lines of code, a statistic tracked automatically by an independent service called codecov, highlighting a strong commitment to software quality (*Codecov*, 2024).

Documentation: Documentation serves as a critical resource for enhancing user experience and adoption, especially for software aimed at users with varying levels of expertise. Currently, plotastic's documentation is focused on basic functionalities. These include detailed installation instructions, example analyses using five test datasets from seaborn that are commonly used in teaching statistics, guidelines on dimension switching with commands like DataAnalysis.switch(), and tutorials on constructing and configuring plots, annotating statistical data, and utilizing multiplot capabilities.

However, the documentation of plotastic could be significantly enhanced. Currently, it lacks a dedicated website, relying instead on GitHub-hosted Jupyter notebooks. While useful, these notebooks are not the most user-friendly or maintainable format for documentation as they can be challenging to navigate and don't update synchronously with software changes. A more robust approach would involve leveraging services like Read the Docs or Sphinx to generate and host documentation directly from the codebase (*Read the Docs*, 2024; *Sphinx*, 2024). This would not only ensure that the documentation remains up-to-date with the latest software developments but also provide a more accessible and navigable user experience, meeting the expectations of users who prefer a dedicated website for software documentation.

Usability for Non-Statisticians: plotastic aims to make statistical analysis more accessible to researchers without extensive statistical training by intuitively mapping plotting con-

cepts to statistical operations. To the author's knowledge, this approach is unique to plotastic and has great potential to make statistics easier and educational for non-statisticians. Still, the software requires responsible and self-critical usage, as emphasized by the thorough disclaimer on its GitHub page regarding the software's statistical robustness, (Kuric, 2024). The disclaimer highlights that while plotastic can facilitate gaining practical experience with statistics and provide a preliminary analysis, it is not a substitute for professional statistical consultation. It is designed to aid users in generating publication-grade figures and performing statistical tests, provided they have a basic understanding of the procedures involved or have their results verified by a statistician. To enhance usability for non-statisticians, plotastic could incorporate a system to suggest appropriate statistical tests based on data characteristics, like parametric tests for normally distributed data. This feature would guide users in selecting the correct tests, thereby augmenting the tool's functionality and broadening its appeal. Additionally, the GitHub page provides critical guidelines for responsible statistical practice, urging users to document their work in detail, understand the limitations of the tests applied, and consult professionals to validate their findings, ensuring that plotastic supports but does not replace thorough statistical analysis (Sandve et al., 2013; Kuric, 2024).

Usability for Non-Programmers: Despite the advantages of plotastic, its adoption among non-programmers in biomedicine may be challenging due to its reliance on a command-line interface (CLI), which is less intuitive for those accustomed to graphical user interfaces (GUIs). However, the integration of advanced artificial intelligence technologies, such as ChatGPT, presents a compelling case for embracing CLI. Indeed, ChatGPT is believed to potentially revolutionize medical research (Ruksakulpiwat et al., 2023).

Unlike GUIs, CLIs are highly compatible with text-based AI technologies, which can significantly lower the barrier to entry. In fact, both ChatGPT-3.5 and -4 demonstrate impressive performance in python (Arefin et al., 2023). This is a game changer, since researchers can now use similair tools as programmers and are only limited by their methodological expertise to formulate a correct prompt (Qureshi et al., 2023)[4]. For instance, when a software is not working as intended, users of a GUI are likely to be stuck without help or further research. Users of a CLI however, could utilize ChatGPT to ask for code-corrections or explanations of the code line-by-line, but also for advice on how to proceed with a statistical analysis and how to implement new features (e.g. editing a plot). Attempts to integrate AI into GUIs however have proven challenging (Gao et al., 2024).

Still, ChatGPT requires responsible use, as it is not sufficient as a standalone tool for statistical analysis (Ordak, 2023). It should also be noted that plotastic is not yet known to

[4]Kelleher (2024): *"You can now recognize and learn the language of almost anything with structure, and you can translate it to anything with structure — so text-protein, protein-text. [...] Everybody is a programmer, and the programming language of the future is called 'human."'*

ChatGPT, but could be included in future versions, depending on the popularity of plotastic.

Overall, the transition to a new data analysis software, especially one that incorporates coding, presents a learning curve. However, the advantages of plotastic in terms of analytical clarity, speed, and depth are anticipated to outweigh these initial challenges.

Adoption and Open-Source Contributions: The adoption rate of plotastic is a critical factor for its sustainability, particularly in the open-source environment where community contributions can significantly support the author in improving and maintaining the software. Since its publication in the Journal of Open Source Software on March 9, 2024, plotastic has garnered attention with 41 visits and 8 *stars* (similar to a 'like' on social media platforms) on its GitHub page. This level of engagement, while modest, shows initial interest and potential for growth. Active involvement from the community is essential for ongoing improvements; hence, efforts are being made to enhance the software's documentation and structure to attract more contributors: plotastic's GitHub page shows a detailed outline of the software's architecture as a class diagram in unified modeling language (UML) format, helping potential contributors orient themselves within plotastic's several modules and classes (shown in Appendix B.1). But further efforts are required, e.g. only few functions are documented with docstrings, which help understanding the purpose and usage of each function. Still, plotastic is a general-purpose data analysis software designed not only for biologists but for a broad range of scientific disciplines, making it a versatile tool with promising potential for wider adoption.

Contributions to Methodological Transparency and Biomedicine: plotastic standardizes statistical analysis by ensuring that it is performed alongside visual representations. This integrated approach simplifies both analysis and interpretation, facilitating smooth replication of analyses. Although it streamlines the data analysis process, it is not a complete solution to the reproducibility crisis in scientific research. Researchers must still possess a basic understanding of data analysis principles and be cautious about their reliance on scripting solutions like Python, which is less familiar to some biomedical researchers.

Statistical literacy and lack of training is a well-documented challenge among clinicians and biomedical researchers, decreasing the confidence in presenting their analyses in detail (Lakhlifi et al., 2023; Federer et al., 2016). Since plotastic alleviates some need for statistical knowledge by automating the configuration of statistical tests, the room for error is reduced, and the user can lay off some responsibility to the software, gaining confidence in presenting their analysis transparently.

Furthermore, plotastic's compatibility with the Jupyter ecosystem leverages *"simple, intuitive ways to both capture and embed computational work directly into our papers"* as advocated by Mesirov (2010). This integration makes plotastic not just a tool for analysis but also a means of enhancing the accessibility and reproducibility of scientific work. As Peng (2011)

suggests, the exploration of data and analysis code may often be sufficient to verify the quality of scientific claims. This seems plausible, given that statistical tests themselves pose rigorous requirements on the data, and the results are often not interpretable without the context data. Combining plotastic with Jupyter Notebooks provides a compelling solution to transparently integrate and document both intermediate results and analytical processes, thus furthering scientific rigor and replicability.

Overall, plotastic is useful statistical tool with the potantial to improve methodological transparency and reproducibility of research in biomedicine.

Summarising Discussion

Time-Lapse Microscopy Added Intuition to Exploratory Cell Biology

- When starting this project, dissemination has not been the main topic. - Surprisingly, Time-lapse identified detaching cells - Hence, Time-lapse proved pivotal for this project, shifting the focus onto in vitro dissemination.

Microscopy holds vast amounts of information. Cell movements themselves add a lot more info. Time-lapse video has proven invaluable for exploratory cell biology

the most important key insight on the mechanism of dissemination identified by timelapse was Cell Division - further insights were multiple time measurements

measuring the minimum time for detachments to begin, or the time required for daughter cells to re-attach to the hMSC monolayer. Such mechanistic insights

Other methods like RNAseq and survival analysis did provide molecular and clinical connections, time-lapse microscopy documented cell interactions as-is, but allowed for a deep and intuitive understanding of cryptic molecular data, placing the conclusions into a context that answer key questions about potential and limits of this study, such as the aggregation behavior of INA-6 cells.

Novel Methods of Isolating Adhering Subpopulations

In this work, innovative *in vitro* methodologies (Well Plate Sandwich Centrifugation and V-Well adhesion Assay) were developed. this was required to fill in gaps of isolating cells with minimized variability introduced by user-bias to clearly separate subpopulations and precisely quantify them.

cite all those methods for cell isolation! - Turning around wellplates: Doesn't allow isolation, just quantification - The author did not show all his washing experiments - Washing is very bad (data not shown): Highly dependent on user: position of cell on well bottom (border cells receive less force), position of pipette tip in well (depth, angle and position on bottom) - This motivated us to explore more reproducible methods

It's a challenge: either quantify cell population, or isolate them! - It's better to specialize in one method, than to do both poorly - Well Plate Sandwich Centrifugation is badly suited for quantification, but possible - we switched to developing V-well adhesion assay for quantification - We realized, V-well isolation allows both ultra precise quantification and isolation of small amounts of cells! - unmatched precision through centrifugation, no washing - But V-well pellets comprise only few cells requiring a lot of technical replicates and an untiring pipetting hand

The Well Plate Sandwich Centrifugation (WPSC) used two different techniques to dissociate MA-INA6 cells from the hMSC monolayer. This had no impact on the ratio of isolated MA-INA6 to nMA-INA6, since nMA-INA6 isolation was performed prior to dissociation using the same protocol consistently. We tried this to test if MACS was really necessary, after all it is costly, time-consuming, introduces an antibody bias and requires cell cold-treatment during antibody: Strong pipetting (*'Wash'*) and repeated Accutase treatment followed by magnetic activated cell sorting (*'MACS'*).

Outlook: High-Value Research Topics for Myeloma Research Arising from this Work

As an Outlook, the Author lists research topics arising from this work that have great potential for breakthroughs in myeloma research.

Cell Division as a Mechanism for Dissemination Initiation: The author describes how the detachment of daughter cells from the mother cell after a cycle of hMSC-(re)attachment and proliferation could be a key mechanism in myeloma dissemination. This mechanism was shown in other studies of extravasation. The author sees great potential in this mechanism as a target for future research. It is probably under-researched due to requirement of sophisticated time-lapse equipment, yet the simplicity of detachment through cell division is intriguing through its simplicity. It implies asymmetric cell division. Cancer cells are known to divide asymmetrically, e.g. moving miRNAs to one daughter cell.

Time as a Key Parameter: The area Thermodynamics of started with scientists measuring how long it takes for gases to cool down. The author claims, by measuring the time it takes for cancer cells to detach could lead to breakthroughs in research of myeloma dissemination.

- Cell adhesion is highly time-dependent. - Cell detachment is required for metastasis and dissemination -

key mechanistic insights

measuring the minimum time for detachments to begin, or the time required for daughter cells to re-attach to the hMSC monolayer. Such mechanistic insights

The author recommends high time resolutions, e.g. 1 frame every 15 min, which is a high resolution for common live cell imaging when compared to Purschke et al. (2010). Time-resolution was mostly limited by available disk space. Investing into more hard drives is worth it, since

Lists of Adhesion Gene Associated With Prolonged Patient Survival: The author lists adhesion genes that are associated with prolonged patient survival. These genes are highly expressed in myeloma samples from patients with longer overall

Conclusion 1: Cancer & Myeloma & Dissemination is bad

lorem ipsum yes yes very bad

Semi-Automation was Critical for Establishing *in vitro* Methods

In vitro research is valued for their speed at creating precise data (Moleiro et al., 2017). In this work, the development and publication of innovative *in vitro* methodologies necessitated the adoption of semi-automated data analysis tools. These novel methods introduced complexities that span multiple experimental parameters, making the results multidimensional. This demanded precise, efficient and standardized data handling capabilities which were facilitated by Python tools like `seaborn` and `plotastic` (Waskom, 2021; Kuric & Ebert, 2024).

Inherent Multidimensionality of Adhesion Studies: Cell adhesion studies often involve multiple independent parameters, posing significant analytical challenges. Two critical dimensions are particularly notable: *'Subpopulation'* and *'Time'*. Analyzing cell adhesion often involves isolation of adherent and non-adherent subpopulations, effectively introducing *'Subpopulation'* as a vital dimension in the dataset (Dziadowicz et al., 2022). This study specifically categorized cells into three levels of MSC-interaction: CM-INA6, nMA-INA6, and MA-INA6. Furthermore, the dynamic nature of cell adhesion processes is profoundly influenced by the factor *'Time'*, making it a crucial experimental parameter for investigation (Rebl et al., 2010; McKay et al., 1997; Bolado-Carrancio et al., 2020). This work includes extensive time-lapse microscopy experiments utilizing a high time resolution (1 frame every 15 min), similar to those time resolutions used by Purschke et al. (2010). This precision was required for key mechanistic insights on hMSC-INA-6 interaction dynamics. These included identifying rolling movements of nMA-INA6 daughter cells around MA-INA6 mother cells, measuring the minimum time for INA-6 detachments to begin, and measuring the time required for daughter cells to re-attach to the hMSC monolayer, etc. Next to mechanistic insights, adhesion time played a crucial methodological role in this study as well: During the V-Well adhesion experiments, we did not

know initially how long INA-6 cells required to form strong adhesion with hMSCs before pelleting nMA-INA6, but required a timepoint with hour precision to capture detachments after cell division that was accelerated through prior cell cycle synchronization at M-Phase.

The extensive facetting features of `seaborn` and `plotastic` were essential for visualizing these multidimensional datasets, allowing for quick exploration of the data (Waskom, 2021).

Further Contributions and Remedies to Multidimensional Complexity: In addition to *'Subpopulation'* and *'Time'*, this study faced additional layers of complexity that were managed through semi-automated analysis.

Experiments typically involved at least three biological replicates and corresponding technical replicates. Although replicates were not treated as independent variables — instead serving for displaying variance — they can add substantially to the data management workload. In this work, semi-automation nullified the manual burdens of handling replicates: `pandas` was used to automate aggregation of technical replicates into means after removing technical outliers thorugh z-score thresholding, while the jupyter notebook format allowed for reviewing filtered data for specific data losses. The removal of technical noise was especially relevant for qPCR data, where low gene expression can lead to sudden increase in Ct value (non-detects). In fact, the decision to remove or impute non-detects is under active discussion, however, available algorithms are hard to understand for non bioinformaticians, but also do not separate biological from technical variance, which is considered bad practice by Motulsky (2018) (McCall et al., 2014; Sherina, 2020). Semi-automation also nullified the burden of handling biological replicates: The automatic aggregation of datapoints during plotting is a key feature of `seaborn`, on which `plotastic` was built. Without such automation, calculating means and standard deviations for simple barplots would have required extensive manual computation in *Microsoft Excel*, or tedious plot configurations in *Graphpad Prism* due to limited facetting functionality of multiple variable tables (*GraphPad Prism 10 User Guide*, 2024).

Replicates can expand datasets as this factor comprises a lot of levels. Similarly, the factor *'Gene'* multiplied the dataset by a total of 30 genes when validating RNAseq data with RT-qPCR. With three subpopulations, one timepoint, eleven biological replicates, and three technical replicates, the qPCR dataset used in this study grew to 2970 datapoints to be statistically analyzed and visualized. This is a manageable size for manual analysis, but the effort involved illustrates the definition of semi-big data.

Methodological variability also introduced additional dimensions: The Well Plate Sandwich Centrifugation (WPSC) used two different techniques to dissociate MA-INA6 cells from the hMSC monolayer: Strong pipetting (*'Wash'*) and repeated Accutase treatment followed by magnetic activated cell sorting (*'MACS'*). These variations, recorded as the factor *'method'*, further complicated the dataset. Although this distinction is not discussed in this work —

rather pooled into one group —, this showcases how protocol changes can add dimensions to the dataset that are not necessarily relevant for the biological question but essential for method optimizations and validation.

Agility During Establishment of V-Well Assay: The concept of agility in laboratory research, inspired by the Agile Manifesto's principle of *"Responding to change over following a plan"* (*Manifesto for Agile Software Development*, 2001), is increasingly relevant in biomedical research (West, 2018; Quanbeck et al., 2022). This adaptive approach was particularly crucial during the development of the V-Well adhesion assay in this study. Semi-automation using python significantly enhanced this agility, allowing rapid statistical testing and visualization of data, which would have taken considerably longer if done manually. This capability enabled real-time adjustments to the experimental technique during live microscopy sessions, integrating raw data tables directly into Python scripts for immediate analysis. Such an agile and adaptive work environment, facilitated by python tools and `seaborn`, proved invaluable for refining the *in vitro* methods being developed. Additionally, the simplicity offered by `seaborn` for complex data plotting, such as the cell cycle profiling shown in Appendix A.1: Fig. 3, which required minimal code to produce a detailed series of 24 histograms, underscores the utility of semi-automation in enhancing laboratory efficiency. While this work does not quantify the full benefits of semi-automation, the author's experiences suggest significant potential impacts on the speed and adaptability of method development in biomedical research.

`plotastic` Exceled in Re-Doing Statistical Analyses and Plots

Establishing new methods of *in vitro* dissemination required not just innovative experimental protocols, but also adaptive ways to visually present complex data. This need for adaptability is crucial during the publication process, where researchers must often experiment with different ways to visually represent their findings to best convey their significance. This process typically involves frequent adjustments to how data is displayed in plots. Such adjustments become especially cumbersome when subsequent adjustments are involved. Traditional tools (*Microsoft Excel* or *Graphpad Prism*) fail at handling semi-big data, while Python packages like *seaborn* reach their limits in terms of adaptability, making the development of `plotastic` a critical step in this work.

`plotastic` addresses these challenges by not only automating statistics, but also by enhancing the adaptability of data visualization as well, making it easier to modify how data is presented without repetitive manual adjustments. The author saw four key steps that required streamlining through `plotastic`:

1. Re-arranging facets

2. Plotting multiple layers of different plot types
3. Statistical Re-Analysis and Re-Annotation
4. Fine-Tuning for publication grade quality

These adjustment steps made re-plotting tedious, since a change in prior steps required a complete re-work of following steps, something which `plotastic` prevented. Its key design feature is the centralized storage of facetting parameters (`x`, `hue`, `col`, `row`). These parameters define which data points are shown on the x-axis, what categories are highlighted by color (`hue`), and how data is grouped into separate plots (by columns and/or rows) into separate plots. This centralization means that once these parameters are set, they not only automate statistical analysis, but also can be automatically applied across all subsequent adjustments made to the plot. This contrasts with `seaborn`, where changing these parameters required adjusting multiple lines of subsequent code.

Re-arranging Facets: `plotastic`'s `.switch()` method allowed for easily shifting the arrangement of plots — for example, switching the data represented on the x-axis with that represented by color — to explore different perspectives of the data quickly. This proved particularly useful when trying to find the most effective way to illustrate complex interactions or trends that might not be immediately apparent. In `seaborn`, changing facets is easy and proved useful during intermediate data analysis, but unfeasable when plots involved multiple layers, sophisticated style edits or statistical annotations, as this can require re-writing subsequent adjustments.

Plotting Multiple Layers of Different Plot Types: Modern journal standards increasingly demand the representation of individual datapoints alongside aggregated data, for example plotting datapoints above a bar- or boxplots. `seaborn` does not automate this, but can require calling two plotting functions in sequence, e.g. `sns.boxplot()` followed by `sns.swarmplot()`. This can be can get repetitive, as adjusting the style of these plots to match each other, e.g. defining the point size or transparency of individual data points to fit into a barplot. `plotastic` was designed for multi-layered plotting, offering single-line functions for plot combinations with pre-configured style-parameters.

Statistical Re-Analysis and Re-Annotation To the author's knowledge, `plotastic`'s capability of streamlining statistical re-analysis is unique and unmatched. `seaborn` alone can not perform this without multiple lines of `statannotations` (Charlier et al., 2022). `plotastic` automates the inclusion of statistical annotations directly into plots. This is a significant enhancement because it ensures that any statistical significance noted in the data is immediately visible and correctly updated whenever the data presentation is changed. This feature proved particularly useful during the peer review process of Kuric et al. (2024), where a reviewer asked for a complete statistical analysis of Chapter 1 5 D, which at that time included only paired

t-tests between selected groups.

Fine-Tuning for Publication Grade Quality: `plotastic` simplified the creation of publication-quality figures by automating style adjustments that are typically manually coded with `matplotlib` when using `seaborn`. These include adjustments like angled x-axis labels or consistent visual styles across multiple figures, which are important for maintaining the professional appearance of published data.

Outlook: Could `plotastic` Address a Re-Analysis Bottleneck? Re-doing analyses and plots is often overlooked bottlenecks in the reproducibility of scientific research, although it does overlap with two principles of the FAIR-guidelines for scientific data management and stewardship: Interoperability[5] and Re-Usability (Wilkinson et al., 2016). This challenge was exemplified during this work's experiments using RT-qPCR. The field of qPCR, where reproducibility issues have been notoriously prevalent. As Bustin (2014) noted, many publications using PCR-based methods have been seriously flawed, underscoring the need for updated approaches (Bustin et al., 2013; Ruiz-Villalba et al., 2021). Furthermore, the evolution of the $\Delta\Delta$Ct formula over recent years highlights the dynamic nature of data analysis standards in biomedicine (Pfaffl, 2001; Ramakers et al., 2003; Ruijter et al., 2021). Despite these challenges, current data analysis infrastructures seldom facilitate a smooth revision or complete redoing of figures, which could hamper efforts to re-analyse and apply the latest techniques to existing datasets, which could be requested e.g. during peer-review (Wilkinson et al., 2016). In response, `plotastic` was specifically designed to streamline the reconfiguration and reanalysis of data visualizations. This work serves as a case study showing that — according to the author's experiences — the manual effort involved was effectively reduced, making the task of re-analysis seem a lot more inviting, especially for handling semi-big data.

Conclusion 2: Demonstrating the Advantages of Semi-Automation in Biomedical Research Methodologies

This thesis illustrates the challenges and solutions associated with managing the inherent complexity of adhesion studies and related methodologies, such as Cell Cycle profiling. These methodologies necessitate sophisticated data handling tools to address two primary challenges: (1) the multidimensionality of semi-big data and (2) the rapid iterative loop of results evaluation and protocol adjustments, a process for which *in vitro* methods are valued.

`seaborn` and `plotastic` have been instrumental in addressing these challenges. `seaborn` facilitated the rapid processing of intermediate results during method development, while `plotastic`

[5] Wilkinson et al. (2016): *"Interoperability — the ability of data or tools from non-cooperating resources to integrate or work together with minimal effort."*

was crucial for crafting publication-grade analyses and figures, filling in the capabilities that `seaborn` lacks. This includes facilitating the easy (re-)design of visualizations and statistical analyses, which are critical for late-stage data processing.

Though this work does not provide empirical evidence quantifying the benefits of semi-automation, it serves as a practical case study demonstrating the transformative potential of such technologies in biomedical research. The integration of semi-automation tools streamlined complex *in vitro* methodologies, significantly enhancing operational agility. This case study bridges biomedical research with bioinformatics, highlighting how semi-automation can reduce data analysis workloads and enable researchers to focus more on exploratory research within the laboratory setting.

To the author's experience, the gained efficiencies not only saved valuable time but also enhanced the clarity and communicative power of the research findings. This is particularly crucial in fields like myeloma dissemination, where precise and transparent data presentation is essential for advancing understanding and treatment strategies. This conclusion suggests a need for further empirical research to validate these benefits more broadly and encourage wider adoption of semi-automation tools in biomedical research.

However, adopting `plotastic` poses its own set of challenges, particularly in the realm of biomedicine where researchers may prefer graphical user interfaces (GUIs) over command-line interfaces (CLIs). While `plotastic` offers a powerful CLI that is efficient and capable of handling complex data manipulation and visualization tasks, the transition from GUIs to CLIs can be intimidating for those accustomed to more visual interaction with software. This barrier can be mitigated by the integration of tools like ChatGPT, which can facilitate the use of CLIs by offering context understanding, code assistance, and error identification.

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.*

Arefin, S. E., Heya, T. A., Al-Qudah, H., Ineza, Y., & Serwadda, A. (2023, July). *Unmasking the giant: A comprehensive evaluation of ChatGPT's proficiency in coding algorithms and data structures* (No. arXiv:2307.05360). arXiv. Retrieved 2024-05-03, from http://arxiv.org/abs/2307.05360 doi: 10.48550/arXiv.2307.05360

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

Bolado-Carrancio, A., Rukhlenko, O. S., Nikonova, E., Tsyganov, M. A., Wheeler, A., Garcia-Munoz, A., … Kholodenko, B. N. (2020, July). Periodic propagating waves coordinate RhoGTPase network dynamics at the leading and trailing edges during cell migration. *eLife*, *9*, e58165. Retrieved 2024-04-25, from https://elifesciences.org/articles/58165 doi: 10.7554/eLife.58165

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

Brankatschk, R., Bodenhausen, N., Zeyer, J., & Bürgmann, H. (2012, June). Simple Absolute Quantification Method Correcting for Quantitative PCR Efficiency Variations for Microbial Community Samples. *Applied and Environmental Microbiology*, *78*(12), 4481–4489. Retrieved 2023-05-27, from https://journals.asm.org/doi/10.1128/AEM.07878-11 doi: 10.1128/AEM.07878-11

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

Bubendorf, L. (2001, August). High-throughput microarray technologies: From genomics to clinics. *European Urology*, *40*(2), 231–238. doi: 10.1159/000049777

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

Bustin, S. A., Benes, V., Garson, J., Hellemans, J., Huggett, J., Kubista, M., … Vandesompele, J. (2013, November). The need for transparency and good practices in the qPCR literature. *Nature Methods*, *10*(11), 1063–1067. Retrieved 2024-05-16, from https://www.nature.com/articles/nmeth.2697 doi: 10.1038/nmeth

.2697

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). *Trevis-md/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

Codecov. (2024). Retrieved 2024-05-02, from https://github.com/codecov

Committee on Strategies for Responsible Sharing of Clinical Trial Data, Board on Health Sciences Policy, & Institute of Medicine. (2015). *Sharing Clinical Trial Data: Maximizing Benefits, Minimizing Risk*. Washington (DC): National Academies Press (US). Retrieved 2024-04-23, from http://www.ncbi.nlm.nih.gov/books/NBK269030/

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

Davidson-Pilon, C. (2019, August). Lifelines: Survival analysis in Python. *Journal of Open Source Software*, *4*(40), 1317. Retrieved 2024-05-02, from https://joss.theoj.org/papers/10.21105/joss.01317 doi: 10.21105/joss.01317

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

Federer, L. M., Lu, Y.-L., & Joubert, D. J. (2016, January). Data literacy training needs of biomedical researchers. *Journal of the Medical Library Association : JMLA*, *104*(1), 52–57. Retrieved 2024-04-24, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4722643/ doi: 10.3163/1536-5050.104.1.008

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

Flier, J. S. (2022). The Problem of Irreproducible Bioscience Research. *Perspectives in Biology and Medicine*, *65*(3), 373–395. doi: 10.1353/pbm.2022.0032

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

Gao, D., Ji, L., Bai, Z., Ouyang, M., Li, P., Mao, D., … Shou, M. Z. (2024, January). *ASSISTGUI: Task-Oriented Desktop Graphical User Interface Automation* (No. arXiv:2312.13108). arXiv. Retrieved 2024-05-16, from http://arxiv.org/abs/2312.13108 doi: 10.48550/arXiv.2312.13108

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

GraphPad Prism 10 User Guide. (2024). Retrieved 2024-05-14, from https://www.graphpad.com/guides/prism/latest/user-guide/multiple-variable-tables.htm

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.

Hannun, A., Digani, J., Katharopoulos, A., & Collobert, R. (2023). *MLX: Efficient and flexible machine learning on Apple silicon.* Retrieved from https://github.com/ml-explore

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

Kelleher, R. (2024, January). *NVIDIA CEO: 'This Year, Every Industry Will Become a Technology Industry'.* Retrieved 2024-05-03, from https://blogs.nvidia.com/blog/nvidia-ceo-ai-drug-discovery-jp-morgan-healthcare-2024/

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

Kim, D., Langmead, B., & Salzberg, S. L. (2015, April). HISAT: A fast spliced aligner with low memory requirements. *Nature methods*, *12*(4), 357–360. Retrieved 2024-04-26, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4655817/ doi: 10.1038/nmeth.3317

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. (2024, April). *Markur4/plotastic.* Retrieved 2024-05-02, from https://github.com/markur4/plotastic

Kuric, M., Beck, S., Schneider, D., Rindt, W., Evers, M., Meißner-Weigl, J., … Ebert, R. (2024, April). Modeling Myeloma Dissemination In Vitro with hMSC-interacting Subpopulations of INA-6 Cells and Their Aggregation/Detachment Dynamics. *Cancer Research Communications*, *4*(4), 1150–1164. Retrieved 2024-05-14, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11057410/ doi: 10.1158/2767-9764.CRC-23-0411

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

Lakhlifi, C., Lejeune, F.-X., Rouault, M., Khamassi, M., & Rohaut, B. (2023, April). Illusion of knowledge in statistics among clinicians: Evaluating the alignment between objective accuracy and subjective confidence, an online survey. *Cognitive Research: Principles and Implications*, *8*, 23. Retrieved 2024-04-24, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10118231/ doi: 10.1186/s41235-023-00474-1

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

Localio, A. R., Goodman, S. N., Meibohm, A., Cornell, J. E., Stack, C. B., Ross, E. A., & Mulrow, C. D. (2018, June). Statistical Code to Support the Scientific Story. *Annals of Internal Medicine*, *168*(11), 828–829. Retrieved 2024-04-23, from https://www.acpjournals.org/doi/10.7326/M17-3431 doi: 10.7326/M17-3431

Love, M. I., Huber, W., & Anders, S. (2014, December). Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biology*, *15*(12), 550. Retrieved 2024-04-26, from https://doi.org/10.1186/s13059-014-0550-8 doi: 10.1186/s13059-014-0550-8

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

Manifesto for Agile Software Development. (2001). Retrieved 2024-05-14, from http://agilemanifesto.org/

McCall, M. N., McMurray, H. R., Land, H., & Almudevar, A. (2014, August). On non-detects in qPCR data. *Bioinformatics*, *30*(16), 2310–2316. Retrieved 2023-04-25, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4133581/ doi: 10.1093/bioinformatics/btu239

McKay, B. S., Irving, P. E., Skumatz, C. M., & Burke, J. M. (1997, November). Cell-cell adhesion molecules and the development of an epithelial phenotype in cultured human retinal pigment epithelial cells. *Experimental Eye Research*, *65*(5), 661–671. doi: 10.1006/exer.1997.0374

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*.

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

Moleiro, A. F., Conceição, G., Leite-Moreira, A. F., & Rocha-Sousa, A. (2017). A Critical Analysis of the Available In Vitro and Ex Vivo Methods to Study Retinal Angiogenesis. *Journal of Ophthalmology*, *2017*, 3034953. doi: 10.1155/2017/3034953

Moran, M. (2003). Arguments for rejecting the sequential Bonferroni in ecological studies. *Oikos*, *100*(2), 403–405. Retrieved 2024-04-24, from https://onlinelibrary.wiley.com/doi/abs/10.1034/j.1600-0706.2003.12010.x doi: 10.1034/j.1600-0706.2003.12010.x

Motulsky, H. (2018). *Intuitive Biostatistics: A Nonmathematical Guide to Statistical Thinking*. Oxford University Press.

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.

Ordak, M. (2023, September). ChatGPT's Skills in Statistical Analysis Using the Example of Allergology: Do We Have Reason for Concern? *Healthcare (Basel, Switzerland)*, *11*(18), 2554. doi: 10.3390/healthcare11182554

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/

Pfaffl, M. W. (2001, May). A new mathematical model for relative quantification in real-time RT–PCR. *Nucleic Acids Research*, *29*(9), e45. Retrieved 2024-05-16, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC55695/

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

Purschke, M., Rubio, N., Held, K. D., & Redmond, R. W. (2010, November). Phototoxicity of Hoechst 33342 in time-lapse fluorescence microscopy. *Photochemical & Photobiological Sciences*, *9*(12), 1634–1639. Retrieved 2022-03-03, from https://pubs.rsc.org/en/content/articlelanding/2010/pp/c0pp00234h doi: 10.1039/C0PP00234H

PyMOL. (2024). Retrieved 2024-04-30, from https://pymol.org/

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

Quanbeck, A., Hennessy, R. G., & Park, L. (2022, November). Applying concepts from "rapid" and "agile" implementation to advance implementation research. *Implementation Science Communications*, *3*(1), 118. doi: 10.1186/s43058-022-00366-3

Qureshi, R., Shaughnessy, D., Gill, K. A. R., Robinson, K. A., Li, T., & Agai, E. (2023, April). Are ChatGPT and large language models "the answer" to bringing us closer to systematic review automation? *Systematic Reviews*, *12*, 72. Retrieved 2024-05-03, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10148473/

doi: 10.1186/s13643-023-02243-z

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

Read the Docs. (2024). Retrieved 2024-05-03, from https://docs.readthedocs.io/en/stable/index.html

Rebl, H., Finke, B., Schroeder, K., & Nebe, J. B. (2010, October). Time-dependent metabolic activity and adhesion of human osteoblast-like cells on sensor chips with a plasma polymer nanolayer. *The International Journal of Artificial Organs*, *33*(10), 738–748.

Rigsby, R. E., & Parker, A. B. (2016, September). Using the PyMOL application to reinforce visual understanding of protein structure. *Biochemistry and Molecular Biology Education: A Bimonthly Publication of the International Union of Biochemistry and Molecular Biology*, *44*(5), 433–437. doi: 10.1002/bmb.20966

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

Rueden, C. T., Schindelin, J., Hiner, M. C., DeZonia, B. E., Walter, A. E., Arena, E. T., & Eliceiri, K. W. (2017, November). ImageJ2: ImageJ for the next generation of scientific image data. *BMC Bioinformatics*, *18*(1), 529. Retrieved 2024-04-25, from https://doi.org/10.1186/s12859-017-1934-z doi: 10.1186/s12859-017-1934-z

Ruijter, J. M., Barnewall, R. J., Marsh, I. B., Szentirmay, A. N., Quinn, J. C., van Houdt, R., … van den Hoff, M. J. B. (2021, June). Efficiency Correction Is Required for Accurate Quantitative PCR Analysis and Reporting. *Clinical Chemistry*, *67*(6), 829–842. Retrieved 2023-05-27, from https://doi.org/10.1093/clinchem/hvab052 doi: 10.1093/clinchem/hvab052

Ruiz-Villalba, A., Ruijter, J. M., & van den Hoff, M. J. B. (2021, May). Use and Misuse of Cq in qPCR Data Analysis and Reporting. *Life*, *11*(6), 496. Retrieved 2023-04-25, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8229287/ doi: 10.3390/life11060496

Ruksakulpiwat, S., Kumar, A., & Ajibade, A. (2023, May). Using ChatGPT in Medical Research: Current Status and Future Directions. *Journal of Multidisciplinary Healthcare*, *16*, 1513–1520. Retrieved 2024-05-03, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10239248/ doi: 10.2147/JMDH.S413470

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

Sherina, V. (2020). Multiple imputation and direct estimation for qPCR data with non-detects.

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

Sphinx. (2024). Retrieved 2024-05-03, from https://docs.readthedocs.io/en/stable/intro/getting-started-with-sphinx.html

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

Sullivan, G. M., & Feinn, R. S. (2021, August). Facts and Fictions About Handling Multiple Comparisons. *Journal of Graduate Medical Education*, *13*(4), 457–460. Retrieved 2024-03-10, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8370375/ doi: 10.4300/JGME-D-21-00599.1

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

Trapnell, C., Roberts, A., Goff, L., Pertea, G., Kim, D., Kelley, D. R., … Pachter, L. (2012, March). Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and Cufflinks. *Nature Protocols*, *7*(3), 562–578. doi: 10.1038/nprot.2012.016

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. *Osteoporo-*

sis 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

West, K. (2018, July). *Reinventing Research: Agile in the Academic Laboratory | Agile Alliance.* Retrieved 2024-05-14, from https://www.agilealliance.org/resources/experience-reports/reinventing-research-agile-in-the-academic-laboratory/

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

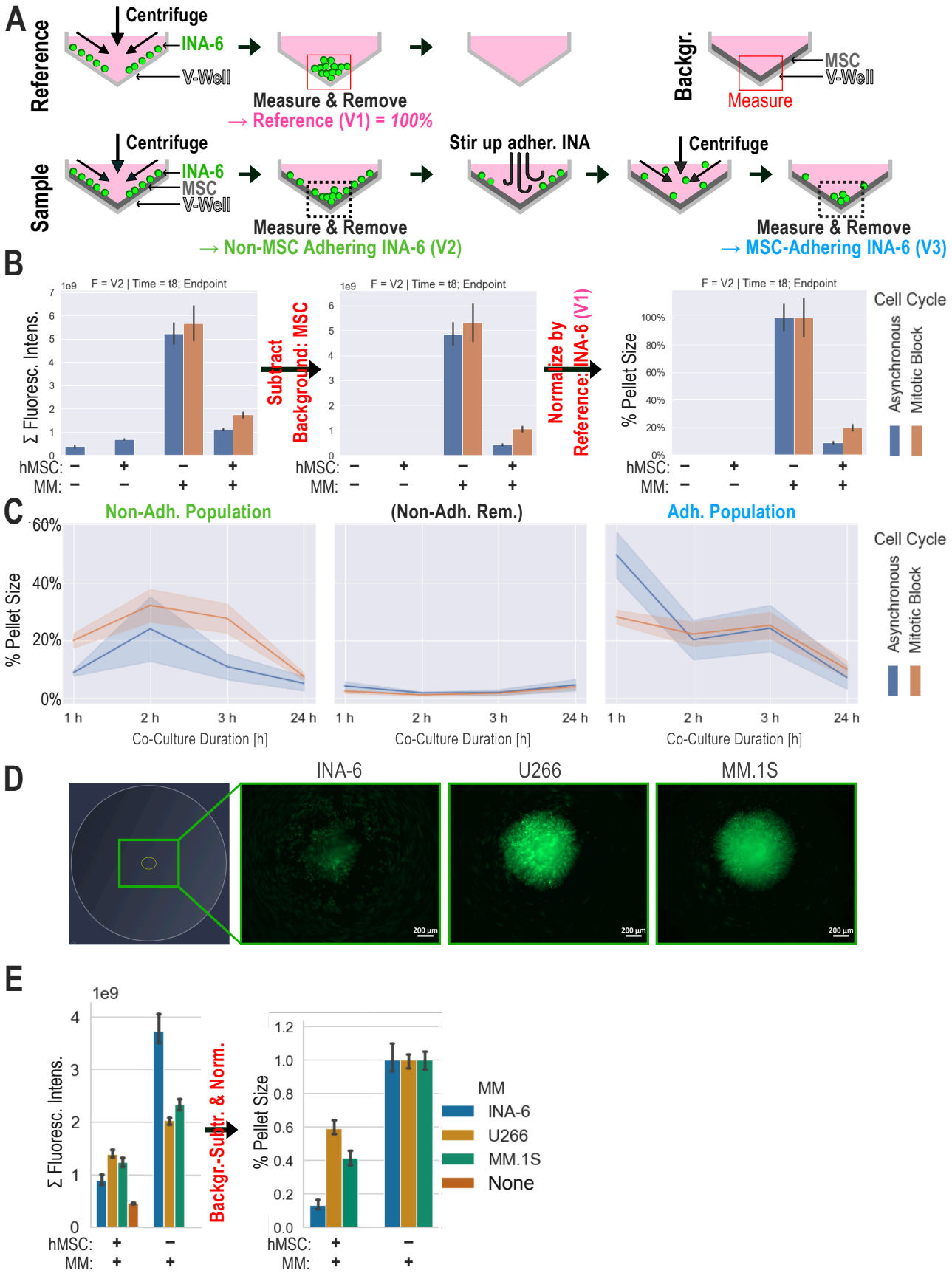
Zhou, Y., Zhou, B., Pache, L., Chang, M., Khodabakhshi, A. H., Tanaseichuk, O., … Chanda, S. K. (2019, April). Metascape provides a biologist-oriented resource for the analysis of systems-level datasets. *Nature Communications*, *10*(1), 1523. Retrieved 2023-02-09, from https://www.nature.com/articles/s41467-019-09234-6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

Ziemann, M., Eren, Y., & El-Osta, A. (2016, August). Gene name errors are widespread in the scientific literature. *Genome Biology*, *17*(1), 177. Retrieved 2024-04-30, from https://doi.org/10.1186/s13059-016-1044-7 doi: 10.1186/s13059-016-1044-7

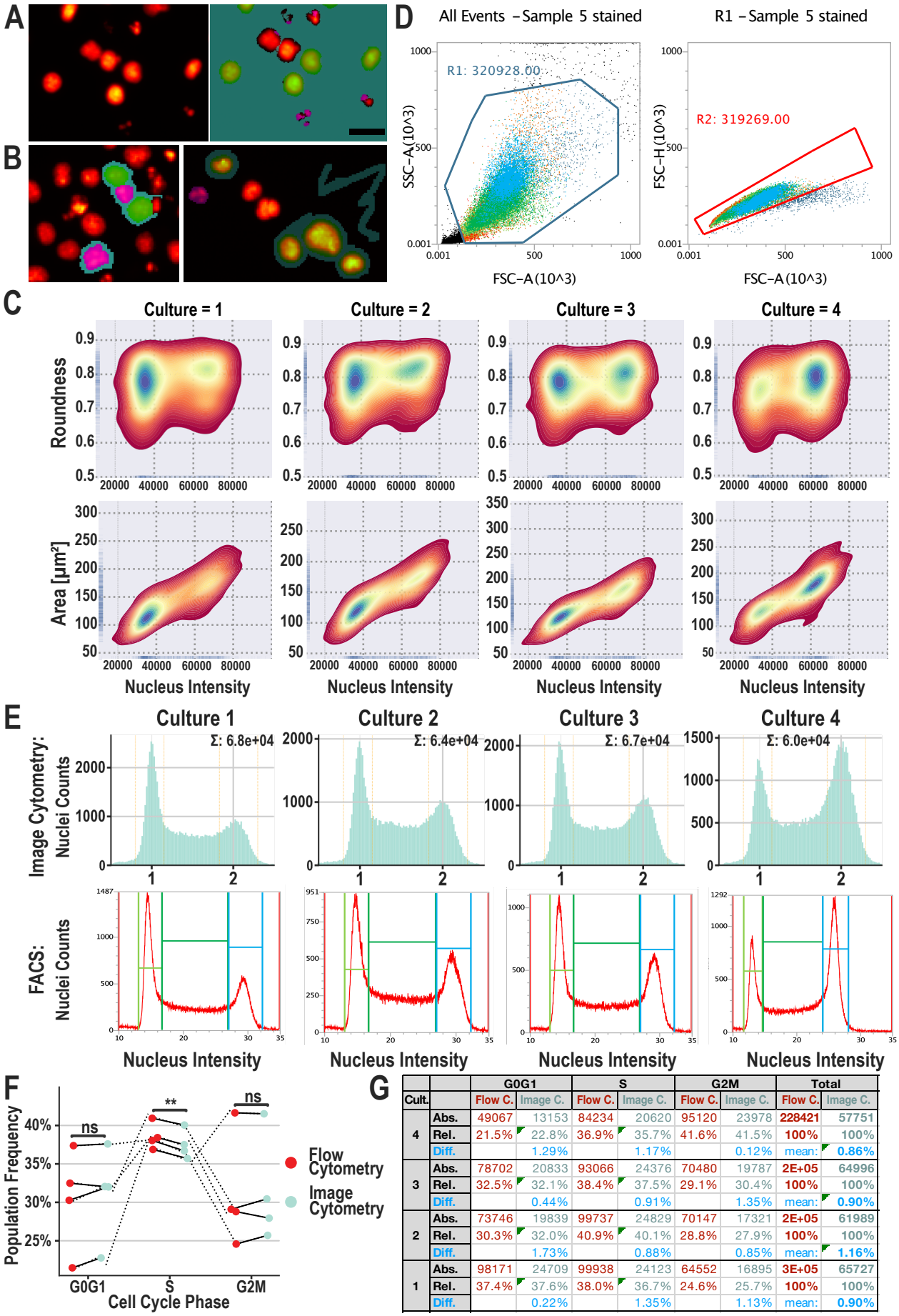
Appendices

A Supplementary Data & Methods

A.1 Figures


A
Reference
Centrifuge
INA-6
V-Well
Measure & Remove
→ Reference (V1) = 100%
Backgr.
MSC
V-Well
Measure
Sample
Centrifuge
INA-6
MSC
V-Well
Measure & Remove
→ Non-MSC Adhering INA-6 (V2)
Stir up adher. INA
Centrifuge
Measure & Remove
→ MSC-Adhering INA-6 (V3)
B
Σ Fluoresc. Intens.
F = V2 | Time = t8; Endpoint
Subtract Background: MSC
Normalize by Reference: INA-6 (V1)
% Pellet Size
Cell Cycle
Asynchronous
Mitotic Block
hMSC:
MM:
C
Non-Adh. Population
(Non-Adh. Rem.)
Adh. Population
% Pellet Size
Co-Culture Duration [h]
Cell Cycle
Asynchronous
Mitotic Block
D
INA-6
U266
MM.1S
200 µm
E
Σ Fluoresc. Intens.
Backgr.-Subtr. & Norm.
% Pellet Size
MM
INA-6
U266
MM.1S
None
hMSC:
MM:

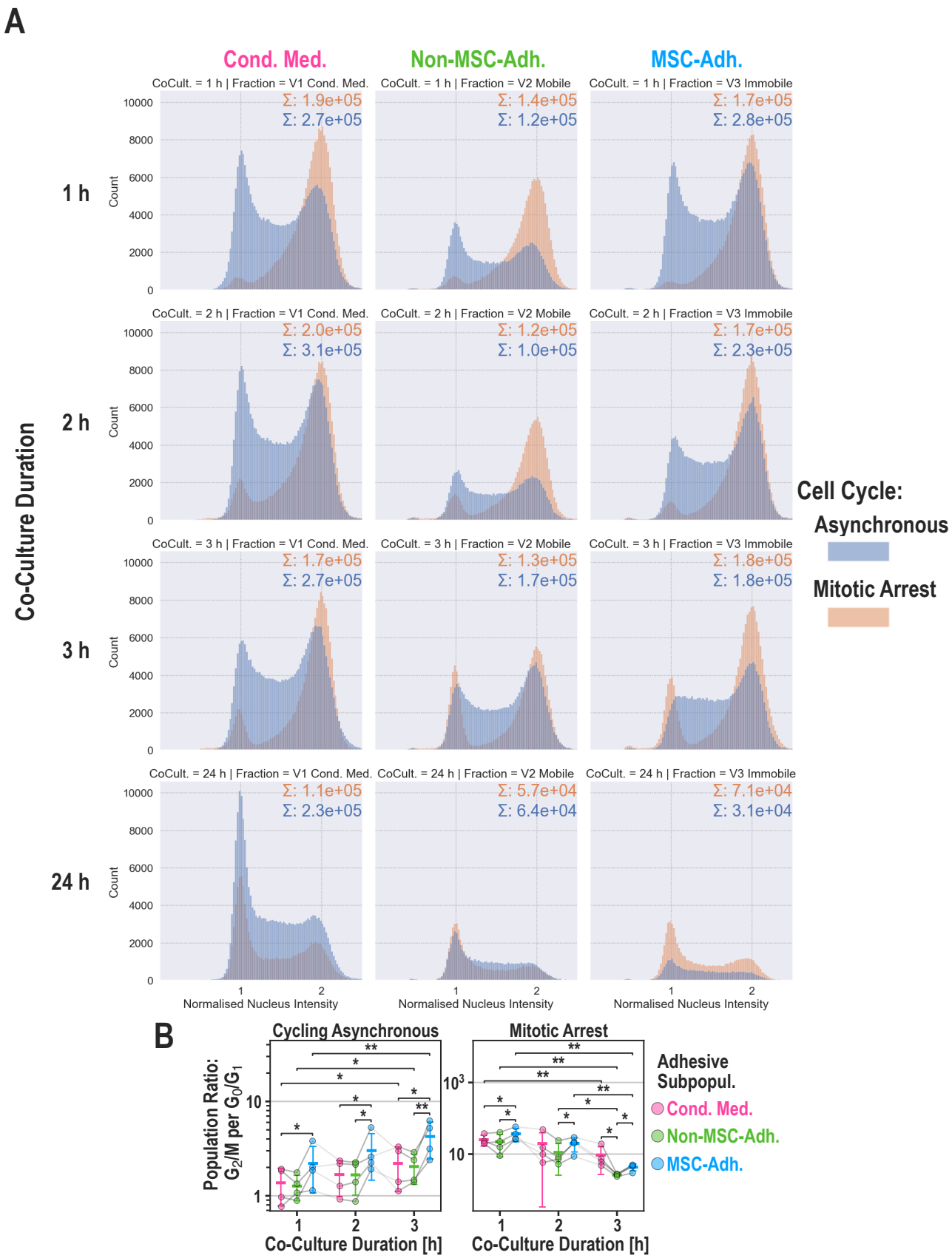
Appendix A Figure 1: Principle and quantification of the V-well adhesion assay of fluorescently labeled myeloma cells adapted by Weetall et al. (2001). **A:** Sample: Subsequent rounds of centrifugation and removal of cell pellet yielded the size of adhesive subpopulations. Fluorescently stained INA-6 cells were added to an hMSC monolayer. Non-adherent INA-6 cells (V2) were pelleted in the well-tip. Pellets were quantified by fluorescence brightness and isolated by pipetting. Immobile INA-6 cells (V3) were manually detached by forceful pipetting. Reference: Omitting adhesive hMSC-layer yielded 100 % non-adherent cells (V1) after the first centrifugation step; Background: hMSC monolayer was used as background signal. **B:** Calculation of the population size relative to total cells starting with pellet intensity. The shown example is the pellet gained by centrifuging mobile subpopulation (V2) after 1 h of co-culture. (see Fig. 3 for context): Intensity values from pellet images were summarized. After subtracting the unlabeled hMSC signal and normalization by a full-size pellet (reference), the resulting values represented the fraction of the adhesive subpopulation. **C:** One of three biological replicates summarized in Fig. 3. Line range shows the standard deviation of four technical replicates. Non.Adh. Rem.: Fluorescence signal after removal of V2. **D:** Example images of myeloma cell lines (INA-6, U266, MM.1S) pelleted in the tip of V-wells. The leftmost image shows the recorded area in a complete V-well. Scale bar = 200 µm. **E:** Results from (D) comparing adhesion strength of three myeloma cell lines to hMSC. Error bars represent technical deviation. MM: Multiple Myeloma.



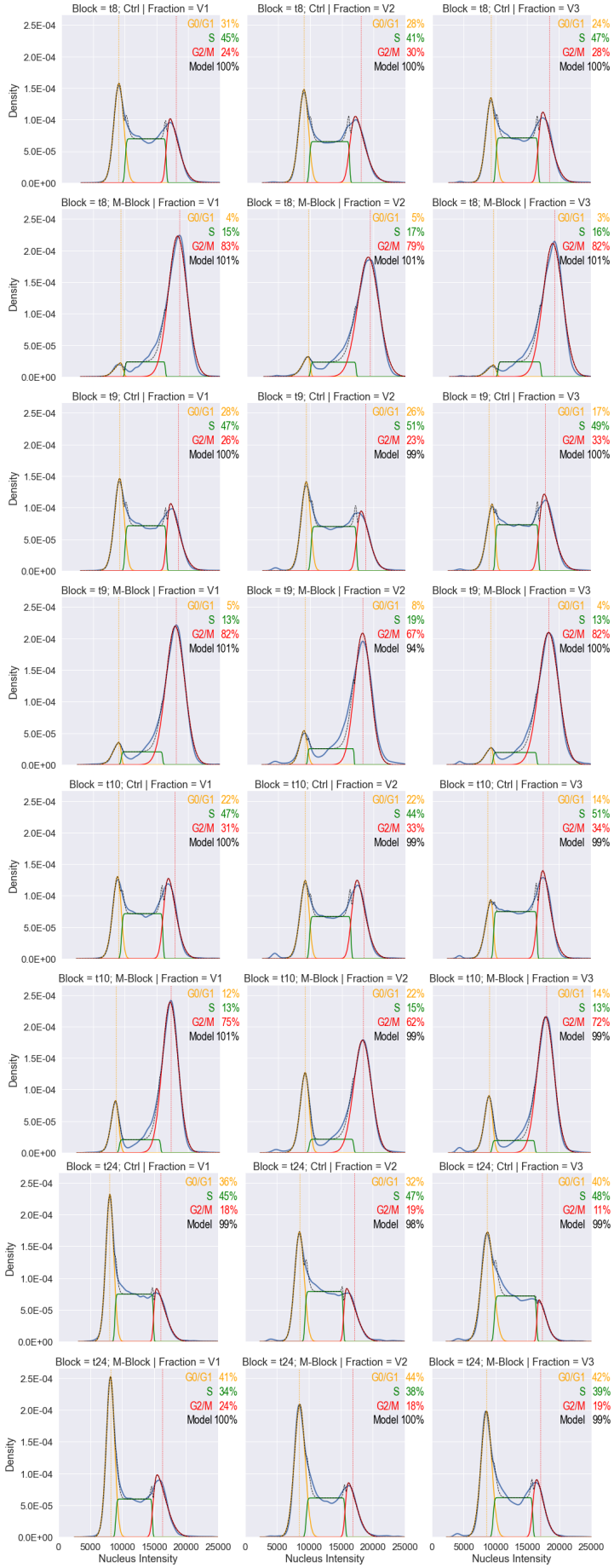
G

Cult.		G0G1		S		G2M		Total	
		Flow C.	Image C.	Flow C.	Image C.	Flow C.	Image C.	Flow C.	Image C.
4	Abs.	49067	13153	84234	20620	95120	23978	228421	57751
	Rel.	21.5%	22.8%	36.9%	35.7%	41.6%	41.5%	100%	100%
	Diff.		1.29%		1.17%		0.12%	mean:	0.86%
3	Abs.	78702	20833	93066	24376	70480	19787	2E+05	64996
	Rel.	32.5%	32.1%	38.4%	37.5%	29.1%	30.4%	100%	100%
	Diff.		0.44%		0.91%		1.35%	mean:	0.90%
2	Abs.	73746	19839	99737	24829	70147	17321	2E+05	61989
	Rel.	30.3%	32.0%	40.9%	40.1%	28.8%	27.9%	100%	100%
	Diff.		1.73%		0.88%		0.85%	mean:	1.16%
1	Abs.	98171	24709	99938	24123	64552	16895	3E+05	65727
	Rel.	37.4%	37.6%	38.0%	36.7%	24.6%	25.7%	100%	100%
	Diff.		0.22%		1.35%		1.13%	mean:	0.90%

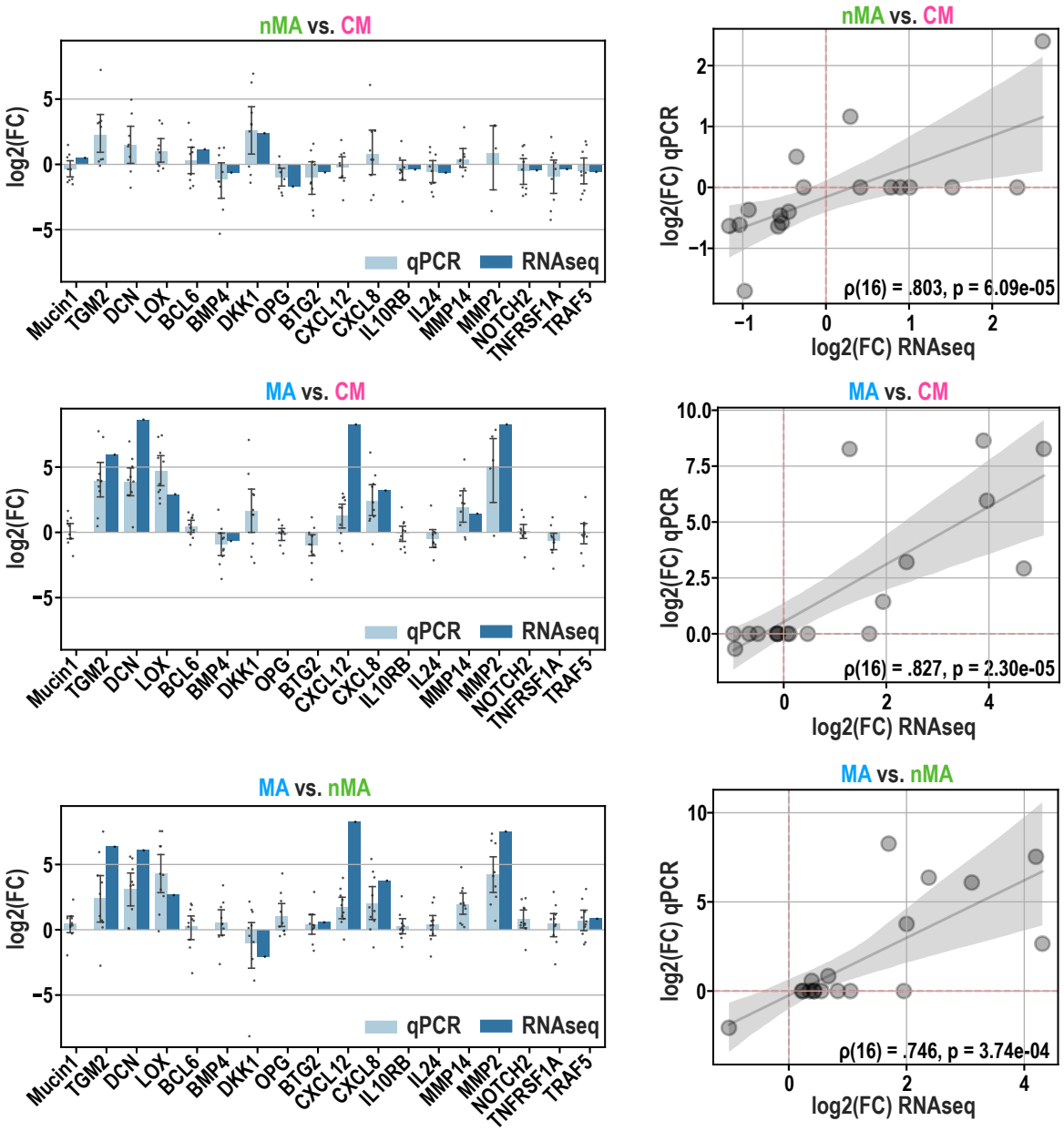
Appendix A Figure 2: Validation of image cytometric analysis of cell cycle in four INA-6 cultures. **A:** Left: Example image cytometric scan: INA-6 cells were stained with Hoechst33342 and scanned by automated fluorescence microscopy. Right: The image was segmented using a convolutional neural network (ZEISS ZEN intellesis) trained to discern healthy nuclei (green) from fragmented ones (magenta). Doublets are excluded by setting an area- and roundness threshold. Scale bar: 20 µm. **B:** Two example images from the training set. **C:** Quality of image cytometric data was ensured by plotting the distribution of nuclei brightnesses vs. the distribution of both nuclei-roundnesses and nuclei-areas. Nuclei with double fluorescence intensity have the same roundness while their area increases, as expected from a cell in G2 phase. **D:** The same samples from (C) were also measured with flow cytometry. Representative example of gating strategy: Left: Dead cells were excluded by setting a minimum threshold for side-scattering (SSC-A). Right: Doublets were excluded by setting a maximum threshold for forward scatter area (FSC-A) (sample "5" represents culture "4" in this figure). **E:** Cell cycle profiles of four independent INA-6 cultures were measured by both image cytometry (top) and flow cytometry (bottom). For both methods, frequencies of G0/G1, S, and G2M were summed up by setting fluorescence intensity thresholds. **F:** Image cytometry yields the same frequencies for G0/G1, S, and G2M when compared to flow cytometry. RM-ANOVA showed that the method has no significant effect on the frequencies of cell cycle populations $[F(1,3) = 1.421, p\text{-unc} = .32]$. **G:** Results from (F) in tabular form. On average, frequencies for G0/G1, S, and G2M measured by Image cytometry differ by 0.95 percent points compared to flow cytometry measurement. Cult.: Culture; C.: Image cytometry; Abs.: Absolute cell count; Rel.: Relative cell count; Diff.: Difference between relative cell counts determined by flow cytometry and image cytometry.


A
Cond. Med.
Non-MSC-Adh.
MSC-Adh.
Co-Culture Duration
1 h
2 h
3 h
24 h
Count
CoCult. = 1 h | Fraction = V1 Cond. Med.
Σ: 1.9e+05
Σ: 2.7e+05
CoCult. = 1 h | Fraction = V2 Mobile
Σ: 1.4e+05
Σ: 1.2e+05
CoCult. = 1 h | Fraction = V3 Immobile
Σ: 1.7e+05
Σ: 2.8e+05
CoCult. = 2 h | Fraction = V1 Cond. Med.
Σ: 2.0e+05
Σ: 3.1e+05
CoCult. = 2 h | Fraction = V2 Mobile
Σ: 1.2e+05
Σ: 1.0e+05
CoCult. = 2 h | Fraction = V3 Immobile
Σ: 1.7e+05
Σ: 2.3e+05
CoCult. = 3 h | Fraction = V1 Cond. Med.
Σ: 1.7e+05
Σ: 2.7e+05
CoCult. = 3 h | Fraction = V2 Mobile
Σ: 1.3e+05
Σ: 1.7e+05
CoCult. = 3 h | Fraction = V3 Immobile
Σ: 1.8e+05
Σ: 1.8e+05
CoCult. = 24 h | Fraction = V1 Cond. Med.
Σ: 1.1e+05
Σ: 2.3e+05
CoCult. = 24 h | Fraction = V2 Mobile
Σ: 5.7e+04
Σ: 6.4e+04
CoCult. = 24 h | Fraction = V3 Immobile
Σ: 7.1e+04
Σ: 3.1e+04
Normalised Nucleus Intensity
Cell Cycle:
Asynchronous
Mitotic Arrest
B
Cycling Asynchronous
Mitotic Arrest
Population Ratio: G2/M per G0/G1
Co-Culture Duration [h]
Adhesive Subpopul.
Cond. Med.
Non-MSC-Adh.
MSC-Adh.

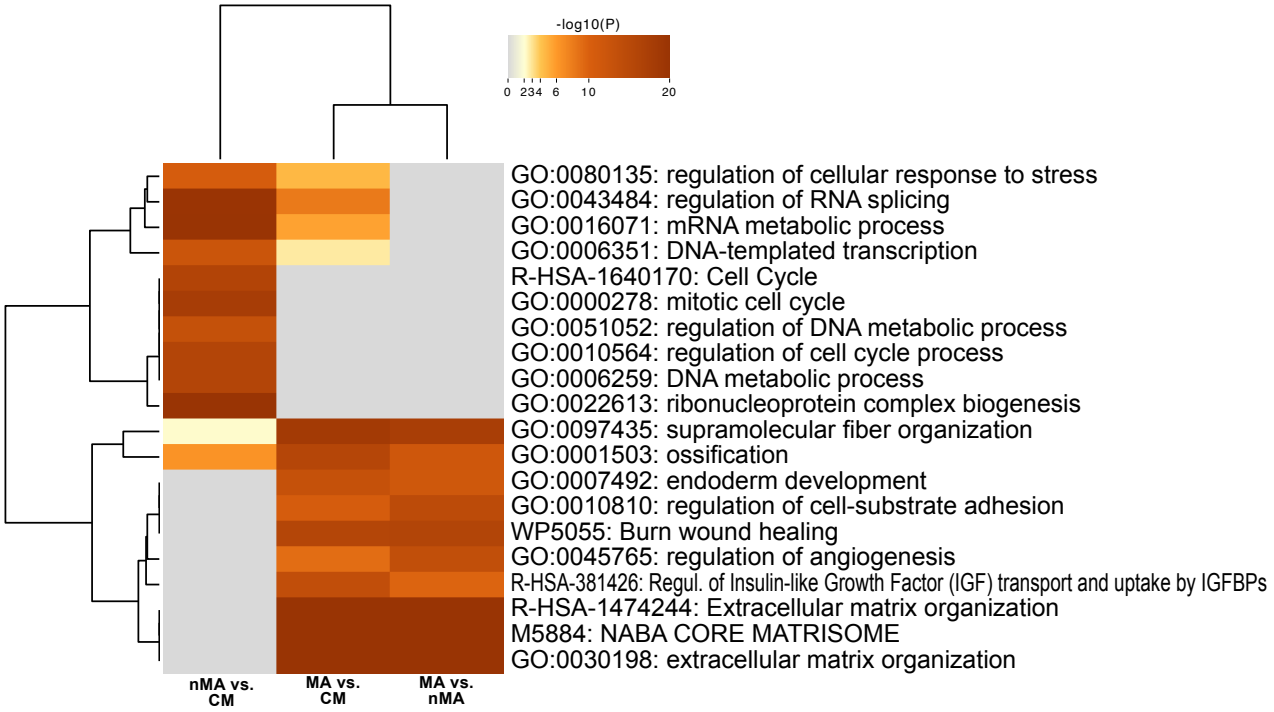
Appendix A Figure 3: Cell cycle analysis of INA-6 pellets gained from V-Well Adhesion assay (Fig. 3). **A:** Cell cycle profiles of MSC-adhering subpopulations. INA-6 cells were synchronized by double thymidine block followed by nocodazole. Cell cycle was released directly before addition to hMSCs. Histograms were normalized and summed up across all biological replicates ($n = 4$). Technical replicates (3) were pooled prior to cell cycle profiling. CoCult. = Co-culture duration. Fraction = Adhesion subpopulations. **B:** Similar figure to Fig. 3C displaying ratio of INA-6 populations (G2/M to G0/G1). **Statistics::** Paired t-test (B).

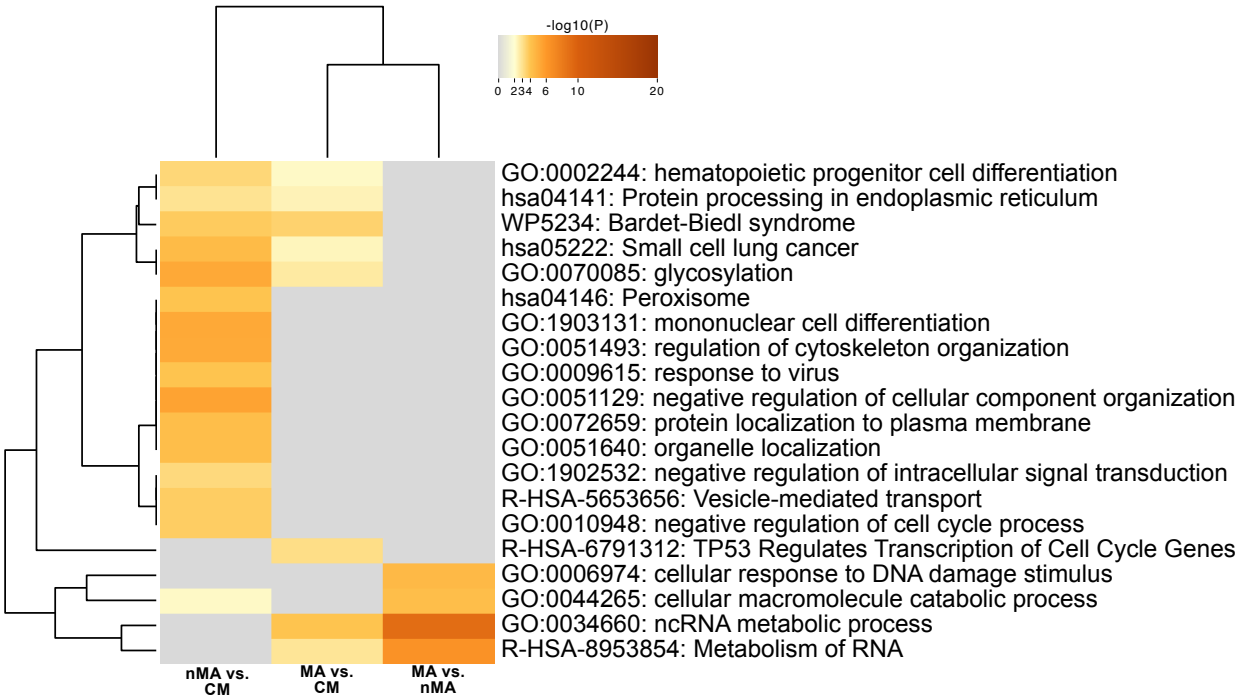

Block = t8; Ctrl | Fraction = V1
G0/G1 31%
S 45%
G2/M 24%
Model 100%
Block = t8; Ctrl | Fraction = V2
G0/G1 28%
S 41%
G2/M 30%
Model 100%
Block = t8; Ctrl | Fraction = V3
G0/G1 24%
S 47%
G2/M 28%
Model 100%
Block = t8; M-Block | Fraction = V1
G0/G1 4%
S 15%
G2/M 83%
Model 101%
Block = t8; M-Block | Fraction = V2
G0/G1 5%
S 17%
G2/M 79%
Model 101%
Block = t8; M-Block | Fraction = V3
G0/G1 3%
S 16%
G2/M 82%
Model 101%
Block = t9; Ctrl | Fraction = V1
G0/G1 28%
S 47%
G2/M 26%
Model 100%
Block = t9; Ctrl | Fraction = V2
G0/G1 26%
S 51%
G2/M 23%
Model 99%
Block = t9; Ctrl | Fraction = V3
G0/G1 17%
S 49%
G2/M 33%
Model 100%
Block = t9; M-Block | Fraction = V1
G0/G1 5%
S 13%
G2/M 82%
Model 101%
Block = t9; M-Block | Fraction = V2
G0/G1 8%
S 19%
G2/M 67%
Model 94%
Block = t9; M-Block | Fraction = V3
G0/G1 4%
S 13%
G2/M 82%
Model 100%
Block = t10; Ctrl | Fraction = V1
G0/G1 22%
S 47%
G2/M 31%
Model 100%
Block = t10; Ctrl | Fraction = V2
G0/G1 22%
S 44%
G2/M 33%
Model 99%
Block = t10; Ctrl | Fraction = V3
G0/G1 14%
S 51%
G2/M 34%
Model 99%
Block = t10; M-Block | Fraction = V1
G0/G1 12%
S 13%
G2/M 75%
Model 101%
Block = t10; M-Block | Fraction = V2
G0/G1 22%
S 15%
G2/M 62%
Model 99%
Block = t10; M-Block | Fraction = V3
G0/G1 14%
S 13%
G2/M 72%
Model 99%
Block = t24; Ctrl | Fraction = V1
G0/G1 36%
S 45%
G2/M 18%
Model 99%
Block = t24; Ctrl | Fraction = V2
G0/G1 32%
S 47%
G2/M 19%
Model 98%
Block = t24; Ctrl | Fraction = V3
G0/G1 40%
S 48%
G2/M 11%
Model 99%
Block = t24; M-Block | Fraction = V1
G0/G1 41%
S 34%
G2/M 24%
Model 100%
Block = t24; M-Block | Fraction = V2
G0/G1 44%
S 38%
G2/M 18%
Model 100%
Block = t24; M-Block | Fraction = V3
G0/G1 42%
S 39%
G2/M 19%
Model 99%
Density
0.0E+00
5.0E-05
1.0E-04
1.5E-04
2.0E-04
2.5E-04
0
5000
10000
15000
20000
25000
Nucleus Intensity

Appendix A Figure 4: Representative (one of the four independent sample sets as seen in Appendix A: Fig. 3) curve fitting analysis of cell cycle profiles generated by Image Cytometry. t8, t9, t10, and t24 refer to 1, 2, 3, and 24 hours after the addition of INA-6 cells to hMSCs.

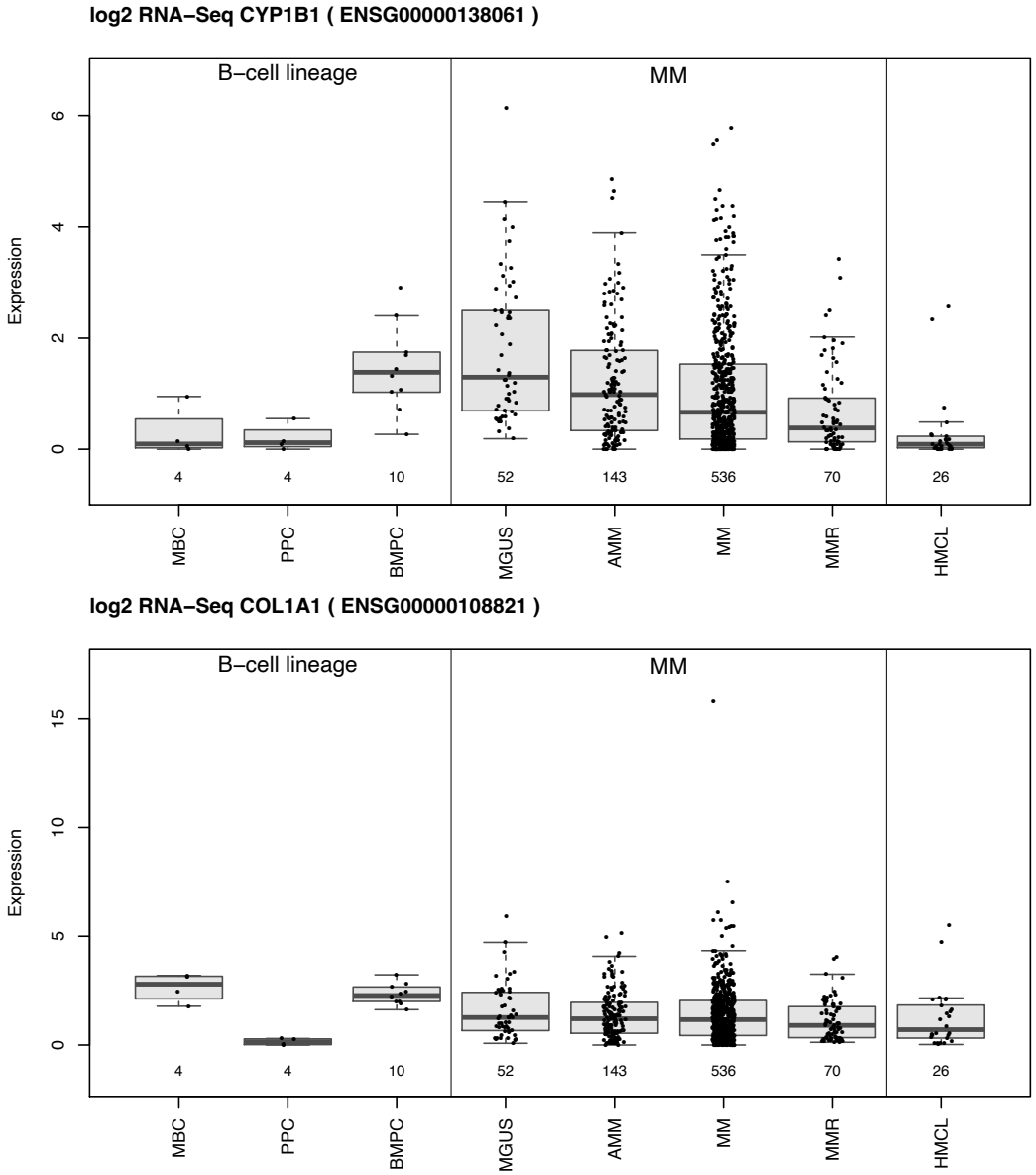


Appendix A Figure 5: Correlation of RNAseq with qPCR Left: Validation of RNAseq results (Fig. 4) with qPCR showing the $\log_2$(foldchange expression) of 18 genes. For qPCR, Datapoints each represent one biological replicate ($n = 10$), which is the mean of technical replicates ($n = 3$). Bar height represents mean of biological replicates, error bars show standard deviation of biological replicates. Right: Correlation between qPCR and RNAseq in terms of $\log_2$(mean foldchange expression per gene). Each dot represents one gene shown in the barplot to the left. Genes measured with qPCR that showed no differential expression in RNAseq were set to have a $-\log_2(FC) = 0$. Shaded area shows the confidence interval of linear regression. Correlation coefficient was calculated using Spearman's rank. $N = 18$ genes. FC: Fold change expression.

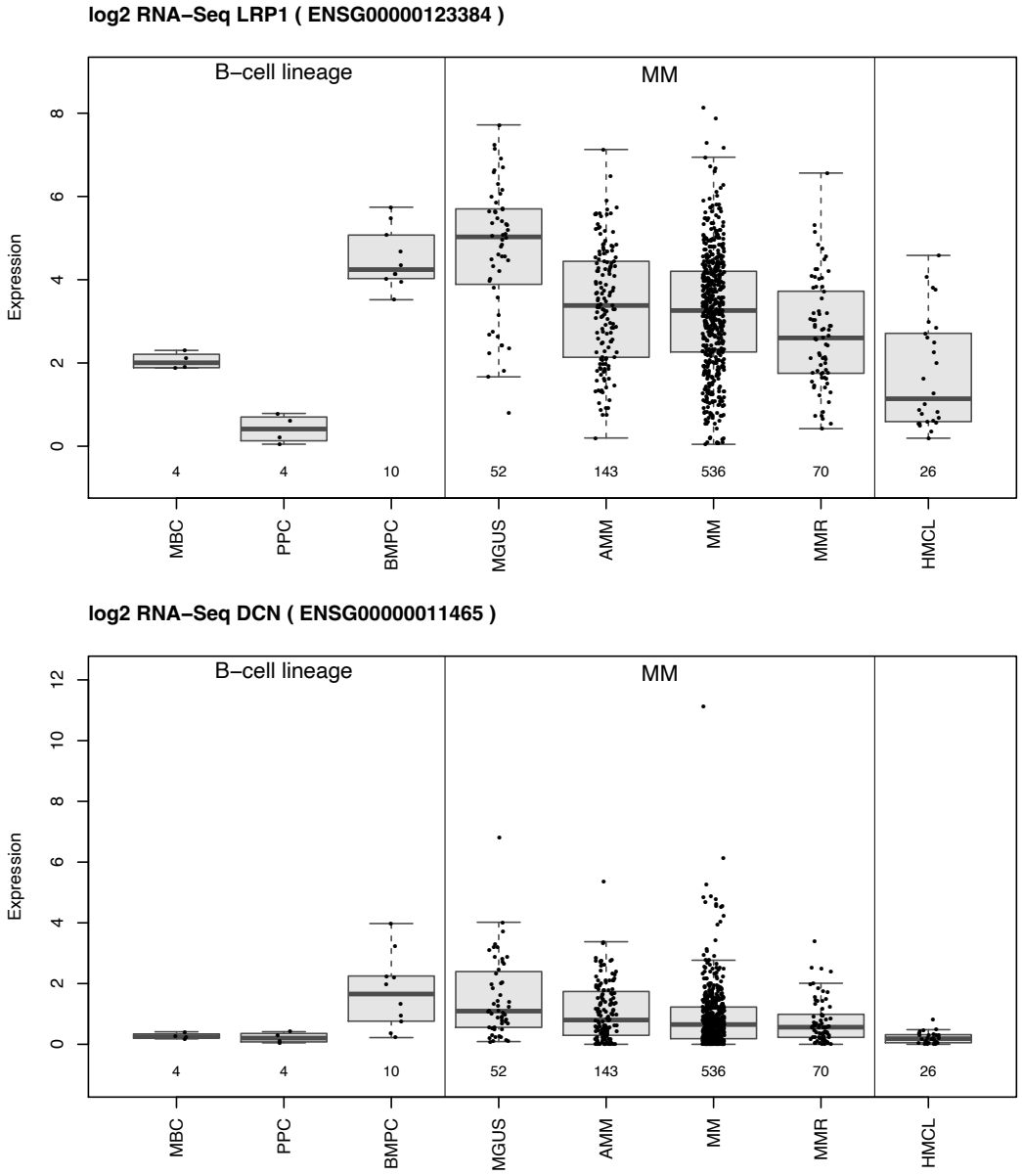




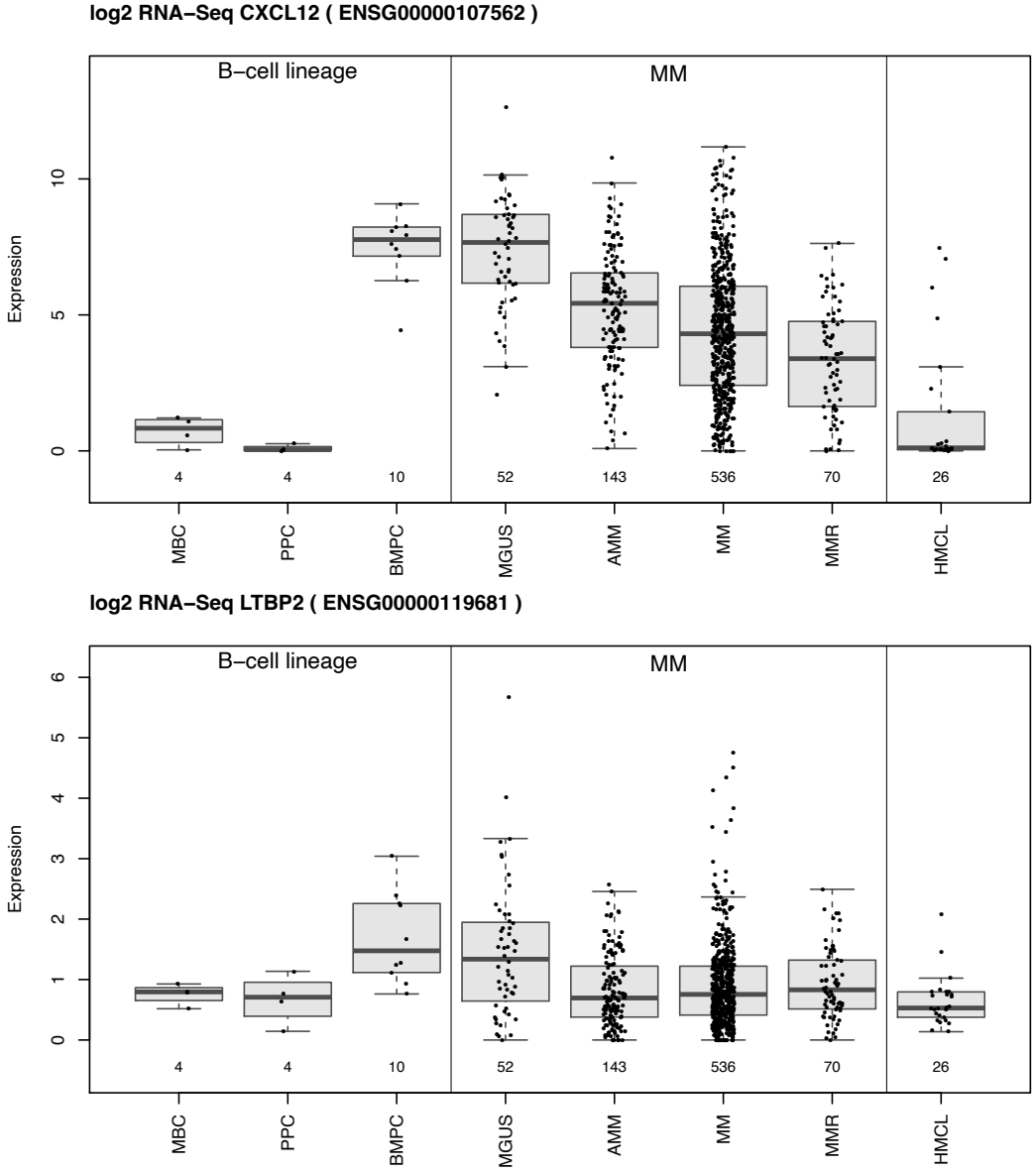
Appendix A Figure 6: Functional enrichment analysis by Metascape using genes that are differentially expressed between MSC-interacting subpopulations. Top: Upregulated genes. Bottom: Downregulated genes.



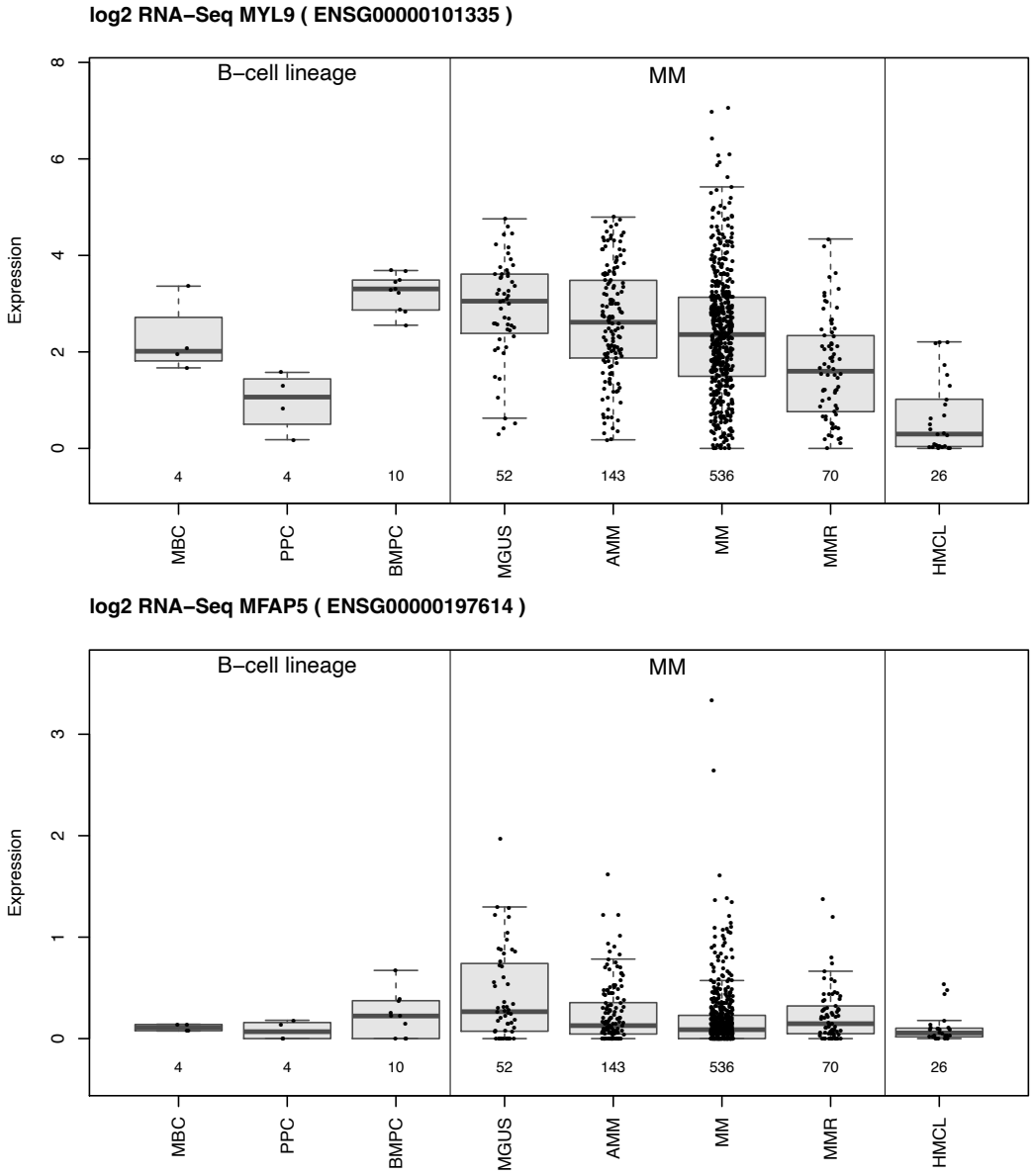
Appendix A Figure 7: Expression levels of adhesion genes that are downregulated and associated with survival ($p < 0.01$). Bone Marrow Plasma Cell (BMPC), Monoclonal Gammopathy of Undetermined Significance (MGUS), Smoldering Multiple Myeloma (sMM), Multiple Myeloma (MM), Multiple Myeloma Relapse (MMR).



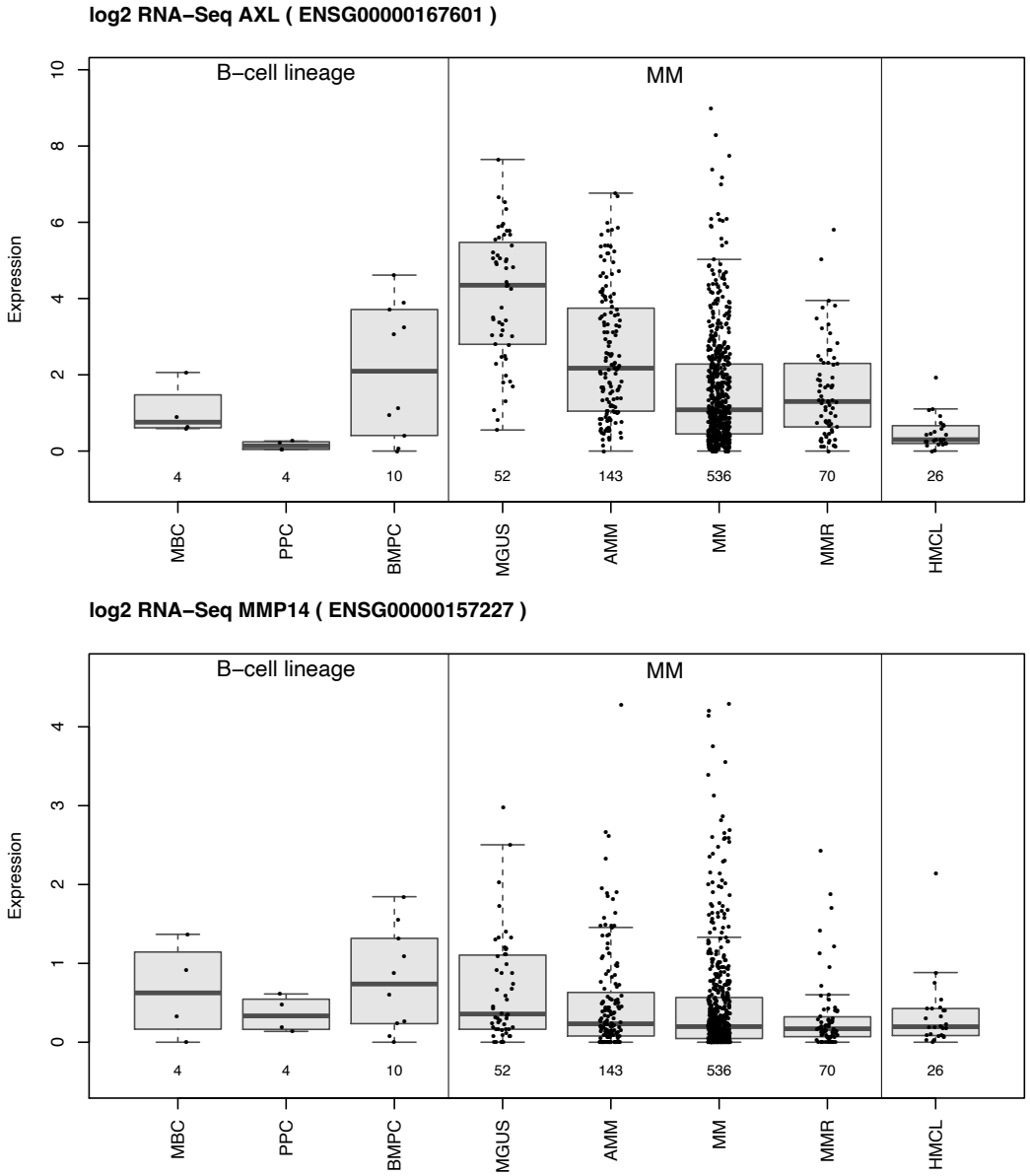
Appendix A Figure 7: *continued from previous page*



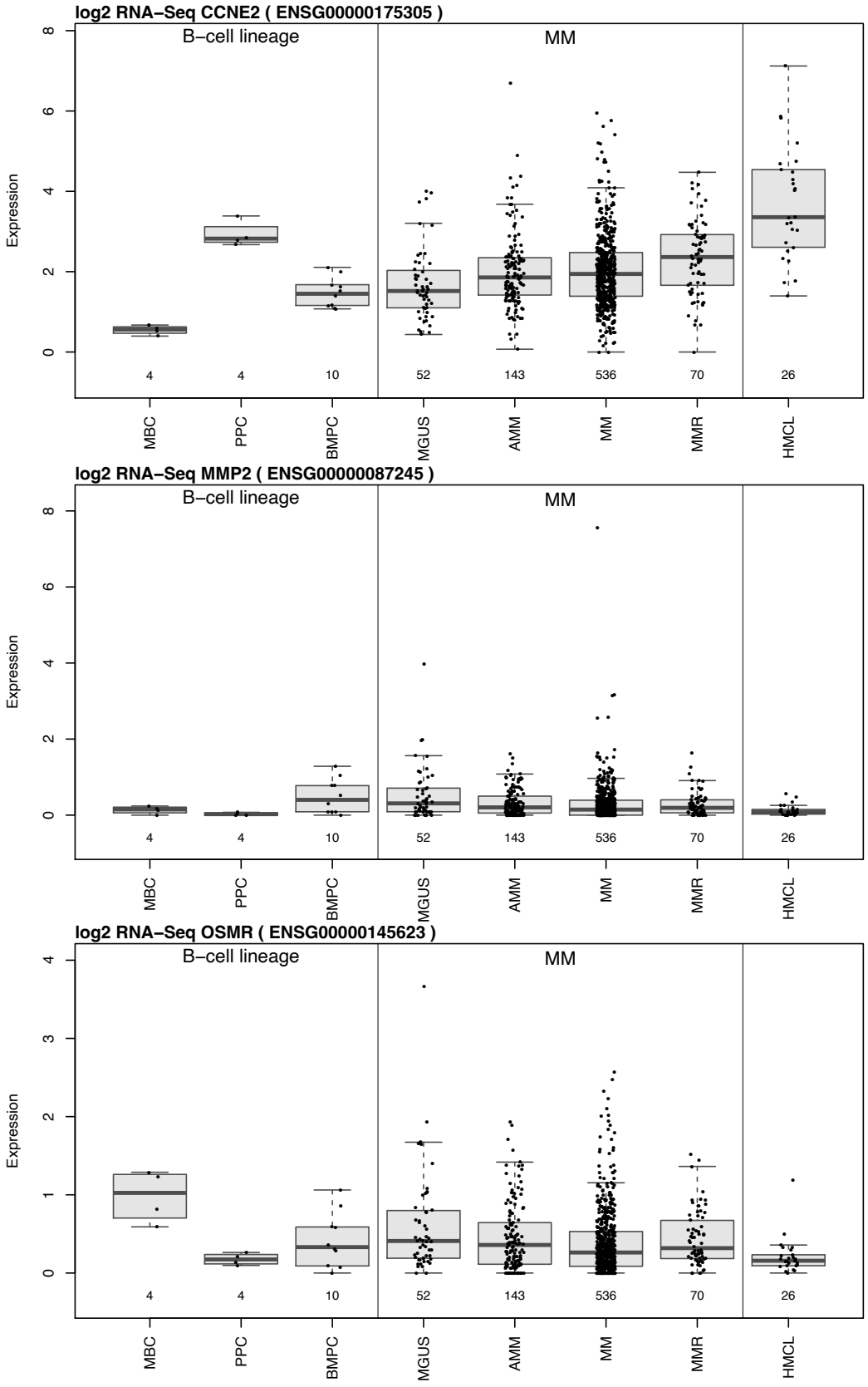
Appendix A Figure 7: *continued from previous page*



Appendix A Figure 7: *continued from previous page*



Appendix A Figure 7: *continued from previous page*


log2 RNA−Seq CCNE2 (ENSG00000175305)
B−cell lineage
MM
Expression
0
2
4
6
8
4
4
10
52
143
536
70
26
MBC
PPC
BMPC
MGUS
AMM
MM
MMR
HMCL
log2 RNA−Seq MMP2 (ENSG00000087245)
B−cell lineage
MM
Expression
0
2
4
6
8
4
4
10
52
143
536
70
26
MBC
PPC
BMPC
MGUS
AMM
MM
MMR
HMCL
log2 RNA−Seq OSMR (ENSG00000145623)
B−cell lineage
MM
Expression
0
1
2
3
4
4
4
10
52
143
536
70
26
MBC
PPC
BMPC
MGUS
AMM
MM
MMR
HMCL

Appendix A Figure 8: Expression levels of adhesion genes that are not downregulated and associated with survival ($p < 0.01$). Bone Marrow Plasma Cell (BMPC), Monoclonal Gammopathy of Undetermined Significance (MGUS), Smoldering Multiple Myeloma (sMM), Multiple Myeloma (MM), Multiple Myeloma Relapse (MMR).

A.2 Tables

Appendix A Table 1: List of hMSC donors, myeloma cell lines, and their mycoplasma test status. If no unique donors were available, hMSC donors were used twice for the same experiment at different passages. WPSC: Well plate sandwich centrifugation.

Cell Type	Donor / Line	Donor Ages	Donor Sex	Date of negative Mycoplasma test	Experiment(s)	Figures
Myeloma Cell Line	**INA-6**	80	m	09.02.22	All	All
	U266			10.10.22	- Validation of V-Well Adhesion Assay	S1E
	MM1.S			24.02.22		
hMSC	**1639**	49	m	not tested	- Validation of V-Well Adhesion Assay	S1E
					- Time-lapse: INA-6 on dispersed hMSC	1D; 2[A-E]
	1571	72	m	not tested	- Saturation of hMSCs	1[A-B]
	1573	47	m	not tested		
	1578	82	m	not tested		
	1842	63	m	not tested	- INA-6 Viability dependent on time and hMSC adhesion surface (INA not washed off)	1E right
	1843	60	m	not tested		
	1537	77	f	not tested		
	1794	82	m	not tested	- INA-6 Viability dependent on time and hMSC adhesion surface (INA washed off)	1[C, E left]
	1779	61	m	not tested		
	1849	69	m	not tested		
	1854	80	f	not tested		
	1605	71	f	not tested	- Time-lapse: INA-6 on dispersed hMSC	1D; 2[A-E]
	1650	57	m	not tested		
	1859	64	f	not tested	- Time-lapse: INA-6 on confluent hMSC	2[G-I]
	1863	79	f	not tested		
	1861	52	f	not tested		
	1818	81	f	not tested	- Cell Cycle Profiling after V-well assay	3C
	1824	82	f	not tested	(Donor measured twice, different passages) - V-well adhesion assay of mitotically blocked INA-6 followed by Cell Cycle Profiling after V-well assay	3[B,C]
	1827	56	m	not tested	- V-well adhesion assay of mitotically blocked INA-6 followed by Cell Cycle Profiling after V-well assay	3[B,C]

continued on next page

Appendix A Table 1 – *continued from previous page*

hMSC	**1501**	59	m	not tested	- INA-6 AI-assisted count during WPSC (INA-6 stained with celltracker green)	4B
	1643	75	f	not tested		
	1718	67	m	not tested		
	1720	58	m	not tested		
	1653	65	m	not tested		
	1591	78	m	not tested	- WPSC (MACS) followed by RNAseq, Metascape analysis, and qPCR validation	4[A,C,D,E]; 5[A-C]
					- WPSC (Wash) followed by qPCR-Validation and Luminescent Viability assays	4[C-E], 4F
	1654	74	m	not tested	- WPSC (MACS) followed by RNAseq, Metascape analysis, and qPCR validation	4[A,C,D,E]; 5[A-C]
					- WPSC (Wash) followed by qPCR-Validation and Luminescent Viability assays	4[C-E], 4F
	1655	78	f	not tested	- WPSC (MACS) followed by RNAseq, Metascape analysis, and qPCR validation	4[A,C,D,E]; 5[A-C]
	1668	80	f	not tested		
	1670	66	f	not tested		
	1701	81	m	not tested	- WPSC (Wash) followed by qPCR-Validation and Luminescent Viability assays	4[C-E], 4F
	1702	79	f	not tested		
	1600	77	m	not tested		
	1681	56	m	not tested	- WPSC (Wash) followed by Luminescent Viability assays	4F
	1672	65	m	not tested	- WPSC (Wash) followed by qPCR-Validation	4[C-E]

Appendix A Table 2: Adhesion genes (from Fig. 6A) categorized by a continuous downregulation across disease progression. Bone Marrow Plasma Cell (BMPC), Monoclonal Gammopathy of Undetermined Significance (MGUS), Smoldering Multiple Myeloma (sMM), Multiple Myeloma (MM), Multiple Myeloma Relapse (MMR). p-adj. = adjusted p-values (Benj.-Hoch.).

Regulation during disease progression	Gene	Ensemble ID	Progression Free / Overall Survival	Better Prognosis with high/low expression	Association of expression with survival	
					[p-unc]	[p-adj]
False	**ADAMTS1**	ENSG00000154734	Prog. Free	low	0.031875	0.084719
			Overall	low	0.048755	0.120104
	ADAMTS2	ENSG00000087116	Prog. Free	high	0.63795	0.767059
			Overall	high	0.811174	0.890528
	BGN	ENSG00000182492	Prog. Free	high	0.38065	0.533967
			Overall	high	0.279004	0.426961
	CAVIN1	ENSG00000177469	Prog. Free	high	0.407479	0.548739
			Overall	high	0.210903	0.3492
	CCDC80	ENSG00000091986	Prog. Free	high	0.002038	0.015833
			Overall	high	0.023743	0.077356
	CCN1	ENSG00000142871	Prog. Free	high	0.285568	0.443729
			Overall	low	0.931563	0.961309
	CCN2	ENSG00000118523	Prog. Free	high	0.030562	0.083425
			Overall	high	0.002889	0.024263
	CCNE2	ENSG00000175305	Prog. Free	low	0.012138	0.046195
			Overall	low	0.000534	0.008638
	CDH11	ENSG00000140937	Prog. Free	high	0.413948	0.550115
			Overall	high	0.044627	0.117163
	CEMIP	ENSG00000103888	Prog. Free	high	0.798984	0.877146
			Overall	low	0.287022	0.428378
	COL12A1	ENSG00000111799	Prog. Free	high	0.340978	0.491983
			Overall	low	0.829338	0.900679
	COL16A1	ENSG00000084636	Prog. Free	low	0.281112	0.443629
			Overall	low	0.162895	0.293792
	COL4A1	ENSG00000187498	Prog. Free	high	0.040286	0.098969
			Overall	high	0.009472	0.039863
	COL4A2	ENSG00000134871	Prog. Free	high	0.0124	0.046195
			Overall	high	0.175895	0.3063
	COL5A1	ENSG00000130635	Prog. Free	high	0.368403	0.524066
			Overall	low	0.860512	0.914414
	COL6A3	ENSG00000163359	Prog. Free	low	0.103315	0.208697
			Overall	low	0.197836	0.336625
	COL8A1	ENSG00000144810	Prog. Free	high	0.680745	0.807636
			Overall	high	0.289334	0.428378
	CREB3L1	ENSG00000157613	Prog. Free	low	0.165978	0.310441
			Overall	low	0.047989	0.120104
	EDIL3	ENSG00000164176	Prog. Free	high	0.863476	0.899083
			Overall	low	0.496663	0.611744
	F3	ENSG00000117525	Prog. Free	high	0.091858	0.197397
			Overall	high	0.009177	0.039863
	FBN1	ENSG00000166147	Prog. Free	high	0.472247	0.603376
			Overall	low	0.401546	0.533633
	FLNC	ENSG00000128591	Prog. Free	high	0.18539	0.329735
			Overall	low	0.474071	0.598515
	FN1	ENSG00000115414	Prog. Free	high	0.843432	0.896701
			Overall	low	0.421268	0.552573
	FBN1	ENSG00000166147	Prog. Free	high	0.472247	0.603376
			Overall	low	0.401546	0.533633
	FLNC	ENSG00000128591	Prog. Free	high	0.18539	0.329735
			Overall	low	0.474071	0.598515
	FN1	ENSG00000115414	Prog. Free	high	0.843432	0.896701

continued on next page

Appendix A Table 2 – *continued from previous page*

			Overall	low	0.421268	0.552573
	FOSB	ENSG00000125740	Prog. Free	low	0.585138	0.712035
			Overall	high	0.942273	0.961309
False	**GJA1**	ENSG00000152661	Prog. Free	high	0.333512	0.491983
			Overall	low	0.34262	0.467631
	GREM1	ENSG00000166923	Prog. Free	high	0.457976	0.59302
			Overall	low	0.591104	0.685623
	HBEGF	ENSG00000113070	Prog. Free	low	0.145103	0.281656
			Overall	low	0.051592	0.124067
	HTRA1	ENSG00000166033	Prog. Free	high	0.01203	0.046195
			Overall	high	0.040407	0.116603
	IGFBP3	ENSG00000146674	Prog. Free	high	0.248011	0.410641
			Overall	low	0.841566	0.904236
	IGFBP7	ENSG00000163453	Prog. Free	low	0.009533	0.043766
			Overall	low	0.024942	0.078722
	ITGA11	ENSG00000137809	Prog. Free	high	0.97438	0.97438
			Overall	low	0.966513	0.976178
	KLF11	ENSG00000172059	Prog. Free	low	0.229416	0.39273
			Overall	low	0.060892	0.133697
	LAMB1	ENSG00000091136	Prog. Free	high	0.477921	0.603376
			Overall	high	0.604163	0.685623
	LOX	ENSG00000113083	Prog. Free	low	0.748901	0.840433
			Overall	low	0.035028	0.104055
	MMP2	ENSG00000087245	Prog. Free	high	2.29E-05	0.002316
			Overall	high	0.044615	0.117163
	NFKBIZ	ENSG00000144802	Prog. Free	high	0.725256	0.832396
			Overall	high	0.310216	0.441293
	NR4A1	ENSG00000123358	Prog. Free	high	0.0214	0.065497
			Overall	high	0.060042	0.133697
	NR4A2	ENSG00000153234	Prog. Free	high	0.275313	0.441375
			Overall	high	0.11176	0.217072
	OSMR	ENSG00000145623	Prog. Free	high	0.000567	0.007153
			Overall	high	0.01287	0.046422
	PDGFRB	ENSG00000113721	Prog. Free	high	0.691005	0.807636
			Overall	high	0.599357	0.685623
	POSTN	ENSG00000133110	Prog. Free	low	0.858041	0.899083
			Overall	low	0.496348	0.611744
	PTX3	ENSG00000163661	Prog. Free	high	0.020943	0.065497
			Overall	high	0.045241	0.117163
	PXDN	ENSG00000130508	Prog. Free	low	0.403966	0.548739
			Overall	low	0.172495	0.305648
	SERPINE1	ENSG00000106366	Prog. Free	low	0.543711	0.669693
			Overall	high	0.869146	0.914414
	SERPINH1	ENSG00000149257	Prog. Free	low	0.001825	0.015833
			Overall	low	0.004399	0.026138
	SIX1	ENSG00000126778	Prog. Free	high	0.784446	0.870649
			Overall	high	0.592089	0.685623
	SMAD3	ENSG00000166949	Prog. Free	low	0.027411	0.0791
			Overall	low	0.016437	0.055338
	SPARC	ENSG00000113140	Prog. Free	high	0.073989	0.162455
			Overall	high	0.244069	0.391285
	SPOCK1	ENSG00000152377	Prog. Free	low	0.531524	0.662765
			Overall	low	0.303273	0.437579
	SULF1	ENSG00000137573	Prog. Free	high	0.190403	0.331564
			Overall	high	0.388706	0.523458
	THBS2	ENSG00000186340	Prog. Free	low	0.318676	0.480392
			Overall	high	0.292654	0.428378
	VPS37B	ENSG00000139722	Prog. Free	low	0.1478	0.281656
			Overall	low	0.199975	0.336625

continued on next page

Appendix A Table 2 – *continued from previous page*

True	**ACTN1**	ENSG00000072110	Prog. Free	high	0.170661	0.313396
			Overall	high	0.007728	0.035478
	ADAM12	ENSG00000148848	Prog. Free	high	0.019179	0.062487
			Overall	high	0.081847	0.168704
	AEBP1	ENSG00000106624	Prog. Free	high	0.010829	0.046195
			Overall	high	0.057228	0.133697
	AXL	ENSG00000167601	Prog. Free	high	0.001496	0.015105
			Overall	high	3.64E-05	0.00184
	CD99	ENSG00000002586	Prog. Free	low	0.916833	0.941333
			Overall	low	0.083964	0.169607
	COL1A1	ENSG00000108821	Prog. Free	high	0.000303	0.004367
			Overall	high	0.000593	0.008638
	COL1A2	ENSG00000164692	Prog. Free	high	0.298023	0.456066
			Overall	high	0.566636	0.673297
	COL3A1	ENSG00000168542	Prog. Free	high	0.025985	0.07719
			Overall	high	0.010794	0.042917
	COL5A2	ENSG00000204262	Prog. Free	low	0.74501	0.840433
			Overall	low	0.99967	0.99967
	COL6A1	ENSG00000142156	Prog. Free	high	0.011972	0.046195
			Overall	high	0.011048	0.042917
	COL6A2	ENSG00000142173	Prog. Free	high	0.261528	0.426037
			Overall	high	0.3235	0.447582
	CXCL12	ENSG00000107562	Prog. Free	high	0.000116	0.002927
			Overall	high	0.000648	0.008638
	CXCL8	ENSG00000169429	Prog. Free	low	0.839416	0.896701
			Overall	high	0.224913	0.366391
	CYP1B1	ENSG00000138061	Prog. Free	high	0.008641	0.041735
			Overall	high	0.000684	0.008638
	DCN	ENSG00000011465	Prog. Free	high	0.004827	0.030473
			Overall	high	0.000247	0.008327
	DUSP1	ENSG00000120129	Prog. Free	high	0.695686	0.807636
			Overall	high	0.454061	0.583718
	FBLN1	ENSG00000077942	Prog. Free	high	0.002676	0.019305
			Overall	high	0.003734	0.026138
	GNB3	ENSG00000111664	Prog. Free	high	0.003748	0.025234
			Overall	high	0.005734	0.03048
	GSTP1	ENSG00000084207	Prog. Free	high	0.972219	0.97438
			Overall	low	0.668091	0.749746
	IGFBP4	ENSG00000141753	Prog. Free	high	0.008677	0.041735
			Overall	high	0.007089	0.034093
	IL1R1	ENSG00000115594	Prog. Free	high	0.126318	0.250159
			Overall	high	0.256501	0.398563
	ITGA5	ENSG00000161638	Prog. Free	high	0.094893	0.19967
			Overall	high	0.159113	0.29219
	ITGAX	ENSG00000140678	Prog. Free	high	0.006717	0.036021
			Overall	high	0.003123	0.024263
	ITGB5	ENSG00000082781	Prog. Free	low	0.436018	0.57192
			Overall	high	0.539497	0.648681
	LAMA4	ENSG00000112769	Prog. Free	high	0.018518	0.062345
			Overall	high	0.104178	0.206314
	LAMB2	ENSG00000172037	Prog. Free	high	0.015472	0.053885
			Overall	high	0.001354	0.013865
	LOXL2	ENSG00000134013	Prog. Free	high	0.808671	0.878235
			Overall	low	0.933264	0.961309
	LRP1	ENSG00000123384	Prog. Free	high	0.006458	0.036021
			Overall	high	0.000434	0.008638
	LTBP2	ENSG00000119681	Prog. Free	high	9.03E-05	0.002927
			Overall	high	0.011656	0.043603
	LUM	ENSG00000139329	Prog. Free	high	0.05158	0.118399
			Overall	high	0.065084	0.139862

continued on next page

Appendix A Table 2 – *continued from previous page*

True	**MAP3K8**	ENSG00000107968	Prog. Free	high	0.000958	0.010755
			Overall	high	0.01617	0.055338
	MAP4K4	ENSG00000071054	Prog. Free	high	0.041155	0.098969
			Overall	high	0.31743	0.445284
	MFAP5	ENSG00000197614	Prog. Free	high	0.000243	0.004094
			Overall	high	0.004269	0.026138
	MMP14	ENSG00000157227	Prog. Free	high	6.93E-05	0.002927
			Overall	high	0.006691	0.033787
	MXRA5	ENSG00000101825	Prog. Free	high	0.034865	0.088035
			Overall	high	0.033819	0.103505
	MYL9	ENSG00000101335	Prog. Free	high	0.000146	0.00295
			Overall	high	1.56E-05	0.001572
	NRP1	ENSG00000099250	Prog. Free	high	0.001888	0.015833
			Overall	high	0.002212	0.020312
	PAPLN	ENSG00000100767	Prog. Free	high	0.034256	0.088035
			Overall	high	0.159113	0.29219
	TEX14	ENSG00000121101	Prog. Free	high	0.237488	0.399771
			Overall	low	0.518581	0.631044
	TGFBI	ENSG00000120708	Prog. Free	high	0.102621	0.208697
			Overall	high	0.004299	0.026138
	TGM2	ENSG00000198959	Prog. Free	high	0.058634	0.131601
			Overall	high	0.119621	0.227958
	THBS1	ENSG00000137801	Prog. Free	high	0.39286	0.543545
			Overall	high	0.456572	0.583718
	TNC	ENSG00000041982	Prog. Free	high	0.012806	0.046195
			Overall	high	0.004752	0.026663
	TNS1	ENSG00000079308	Prog. Free	high	0.338737	0.491983
			Overall	high	0.757617	0.840872
	TPM1	ENSG00000140416	Prog. Free	high	0.029263	0.0821
			Overall	high	0.001373	0.013865
	TUBA1A	ENSG00000167552	Prog. Free	low	0.006776	0.036021
			Overall	low	0.042929	0.117163
	TUBB6	ENSG00000176014	Prog. Free	low	0.186088	0.329735
			Overall	low	0.060071	0.133697
	VCAN	ENSG00000038427	Prog. Free	high	0.042782	0.100487
			Overall	high	0.080757	0.168704
	ZFP36L1	ENSG00000185650	Prog. Free	high	0.922693	0.941333
			Overall	high	0.24957	0.393852

Appendix A Table 3: List of primers. Some primers required a melting step to be performed before fluorescent readout to remove byproducts.

Primer	**Sequence 5' - 3'**	**base pairs [bp]**	**annealing temp. [°C]**
36B4_s	tgcatcagtaccccattctatcat	122	60
36B4_as	aggcagatggatcagccaaga		
BCL6_s	tagagcccataaaacggtcctcat	221	55 + Melting Step at 77 °C
BCL6_as	cgcaaattgagccgagatgtgt		
BMP4_s	tacatgcgggatctttaccg	132	58
BMP4_as	atgttcttcgtggtggaagc		
BTG2_s	gtattcttgtagggccgacactaa	264	60 + Melting Step at 78 °C
BTG2_as	tcttaaggtgattcggtttgggaa		
CXCL8_s	actgagagtgattgagagtggacc	251	55 + Melting Step at 77 °C
CXCL8_as	ccctacaacagacccacacaatac		
CXCL12_s	gattcttcgaaagccatgttgcca	119	56
CXCL12_as	caatgcacacttgtctgttgttgt		
DCN_s	caacaacaagcttaccagagtacct	160	57
DCN_as	tgaaaagactcacacccgaataaga		
DKK1_s	gcactgatgagtactgcgctag	129	56
DKK1_as	ttttgcagtaattcccggggc		
IL10RB_s	gagtgagcctgtctgtgagcaa	139	55
IL10RB_as	cttgtaaacgcaccacagcaag		
IL24_s	caaacagttggacgtagaagcagc	149	55
IL24_as	tgaaatgacacagggaacaaacca		
LOX_s	ctgctcagatttccccaaag	125	57
LOX_as	tggcatcaagcaggtcatag		
MMP2_s	ttgtatttgatggcatcgctcaga	155	56
MMP2_as	cgtataccgcatcaatcttttccg		
MMP14_s	cgacaagattgatgctgctc	140	57
MMP14_as	tcccttcccagactttgatg		
MUC1_s	gcagcctctcgatataacctg	200	58
MUC1_as	gtaggtggggtactcgctca		
NOTCH2_s	gtgcttgttgaacacttgtgcc	185	55
NOTCH2_as	cactcgcatctgtatccaccaatg		
OPG_s	no sequence available (Proprietary primers from Qiagen: QT00014294 TNFRSF11B_1_SG)		60
OPG_as			
PRICKLE1_s	cagaggtatatcatgaaggacggc	102	56
PRICKLE1_as	gtccacaccaatatgttccccac		
TGM2_s	caaccttctcatcgagtacttccg	100	58
TGM2_as	tcatccacgactccacccag		
TNFRSF1A_s	ctccttcaccgcttcagaaaacc	153	55
TNFRSF1A_as	ttcactccaataatgccggtactg		
TRAF5_s	tgccctgtagataaagaggtcatca	177	56
TRAF5_as	aacactgcacaggttgaaataagc		

A.3 Materials & Methods

Isolation and Culturing of Primary Human Bone Marrow-Derived Mesenchymal Stromal Cells

Primary human Mesenchymal Stromal Cells (MSCs) were obtained from the femoral head of patients (Appendix A: Tab. 1) undergoing elective hip arthroplasty. Material was collected with written informed consent of all patients and the procedure was approved by the local Ethics Committee of the University of Würzburg (186/18). In brief, bone marrow was washed with MSC-medium (Dulbecco's modified Eagle's medium (DMEM/F12, Thermo Fisher Scientific, Darmstadt, Germany) supplemented with 10 % Fetal Calf Serum (FCS, Bio&Sell GmbH, Feucht, Germany, Fernandez-Rebollo et al. (2017)), 100 U/ml penicillin, 0.1 mg/ml streptomycin (Thermo Fisher Scientific), 50 µg/ml ascorbate and 100 nmol/l sodium selenite (both Sigma-Aldrich GmbH, Munich, Germany)) and centrifuged at 250 g for 5 min. The pellet was washed four times with MSC-medium and resulting supernatants containing released cells were collected. Cells were pelleted and cultured at a density of 1×10^9 cells per 175 cm^2 culture flask. After two days non-attached cells were washed away and adherent ones were cultivated in MSC-medium until confluence. Then, they were either frozen in liquid nitrogen or directly utilized for experiments. hMSC cultures were sustained for a maximum of two passages. All cells were cultured at 37 °C and at 5 % CO_2.

Culturing of Myeloma Cell Lines

The plasmacytoma cell line INA-6 [*RRID:CVCL_5209*; DSMZ, Braunschweig, Germany, authenticated by DSMZ in 2014 (Burger, Guenther, et al., 2001; Gramatzki et al., 1994)] was cultivated in RPMI1640 medium (Thermo Fisher Scientific) supplemented with 20 % (v/v) FCS, 100 µg/ml gentamicin, 2 mmol/l L-glutamine (both Thermo Fisher Scientific), 1 mmol/l sodium pyruvate, 100 nmol/l sodium selenite (both Sigma Aldrich GmbH) and 2 ng/ml recombinant human interleukin-6 (IL-6; Miltenyi Biotec, Bergisch Gladbach, Germany). INA-6 were passaged three times per week by diluting them to 1×10^5 cells/ml, 2×10^5 cells/ml, or 4×10^5 cells/ml for 3, 2, and 1 days of culturing, respectively. MM.1S [*RRID:CVCL_8792*] (Greenstein et al., 2003), and U266 cells [*CVCL_0566*] (Nilsson et al., 1970) were propagated and cultivated in RPMI1640 medium comprising 10 % (v/v) FCS, 100 U/ml penicillin, 100 µg/ml streptomycin, 2 mmol/l L-glutamine, and 1 mmol/l sodium pyruvate. All cells were cultured at 37 °C and at 5 % CO_2.

Co-Culturing of Primary hMSCs and INA-6 and MSC-Conditioning of Medium

For each co-culture, hMSCs were seeded out 24 h prior to INA-6 addition to generate MSC-conditioned medium (CM). CM from different donors was collected separately and used imme-

diately when adding INA-6. To ensure that CM was free of hMSCs, it was strained (40 µm) and centrifuged for 15 min at 250 g. INA-6 cells were washed with PBS (5 min, 1200 rpm), resuspended in MSC-medium and added to hMSCs such that co-culture comprised 33 % (v/v) of CM gathered directly from the respective hMSC-donor. Co-cultures did not contain IL-6 (Chatterjee et al., 2002).

Collagen I Coating

Collagen I solution (isolated from rat tail, Corning, NY, USA) was diluted 1:2 (75 ng/mL) in acetic acid (0.02 N), applied to 96-well plates (30 µL in each well) and incubated for 2 h at room temperature. Acetic acid was removed and wells were washed once with 100 µL of PBS. Coated plates were stored dry at 4 °C.

Fluorescent Staining of Cells

For each live staining, cells were strained (70 µm) to remove clumps and washed (5 min, 250 g) once with the respective media (without FCS) and then resuspended in staining reagents. For *CellTracker™Green CMFDA Dye* and *CellTracker™Deep Red Dye* (Thermo Fisher Scientific) staining, 1 mL staining solution for a maximum of 1×10^6 cells was prepared. Staining was done at room temperature (RT) for 15 min using 5 µM CMFDA (5-Chlormethyl-fluoresceindiacetat) and 5 min of 1–2 µM DeepRed. To reduce background, stained cells were pelleted, resuspended in cell medium (containing FCS), incubated for 30 min (37 °C, 5 % CO_2), washed in cell medium, resuspended in 100–1000 µL and counted.

For PKH26 staining (Sigma Aldrich GmbH), a maximum of 1×10^4 cells was resuspended in 500 µL diluent C before swiftly adding 500 µL of staining solution (1 µL diluted in 500 µL diluent C) and incubating cells for 5 min at RT. The staining reaction was stopped by adding 1 mL of FCS-containing medium and adding 3 mL of FCS-free medium. Cells were washed with 10 mL of FCS-containing medium, resuspended in 100–1000 µL cell medium, and counted.

For Calcein-AM (Calcein-O,O -diacetat-tetrakis-(acetoxymethyl)-ester) (Thermo Fisher Scientific) staining, end concentrations of 0.5 µM were used. 12.5 µL of diluted stock solution (2.5 µM) was carefully added to 50 µL of the co-culture and incubated for 10 min at 37 °C.

For Hoechst 33342 staining, cells were washed once with PBS, resuspended in a maximum of 500 µL of PBS, and fixed with 5 mL of ice-cold ethanol (70 % v/v) by vigorously pipetting up and down to dissociate aggregates. Cells were washed once with PBS and stained with 2.5 µg/mL Hoechst 33342 (Thermo Fisher Scientific) diluted in PBS for 1 hour at 37 °C.

Automated Fluorescence Microscopy

To remove clumps for microscopic applications, we cultured cells in 40 µm strained medium containing FCS. To reduce background fluorescence and phototoxicity, we used phenol-red free versions of the respective medium, if available.

All microscopy equipment was acquired from ZEISS. The microscope was an *Axio Observer 7* with confocal *ApoTome.2* equipped with a motorized reflector revolver and motorized scanning table (130×100 mm). The microscope was mounted on an Antivibrations-Set (Axio Observer (D)) with two antivibration carrier plates, each equipped with two vibration dampening feet. The light source was a *microLED 2* for transmission light and (for fluorescence) *Colibri 7* (R[G/Y]B-UV) for five channels of incident light (385 nm, 475 nm, 555 nm, 590 nm and 630 nm). For excitation (EX) and emission (EM) light filtering and beam splitting (BS) we used the following reflectors: *96 HE BFP shift free* (E) (EX: 390 / 40 nm, BS: 420 nm, EM: 450 / 40 nm), *43 HE Cy 3 shift free* (E) (EX: 550 / 25 nm, BS: 570 nm, EM: 605 / 70 nm), *38 HE eGFP shift free* (E) (EX: 470 / 40 nm, BS: 495 nm, EM: 525 / 50 nm) and *90 HE LED* (E) (EX: 385 + 475 + 555 + 630 nm, BS: 405 + 493 + 575 + 653 nm, EM: 425/30 + 514/30 + 592/30 + 709/100 nm). We used the black and white camera *Axiocam 506 mono* (D) and if not stated otherwise, 2×2 binning was used for fluorescence imaging. For mosaic acquisitions ("tiles") we used a tiling overlap of 8–10 % and image tiles were not stitched. Images were magnified 5x and 10x (*Fluar 5x/0.25 M27* and *EC Plan-Neofluar 10x/0.3 Ph1 M27*).

Cell Viability and Apoptosis Assay

To examine cell viability and apoptosis, cells were seeded in a 96-well plate (1×10^4 cells per well) to be measured inside culture wells after respective incubation time immediately. ATP-amount and Caspase 3/7 activity were used as a proxy for viability and apoptosis rates, respectively. They were assessed using the *CellTiter-Glo Luminescent Cell Viability Assay* and the *Caspase-Glo 3/7 Assay*, respectively (Promega GmbH, Mannheim, Germany), according to the manufacturer's instructions. Luminescence was measured with an Orion II Luminometer (Berthold Detection Systems, Pforzheim, Germany).

Microscopic Characterization of hMSC Saturation

For saturating hMSC with INA-6, hMSCs were stained with *CellTracker Green*, plated out on 384-well plates (Greiner Bio-One GmbH, Frickenhausen, Germany) at 5×10^3 hMSC/cm^2 and cultured for 24 h. INA-6 cells were stained with *CellTracker DeepRed*, resuspended in MSC-medium, added to adhering hMSCs in different amounts (5×10^3 INA6/cm^2, 1×10^3 INA6/cm^2,

2×10^3 INA6/cm^2) and co-cultured for 24 h and 48 h. The complete co-culture was scanned and the number of INA-6 cells adhering to one hMSC was counted manually for 100 MSCs for each technical replicate. Fluorescent images were digitally re-stained (INA-6 green, hMSC inverse black).

Analysis of INA-6 Survival and Aggregation Depending on hMSC Confluence

To describe aggregate growth and survival of INA-6 depending on hMSC density, unstained hMSCs were seeded out into 96-well plates (white, clear bottom, Greiner) at different densities (see Seeding Table). To ensure nutrient supply, we used lower cell densities for longer co-culturing durations while maintaining constant ratios of INA-6 to adhesion surface provided by hMSCs. Those plates that were to be assessed after 72 h of co-culturing received an additional 100 µL of fresh MSC-medium after 24 h of co-culturing (total volume of 300 µL), and after 48 h of co-culturing, 100 µL was gently removed from the co-culture and carefully replaced with fresh MSC-medium without disturbing the co-culture on the bottom.

To describe aggregate growth, complete wells were scanned using 10x magnification, phase contrast, 2×2 binning, and autofocus focusing on each tile both before and after harvesting. Afterwards, INA-6 cells were harvested for measuring viability and apoptosis.

Seeding Table: Seeding densities for describing growth and survival of INA-6 depending on hMSC density. Co-cult. dur.: Co culturing duration; MSC-adh. surface: adhesion surface provided by hMSCs; vol.: volume.

Co-cult. dur. [h]	**hMSC density [1000 hMSC/cm²]**	**INA-6 density [1000 INA6/cm²]**	**Ratios INA : MSC (adh. surface)**	**Seeding vol. [µL]**	**End vol. [µL]**
24	2, 10, 40	10	1:0.2, 1:1, 1:confluent	200	200
48	1, 5, 40	5	1:0.2, 1:1, 1:confluent	200	200
72	1, 5, 40	5	1:0.2, 1:1, 1:confluent	200	300 [after 24 h: +100], [after 48 h: exchange 100]

For luminescent assessment of cell survival, INA-6 cells were harvested by removing co-culture medium, adding 150 µL of MSC-medium, and then stirred by strongly pipetting up and down twice while aiming the pipette tip at the upper corner, lower left, and lower right of the well bottom ('Mercedes star'). Washing and stirring was repeated once before washing wells again with 150 µL MSC-medium. Harvested INA-6 cells were strained (40 µm filter), pelleted, and resuspended in 200 µL MSC-medium. Cells were counted using Neubauer chambers, re-distributed into 96-well plates (white, clear bottom) with 1×10^5 INA-6 cells per well, and then subjected to viability and apoptosis assays.

To minimize the loss of sensitive apoptotic cells, another approach was used to measure

viability and apoptosis without harvesting INA-6 cells. hMSCs and INA-6 were seeded out individually in parallel to the co-cultures. Prior to measuring viability and apoptosis, culture volume was adjusted to 150 µL by removing 50 µL or 150 µL for the timepoints 48 h or 72 h, respectively (carefully not to stir up culture on bottom). 100 µL of luminescent reagents were then added directly to 150 µL of co-culture. The fold change of viability or apoptosis that is due to MSC interaction ($FC_{\text{MSC interaction}}$) was then calculated using the following formula, with L being the mean of four technical replicates measured in relative luminescent units per seconds [RLU/s], and $L_{\text{Co Culture}}$, L_{MSC}, L_{INA6} the luminescence measured in the co-culture, hMSCs alone, and INA-6 alone, respectively.

$$FC_{\text{MSC Interaction}} = \frac{L_{\text{Co Culture}}}{L_{\text{MSC}}+L_{\text{INA-6}}}$$

Time-Lapse Characterization of INA-6 Aggregation, Detachment and Division

In order to record the aggregation and detachment of INA-6 in contact with hMSCs, hMSCs (5×10^3 cells/cm^2) were fluorescently stained with PKH26 and plated onto 8-well μ-Slides (ibidi, Gräfelfing, Germany). hMSCs were incubated for 24 h before being placed into an ibidi Stage Top Incubation System and were equilibrated to the incubation system for a minimum of 3 h (80 % humidity and 5 % CO2). INA-6 cells (2×10^4 cells/cm^2) were washed and resuspended in 33 % (v/v) MSC-conditioned medium before adding them directly before acquisition start in a small volume (10 µL). Brightfield and fluorescence images of 13 mm^2 of co-culture were acquired every 15 minutes for 63 h. Movement speed of the motorized table was adjusted to the lowest setting that allows acquisition of the complete region within 15 minutes.

Respective events of interest were analyzed manually and categorized into defined event parameters. Events were binned across the time axis using these boundaries: [0.0, 12.85, 25.7, 38.55, 51.4, 64.25]. We collected a minimum of events per recording and analysis so that each time bin contained at least 5 values, except when analyzing detachment events, since these did not appear before 20 h of incubation for some replicates. For each recording and event parameter, the event count was normalized by dividing by the total number of events per time bin.

We determined the frequency and the cause of aggregation by looking for two interacting INA-6 cells and went backward in time to see if they were two daughter cells or if two independent INA-6 cells had collided. We determined the frequency of aggregates with detaching cells by tracing their growth across the complete time-lapse and looking for detachment events. We picked random 100 aggregates by including aggregates from both the border and center of the well.

We characterized detachment events by noting multiple parameters manually: Time point

of detachment, aggregate size (at the time of detachment), the last interaction partner, and the number of detaching INA-6 cells.

For characterizing cell division events, we recorded a new set of time-lapse videos using unstained hMSCs that were grown to confluence for 24 h (4×10^4 hMSCs/cm^2) to provide for unlimited adhesion surface. We categorized daughter cells in terms of their mobility (mobility being the speed of putative movements or "rolling"). The mobility criteria were met if one INA-6 daughter cell moved farther than half a cell radius within one frame (15 min) relative to the MSC-adherent INA-6 cell which was required to stand still in-between respective frames. We measured the "rolling" duration by subtracting the time point of the last perceived movement from the time point of division. We excluded those division events from the measurement of rolling duration if INA-6 cells underwent apoptosis shortly after division.

Cell Cycle Synchronization at M-Phase

INA-6 cells were arrested at mitosis by double thymidine (2 mM) treatments followed by 5 h of nocodazole (500 ng/mL) incubation. In detail: 3×10^5 cells/mL INA-6 in 4 mL were treated with 2 mM thymidine (Sigma Aldrich GmbH) for 16.5 h. Cells were released by washing them in INA-6 medium once and allowed to cycle for 9 h before treating them with 2 mM thymidine for 18 h a second time. Afterwards, cells were released and allowed to cycle for 2 h before treating them with 100 ng/mL nocodazole (Sigma Aldrich GmbH) for 5 h. Arrested INA-6 were released by washing them once and resuspending them in MSC-medium with 33 % MSC-conditioned medium. Cell cycle profile was checked using image cytometry (Appendix A: Fig. 2).

V-Well Adhesion Assay

This assay was modified from (Weetall et al., 2001). 96 v-well plates were coated with collagen I (rat tail, Corning). Collagen coating ensures that confluent hMSCs withstand centrifugation even after hMSCs in the well tip were removed. hMSCs (4×10^4 cells/cm^2) were seeded out and grown to confluence for 24 h in collagen-coated v-well plates. To ensure that only INA-6 are pelleted in the v-well tip, hMSCs were removed from the well-tip by touching the well-ground with a 10 µL pipette and roughly pipetting hMSCs away.

Arrested INA-6 (1×10^4 cells/cm^2) were released by washing them once in PBS and resuspending them in 33 % (v/v) MSC-conditioned medium before adding them on top of confluent hMSCs. INA-6 adhered for 1, 2, 3, and 24 h before the complete co-culture was stained with 0.5 µM Calcein-AM (10 min at 37 °C). Non-adherent INA-6 were pelleted by centrifugation using a Hettich 1460 rotor ($r = 124$ mm) at 2000 rpm (555 g) for 10 min.

The well tip was imaged by fluorescence microscopy with 5x magnification, *96 HE* emission

filter, autofocus configured for maximum signal intensity, 2×2 binning and 14 bit grayscale depth. Pellet brightness was analyzed in ZEN 2.6 (Zeiss) by summing up pixel brightnesses across the complete pellet image. Background brightness was acquired from a cell culture with only hMSCs. Reference brightness was acquired from a cell culture with only INA-6, defining 100 % pellet brightness without adhesion. Background intensity was subtracted before normalizing by reference. Outliers were removed from technical replicates ($n = 4$) if their z-score was larger than 1.5 σ technical variation.

After measuring pellet brightnesses, the cell pellet was removed by pipetting 10 µL from the well tip. Pellets of the same technical replicates were pooled, washed in PBS, resuspended in 200 µL PBS, added to 1.8 mL ice-cold 70 % ethanol, and stored at −20 °C. Remaining non-MSC-adhering INA-6 cells were removed by replacing culture medium with 100 µL of medium. MSC-adherent INA-6 were manually detached by rapid pipetting and equally pelleted, analyzed, and isolated.

Cell Cycle Profiling

INA-6 cells were fixed in 70 % ice-cold ethanol, washed, resuspended in PBS, distributed in 96-well plates, and stained with Hoechst 33342 (2.5 µg/mL in PBS) for 1 h at 37 °C. For image cytometric cell cycle profiling, plates were scanned completely using automated fluorescence microscopy with 5x magnification, *96 HE* emission filter, 1×1 binning, 14 bit depth, and an illumination time that fills 70 % of the grayscale range. The autofocus was configured to re-adjust every second tile. A pre-trained convolutional neural network ("DeepFeatures 2 reduced", *Intellesis*, Zeiss) was fine-tuned to segment scans into background, single nuclei, and fragmented nuclei. Nuclei were filtered to exclude fragmented nuclei and those nuclei with extreme size (within the range of 50–500 µm^2) and roundness (within the range of 0.4–1.0). Cell cycle profiles were normalized by the mode of the nucleus intensities within the G0/G1 peak. To retrieve frequencies of cells cycling in G0/G1, S, and G2 phase, the brightness distribution of all single nuclei was fitted to the sum of three Gaussian curves ("Skewed Gaussian Model" for G0G1 and G2 phase, and "Rectangle Model" for S phase) using the python package LMFIT (Newville et al., 2014) (Appendix A: Fig. 4). The Gaussian curves were used to calculate the cell frequencies for each cell cycle phase by integration using the composite trapezoidal rule implemented by numpy.trapz (Harris et al., 2020).

For validation of image cytometry, 5 mL of INA-6 stock culture was removed and ethanol fixed as described above. Flow cytometry analyses were performed using an *Attune Nxt Flow* Cytometer (Thermo Fisher Scientific). Data analyses were performed using FlowJo V10 software (TreeStar, USA).

Protocol: Well Plate Sandwich Centrifugation (WPSC)

96-well plates (flat bottom, clear) were coated with collagen I (rat tail, Corning) to ensure that confluent hMSCs withstand centrifugation and repeated washing. hMSCs (2×10^4 cells/cm^2) were seeded out and grown to confluence for 72 hours in collagen-coated 96-well plates. To remove aggregates from the medium and prevent clogging of magnetic columns, any FCS-containing fluid was strained with a 40 µm cell strainer.

- **Collect MSC-Conditioned Medium and Add INA-6:**
  1. Collect hMSC-conditioned medium (CM) from the well plates and replace it with 100 µL of fresh hMSC medium. Collect CM from different donors separately.
  2. Strain CM (40 µm) and centrifuge for 15 minutes at 250 g to ensure that CM does not contain hMSCs.
  3. Dilute CM by mixing 2 parts of CM with 1 part of MSC-medium (dilute 1.5 fold).
  4. Count INA-6 cells and retrieve enough cells to fill all 96 wells with 2×10^4 INA6/cm^2 (6.8×10^4 cells per well, covering approximately 65 % of the well bottom).
  5. Centrifuge INA-6 (5 minutes, 250 g) and resuspend them in a volume of diluted CM to reach a concentration of 6.8×10^5 INA6/mL.
  6. Add 100 µL INA-6 suspension to hMSCs (end volume: 200 µL; end concentration: 33 % (v/v) hMSC-conditioned medium).
  7. Incubate for 24 hours at 37 °C and 5 % CO_2.
- **Prepare CM-INA6 Reference:**
  8. Add 100 µL of fresh MSC-medium into each well of an empty 96-well plate (not coated).
  9. Add 100 µL of INA-6 suspension (6.8×10^5 INA-6/mL in diluted CM).
  10. Incubate for 24 hours at 37 °C and 5 % CO_2.
- **Collect CM-INA6 and nMA-INA6:**
  11. Pre-warm well plate centrifuge to 37 °C.
  12. Prepare a counter-weight by filling 200 µL of water into all wells of an empty 96-well plate.
  13. Prepare well-plate sandwiches:
      a. Turn an empty 96-well plate ("catching plate") upside down and place one on top of the co-culture-plate, the CM-INA6 reference plate, and the counter-weight so that all well openings align.
      b. Fix well plates using tape with reusable adhesive (e.g., *Leukofix*).
  14. Turn both plates around. Medium will spill from the co-culture plate into the catching plate.
  15. Centrifuge plate for 40 seconds at 1000 rpm with the catching plate facing the ground.

16. Remove the adhesive tape and the co-culture plate.
17. Turn the co-culture plate around and add 30 µL of washing medium (MSC-medium 0% FCS, 3 mM EDTA) gently by touching the wall of each well and pressing the pipette slowly.
    a. Work quickly to ensure that co-culture does not dry. We recommend using a multipette (Eppendorf).
    b. Many nMA-INA6 are removed by physical force applied by adding 30 µL of medium and not just by centrifugation. Hence, it is critical to apply the same dispensing technique across all replicates. We recommend using a multipette (Eppendorf) that can apply 30 µL with controllable pressure, since its push-button retains a long pushing path even for dispensing small volumes, unlike push-buttons from the usual 100 µL pipettes that reduce the pushing-path for smaller volumes.
    c. Centrifugation minimizes technical variability by replacing one step of manual pipetting. Also, it ensures that confluent MSCs remain unharmed. Manual pipetting on the other hand would require touching the well-bottom to remove all fluids which damages the adhesive hMSC layer.
18. Turn the co-culture plate upside down, place it onto the catching plate and re-apply adhesive tape to fix the well plate sandwich.
19. Repeat steps 14-18 two more times until the catching plate contains 290 µL of medium in each well.
20. Pool CM-INA6 from the catching plate that was fixed to the reference plate.
21. Pool nMA-INA6 from the catching plate that was fixed to the co-culture plate.
22. Collect remaining INA-6 by adding 100 µL of PBS into each well of the catching plates, collect and pool with CM-INA6 or nMA-INA6.
23. Strain CM-INA6 and nMA-INA6 using 40 µm cell strainer.
24. Isolate MA-INA6 by continuing with either accutase dissociation or rough pipetting.

- **Collect MA-INA6 by Accutase Dissociation Followed by MAC Sorting:**

25. Block 2 mL tubes with sorting buffer (PBS, 2 mM EDTA, 1 % BSA) for 1 hour at 4 °C.
26. Dilute accutase (Sigma Aldrich GmbH) (400 to600 units/mL) 4-fold in cold PBS. Always keep accutase on ice, since accutase loses activity at room temperature.
27. Add 50 µL of cold accutase (directly after the last centrifugation step) and incubate co-culture plate for 5 minutes at 37 °C.
28. Place a co-culture plate onto a shaker and shake for 1 minute at 300 rpm.
29. Collect cell suspension from wells and stop the reaction by adding 500 µL of FCS to pooled cell suspension.

30. Evaluate presence of adherent INA-6 cells and the integrity of confluent hMSCs under the microscope.
31. Repeat steps 27-30 until all INA-6 cells have dissociated or until confluent hMSCs start to tear.
32. Strain cell suspension (30 µm). This yields MA-INA6.
33. Pellet MA-INA6, nMA-INA6, and CM-INA6 (1200 rpm, 10 minutes).
34. Resuspend MA-INA6 in 86 µL sorting buffer (PBS, 2 mM EDTA, 1 percent BSA).
35. Resuspend CM-INA6 and nMA-INA6 in 300 µL cold diluted accutase and incubate for 3 minutes at 37 °C to ensure equal treatment for all samples.
36. Stop accutase by adding 200 µL of FCS (100 percent).
37. Pellet CM-INA6 and nMA-INA6 (1200 rpm, 10 minutes) and resuspend in 86 µL sorting buffer (PBS, 2 mM EDTA, 1 percent BSA).
38. Transfer samples into 2 mL tubes that were blocked with sorting buffer.
39. Add 10 µL of CD45 coated magnetic beads (Miltenyi Biotec B.V. & Co. KG, Bergisch Gladbach).
40. Place tubes into rotator and incubate for 15 minutes at 4 °C.
41. Continue with MAC sorting according to the manual. Use an MS column and wash 3 times.
42. Improve purity of eluted MA-INA6 by straining eluate (30 µm) (wash strainer using 1 mL of sorting buffer) and applying it onto an MS column a second time. Wash three times.
43. Collect 20 µL per eluate and apply it onto a 96 well plate to evaluate purity.
    a. Incubate plate for 24 hours.
    b. Count the number of adherent cells (hMSCs) per INA-6 using phase contrast microscopy.
    c. We reached a mean purity of 3.2×10^{-4} ($\pm 2.2 \times 10^{-4}$) hMSCs per MA-INA6.
    d. hMSC contamination did not have an impact on RNAseq, since those genes that are highly expressed in hMSCs (VCAM1, ALPL, FGF5, FGFR2), did not appear as differentially expressed in MA-INA6 (data not shown). RNAseq detected 0.44 ±0.16 CPM-normalized counts of VCAM1 transcripts in MA-INA6, however, it was excluded like all genes with less than 1 count in at least 2 of 5 replicates.
44. Count cells using a Neubauer chamber.
45. Pellet samples (250 g for 5 minutes).
46. Resuspend in respective medium or lysis buffer (e.g., RA1 for RNA extraction).

- **Collect MA-INA6 by Rough Pipetting (No MAC Sorting):**

47. After the last centrifugation step, add hMSC-medium to each well of the co-culture

plate to reach a volume of 150 µL.

a. Since the yield of MA-INA6 was large, we dissociated MA-INA6 cells from hMSCs by vigorous pipetting (for further samples after RNAseq, see Supplementary Table 1). Since no enzymatic digestion is used, we reckoned that there would be no need for MAC sorting. Confluent hMSCs withstand this procedure and don't dissociate as single cells, which can be removed by straining cells (30 µm). We reached similar purities as for MAC-sorting (data not shown).

48. Using a multi-channel pipette (100 µL), gently raise 90 µL into the tips.
49. Lean pipette tip on the upper well-border and roughly pipette up and down once.
50. Repeat step 48 at the lower right and lower left well border (Total of 3 pipetting steps "Mercedes Star").
51. Attach a catching plate onto the co-culture and centrifuge for 40 seconds at 500 rpm (28 g).
52. Repeat steps 46-50 until a sufficient amount of MA-INA6 is removed.
53. Control purity of MA-INA6 by placing out aliquot onto an empty 96 well plate.
54. Collect MA-INA6 from catching plate.
55. Remove hMSCs by straining cell suspension (30 µm).
56. Count cells using a Neubauer chamber.
57. Pellet MA-INA6 (250 g for 5 minutes).
58. Resuspend in respective medium or lysis buffer.

Centrifugal Force: We used a Hettich 1460 rotor ($r = 124$ mm) (Hettich GmbH & Co. KG, Tuttlingen, Germany). For calculating the centrifugal force that acts onto the co-culture within well plate sandwiches, we subtracted the height of the catching plate (14.4 mm, Greiner 96-well plate) and the depth of each well (10.9 mm). This yields a radius of 98.7 mm, which translates to the following centrifugal forces: 500 rpm: 28 g; 1000 rpm: 110 g; 2000 rpm: 441 g.

Washing medium containing EDTA: Washing medium containing EDTA: EDTA removes calcium from integrins, which are required for adhesion. It is not strong enough to dissociate INA-6 from hMSCs, but could help with removing INA-6 from other INA-6. For generating samples for RNAseq, we added 3 mM of EDTA to the washing medium. For further samples, we did not add EDTA to the washing medium, since we found that it does not increase yield for all biological replicates consistently (data not shown). We suspect that integrin-mediated adhesion depends on hMSC donor or internal variance of INA-6. We recommend using 3 mM of EDTA, however, this requires further optimizations like including an incubation time at 37 °C after the addition of washing medium to account for biological variance. However, this could take long incubation times of up to 60 minutes (Lai et al., 2022).

Track Cell Number During WPSC

To track the cell count during WPSC, INA-6 cells were stained with *CellTracker green*, and both co-culturing and catching plates were scanned after each centrifugation step. For each round of centrifugation, an empty catching plate was used. A pre-trained convolutional neural network (*Intellesis*, Zeiss) was fine-tuned to segment the scans into background, cells, and cell borders. Single cells were counted, and the cumulative sum for each catching plate was calculated.

Sub-Culturing After WPSC of MSC-Interacting INA-6 Subpopulations

After CM-INA6, nMA-INA6, and MA-INA6 were isolated, they were counted with a Neubauer chamber using all nine quadrants and diluted to 1×10^5 cells/mL in MSC-medium (10% FCS, no IL-6 except for control). 100 µL of cell suspension was applied to 96-well plates, incubated for 48 hours at 37 °C and 5 % CO_2, and then subjected to viability and apoptosis assays.

RNA Isolation

Total RNA was isolated from INA-6 cells using the *NucleoSpin RNA II Purification* Kit (Macherey-Nagel, Düren, Germany) according to the manufacturer's instructions.

RNAseq, Differential Expression and Functional Enrichment Analysis of INA-6 cells

FASTQ files were merged to the respective sample. The quality of FASTQ files was assessed with `FastQC` (Andrews, 2010) and a joint report was created with `MultiQC` (Ewels et al., 2016). FASTQ files were aligned with `STAR` (Dobin et al., 2013) to the GRCh38 reference genome build (Zerbino et al., 2018). Quality and alignment statistics of final BAM files were assessed with `samtools stats` (Li et al., 2009), and a joint report with `FastQC` reports by `MultiQC` was generated.

Raw read counts were generated with `HTSeq` (Anders et al., 2015) using the union method. `HTSeq` runs internally in `STAR`. Differential gene expression analysis was performed with `edgeR` (Robinson et al., 2010) in R 3.6.3 (R Core Team, 2018), according to the `edgeR` manual.

Counts were merged and genes with zero counts in all samples were removed (number of genes: 36380). The whole count table was annotated with R Bioconductor (Gentleman, n.d.) human annotation data package `org.Hs.eg.db` (Carlson, 2016). A DGEList Element was created with the raw counts, gene information, i.e., Ensembl GeneIDs, HUGO Symbol, Genename, and ENTREZ GeneIDs and a sample grouping metadata table.

```
1 y <- DGEList(counts=ct2[,-1:4], group=meta.data$group, genes=ct2[,1:4])
```

Counts were filtered to keep only those genes which have at least 1 read per million in at least 2 samples (number of genes: 14136). Afterwards, normalization factors were recalculated.

```
keep <- rowSums(cpm(y)>1) >=2
y <- y[keep, , keep.lib.size=FALSE]
y1 <- calcNormFactors(y)
```

A design matrix was created with the grouping factor by treatment condition (group=F1, F2, F3, which are abbreviations for CM-INA6, nMA-INA6, MA-INA6, respectively)

```
design = model.matrix(~0+group)
```

Dispersion was estimated, the resulting coefficient of biological variation (BCV) is 0.135, i.e., BCV expression values vary up and down by 13.5% between samples.

```
y1.1 <- estimateDisp(y1, design)
BCV <- sqrt(model.F$y1.1$common.dispersion)
```

A generalized linear model (`glmQLFit` function) was fitted.

```
fit <- glmQLFit(y1.1, design)
```

and pairwise comparisons were made, e.g.

```
F1vsF2 <- glmQLFTest(fit, contrast=makeContrasts(groupF1 - groupF2, levels=design))
DE.F1vsF2 <- topTags(F1vsF2, n=nrow(F1vsF2), p.value=0.05)
```

Afterwards, gene list of differentially expressed genes were used for functional enrichment analysis with `Metascape` (Y. Zhou et al., 2019).

RT-qPCR

For cDNA synthesis, 1 µg of total RNA was reverse transcribed with Oligo(dT)15 primers and Random Primers (both Promega GmbH) and *Superscript IV* reverse transcriptase (Thermo Fisher Scientific) according to the manufacturer's instructions. For quantitative PCR, the cDNA was diluted 1:10, and qPCR was performed in 20 µL using 2 µL of cDNA, 10 µL of *GoTaq qPCR Master Mix* (Promega GmbH), and 5 pmol of sequence-specific primers obtained from biomers.net GmbH (Ulm, Germany) or Qiagen GmbH (Hilden, Germany) (see Supplementary Table 3 for primer sequences and PCR conditions). qPCR conditions were as follows: 95 °C for 3 minutes; 40 cycles: 95 °C for 10 seconds; respective annealing temperature for 10 seconds; 72 °C for 10 seconds; followed by melting curve analysis for the specificity of qPCR products using an *TOptical Gradient 96 PCR Thermal Cycler* (Analytik Jena AG, Jena, Germany). Samples that showed unspecific byproducts were discarded. Ct values were measured in three technical replicates (triplicates). Non-detects were discarded. One of three technical replicates was treated as an outlier and excluded if its z-score crossed 1.5 σ technical variation. We normalized

expression by the housekeeping gene *36B4*. Efficiencies were determined in each reaction by linear regression of log-transformed amplification curves (Ramakers et al., 2003). Differential expression was calculated based on a modified $\Delta\Delta Ct$ formula that separated exponents to apply individual efficiencies to each Ct value:

$$\text{Fold Change} = \frac{E_{\text{tar}}^{\Delta Ct_{\text{tar}(co\ -\ treated)}}}{E_{\text{ref}}^{\Delta Ct_{\text{ref}(co\ -\ treated)}}} = \frac{E_{\text{tar,co}}^{Ct_{\text{tar,co}}} \div E_{\text{tar,treated}}^{Ct_{\text{tar,treated}}}}{E_{\text{ref,co}}^{Ct_{\text{ref,co}}} \div E_{\text{ref,treated}}^{Ct_{\text{ref,treated}}}}$$

- $E_{\text{tar,co}}$ = Efficiency of the target gene measured in the control sample
- $Ct_{\text{tar,co}}$ = Ct value of the target gene measured in the control sample
- tar = Target Gene
- ref = Reference Gene
- $treated$ = Treated sample
- co = Control Sample

Fold change expression was normalized by the median of CM-INA6 (and not sample-wise, as commonly used in $\Delta\Delta C_t$) since some genes were not expressed without direct hMSC contact (e.g., *MMP2*), and also in order to display variation of CM-INA6 next to nMA-INA6 and MA-INA6.

Statistics

For molecular analyses, each data point represents one biological replicate, defined as the mean of all technical replicates of co-cultures seeded from the same batch of hMSCs and/or INA-6 cells on the same day. For analyses of time-lapse recordings, each data point represents the normalized event count from one co-culture recording. Unique hMSCs were prioritized for each biological replicate or recording (see Appendix A: Tab. 1). Bars and lines represent the mean, and error bars represent the standard deviation of all hMSC donors or recordings (= all biological replicates).

Metric, normally distributed, dependent data were analyzed using factorial RM-ANOVA and paired Student's t-tests. Results of RM-ANOVA are reported as follows

$$[F(df_1, df_2) = F; p = p\text{-value}],$$

where df_1 is the degrees of freedom of the observed effect, df_2 is the degrees of freedom of the error, and F is the F-statistic (Vallat, 2018). If sphericity was met, p-values were not corrected using the Greenhouse-Geisser method (p-unc).

$$df_1 = k - 1 \qquad \text{(k = number of groups)}$$

$$df_2 = (k-1)(n-1) \qquad \text{(n = number of samples in each group)}$$

$$F = \frac{SS_{\text{Effect}}/df_1}{SS_{\text{Error}}/df_2} \qquad \text{(SS = Sums of squares for effect or error)}$$

If data points within dependent sample pairs were missing, such pairs were excluded from paired t-tests while others from the same subject remained.

Metric non-normal distributed, independent data was analyzed using the Kruskal-Wallis H-test and Mann–Whitney U tests. Results of the Kruskal-Wallis H-test are reported as

$$H(df) = H$$

, with df being the degrees of freedom and H being the Kruskal-Wallis H statistic, corrected for ties (Vallat, 2018).

Metric bivariate non-normal distributed data was correlated using Spearman's rank correlation and reported as:

$$\rho(df) = \rho, p = p\text{-value}$$

, where ρ is Spearman's rank correlation coefficient. df is calculated as

$$df = n - 2 \quad (\text{n = number of observations})$$

These tests were applied using Python (3.10) packages `pingouin` (0.5.1) and `statsmodels` (0.14.0) (Seabold & Perktold, 2010; Vallat, 2018). Data was plotted using `seaborn` (Waskom, 2021) and `plotastic` (Kuric & Ebert, 2024). Sphericity was ensured by Mauchly's test. Normality was checked with the Shapiro-Wilk test for $n > 3$.

Data points were log10 transformed to convert the scale from multiplicative ("fold change") to additive, or to fulfill sphericity requirements. P-values derived from patient survival data were corrected using the Benjamini-Hochberg procedure. For other post-hoc analyses, p-values were not adjusted for family-wise error rate in order to minimize type I errors. To prevent type II errors, the same conclusions were validated by different experimental setups and through varying hMSCs donors across experiments (see Appendix A: Tab. 1).

Significant p-values from pairwise tests were annotated as stars between data groups

$$p - value = 0.05 > * > 0.01 > ** > 10^{-3} > *** > 10^{-4} > ****.$$

If too many significant pairs were detected, only those pairs of interest were annotated.

No power calculation was performed to determine sample size since samples were limited by availability of primary hMSC donors. Experiments were repeated until a minimum of three biological replicates were gathered.

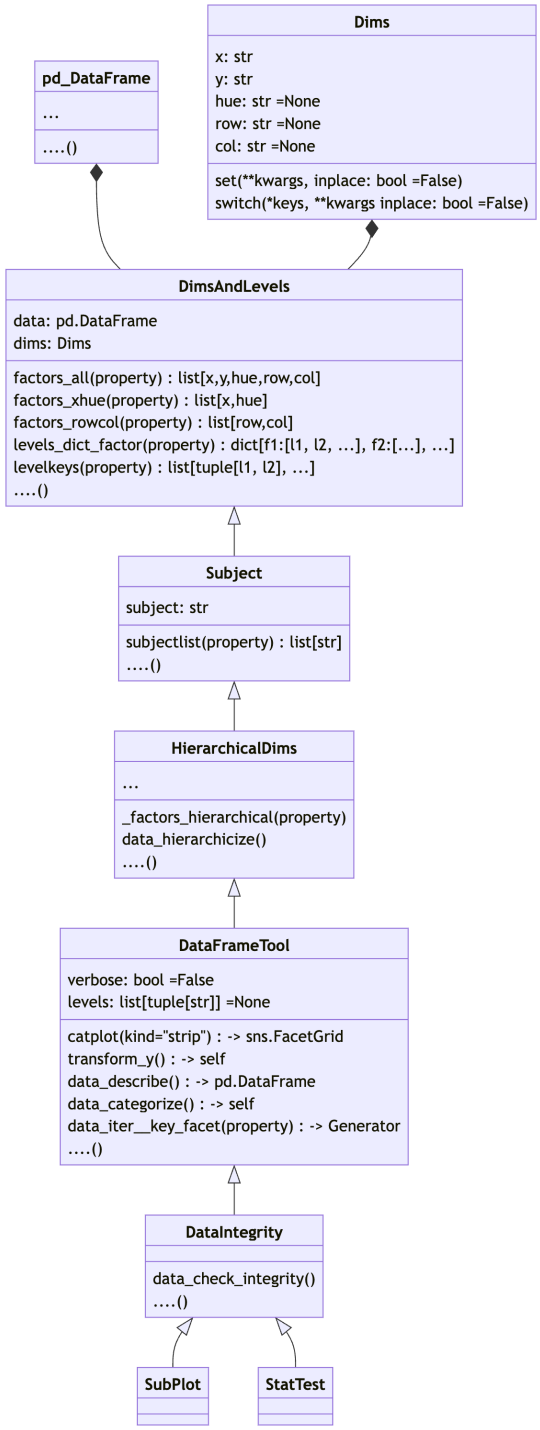
Patient Cohort, Analysis of Survival and Expression

Patient samples ($n = 873$) were collected at the University Hospital Heidelberg and processed as described (Seckinger et al., 2017, 2018), and are available at the European Nucleotide Archive (ENA) via accession numbers PRJEB36223 and PRJEB37100. Consecutive patients with monoclonal gammopathy of unknown significance (MGUS) ($n = 62$), asymptomatic ($n = 259$), symptomatic, therapy-requiring ($n = 764$), and relapsed/refractory myeloma ($n = 90$), as well as healthy donors ($n = 19$) as comparators were included in the study approved by the ethics committee (#229/2003, #S-152/2010) after written informed consent.

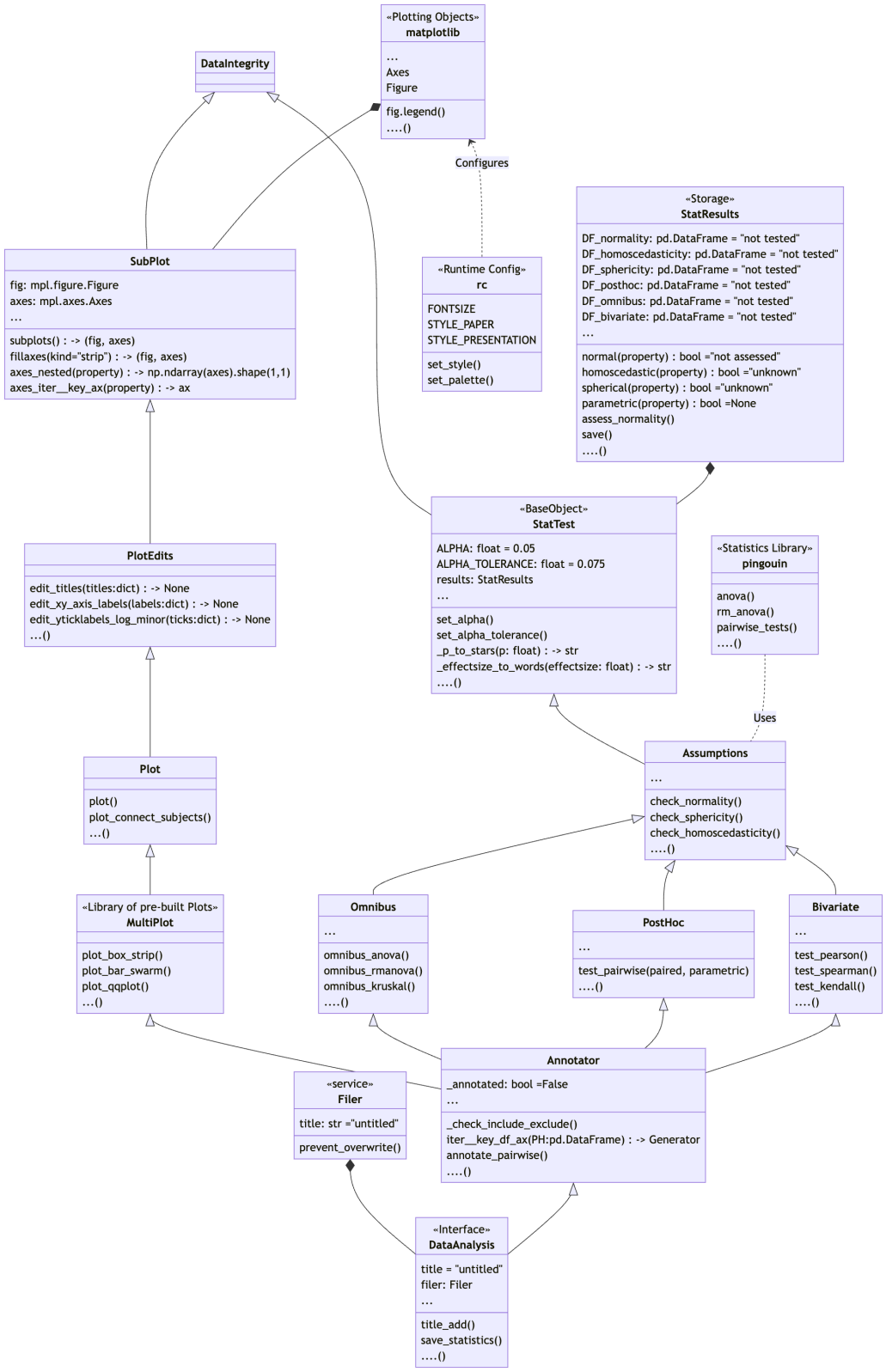
Gene expression was measured by RNA sequencing as previously described (Seckinger et al., 2018). Gene expression is defined as the $\log_2$ transformed value of normalized counts + 1 (as pseudocount). Progression-free (PFS) and overall survival (OS) were analyzed for the subset of previously untreated symptomatic myeloma patients. For delineating "high" and "low" expression of target adhesion ($n = 101$) and cell cycle ($n = 173$) genes, thresholds per gene were calculated with maximally selected rank statistics by the `maxstat` package in R (Hothorn & Lausen, n.d.). PFS and OS were analyzed for high vs. low expression with the Kaplan-Meier method (Kaplan & Meier, 1958). Significant differences between the curves were analyzed with log-rank tests (Harrington & Fleming, 1982). P-values were corrected for multiple testing by the Benjamini-Hochberg method. Analyses were performed with R version 3.6.3 (R Core Team, 2018).

B Documentation of plotastic

B.1 Class Diagram



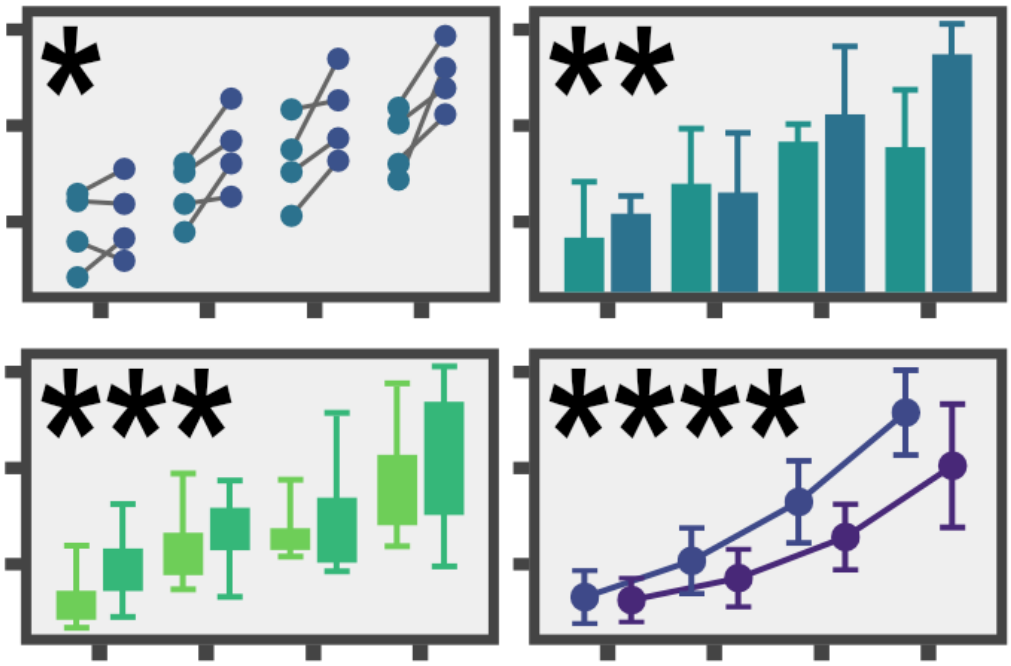
Appendix B Figure 1: **Class diagram of `plotastic` (upper part):** The architecture of `plotastic` begins with classes that are related to handling a `pandas.DataFrame` object which stores the data, and defining dimensions to group the data (y, x, hue, col, row). This diagram ends with the classes `SubPlot` and `StatTest` and is continued on the next page. Arrow shapes follow the UML (unified modeling language): A hollow triangle indicates inheritance (*"is a"*) and a filled diamond indicates composition (*"has a"*).



Appendix B Figure 1: *continued from previous page* The architecture of `plotastic` continues after the class `DataIntegrity` with classes for plotting (`SubPlot`) and statistical testing (`StatTest`) and ends with the class `DataAnalysis`, which serves as the main user interface. Arrow shapes follow the UML (unified modeling language): A hollow triangle indicates inheritance (*"is a"*) and a filled diamond indicates composition (*"has a"*).

B.2 Readme

The following pages are the README.md of plotastic found in the Python Package Index (PyPi) (pypi.org/project/plotastic), and on GitHub (github.com/markur4/plotastic).



---

plotastic: Bridging Plotting and Statistics

📦 Installation

Install from PyPi:

```
pip install plotastic
```

Install from GitHub: (experimental, check CHANGELOG.md)

```
pip install git+https://github.com/markur4/plotastic.git
```

Requirements

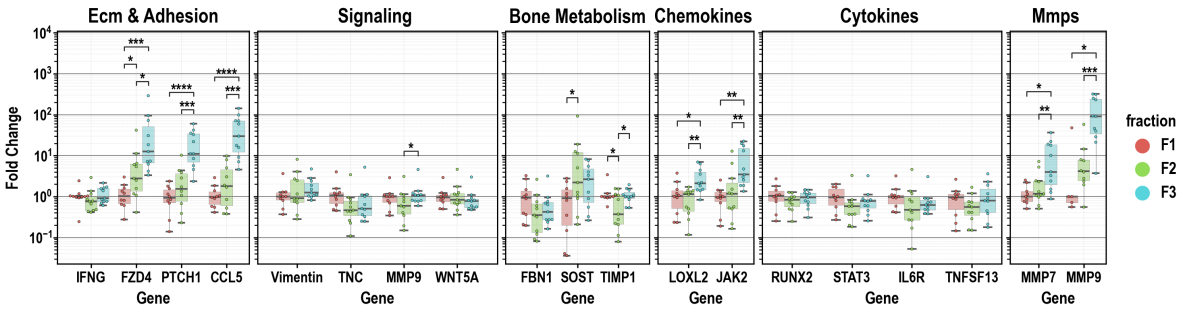
- Python >= 3.11 *(not tested with earlier versions)*

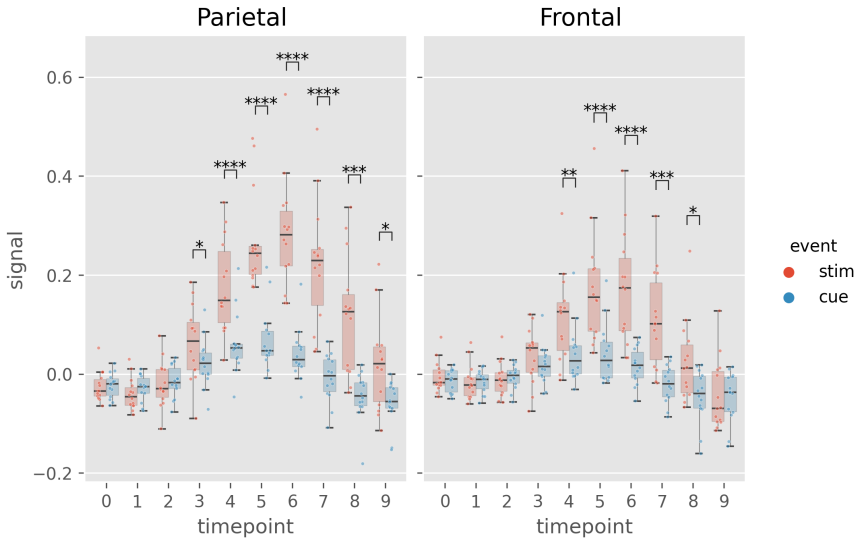
- pandas == 1.5.3 *(pingouin needs this)*
- seaborn <= 0.12.2 *(later versions reworked hue)*

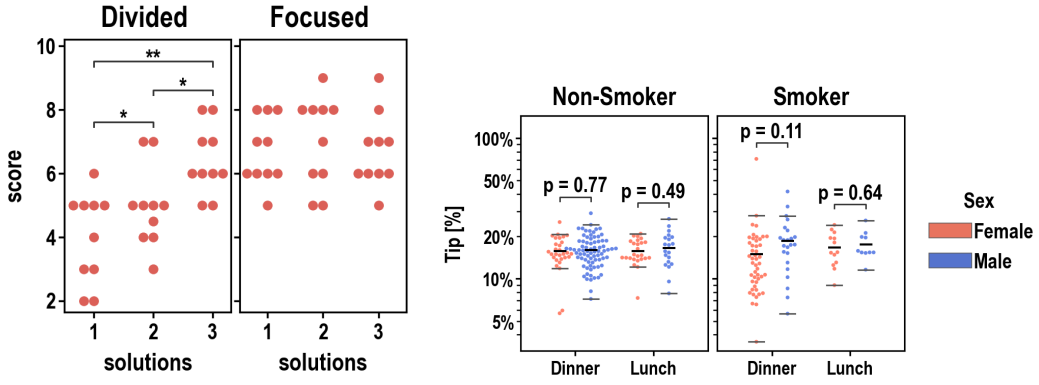
📷 Example Gallery

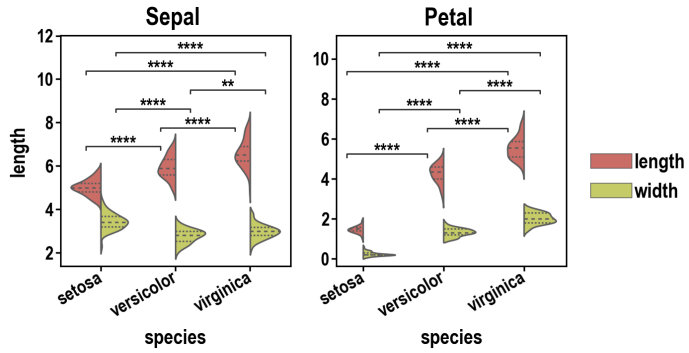
▶ ***(click to unfold)***

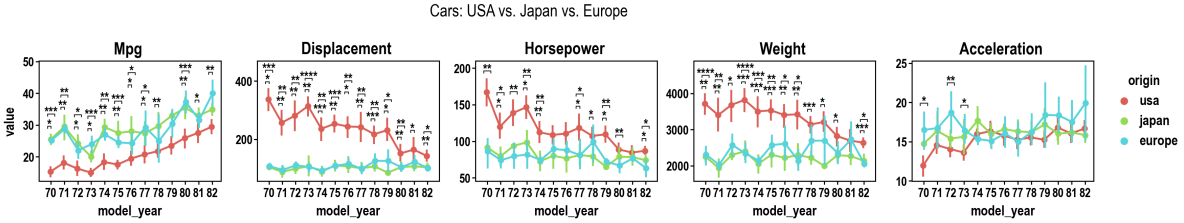
🐁 Click on Images for Code! 🐁

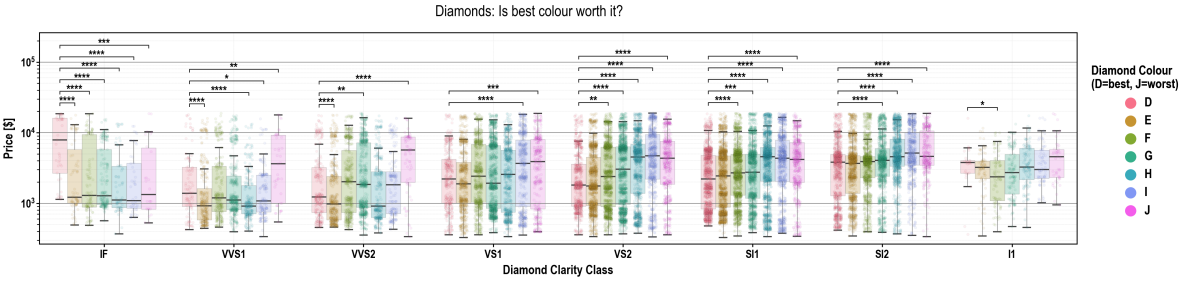




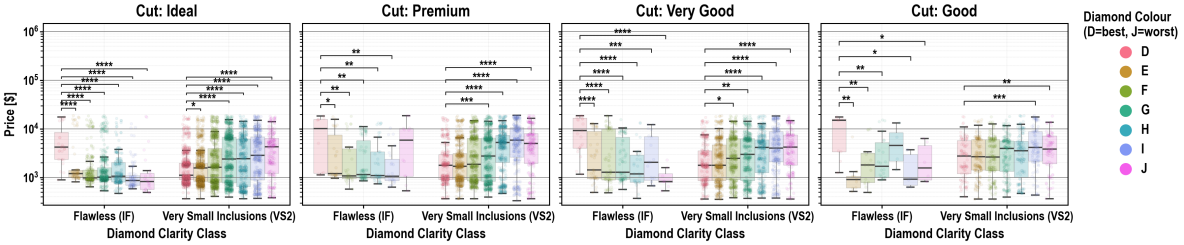








>> An Ideally cut diamond with minor flaws and worst color
is as expensive as a perfect diamond with best color and best clarity



About plotastic

▶ 🤔 ***Summary***

plotastic addresses the challenges of transitioning from exploratory data analysis to hypothesis testing in Python's data science ecosystem. Bridging the gap between seaborn and pingouin, this library offers a unified environment for plotting and statistical analysis. It simplifies the workflow with a user-friendly syntax and seamless integration with familiar seaborn parameters (y, x, hue, row, col). Inspired by seaborn's consistency, plotastic utilizes a DataAnalysis object to intelligently pass parameters to pingouin statistical functions. The library systematically groups the data according to the needs of statistical tests and plots, conducts visualisation, analyses and supports extensive customization options. In essence, plotastic establishes a protocol for configuring statical analyses through plotting parameters.

This approach streamlines the process, translating `seaborn` parameters into statistical terms, providing researchers and data scientists with a cohesive and user-friendly solution in python.!

Workflow:

1. 🧱 **Import & Prepare your pandas DataFrame**
   - We require a long-format pandas dataframe with categorical columns
   - If it works with seaborn, it works with plotastic!
2. 🔀 **Make a DataAnalysis Object**
   - `DataAnalysis(DataFrame, dims={x, y, hue, row, col})`
   - Check for empty data groups, differing samplesizes, NaN-count, etc. automatically
3. ✅ **Explore Data**
   - Check Data integrity, unequal samplesizes, empty groups, etc.
   - Quick preliminary plotting with e.g. `DataAnalysis.catplot()`
4. 🔨 **Adapt Data**
   - Categorize multiple columns at once
   - Transform dependent variable
   - Each step warns you, if you introduced NaNs without knowledge!
   - etc.
5. ✨ **Perform Statistical Tests** ✨
   - Check Normality, Homoscedasticity, Sphericity
   - Perform Omnibus tests (ANOVA, RMANOVA, Kruskal-Wallis, Friedman)
   - Perform PostHoc tests (Tukey, Dunn, Wilcoxon, etc.) based on `pg.pairwise_tests()`
6. 📊 **Plot figure**
   - Use pre-defined and optimized multi-layered plots with one line (e.g. strip over box)!
   - Annotate statistical results (p-values as *, **, ***, etc.) with full control over which data to include or exclude!
7. 💿 **Save all results at once!**
   - One DataAnalysis object holds:
     - One DataFrame in `self.data`
     - One Figure in `self.fig`, `self.axes`
     - Multiple statistical results: `self.results`
   - Use `DataAnalysis.save_statistics()` to save all results to different sheets collected in one .xlsx filesheet per test

▶ 📊 ***Translating Plots into Statistics!***

In Principle:

- Categorical data is separable into `seaborn`'s categorization parameters: ***x***, ***y***, ***hue***, ***row***, ***col***. We call those *"dimensions"*.
- These dimensions are assigned to statistical terms:
  - ***y*** is the ***dependent variable*** (***DV***)
  - ***x*** and ***hue*** are ***independent variables*** (***IV***) and are treated as ***within/between factors*** (categorical variables)
  - ***row*** and ***col*** are ***grouping variables*** (categorical variables)
  - A ***subject*** may be specified for within/paired study designs (categorical variable)

- For each level of ***row*** or ***col*** (or for each combination of ***row***- and ***col*** levels), statistical tests will be performed with regards to the two-factors ***x*** and ***hue***

Example with ANOVA:

- Imagine this example data:
  - Each day you measure the tip of a group of people.
  - For each tip, you note down the ***day***, ***gender***, ***age-group*** and whether they ***smoke*** or not.
  - Hence, this data has 4 categorical dimensions, each with 2 or more *levels*:
    - ***day***: 4 levels (*monday*, *tuesday*, *wednesday*, *Thursday*)
    - ***gender***: 2 levels (*male*, *female*)
    - ***smoker***: 2 levels (*yes*, *no*)
    - ***age-group***: 2 levels (*young*, *old*)
- Each category is assigned to a place of a plot, and when calling statistical tests, we assign them to statistical terms (in comments):
  - ```
    # dims is short for dimensions
    dims = dict(          # STATISTICAL TERM:
        y = "tip",        # y-axis, dependent variable
        x = "day",        # x-axis, independent variable (within-
    subject factor)
        hue = "gender",   # color,  independent variable (within-
    subject factor)
        col = "smoker",   # axes,   grouping variable
        row = "age-group" # axes,   grouping variable
    )
    ```
- We perform statistical testing groupwise:
  - For each level-combinations of ***smoker*** and ***age-group***, a two-way ANOVA will be performed (with ***day*** and ***gender*** as ***between*** factors for each datagroup):
    - 1st ANOVA assesses datapoints where ***smoker***=*yes* AND ***age-group***=*young*
    - 2nd ANOVA assesses datapoints where ***smoker***=*yes* AND ***age-group***=*old*
    - 3rd ANOVA assesses datapoints where ***smoker***=*no* AND ***age-group***=*young*
    - 4th ANOVA assesses datapoints where ***smoker***=*no* AND ***age-group***=*old*
  - Three-way ANOVAs are not possible (yet), since that would require setting e.g. ***col*** as the third factor, or implementing another dimension (e.g. ***hue2***).

▶ ! ***Disclaimer about Statistics***

This software was inspired by ...

- ... ***Intuitive Biostatistics*** - Fourth Edition (2017); Harvey Motulsky
- ... ***Introduction to Statistical Learning with applications in Python*** - First Edition (2023); Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, Jonathan Taylor
- ... talking to other scientists struggling with statistics

✅ `plotastic` **can help you with...**
    ```

- ... gaining some practical experience when learning statistics
- ... quickly gain statistical implications about your data without switching to another software
- ... making first steps towards a full statistical analysis
- ... plotting publication grade figures (check statistics results with other software)
- ... publication grade statistical analysis **IF** you really know what you're doing OR you have back-checked your results by a professional statistician
- ... quickly test data transformations (log)

🚫 `plotastic` can NOT ...

- ... replace a professional statistician
- ... teach you statistics, you need some basic knowledge (but is awesome for practicing!)
- ... test for multicolinearity (Absence of multicolinearity is required by ANOVA!)
- ... perform stringent correction for multiple testing (e.g. bonferoni), as statistical tests are applied to sub-facets of the whole dataframe for each axes, which depends on the definition of x, hue, col, etc. Hence, corrected p-values might over-estimate the significance of your results.

🟡 Be critical and responsible with your statistical analysis!

- **Expect Errors:** Don't trust automated systems like this one!
- **Document your work in *ridiculous detail***:
  - Include the applied tests, the number of technical replicates and the number of biological/independent in each figure legend
  - State explicitly what each datapoint represents:
    - 1 datapoint = 1 Technical replicate?
    - 1 datapoint = The mean of all technical replicate per independent replicate/subject?
  - State explicitly what the error-bars mean: Standard deviation? Confidence interval?
  - (Don't mix technical with biological/independent variance)
  - Report if/how you removed outliers
  - Report if you did or did not apply correction methods (multiple comparisons, Greenhouse Geyser, etc.) and what your rationale is (exploratory vs. confirmatory study? Validation through other methods to reduce Type I error?)
- **Check results with professionnals:**
  - *"Here is my data, here is my question, here is my analysis, here is my interpretation. What do you think?"*

▶ ✅ *Feature List*

- ✅ **: Complete and tested**
- 👍 **: Complete**
- 📆 **: Planned or unfinished (no date)**
- 🤷 **: Maybe..? (Rather not...)**
- 🚫 **: Not planned, don't want**
- 😖 **: Help Please..?**

▼ *Plotting*

- 👍 Make and Edit Plots: *Implemented* ✅

    - *All (non-facetgrid) seaborn plots should work, not tested*

- 📆 QQ-Plot
- 📆 Kaplan-Meyer-Plot
- 🤹 Interactive Plots (where you click stuff and adjust scale etc.)
    - *That's gonna be alot of work!*
- 🚫 Support for `seaborn.FacetGrid`
    - *Why not? - `plotastic` uses matplotlib figures and fills its axes with seaborn plot functions. In my opinion, that's the best solution that offers the best adaptibility of every plot detail while bieng easy to maintain*
- 🚫 Support for `seaborn.objects` (same as Facetgrid)
    - *Why not? - I don't see the need to refactor the code*
- 😖 **NEED HELP WITH:** The hidden state of `matplotlib` figures/plots/stuff that gets drawn:
    - *I want to save the figure in `DataAnalysis.fig` attribute. As simple as that sounds, matplotlib does weird stuff, not applying changes after editing the plot.*
    - *It'd be cool if I could control the changes to a DataAnalysis object better (e.g. using `inplace=True` like with `pd.DataFrames`). But I never figured out how to control matplotlib figure generation, even with re-drawing the figure with canvas. It's a mess and I wasted so much time already.*

▼ ***Multi-Layered Plotting***

- ✅ Box-plot + swarm
- 👍 Box-plot + strip
- 📆 Violin + swarm/strip

▼ ***Statistics***

- Assumption testing
    - ✅ Normality (e.g. Shapiro-Wilk)
    - ✅ Homoscedasticity (e.g. Levene)
    - ✅ Sphericity (e.g. Mauchly)
- Omnibus tests
    - ✅ ANOVA, RMANOVA, Kruskal-Wallis, Friedman
    - 📆 Mixed ANOVA
    - 📆 Annotate Results into Plot
- PostHoc
    - ✅ `pg.pairwise_tests()`
        - *Works with all primary options. That includes all parametric, non-parametric, paired, unpaired, etc. tests (t-test, paired t-test, MWU, Wilcoxon, etc.)*
    - ✅ Annotate Stars into plots (*, **, etc.)
        - *Specific pairs can be included/excluded from annotation*
    - 📆 Make correction for multiple testing go over complete DataFrame and not Facet-wise:
- Bivariate
    - 📆 Find and Implement system to switch between numerical and categorical x-axis
        - *Function to convert numerical data into categorical data by binning?*
    - 📆 Pearson, Spearman, Kendall

▼ ***Analysis Pipelines***

Idea: Put all those statistical tests into one line. I might work on this only after everything's implemented and working confidently and well!

- 🐝 `between_samples(parametric=True)`: ANOVA + Tukey (if Normality & Homoscedasticity are given)
- 🐝 `between_samples(parametric=False)`: Kruskal-Wallis + Dunn
- 🐝 `within_samples(parametric=True)`: RM-ANOVA + multiple paired t-tests (if Normality & Sphericity are given)
- 🐝 `within_samples(parametric=False)`: Friedman + multiple Wilcoxon

📔 How To Use

Documentations

1. **Example Gallery**
    1. Quick Example: FMRI
    2. qPCR (paired, parametric)
    3. Cars (unpaired, non-parametric)
    4. Diamonds (unpaired, non-parametric)
    5. Attention (paired/mixed, parametric)
    6. Iris (unpaired, parametric)
    7. Tips (unpaired, parametric)
2. **Data**
    1. Set/Switch Dimensions
3. **Plotting**
    1. Quick & Simple: MultiPlots
    2. Constructing Plots
    3. Legends
    4. Styles

Quick Example

Import plotastic and example Data

```
import matplotlib.pyplot as plt
import plotastic as plst

# Import Example Data (Long-Format)
DF, _dims = plst.load_dataset("fmri", verbose = False)
DF.head()
```

Assign each column to a dimension (y, x, hue, col, row):

```
dims = dict(
    y = "signal",    # y-axis, dependent variable
    x = "timepoint", # x-axis, independent variable & within-subject
factor
    hue = "event",   # color,  grouping variable & within-subject factor
    col = "region"   # axes,   grouping variable
)
```

Initialize DataAnalysis Object

```
DA = plst.DataAnalysis(
    data=DF,           # Dataframe, long format
    dims=dims,         # Dictionary with y, x, hue, col, row
    subject="subject", # Datapoints are paired by subject (optional)
    verbose=False,     # Print out info about the Data (optional)
)
```

Perform Statistics

No arguments need to be passed, although `**kwargs`, are passed to respective `pingouin` functions.

```
DA.check_normality()  # Normal Distribution?
DA.check_sphericity() # Sphericity?
DA.omnibus_rm_anova() # Repeated Measures ANOVA
DA.test_pairwise()    # Post-hoc tests
```

Save Results:

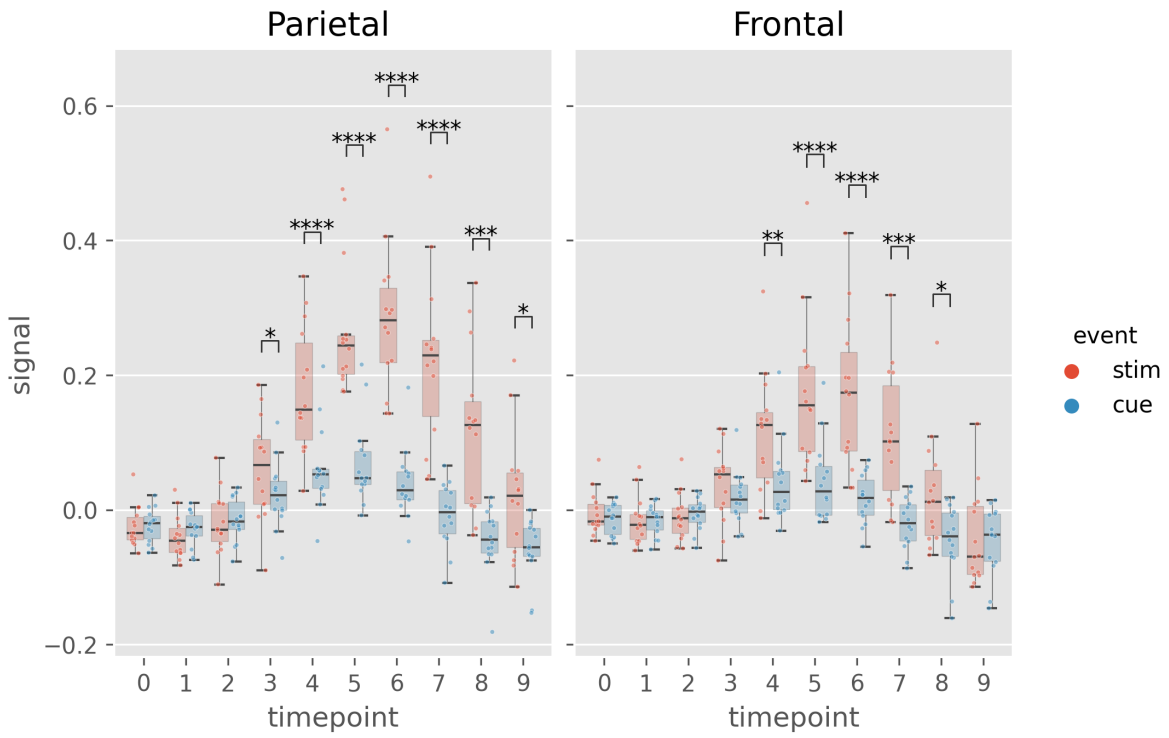
Output is one excel file containing results of all performed tests (normality, anova, t-tests, etc.) in different sheets

```
DA.save_statistics("example.xlsx")
```

Annotate post-hoc results into plot:

```
(DA
 .plot_box_strip()   # Pre-built plotting function initializes plot
 .annotate_pairwise( # Annotate results from DA.test_pairwise()
     include="__HUE" # Use only significant pairs across each hue
     )
)
```

```
# Saving the plot like matplotlib!
plt.savefig("example.png", dpi=200, bbox_inches="tight")
```



🧪 Testing

▶ ***(click to unfold)***

- Download/Clone repository
- Install development tools `pip install .[dev]`
- Run tests
  - Run `pytest ./tests`
  - To include a coverage report run `pytest ./tests -cov--cov-report=html` and open `./htmlcov/index.html` with your browser.

🤝 Community Guidelines

▶ ***(click to unfold)***

When interacting with the community, you must adhere to the Code of Conduct

Contribute

I am grateful for pull requests!

- Make sure to understand the code (e.g. see Class diagram in this Readme)
- Run tests before submitting a pull request

Reporting Issues & Problems

If you need help, please open an issue on this repository.

- Please provide a minimal example to reproduce the problem.

Support

If you need help, please open an issue on this repository.

✍🏻 Cite These!

▶ ***(click to unfold)***

Kuric et al., (2024). plotastic: Bridging Plotting and Statistics in Python. Journal of Open Source Software, 9(95), 6304, https://doi.org/10.21105/joss.06304

Vallat, R. (2018). Pingouin: statistics in Python. Journal of Open Source Software, 3(31), 1026, https://doi.org/10.21105/joss.01026

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

```
@article{Kuric2024,
  doi = {10.21105/joss.06304},
  url = {https://doi.org/10.21105/joss.06304},
  year = {2024}, publisher = {The Open Journal},
  volume = {9},
  number = {95},
  pages = {6304},
  author = {Martin Kuric and Regina Ebert},
  title = {plotastic: Bridging Plotting and Statistics in Python},
  journal = {Journal of Open Source Software}
}


@article{Waskom2021,
    doi = {10.21105/joss.03021},
    url = {https://doi.org/10.21105/joss.03021},
    year = {2021},
    publisher = {The Open Journal},
    volume = {6},
    number = {60},
    pages = {3021},
    author = {Michael L. Waskom},
    title = {seaborn: statistical data visualization},
    journal = {Journal of Open Source Software}
 }

 @article{Vallat2018,
  title    = "Pingouin: statistics in Python",
  author   = "Vallat, Raphael",
  journal  = "The Journal of Open Source Software",
  volume   =  3,
```

```
    number   =  31,
    pages    = "1026",
    month    =  nov,
    year     =  2018
  }
```

B.3 Example Analysis “qpcr”

The following pages are a jupyter notebook from an example analysis using plotastic. This notebook and further analyses examples are found on GitHub (github.com/markur4/plotastic). The qPCR dataset was derived from the experiments described in Chapter 1 Fig. 4 and was changed into a public test dataset by exchanging the original gene names with random ones while preserving gene classes and quantitative fold changes.

qpcr

April 9, 2024

[ ]:
```
import plotastic as plst
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
```

[ ]:
```
# Set Plot Style
plst.set_style("paper")
# plst.set_palette("hls", verbose=True)
plst.set_palette(["#db5f57", "#91db57", "#57d3db"])
```

```
#! You chose this color palette: ['#db5f57', '#91db57', '#57d3db', '#db5f57',
'#91db57', '#57d3db', '#db5f57', '#91db57']
```

```
['#db5f57',
 '#91db57',
 '#57d3db',
 '#db5f57',
 '#91db57',
 '#57d3db',
 '#db5f57',
 '#91db57']
```

1 Example Analysis: qPCR

Raw Data: https://github.com/markur4/plotastic/tree/main/src/plotastic/example_data/data

Original Source: (unpublished)

[ ]:
```
# Import Example Data
DF, _dims = plst.load_dataset("qpcr", verbose=False)
dims = dict(
    y="fc",
    x="gene",
    hue="fraction",
    # col= 'method',
    row="class",
)
DA = plst.DataAnalysis(DF, dims, subject="subject", verbose=False)
```

[ ]:
```
DA.transform_y("log10", inplace=True)  # Log transform
DA.check_normality()  # -> Only few groups are not normal -> parametric
```

[ ]:

class	gene	fraction	W	pval	normal	n
Bone Metabolism	F1	FBN1	0.936873	0.518768	True	10
		SOST	0.880395	0.131862	True	10
		TIMP1	0.745494	0.004807	False	9
	F2	FBN1	0.954764	0.705148	True	11
		SOST	0.967810	0.863610	True	11
		TIMP1	0.914325	0.274168	True	11
	F3	FBN1	0.915247	0.281020	True	11
		SOST	0.923112	0.345415	True	11
		TIMP1	0.937230	0.488505	True	11
Chemokines	F1	LOXL2	0.930358	0.451421	True	10
		JAK2	0.897331	0.204749	True	10
	F2	LOXL2	0.874630	0.088876	True	11
		JAK2	0.960025	0.772006	True	11
	F3	LOXL2	0.943678	0.564652	True	11
		JAK2	0.878406	0.099301	True	11
Cytokines	F1	RUNX2	0.947142	0.634825	True	10
		STAT3	0.933422	0.482382	True	10
		IL6R	0.927258	0.421472	True	10
		TNFSF13	0.907481	0.264130	True	10
	F2	RUNX2	0.915611	0.283765	True	11
		STAT3	0.907354	0.226836	True	11
		IL6R	0.985709	0.989621	True	11
		TNFSF13	0.958855	0.757330	True	11
	F3	RUNX2	0.924060	0.353917	True	11
		STAT3	0.932663	0.438418	True	11
		IL6R	0.826181	0.020798	False	11
		TNFSF13	0.970421	0.890746	True	11
ECM & Adhesion	F1	IFNG	0.715267	0.001349	False	10
		FZD4	0.981633	0.973303	True	10
		PTCH1	0.911578	0.292008	True	10
		CCL5	0.969121	0.882582	True	10
	F2	IFNG	0.899109	0.180269	True	11
		FZD4	0.979590	0.963841	True	11
		PTCH1	0.986610	0.990734	True	10
		CCL5	0.925780	0.407685	True	10
	F3	IFNG	0.905665	0.216509	True	11
		FZD4	0.923819	0.351743	True	11
		PTCH1	0.957827	0.744318	True	11
		CCL5	0.940093	0.521596	True	11
MMPs	F1	MMP7	0.955749	0.752957	True	9
		MMP9	0.675286	0.005186	False	5
	F2	MMP7	0.926078	0.372552	True	11

```
                     MMP9      0.971128  0.901100    True  10
               F3    MMP7      0.924886  0.361455    True  11
                     MMP9      0.913554  0.268549    True  11
  Signaling        F1 Vimentin  0.919696  0.354424    True  10
                     TNC       0.928589  0.434161    True  10
                     NOTCH1    0.922084  0.374662    True  10
                     WNT5A     0.903581  0.239742    True  10
               F2    Vimentin  0.957763  0.743507    True  11
                     TNC       0.959813  0.769352    True  11
                     NOTCH1    0.977556  0.951045    True  11
                     WNT5A     0.937156  0.487661    True  11
               F3    Vimentin  0.910924  0.250109    True  11
                     TNC       0.884194  0.117578    True  11
                     NOTCH1    0.779982  0.005132   False  11
                     WNT5A     0.812114  0.013581   False  11
```

[ ]:
```
DA.check_sphericity()
```

[ ]:
```
                                spher         W       chi2  dof      pval  \
class           fraction
Bone Metabolism F1       0   True  0.592922   3.658847    2  0.160506
                F2       0   True  0.703252   3.168356    2  0.205116
                F3       0   True  0.832864   1.645964    2  0.439120
Chemokines      F1       0   True       NaN        NaN    1  1.000000
                F2       0   True       NaN        NaN    1  1.000000
                F3       0   True       NaN        NaN    1  1.000000
Cytokines       F1       0   True  0.629185   3.577934    5  0.614197
                F2       0  False  0.262747  11.657816    5  0.040987
                F3       0  False  0.210032  13.610980    5  0.019012
ECM & Adhesion  F1       0   True  0.486690   5.560987    5  0.354712
                F2       0   True  0.295164   8.202615    5  0.149255
                F3       0   True  0.297080  10.586623    5  0.061736
MMPs            F1       0   True       NaN        NaN    1  1.000000
                F2       0   True       NaN        NaN    1  1.000000
                F3       0   True       NaN        NaN    1  1.000000
Signaling       F1       0   True  0.536227   4.812474    5  0.442437
                F2       0   True  0.554009   5.151113    5  0.400336
                F3       0  False  0.117602  18.669462    5  0.002375

                                group count       n per group
class           fraction
Bone Metabolism F1       0            3       [10, 10, 9]
                F2       0            3      [11, 11, 11]
                F3       0            3      [11, 11, 11]
Chemokines      F1       0            2          [10, 10]
                F2       0            2          [11, 11]
                F3       0            2          [11, 11]
```

```
Cytokines       F1       0            4  [10, 10, 10, 10]
                F2       0            4  [11, 11, 11, 11]
                F3       0            4  [11, 11, 11, 11]
ECM & Adhesion  F1       0            4  [10, 10, 10, 10]
                F2       0            4  [10, 11, 11, 10]
                F3       0            4  [11, 11, 11, 11]
MMPs            F1       0            2           [9, 5]
                F2       0            2         [11, 10]
                F3       0            2         [11, 11]
Signaling       F1       0            4  [10, 10, 10, 10]
                F2       0            4  [11, 11, 11, 11]
                F3       0            4  [11, 11, 11, 11]
```

[ ]:
```
# Default is (paired) t-test, and since DA has subject: paired=True
DA.test_pairwise()
```

[ ]:
```
                                        gene     A      B   mean(A)  \
class          fraction Contrast
ECM & Adhesion -        gene               -  CCL5   FZD4  0.591713
                        gene               -  CCL5   IFNG  0.591713
                        gene               -  CCL5  PTCH1  0.591713
                        gene               -  FZD4   IFNG  0.622994
                        gene               -  FZD4  PTCH1  0.622994
…                                        …  …    …       …
MMPs           NaN      gene * fraction  MMP9    F1     F3  0.256111
                        gene * fraction  MMP9    F2     F3  0.677357
               F1       fraction * gene   NaN  MMP7   MMP9  0.032549
               F2       fraction * gene   NaN  MMP7   MMP9  0.185211
               F3       fraction * gene   NaN  MMP7   MMP9  0.742060

                                          std(A)   mean(B)    std(B) Paired  \
class          fraction Contrast
ECM & Adhesion -        gene             0.253752  0.622994  0.266747   True
                        gene             0.253752 -0.026656  0.149430   True
                        gene             0.253752  0.469495  0.330886   True
                        gene             0.266747 -0.026656  0.149430   True
                        gene             0.266747  0.469495  0.330886   True
…                                           …        …        …    …
MMPs           NaN      gene * fraction  0.802159  1.845550  0.600687   True
                        gene * fraction  0.546148  1.845550  0.600687   True
               F1       fraction * gene  0.228544  0.256111  0.802159   True
               F2       fraction * gene  0.361750  0.677357  0.546148   True
               F3       fraction * gene  0.567249  1.845550  0.600687   True

                                        Parametric          T   dof  \
class          fraction Contrast
ECM & Adhesion -        gene                 True  -0.327586  10.0
```

```
                        gene                  True   7.882620  10.0
                        gene                  True   1.320783  10.0
                        gene                  True   7.532512  10.0
                        gene                  True   2.105924  10.0
…                                            …        …   …
MMPs            NaN     gene * fraction       True  -3.513968   4.0
                        gene * fraction       True  -5.680475   9.0
                F1      fraction * gene       True  -0.543884   4.0
                F2      fraction * gene       True  -3.811156   9.0
                F3      fraction * gene       True -15.767066  10.0

                                          alternative         p-unc       BF10  \
class           fraction Contrast
ECM & Adhesion  -        gene               two-sided  7.499799e-01      0.312
                         gene               two-sided  1.339935e-05   1643.947
                         gene               two-sided  2.160003e-01      0.598
                         gene               two-sided  1.987311e-05   1165.781
                         gene               two-sided  6.146203e-02      1.461
…                                              …          …        …
MMPs            NaN      gene * fraction    two-sided  2.458360e-02      3.686
                         gene * fraction    two-sided  3.016844e-04    111.751
                F1       fraction * gene    two-sided  6.154168e-01      0.448
                F2       fraction * gene    two-sided  4.145762e-03     12.636
                F3       fraction * gene    two-sided  2.163081e-08  4.845e+05

                                            hedges **p-unc    Sign.  \
class           fraction Contrast
ECM & Adhesion  -        gene             -0.115597      ns    False
                         gene              2.856874    ****  signif.
                         gene              0.398765      ns    False
                         gene              2.890772    ****  signif.
                         gene              0.491362   0.061   toler.
…                                             …     …      …
MMPs            NaN      gene * fraction -2.179426       *  signif.
                         gene * fraction -2.024511     ***  signif.
                F1       fraction * gene -0.255634      ns    False
                F2       fraction * gene -0.939861      **  signif.
                F3       fraction * gene -1.817138    ****  signif.

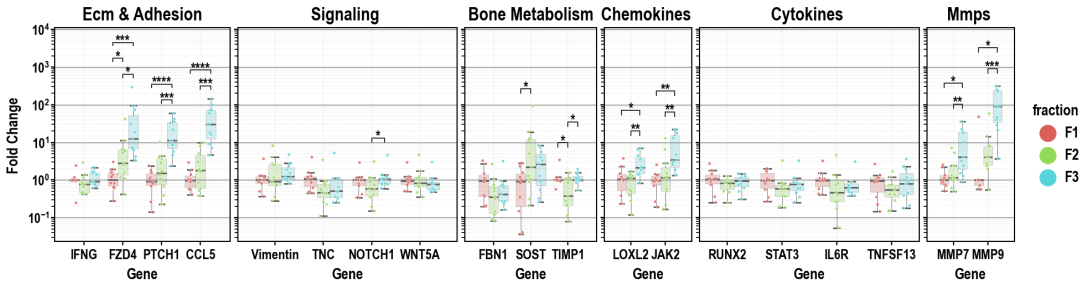
                                                          pairs cross
class           fraction Contrast
ECM & Adhesion  -        gene                        (CCL5, FZD4)     x
                         gene                        (CCL5, IFNG)     x
                         gene                       (CCL5, PTCH1)     x
                         gene                        (FZD4, IFNG)     x
                         gene                       (FZD4, PTCH1)     x
…                                                          …   …
```

```
MMPs            NaN      gene * fraction  ((MMP9, F3), (MMP9, F1))   hue
                         gene * fraction  ((MMP9, F3), (MMP9, F2))   hue
                F1       fraction * gene  ((MMP9, F1), (MMP7, F1))     x
                F2       fraction * gene  ((MMP9, F2), (MMP7, F2))     x
                F3       fraction * gene  ((MMP9, F3), (MMP7, F3))     x

[167 rows x 19 columns]
```

[ ]:
```python
# Plot
(
    DA.switch("row", "col", verbose=False)
    .set(y="fc", inplace=False)  # set y back to to fc to display non-log values
    .plot_box_strip(
        subplot_kws=dict(
            figsize=(10, 2.5),
            width_ratios=[4, 5, 3, 2, 5, 2],
        ),
        strip_kws=dict(alpha=0.8),
    )
    .edit_grid()
    .edit_y_scale_log(10)
    .edit_xy_axis_labels(y_leftmost_col="Fold Change", x="Gene")
    .annotate_pairwise(include="__HUE")
)

plt.savefig("qpcr1.png", dpi=300, bbox_inches="tight")
```



C Submission Forms & Documents

C.1 Author Contributions

Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Graduate School
Life Sciences

Statement of individual author contributions and of legal second publication rights to manuscripts included in the dissertation

Manuskript 1: Research Article (submitted, under revision)					
Martin Kuric (MK), Susanne Beck, Doris Schneider, Wyonna Rindt, Marietheres Evers, Jutta Meißner-Weigl, Sabine Zeck, Melanie Krug, Marietta Herrmann, Tanja Nicole Hartmann, Ellen Leich, Maximilian Rudert, Denitsa Docheva, Anja Seckinger, Dirk Hose, Franziska Jundt, Regina Ebert (RE) (2024): Keep it Together: Describing Myeloma Dissemination *in vitro* with hMSC-Interacting Subpopulations and their Aggregation/Detachment Dynamics, **Cancer Research Communications**					
Participated in	**Author Initials,** Responsibility decreasing from left to right				
Study Design	MK	Regina Ebert	Wyonna Rindt		
Methods Development	MK	Doris Schneider			
Data Collection	MK	Doris Schneider			
Data Analysis and Interpretation	MK	Susanne Beck	Regina Ebert		
Manuscript Writing Writing of Introduction Writing of Materials & Methods Writing of Discussion Writing of First Draft	MK	Regina Ebert			

Explanations: The content of this publication exceeds the usual scope (~29 pages Supplemental). It includes not only research findings but also survival data and protocols of new, established methods and their validations. The contribution of Martin Kuric was pivotal and predominant in all aspects of this work. Doris Schneider assisted in the experimental procedures. Susanne Beck analyzed the raw data from RNAseq and survival data, which were interpreted, depicted, and summarized by Martin Kuric.

Manuscript 2: Data Analysis Software (submitted, passed peer-review, under revision)					
Martin Kuric (MK), Regina Ebert (2024): plotastic: Bridging Plotting and Statistics in Python, **Journal of Open Source Software**					
Participated in	**Author Initials,** Responsibility decreasing from left to right				
Idea, Architectural Design	MK				
Software Development Feature Implementation Testing	MK				
Distribution of Software Documentation Version Control (GitHub) Deployment (PyPi)	MK				
Manuscript Writing Writing of Statement of Need Writing of Example Writing of Overview	MK	Regina Ebert			

Explanations: The software was entirely created by Martin Kuric, comprising more than 8000 total lines (including ~2000 testable lines) and is comparable in size to a typical web application. The release of this software involved version control using GitHub, packaging and deployment on PyPi. Regina Ebert gave feedback on submitted manuscript.

Manuscript 3: Research Letter (published)

Daniela Simone Maichl, Julius Arthur Kirner, Susanne Beck, Wen-Hui Cheng, Melanie Krug, Martin Kuric (MK), Carsten Patrick Ade, Thorsten Bischler, Franz Jakob, Dirk Hose, Anja Seckinger, Regina Ebert & Franziska Jundt (2023): Identification of NOTCH-driven matrisome-associated genes as prognostic indicators of multiple myeloma patient survival, **Blood Cancer Journal 13:134**

Participated in	**Author Initials,** Responsibility decreasing from left to right				
Study Design Methods Development	Daniela Simone		Franziska Jundt		
Data Collection	Daniela Simone		Franziska Jundt		
Data Analysis and Interpretation	Daniela Simone	Susanne Beck	Franziska Jundt	MK	
Manuscript Writing Writing of Introduction Writing of Materials & Methods Writing of Discussion Writing of First Draft	Daniela Simone		Franziska Jundt	MK	

Explanations: This co-authorship is not a chapter in this dissertation. Martin Kuric produced figures of processed but complex-to-visualize data and gave feedback on submitted manuscript.

Manuscript 4: Research Paper (under peer-review)

Wyonna Rindt, Melanie Krug, Shuntaro Yamada, Franziska Sennefelder, Louisa Belz, Wen-Hui Cheng, Azeem Muhammad, Martin Kuric (MK), Marietheres Evers, Ellen Leich, Tanja Nicole Hartmann, Ana Rita Pereira, Marietta Hermann, Jan Hansmann, Mohammed Ahmed Yassin, Kamal Mustafa, Regina Ebert, and Franziska Jundt (2024): A 3D bioreactor model to study osteocyte differentiation and mechanobiology under perfusion and compressive mechanical loading, **Acta Biomaterialia**

Participated in	**Author Initials,** Responsibility decreasing from left to right				
Study Design Methods Development	Wyonna Rindt	Franziska Jundt		MK	
Data Collection	Wyonna Rindt	Franziska Jundt		MK	
Data Analysis and Interpretation	Wyonna Rindt	Franziska Jundt		MK	
Manuscript Writing Writing of Introduction Writing of Materials & Methods Writing of Discussion Writing of First Draft	Wyonna Rindt	Franziska Jundt		MK	

Explanations: This co-authorship is not a chapter in this dissertation. Martin Kuric contributed by counseling during weekly meetings in tight collaboration with Franziska Jundt's group, assisting Wyonna Rindt during laboratory experiments, image analysis and giving feedback on submitted manuscript.

Manuscript 5: Research Paper (under revision)

Marietta Herrmann, Jutta Schneidereit, Susanne Wiesner, Martin Kuric (MK), Maximilian Rudert, Martin Lüdemann, Mugdha Srivastava, Norbert Schütze, Regina Ebert, Denitsa Docheva, Franz Jakob (2024): Peripheral blood cells enriched by adhesion to CYR61 are heterogenous myeloid modulators of tissue regeneration with early endothelial progenitor characteristics, **European Cells and Materials**

Participated in	**Author Initials,** Responsibility decreasing from left to right				
Study Design Methods Development	Marietta Herrmann				
Data Collection	Marietta Herrmann			MK	

Data Analysis and Interpretation	Marietta Herrmann				
Manuscript Writing Writing of Introduction Writing of Materials & Methods Writing of Discussion Writing of First Draft	Marietta Herrmann			MK	

Explanations: This co-authorship is not a chapter in this dissertation. Martin Kuric contributed by establishing and measuring large automated microscopy scans of stained cells for quantifying osteogenic differentiation and giving feedback on submitted manuscript.

Manuscript 6: Research Letter (published)

Marietheres Evers, Martin Schreder, Thorsten Stühmer, Franziska Jundt, Regina Ebert, Tanja Nicole Hartmann, Michael Altenbuchinger, Martina Rudelius, Martin Kuric (MK), Wyonna Darleen Rindt, Torsten Steinbrunn, Christian Langer, Sofia Catalina Heredia-Guerrero, Hermann Einsele, Ralf Christian Bargou, Andreas Rosenwald, Ellen Leich (2023): Prognostic value of extracellular matrix gene mutations and expression in multiple myeloma, **Blood Cancer J. 13(1):43**

Participated in	**Author Initials,** Responsibility decreasing from left to right				
Study Design Methods Development	Marietheres Evers				
Data Collection	Marietheres Evers				
Data Analysis and Interpretation	Marietheres Evers			MK	
Manuscript Writing Writing of Introduction Writing of Materials & Methods Writing of Discussion Writing of First Draft	Marietheres Evers			MK	

Explanations: This co-authorship is not a chapter in this dissertation. Martin Kuric contributed by counseling during regular meetings with Ellen Leich's group and giving feedback on submitted manuscript.

If applicable, the doctoral researcher confirms that she/he has obtained permission from both the publishers (copyright) and the co-authors for legal second publication.

The doctoral researcher and the primary supervisor confirm the correctness of the above mentioned assessment.

		Würzburg	
Doctoral Researcher's Name	Date	Place	Signature

		Würzburg	
Primary Supervisor's Name	Date	Place	Signature

Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Graduate School
Life Sciences

Statement of individual author contributions to figures/tables of manuscripts included in the dissertation

Manuskript 1: Research Article (submitted, under revision)

Martin Kuric (MK), Susanne Beck, Doris Schneider, Wyonna Rindt, Marietheres Evers, Jutta Meißner-Weigl, Sabine Zeck, Melanie Krug, Marietta Herrmann, Tanja Nicole Hartmann, Ellen Leich, Maximilian Rudert, Denitsa Docheva, Anja Seckinger, Dirk Hose, Franziska Jundt, Regina Ebert1 (2024): Keep it Together: Describing Myeloma Dissemination *in vitro* with hMSC-Interacting Subpopulations and their Aggregation/Detachment Dynamics, **Cancer Research Communications**

Figure	**Author Initials,** Responsibility decreasing from left to right				
1	MK	Doris Schneider			
2	MK	Doris Schneider			
3	MK	Doris Schneider	Sabine Zeck	Wyonna Rindt	Melanie Krug
4	MK	Doris Schneider	Susanne Beck		
5	MK	Susanne Beck			
6	MK	Susanne Beck			
7	MK				
S1	MK	Doris Schneider	Sabine Zeck	Wyonna Rindt	Melanie Krug
S2	MK	Doris Schneider	Marietta Herrmann		
S3	MK	Doris Schneider	Sabine Zeck		
S4	MK				
S5	MK				
S6	MK	Susanne Beck			
Table	**Author Initials,** Responsibility decreasing from left to right				
1	MK	Susanne Beck			
2	MK	Susanne Beck			
S1	MK	Doris Schneider			
S2	MK	Susanne Beck			
S3	MK				
S4	MK	Doris Schneider			

Manuscript 2: Data Analysis Software (submitted, passed peer-review, under revision)

Martin Kuric, Regina Ebert (2024): plotastic: Bridging Plotting and Statistics in Python, **Journal of Open Source Software**

Figure	**Author Initials,** Responsibility decreasing from left to right				
1	MK				
Table	**Author Initials,** Responsibility decreasing from left to right				
1	MK				
2	MK				

Documentation	**Author Initials,** Responsibility decreasing from left to right				
README	MK				
Example Gallery	MK				
Features	MK				
Testing	**Author Initials,** Responsibility decreasing from left to right				
Test-Code (Pytest)	MK				
Continuous Integration	MK				

Explanations: All files are available on GitHub (https://github.com/markur4/plotastic) and installable via pypi.com. Documentations are found in the Readme, including example gallery and feature explanation. Software tests was written using pytest. Coverage of code by tests is reviewable with codecov (https://app.codecov.io/gh/markur4/plotastic). Continuous Integration is implemented using GitHub actions.

Manuscript 3: Research Letter (published)

Daniela Simone Maichl, Julius Arthur Kirner, Susanne Beck, Wen-Hui Cheng, Melanie Krug, Martin Kuric, Carsten Patrick Ade, Thorsten Bischler, Franz Jakob, Dirk Hose, Anja Seckinger, Regina Ebert & Franziska Jundt (2023): Identification of NOTCH-driven matrisome-associated genes as prognostic indicators of multiple myeloma patient survival, **Blood Cancer Journal 13:134**

Figure	**Author Initials,** Responsibility decreasing from left to right				
1 a	Daniela Simone			Susanne Beck	
1 b	Daniela Simone			Susanne Beck	
1 c	Daniela Simone			Susanne Beck	MK
1 d	Daniela Simone			Susanne Beck	MK
Table	**Author Initials,** Responsibility decreasing from left to right				
1	Daniela Simone				

Explanations: Martin Kuric plotted multidimensional diagrams using python and fine-adjusted them using professional design software (Affinity Publisher, Serif Ltd).

Manuscript 4: Research Paper (under peer-review)

Wyonna Rindt, Melanie Krug, Shuntaro Yamada, Franziska Sennefelder, Louisa Belz, Wen-Hui Cheng, Azeem Muhammad, Martin Kuric (MK), Marietheres Evers, Ellen Leich, Tanja Nicole Hartmann, Ana Rita Pereira, Marietta Hermann, Jan Hansmann, Mohammed Ahmed Yassin, Kamal Mustafa, Regina Ebert, and Franziska Jundt (2024): A 3D bioreactor model to study osteocyte differentiation and mechanobiology under perfusion and compressive mechanical loading, **Acta Biomaterialia**

Figure	**Author Initials,** Responsibility decreasing from left to right				
1	Wyonna Rindt				MK
2	Wyonna Rindt				
3	Wyonna Rindt				
4	Wyonna Rindt				
5	Wyonna Rindt				MK
6	Wyonna Rindt				MK
7	Wyonna Rindt				MK

Explanations: Martin Kuric contributed by counseling on experimental procedures and data analysis, such as quantifying normalized fluorescence intensity of immunohistochemistry and qPCR.

Manuscript 5: Research Paper (under revision)

Marietta Herrmann, Jutta Schneidereit, Susanne Wiesner, Martin Kuric (MK), Maximilian Rudert, Martin Lüdemann, Mugdha Srivastava, Norbert Schütze, Regina Ebert, Denitsa Docheva, Franz Jakob (2024): Peripheral blood cells enriched by adhesion to CYR61 are heterogenous myeloid modulators of tissue regeneration with early endothelial progenitor characteristics, **European Cells and Materials**

Figure	**Author Initials,** Responsibility decreasing from left to right				
1	Marietta Herrmann				
2	Marietta Herrmann				
3	Marietta Herrmann				
4	Marietta Herrmann				
5	Marietta Herrmann				
6	Marietta Herrmann				
7	Marietta Herrmann				MK

Explanations: Martin Kuric scanned osteogenically differentiated MSCs in Fig. 7 for quantification of alizarin red staining.

Manuscript 6: Research Letter (published)

Marietheres Evers, Martin Schreder, Thorsten Stühmer, Franziska Jundt, Regina Ebert, Tanja Nicole Hartmann, Michael Altenbuchinger, Martina Rudelius, Martin Kuric (MK), Wyonna Darleen Rindt, Torsten Steinbrunn, Christian Langer, Sofia Catalina Heredia-Guerrero, Hermann Einsele, Ralf Christian Bargou, Andreas Rosenwald, Ellen Leich (2023): Prognostic value of extracellular matrix gene mutations and expression in multiple myeloma, **Blood Cancer J. 13(1):43**

Figure	**Author Initials,** Responsibility decreasing from left to right				
1	Marietheres Evers				
2	Marietheres Evers				

Explanations: Martin Kuric contributed indirectly through counseling and feedback on submitted manuscript.

I also confirm my primary supervisor's acceptance.

Doctoral Researcher's Name Date Place Signature

C.2 Affidavit

Affidavit

I hereby confirm that my thesis entitled "Development and Semi-Automated Analysis of an in vitro Model for Myeloma Cells Interacting with Mesenchymal Stromal Cells" is the result of my own work. I did not receive any help or support from commercial consultants. All sources and / or materials applied are listed and specified in the thesis.

Furthermore, I confirm that this thesis has not yet been submitted as part of another examination process neither in identical nor in similar form.

Würzburg
Place, Date Signature

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides statt, die Dissertation "Entwicklung und semi-automatisierte Analyse eines in vitro-Modells für Myelomzellen in Interaktion mit mesenchymalen Stromazellen" eigenständig, d.h. insbesondere selbständig und ohne Hilfe eines kommerziellen Promotionsberaters, angefertigt und keine anderen als die von mir angegebenen Quellen und Hilfsmittel verwendet zu haben.

Ich erkläre außerdem, dass die Dissertation weder in gleicher noch in ähnlicher Form bereits in einem anderen Prüfungsverfahren vorgelegen hat.

Würzburg
Ort, Datum Unterschrift

C.3 Curriculum Vitae

MARTIN KURIC

Cell Biologist | Data Scientist

97072 Würzburg
+49 15773816443
martin.kur4@gmail.com
martin-kuric
github.com/markur4

WHO AM I?

As a cancer cell biologist with a strong passion for data analysis and machine learning, I am seeking a position where I can utilize my creativity to automate tasks, solve complex problems or handle big data.

Microscopy	100%	Python	100%
Cancer Biology	90%	Image Analysis	90%
Stem Cells	80%	Statistics	80%
RNA Biochemistry	75%	ML/AI	75%

SELECTED PROJECTS

2024 **Python Software *"ImageP"*** **GitHub Repository**
Accelerates batch processing of images of different sizes and types by >100%.
`numpy / skimage / scipy`

2020-2024 **Python Software *"plotastic"*** **Journal of Open Source Software | GitHub Repository**
Published a statistical library that self-configures based on intuitive plotting parameters.
`pandas / matplotlib / pingouin / seaborn`

2018-2024 **Cancer Research Project** **Journal: Cancer Research Communications**
Worked in a team with up to three technical assistants and published a list of genes with relevance for survival of myeloma patients (under peer-review).
`Time-Lapse Microscopy / RNAseq / Analysis of Patient Survival`

26.05.2022 **Deep-Learning Assisted Image Cytometry** **Poster at "Achilles Conference"**
Measurement of per-cell parameters from large automated microscopy scans.
`Convolutional Neural Networks / Image Segmentation`

EDUCATION

28.01.2019 – 2024 **Dr. rer. nat. in Biomedicine** **Prof. Dr. Regina Ebert | University of Würzburg**
Research focus: Dissemination of multiple myeloma & mesenchymal stromal cell interactions

01.04.2017 – 2024 (parallel to M.Sc. & PhD) **Elite Biological Physics** **University of Bayreuth**
Interdisciplinary & international study program for exceptional students of physics or biology.

01.10.15 – 15.08.18 **M.Sc. in Biochemistry & Molecular Biology** **Prof. Dr. Claus-D. Kuhn | University of Bayreuth**
Research focus: RNA biochemistry, small RNAseq, stem cells & piRNAs in *S. mediterranea*

01.10.12 – 14.12.15 **B.Sc. in Biochemistry** **Prof. Dr. Benedikt Westermann | University of Bayreuth**
Research focus: Cell biology, mitochondrial inheritance in *S. cerevisiae*

LANGUAGES

German, English - C2
Slovakian - passive
French, Spanish - A2

SOFT SKILLS

Quality Management
Project Management
Violent Free Communication

HOBBIES

Coding - Python
Music - Piano & Guitar
Gym - Lift. Grow. Repeat

Würzburg (Location) 05.03.2024 (Date) Martin Kuric (Signature)