Novel bioinformatic tools and methods to study Next Generation Sequencing data with a focus on DNA repair and genome stability

Dissertation
Zur Erlangung des Grades
Doktor der Naturwissenschaften

Am Fachbereich Biologie
Der Johannes Gutenberg-Universität Mainz

Sergi Sayols Puig
geb. am 08.08.1980 in Cassa de la Selva, Spain

Mainz, 2024

Dekan:
1. Berichterstatter:
2. Berichterstatter:
Tag der mündlichen Prüfung: Dienstag, den 15.07.2025

Johannes Gutenberg University of Mainz
Faculty: Fachbereich Biologie
Chair or Institute: Institute of Organismic and Molecular Evolution
1st Supervisor and Reviewer:
2nd Supervisor:

Novel bioinformatic tools and methods to study Next Generation Sequencing data with a focus on DNA repair and genome stability

In Partial Fulfillment
of the Requirements for the Degree
Doktor der Naturwissenschaften

Submitted by: **Sergi Sayols Puig**
Email: sergisayolspuig@imb-mainz.de

Mainz, December 2024

Preface

Jurassic Park inspired more people to go into biotech than any academic paper. The Matrix inspired more people to go into computer science than any Github repo. The Martian inspired more people to go into aerospace engineering than any industry trend report.

Science fiction doesn't predict the future, it does something much more interesting: tell stories about technology so compelling that people dedicate their lives to advancing the frontier.

Eliot Peper

Abstract

Next Generation Sequencing is a widely used technology that enables precise identification and quantification of nucleic acids. Advanced sequencing-based experimental protocols have enabled the investigation of their modifications, organization, interaction, and regulation, among others. This thesis introduces three novel methodologies implemented as software packages for facilitating the comprehensive analysis, visualization and interpretation of *omics* sequencing data.

In *Chapter 1* we describe the problem of PCR clonal artefacts in RNA-seq and enrichment-based assays, such as ChIP-seq. We present the tool *dupRadar*, a novel method to tell apart those PCR artifacts from normal read duplication due to natural over-sequencing of highly expressed genes or enriched loci. We apply our method to detect over-sequenced libraries of limited complexity in cases of little input material in a synthetic dataset and also in public datasets of bulk RNA-seq and single-cell RNA-seq. We found that datasets generated from lower input material exhibit limited library complexity, leading to increased duplication rates even among lowly expressed genes. Finally, we run differential expression analysis to demonstrate that even low levels of PCR artifacts can have an influence on downstream analysis and data interpretation.

Chapter 2 introduces *rrvgo*, a novel tool for interpreting large lists of Gene Ontology terms. The package gives access to several semantic similarity methods; here, I apply the *Relevance* method to GO terms significantly enriched in the publicly available gene expression data from the breast cancer study published by Schmidt et al. in 2008, comparing grade III to grade I breast cancer patients. This approach identifies clusters of potentially redundant terms with high correlation of information content within the set of GO terms. I further demonstrate the utility of rrvgo's visualizations, which facilitate the detection and refinement of a non-redundant set of GO terms for more focused biological interpretation.

Chapter 3 introduces *BreakTag*, an innovative approach for genome-wide identification and quantification of DNA double-strand breaks and their structural characteristics at single nucleotide resolution using high-throughput sequencing. Additionally, we developed *breakinspectoR*, a bioinformatics pipeline designed to detect, quantify and study the end structure of Cas9-induced DSBs in BreakTag data. Using BreakTag, we analyzed cleavage patterns by SpCas9 across three genome-wide CRISPR libraries, comprising 3,500 distinct single-guide RNAs, and identified over 150,000 on- and off-target cleavage sites. Analysis of DSB break ends revealed that approximately 35% of the identified breaks exhibit staggered ends. A machine learning model trained using target site sequence composition and DSB end structure data revealed that protospacer sequence significantly influences Cas9 incision patterns. Furthermore, by examining matched datasets of Cas9 cleavage sites and subsequent

repair outcomes, we found a link between staggered breaks and single-nucleotide insertions. In conclusion, these findings demonstrate that the structure of Cas9 DSB ends is sequence-dependent, suggesting that guide RNAs can be strategically designed to produce precise, predictable repair outcomes. This approach may provide new opportunities for correcting diseases caused by single-nucleotide deletions.

Overall during my PhD, in collaboration with wet-lab researchers, I have developed novel tools and methods to a broad range of applications of *omics* sequencing data, with special focus on the study of DNA repair and genome stability.

Statement of contributons

In the project included in *Chapter 1*, [redacted] and I conceived the project, designed and tested the software. [redacted], [redacted] and I implemented the software, designed and wrote the vignette. [redacted] and I drafted the manuscript.

In the project included in *Chapter 2*, I conceived the project, designed and implemented the software. I wrote the vignette and set up the the webservice running the interactive web app. I wrote the manuscript.

The first authorship of the publication in *Chapter 3* is shared between [redacted] and myself. [redacted], [redacted] and I conceived and designed the study. [redacted] conceived BreakTag and performed all BreakTag, AmpliSeq and other experiments in the lab. I designed and implemented the BreakTag pipeline for processing BreakTag data and the breakinspectoR package for nominating, quantifying and studying end structure of Cas9-induced DSBs in BreakTag data. I wrote the vignette and manuals for the software. I processed the high-throughput data, and Gabe and I performed the bioinformatics analyses and prepared the plots and figures included in the publication. [redacted] performed the transduction of gRNA-target lentiviral pools in the Cas9-expressing cells and library preparation for amplicon sequencing. [redacted] and [redacted] cloned and produced the recombinant engineered Cas9 variants, and [redacted] provided expertise. [redacted] and [redacted] wrote the original draft with input from all authors.

List of publications

List of publications included in this thesis as separate chapters:

- **dupRadar (chapter 1):** Sayols, S., [redacted], & [redacted] (2016). dupRadar: a Bioconductor package for the assessment of PCR artifacts in RNA-Seq data. *BMC Bioinformatics*, 17(1), 428. doi:10.1186/s12859-016-1276-2
- **rrvgo (chapter 2):** Sayols, S. (2023). rrvgo: a Bioconductor package for interpreting lists of Gene Ontology terms. *microPublication Biology*, 2023. doi:10.17912/micropub.biology.000811
- **BreakTag (chapter 3):** [redacted]*, Sayols, S.*, [redacted], [redacted], [redacted], [redacted], & [redacted] (2024). Linking CRISPR-Cas9 double-strand break profiles to gene editing precision with BreakTag. *Nature Biotechnology*. doi:10.1038/s41587-024-02238-8

Other publications:

- Longo, G.M.C.*, Sayols, S.*, Stefanova, M.E.*, Xie, T.*, Elsayed, W.*, ... Roukos, V. (2024). Type II topoisomerases shape multi-scale 3D chromatin folding in regions of positive supercoils. *Molecular Cell* [Preprint]. doi:10.1016/j.molcel.2024.10.007.
- García-Valverde, A., Rosell, J., Sayols, S., Gómez-Peregrina, D., Pilco-Janeta, D. F., Olivares-Rivas, I., ... Serrano, C. (2021). E3 ubiquitin ligase Atrogin-1 mediates adaptive resistance to KIT-targeted inhibition in gastrointestinal stromal tumor. *Oncogene*, 40(48), 6614–6626. doi:10.1038/s41388-021-02049-0
- Hernandez-Meza, G., von Felden, J., Gonzalez-Kozlova, E. E., Garcia-Lezana, T., Peix, J., Portela, A., Craig, A.J., Sayols, S., ... Villanueva, A. (2021). DNA methylation profiling of human hepatocarcinogenesis. *Hepatology (Baltimore, Md.)*, 74(1), 183–199. doi:10.1002/hep.31659
- Bouwman, B. A. M., Agostini, F., Garnerone, S., Petrosino, G., Gothe, H. J.,Sayols, S., ... Crosetto, N. (2020). Genome-wide detection of DNA double-strand breaks by in-suspension BLISS. *Nature Protocols*, 15(12), 3894–3941. doi:10.1038/s41596-020-0397-2
- Casas-Vila, N., Sayols, S., Pérez-Martínez, L., Scheibe, M., & Butter, F. (2020). The RNA fold interactome of evolutionary conserved RNA structures in S. cerevisiae. *Nature Communications*, 11(1), 2789. doi:10.1038/s41467-020-16555-4
- Strobel, B., Spöring, M., Klein, H., Blazevic, D., Rust, W., Sayols, S., ... Kreuz, S. (2020). High-throughput identification of synthetic riboswitches by

barcode-free amplicon-sequencing in human cells. *Nature Communications*, 11(1), 714. doi:10.1038/s41467-020-14491-x

- Gothe, H. J., Bouwman, B. A. M., Gusmao, E. G., Piccinno, R., Petrosino, G., Sayols, S., ... Roukos, V. (2019). Spatial chromosome folding and active transcription drive DNA fragility and formation of oncogenic MLL translocations. *Molecular Cell*, 75(2), 267-283.e12. doi:10.1016/j.molcel.2019.05.015
- Becker, K., Bluhm, A., Casas-Vila, N., Dinges, N., Dejung, M., Sayols, S., ... Legewie, S. (2018). Quantifying post-transcriptional regulation in the development of Drosophila melanogaster. *Nature Communications*, 9(1), 4970. doi:10.1038/s41467-018-07455-9
- Kaymak, A., Sayols, S., Papadopoulou, T., & Richly, H. (2018). Role for the transcriptional activator ZRF1 in early metastatic events in breast cancer progression and endocrine resistance. *Oncotarget*, 9(47), 28666–28690. doi:10.18632/oncotarget.25596
- Sanchez-Mut, Jose V., Heyn, H., Silva, B. A., Dixsaut, L., Garcia-Esparcia, P., Vidal, E., Sayols, S., ... Gräff, J. (2018). PM20D1 is a quantitative trait locus associated with Alzheimer's disease. *Nature Medicine*, 24(5), 598–603. doi:10.1038/s41591-018-0013-y
- Tristán-Flores, F. E., Guzmán, P., Ortega-Kermedy, M. S., Cruz-Torres, G., de la Rocha, C., Silva-Martínez, G. A., Rodríguez-Ríos, D., Alvarado-Caudillo, Y., Barbosa-Sabanero, G., Sayols, S., ... Zaina, S. (2018). Liver X receptor-binding DNA motif associated with atherosclerosis-specific DNA methylation profiles of Alu elements and neighboring CpG islands. *Journal of the American Heart Association*, 7(3). doi:10.1161/JAHA.117.007686
- Dzama, M. M., Nigmatullina, L., Sayols, S., Kreim, N., & Soshnikova, N. (2017). Distinct populations of embryonic epithelial progenitors generate Lgr5+ intestinal stem cells. *Developmental Biology*, 432(2), 258–264. doi:10.1016/j.ydbio.2017.10.012
- Berdasco, M., Gómez, A., Rubio, M. J., Català-Mora, J., Zanón-Moreno, V., Lopez, M., Hernández, C., Yoshida, S., Nakama, T., Ishikawa, K., Ishibashi, T., Boubekeur, A. M., Louhibi, L., Pujana, M. A., Sayols, S., ... Esteller, M. (2017). DNA methylomes reveal biological networks involved in human eye development, functions and associated disorders. *Scientific Reports*, 7(1), 11762. doi:10.1038/s41598-017-12084-1
- Vidal, E., Sayols, S., Moran, S., Guillaumet-Adkins, A., Schroeder, M. P., Royo, R., ... Esteller, M. (2017). A DNA methylation map of human cancer at single base-pair resolution. *Oncogene*, 36(40), 5648–5657. doi:10.1038/onc.2017.176
- Jahn, A., Rane, G., Paszkowski-Rogacz, M., Sayols, S., Bluhm, A., Han, C.-T., ... Kappei, D. (2017). ZBTB48 is both a vertebrate telomere-binding

protein and a transcriptional activator. *EMBO Reports*, 18(6), 929–946. doi:10.15252/embr.201744095

- Kazakevych, J., Sayols, S., Messner, B., Krienke, C., & Soshnikova, N. (2017). Dynamic changes in chromatin states during specification and differentiation of adult intestinal stem cells. *Nucleic Acids Research*, 45(10), 5770–5784. doi:10.1093/nar/gkx167
- Sanchez-Mut, Jose Vicente, Heyn, H., Vidal, E., Delgado-Morales, R., Moran, S., Sayols, S., ... Gräff, J. (2017). Whole genome grey and white matter DNA methylation profiles in dorsolateral prefrontal cortex. *Synapse (New York, N.Y.)*, 71(6). doi:10.1002/syn.21959
- Nigmatullina, L., Norkin, M., Dzama, M. M., Messner, B., Sayols, S., & Soshnikova, N. (2017). Id2 controls specification of Lgr5+ intestinal stem cell progenitors during gut development. *The EMBO Journal*, 36(7), 869–885. doi:10.15252/embj.201694959
- Papadopoulou, T., Kaymak, A., Sayols, S., & Richly, H. (2016). Dual role of Med12 in PRC1-dependent gene repression and ncRNA-mediated transcriptional activation. *Cell Cycle (Georgetown, Tex.)*, 15(11), 1479–1493. doi:10.1080/15384101.2016.1175797
- Heyn, H., Vidal, E., Ferreira, H. J., Vizoso, M., Sayols, S., Gomez, A., ... Esteller, M. (2016). Epigenomic analysis detects aberrant super-enhancer DNA methylation in human cancer. *Genome Biology*, 17(1), 11. doi:10.1186/s13059-016-0879-2
- Sanchez-Mut, J. V., Heyn, H., Vidal, E., Moran, S., Sayols, S., Delgado-Morales, R., ... Esteller, M. (2016). Human DNA methylomes of neurodegenerative diseases show common epigenomic patterns. *Translational Psychiatry*, 6(1), e718. doi:10.1038/tp.2015.214
- Boque-Sastre, R., Soler, M., Oliveira-Mateos, C., Portela, A., Moutinho, C., Sayols, S., ... Guil, S. (2015). Head-to-head antisense transcription and R-loop formation promotes transcriptional activation. *Proceedings of the National Academy of Sciences of the United States of America*, 112(18), 5785–5790. doi:10.1073/pnas.1421197112
- Valencia-Morales, M. del P., Zaina, S., Heyn, H., Carmona, F. J., Varol, N., Sayols, S., ... Esteller, M. (2015). The DNA methylation drift of the atherosclerotic aorta increases with lesion progression. *BMC Medical Genomics*, 8(1), 7. doi:10.1186/s12920-015-0085-1
- Blanco, I., Kuchenbaecker, K., Cuadras, D., Wang, X., Barrowdale, D., de Garibay, G. R., ... Sayols, S., ... Pujana, M. A. (2015). Assessing associations between the AURKA-HMMR-TPX2-TUBG1 functional module and breast cancer risk in BRCA1/2 mutation carriers. *PloS One*, 10(4), e0120020. doi:10.1371/journal.pone.0120020

- Cornella, H., Alsinet, C., Sayols, S., Zhang, Z., Hao, K., Cabellos, L., ... Llovet, J. M. (2015). Unique genomic profile of fibrolamellar hepatocellular carcinoma. *Gastroenterology*, 148(4), 806-18.e10. doi:10.1053/j.gastro.2014.12.028
- Stefansson, O. A., Moran, S., Gomez, A., Sayols, S., Arribas-Jorba, C., Sandoval, J., ... Esteller, M. (2015). A DNA methylation-based definition of biologically distinct breast cancer subtypes. *Molecular Oncology*, 9(3), 555–568. doi:10.1016/j.molonc.2014.10.012
- Carmona, F. J., Davalos, V., Vidal, E., Gomez, A., Heyn, H., Hashimoto, Y., Vizoso, M., Martinez-Cardus, A., Sayols, S., ... Esteller, M. (2014). A comprehensive DNA methylation profile of epithelial-to-mesenchymal transition. *Cancer Research*, 74(19), 5608–5619. doi:10.1158/0008-5472.CAN-13-3659
- Zaina, S., Heyn, H., Carmona, F. J., Varol, N., Sayols, S., Condom, E., ... Esteller, M. (2014). DNA methylation map of human atherosclerosis. *Circulation. Cardiovascular Genetics*, 7(5), 692–700. doi:10.1161/CIRCGENETICS.113.000441
- Sandoval, J., Mendez-Gonzalez, J., Nadal, E., Chen, G., Carmona, F. J., Sayols, S., ... Esteller, M. (2013). A prognostic DNA methylation signature for stage I non-small-cell lung cancer. *Journal of Clinical Oncology: Official Journal of the American Society of Clinical Oncology*, 31(32), 4140–4147. doi:10.1200/JCO.2012.48.5516
- Heyn, H., Moran, S., Hernando-Herraez, I., Sayols, S., Gomez, A., Sandoval, J., ... Esteller, M. (2013). DNA methylation contributes to natural human variation. *Genome Research*, 23(9), 1363–1372. doi:10.1101/gr.154187.112
- Rodriguez-Paredes, M., Martinez de Paz, A., Simó-Riudalbas, L., Sayols, S., Moutinho, C., Moran, S., ... Esteller, M. (2014). Gene amplification of the histone methyltransferase SETDB1 contributes to human lung tumorigenesis. *Oncogene*, 33(21), 2807–2813. doi:10.1038/onc.2013.239
- Petazzi, P., Sandoval, J., Szczesna, K., Jorge, O. C., Roa, L., Sayols, S., ... Esteller, M. (2013). Dysregulation of the long non-coding RNA transcriptome in a Rett syndrome mouse model. *RNA Biology*, 10(7), 1197–1203. doi:10.4161/rna.24286
- Heyn, H., Ferreira, H. J., Bassas, L., Bonache, S., Sayols, S., Sandoval, J., ... Larriba, S. (2012). Epigenetic disruption of the PIWI pathway in human spermatogenic disorders. *PloS One*, 7(10), e47892. doi:10.1371/journal.pone.0047892
- Krausz, C., Sandoval, J., Sayols, S., Chianese, C., Giachini, C., Heyn, H., & Esteller, M. (2012). Novel insights into DNA methylation features in spermatozoa: stability and peculiarities. *PloS One*, 7(10), e44479. doi:10.1371/journal.pone.0044479

- Heyn, H., Carmona, F. J., Gomez, A., Ferreira, H. J., Bell, J. T., Sayols, S., ... Esteller, M. (2013). DNA methylation profiling in breast cancer discordant identical twins identifies DOK7 as novel epigenetic biomarker. *Carcinogenesis*, 34(1), 102–108. doi:10.1093/carcin/bgs321
- Heyn, H., Vidal, E., Sayols, S., Sanchez-Mut, J. V., Moran, S., Medina, I., ... Esteller, M. (2012). Whole-genome bisulfite DNA sequencing of a DNMT3B mutant patient. *Epigenetics: Official Journal of the DNA Methylation Society*, 7(6), 542–550. doi:10.4161/epi.20523

List of Abbreviations

ATAC-seq	Assay for transposase-accessible chromatin followed by sequencing
CAR-T	Chimeric antigen receptor T cells
CNV	Copy number variation
CRISPR	Clustered regularly interspaced short palindromic repeats
CWL	Common Workflow Language
ChIP-seq	Chromatin immunoprecipitation followed by sequecing
DAG	Directed acyclic graph
DL	Deep learning
DNA	Deoxyribonucleic acid
DO	Disease ontology
DSB	double-strand break
FDR	False discovery rate
FRiP	Fraction of reads in peaks
GO	Gene ontology
GUI	Graphic user interface
HDR	Homology directed repair
HPC	High-performance computing
IC	Information content
IDR	Irreproducibility discovery rate
JVM	Java virtual machine
MICA	Most informative common ancestor
ML	Machine learning
MNV	Multiple nucleotide variation
MeRIP-seq	m(6)A-specific methylated RNA immunoprecipitation followed by sequencing
MeSH	Medical subject headings
NGS	Next generation sequencing
NHEJ	Non-homologous end joining
NLM	US National library of Medicine
NRF	Non-redundant fraction
PAM	Protospacer adjacent motif
PBC	PCR bottleneck coefficient

PCR	Polymerase chain reaction
RIP-seq	RNA immunoprecipitation followed by sequencing
RNA	Ribonucleic acid
RNA-seq	RNA sequencing
SAM	Sequence alignment map
SBS	Sequencing by synthesis
SNP	Single nucleotide polymorphism
SNV	Single nucleotide variation
TALE	Transcription activator-like effector
TALEN	TALE nuclease
UMI	Unique molecular identifier
VAF	Variant allele frequency
WES	Whole exome sequencing
WGBS	Whole genome bisulfite sequencing
WGS	Whole genome sequencing
ZF	Zinc finger
ZFN	ZF nuclease
bp	base-pair
cDNA	Complementary DNA
gDNA	Genomic DNA
gRNA	Guide RNA
mRNA	Messenger RNA
rRNA	ribosomal RNA
scRNA-seq	single-cell RNA sequencing
ssDNA	Single-stranded DNA

Table of Contents

List of Tables

List of Figures

Introduction

Next Generation Sequencing

Background

Next-generation sequencing (NGS) is the term coined, almost 2 decades ago, to describe the high-output sequencing methods that produce data beyond the genome scale. It represented a new generation of instruments that enabled the study of biological systems with unprecedented depth and breath.

Since the development of foundational methods such as Maxam-Gilbert (Maxam & Gilbert, 1977) and Sanger sequencing (Sanger & Coulson, 1975) to study the composition of the DNA sequences, the field has evolved quickly in terms of applicability, cost reduction and throughput. A number of improvements made to Sanger sequencing allowed the process to be increasingly automated and this led to the development of the first commercial DNA sequencing instruments (figure 1a).

After the milestone that represented the publication of the first draft of the human genome in 2001 (Lander et al., 2001; Venter et al., 2001), a new generation of instruments appeared in the market in 2006. The most notable being Roche's 454 pyrosequencing technology (Margulies et al., 2005), Ion Torrent (Rothberg et al., 2011) and Solexa (Ju et al., 2006) (figure 1b), all based on sequencing-by-synthesis (SBS) methods involving a DNA-polymerase. They could process more sequences in parallel and output a greater number of short reads, up to 400-500 base pairs, thanks to the improvements in microfabrication and high-resolution imaging. These led to increased throughput and cost reduction, representing a real shift in the market. These improvements came to define this second generation of sequencers and coined the term *Next Generation Sequencing*. Since then, a surge in novel methods, techniques, and protocols has emerged, facilitating the investigation of virtually any question in the field of genetics.

As of 2023, the second-generation still dominates the market, thanks to continuous improvements pushing its range of applicability, throughput and costs to the limit. The launch of Illumina's NovaSeq 6000 in 2017 brought the cost of sequencing a complete human genome under $100. Yet, the limitations and biases of the different platforms, like the output of relatively short reads, hinders the study of certain topics such as the *de novo* assembly of genomes and the quantitative analysis of complete transcriptomes.

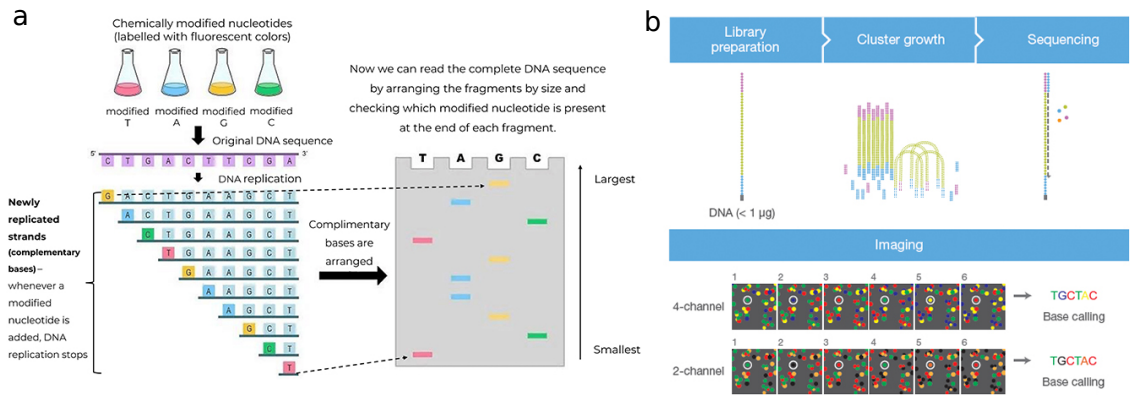


Figure 1: First and Second generation sequencing technologies. (a) Sanger's method of sequencing. (b) 2-Channel SBS Technology Image Detection and Base Calling (Illumina). Adapted from Illumina(C) and (Barua, Bandopadhyay, Biswas, & Gupta, 2022)

A new generation sequencing technologies (third-generation sequencing) emerged in 2014, enabling researchers to obtain reads in the range of kilobases from a single molecule in real time. An important advantage of these technologies is that clonal amplification is avoided, allowing direct sequencing of native, and potentially unmodified, DNA. First systems were commercialized by Pacific Biosciences (Eid et al., 2009), and significant progress was made in recent years with Oxford Nanopore Technologies (Clarke et al., 2009; Eisenstein, 2012) leading the development and commercialization.

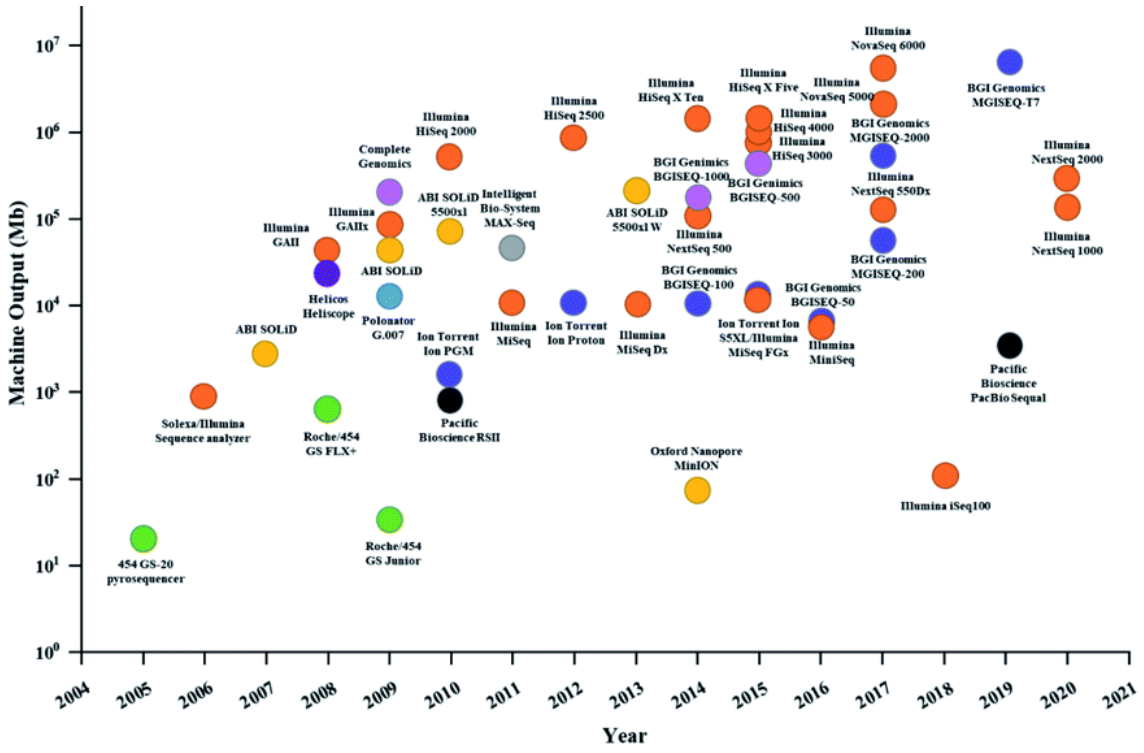


Figure 2: Timeline of commercial NGS instruments. Colors represent the different vendors. Adapted from (Zhou et al., 2021)

Applications of NGS

Whole Genome Sequencing (WGS) is one of the most widely utilized applications in NGS, providing a comprehensive view of genomic information and its biological implications. Recent advancements in NGS platforms, as exemplified by large-scale initiatives such as the 1000 Genomes Project (Durbin et al., 2010) and the UK Biobank (Bahcall, 2018)—which aimed to incorporate genetic data from 500,000 volunteer participants—have facilitated the sequencing of thousands of individuals. These efforts have yielded unprecedented insights into human genetic variation at the population level. Whole Exome Sequencing (WES), which focuses on regions of the genome directly associated with phenotypic consequences, offers the advantage of sequencing more individual samples within a single sequencing run or achieving greater depth of coverage per region. In addition, targeted sequencing enables extremely high coverages, often exceeding 10,000x, which is particularly valuable for validating rare variants (Griffith, Walker, Spies, Ainscough, & Griffith, 2015).

A surge of new methods to interrogate virtually any question in the field of genetics has emerged since the inception of NGS (Illumina, 2017). Currently it's possible to use NGS in order to obtain information of sequence rearrangements, mapping of DNA breaks, epigenetics and chemical modifications of DNA/RNA or proteins interacting with them, interaction between proteins and RNA/DNA or other proteins. Thus, not only the actual sequence of a nucleic acid can be determined by NGS, but also have an insight into the regulatory mechanisms of the genome. Chromatin Immunoprecipitation followed by Sequencing (ChIP-seq) (Park, 2009; Solomon, Larsen, & Varshavsky, 1988) is a popular method to determine DNA-protein interactions. In this method, DNA-protein complexes are crosslinked *in vivo*. Protein-specific antibodies are used to immunoprecipitate the complexes, followed by DNA extraction, purification, and sequencing. This yields high-resolution sequences of the protein-binding sites (figure 3a). ATAC-Seq uses the Tn5 transposome to detect nucleosome-free regions of the genome (Buenrostro, Giresi, Zaba, Chang, & Greenleaf, 2013) (figure 3b). Whole Genome Bisulfite Sequencing (WGBS) is a well-established protocol to detect methylated cytosines at single base resolution by treating genomic DNA (gDNA) with sodium bisulfite before sequencing (Feil, Charlton, Bird, Walter, & Reik, 1994). Upon bisulfite treatment, unmethylated cytosines are deaminated to uracils, while methylated cytosines resist deamination (figure 3c).

NGS can also be effectively used to determine sequences of other nucleic acids such as RNA. As for DNA, quantitative methods to characterize RNA transcription, modifications, structure or RNA-protein interactions have been developed since the inception of NGS. RNA sequencing (RNA-seq) describes the abundance and sequence of RNA transcripts (Marioni, Mason, Mane, Stephens, & Gilad, 2008; Mortazavi, Williams, McCue, Schaeffer, & Wold, 2008). It is based on the use of a reverse transcriptase to convert the RNA to complementary DNA (cDNA) before sequencing. Read output provides an effective measure of transcript abundance levels. RIP-Seq identifies RNA sequences at which proteins bind forming RNA-protein complexes (J. Zhao et al., 2010). MeRIP-Seq

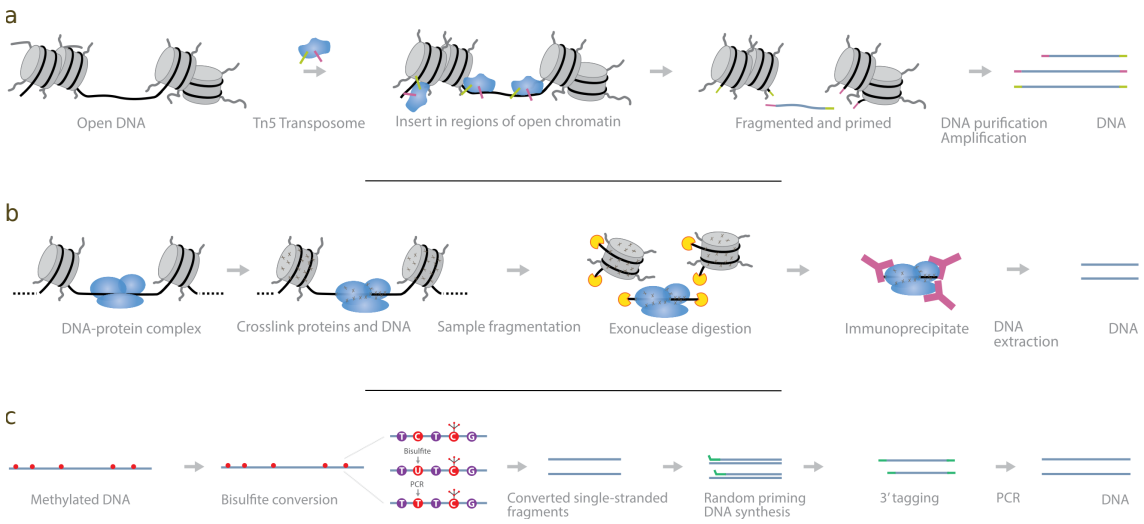


Figure 3: A schematic overview of 3 prominent methods to interrogate the epigenetic landscape. (a) Chip-Seq; (b) ATAC-seq; (c) WGBS; Adapted from (Illumina, 2017)

uses m6A-specific antibodies to immunoprecipitate RNA and map post-transcriptionally modified m6A-methylated RNA (Meyer et al., 2012). Long reads provided by the newer generation of sequencers offer superior performance for detecting full length isoforms. However, they lack the ability to effectively measure transcript abundance.

Perspectives of NGS

Since the first single-cell whole-transcriptome profiling was first published by Tang et al. in 2009 (Tang et al., 2009), new methods emerged to simultaneously capture multi-modal information from single cells. The epigenetic landscape of every cell type is shaped by a unique combination of histone marks, transcription factors, open chromatin and DNA methylation. However, the cell type specific epigenetic background is specially challenging to capture at single cell level. Protocols traditionally used in bulk cells for epigenomic profiling, such as ATAC-seq (Buenrostro et al., 2013) ChIP-seq (Park, 2009) or WGBS (Feil et al., 1994), require of multiple cells and are particularly challenging at single cell level as the template is limited to one set per cell. Rapidly advancing single-cell multi-modal omics (scMulti-omics) technologies enable the measurement of DNA methylation, chromatin accessibility, RNA expression, protein abundance, gene perturbation, and spatial information, all obtained from a single cell ("Method of the Year 2019," 2020; Packer & Trapnell, 2018; Stuart & Satija, 2019). Nevertheless, there are yet many layers which cannot be captured simultaneously from the same cell in order to obtain a complete picture of a single cell's omics profile (Ma, McDermaid, Xu, Chang, & Ma, 2020) (figure 4), opening the field to develop further in the coming years.

Through evolution, genomes have become highly complex entities with long stretches of repetitive elements, structural variation and chemical modifications that dictate their regulation. While short read sequencers provide an accurate and cost-effective platform

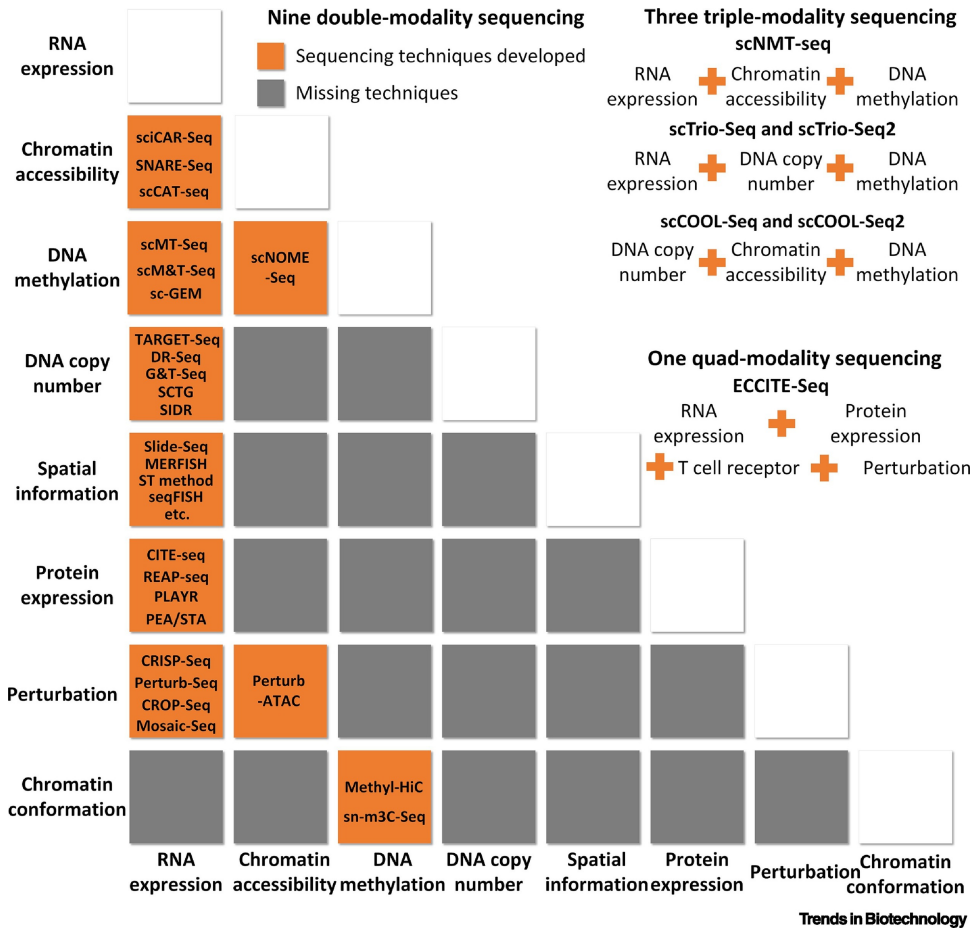


Figure 4: Existing scMulti-omics combinations and representative sequencing techniques. Adapted from (Ma et al., 2020)

to interrogate virtually any question in the field of genetics, sequencing long nucleic acid polymers in short amplified fragments complicates the task of reconstructing and counting the original molecules. Long read sequencing platforms, usually referred as third generation sequencing, cover this niche and allow researches to read full length transcripts or assemble genomes end-to-end without gaps in highly repetitive regions like telomeric or centromeric regions. Recently, the Telomere-to-Telomere Consortium has released the first complete sequence of the human genome (Nurk et al., 2022), resolving previously unknown regions including segmental duplications, ribosomal rRNA gene arrays, and satellite arrays that harbor unexplored variation of unknown consequence. There are, however, several notable limitations in the currently commercially available platforms, namely error rates as high as 15% for PacBio (Carneiro et al., 2012) and 30% for Oxford Nanopore (ONT) (Goodwin et al., 2015) with indel errors dominating, or difficulties to effectively read homopolymers. Recent improvements in the chemistry and the base calling algorithms are improving accuracy.

Nanopore sequencers directly detect the DNA composition of a native single-stranded DNA (ssDNA) molecule. As the DNA slides across the pore, a voltage blockade occurs that modulates the current passing through the pore. Shifts in voltage are characteristic

Table 2: Key steps and processes in analysis of NGS data.

Stage	Task
Pre-processing of raw data	Base calling QC metrics of the sequencing run Trimming adapter sequences Demultiplexing of samples into individual FASTQ
Sequence alignment and quantification	Trimming low-quality sequence reads and/or portions of reads Sequence alignment *de novo* assembly Identification of mutations Gene/transcript quantification Identification of isoforms
Annotation and interpretation	Annotation of mutations and structural variants Predict functional impact of protein mutations Differential expression analysis Differential splicing analysis/isoform abundance Gene set enrichment analsyis

of the particular DNA sequence in the pore. Currently the instrument can detect over 1,000 possible signal profiles representing all possible k-mer and its chemical modifications (Clarke et al., 2009), opening the door to detect base modifications without additional treatment or specific protocols.

Finally, advances in the field led to an abrupt drop in time/cost of sequencing beyond Moore's Law. This has facilitated the incorporation of sequencing as a rutinary technique in the lab and, in addition, expanded the scale and scope of all research projects (Wetterstrand, 2021).

Analysis of NGS data

Processing the data generated by an NGS instrument usually means following one or various workflows consisting of multiple steps which involve many off-the-shelf bioinformatic tools engineered to perform a specific task. A broad understanding of the biological question, library preparation protocol, as well as some level of coding skills and the ability to interact with a high performance computing environment will be necessary (Dudley & Butte, 2009; Shade & Teal, 2015). Given the volume of data generated in a typical NGS run, converting the raw data into knowledge poses an exceptional computational challenge.

Generally, this process can be conceptually split into the 3 parts as summarized in table 2.

Primary analysis: pre-processing of raw data

In Illumina sequencing, the software present in the instrument generates binary files containing the base calls for each cycle. These files are then separated into multiple FASTQ files, each corresponding to one unique sample, using index reads ("Bcl2fastq2 Conversion Software v2.20 Software Guide (15051736)," n.d.). FASTQ is the *de-facto* standard format for storing sequencing data, consisting of a file presenting one or multiple nucleic

acid sequences along with the numeric quality score associated with each nucleotide in the sequence (Cock, Fields, Goto, Heuer, & Rice, 2010).

Importantly, some metrics describing the quality can be obtained at this step:

- Density of clonal clusters generated in the surface of the flow cell. Optimal clustering has a direct implication in the read throughput of the run.
- Fraction of clusters passing filter, or the number of clusters which were uniquely identifiable and not overlapping other clusters.
- Total number of bases sequenced.
- Distribution of quality scores assigned to each base sequenced. Quality scores encode the probability of an incorrect base call as $Q = -10 \log_{10} P$.
- Fraction of reads aligned to an exogenous control sequence, usually the PhiX genome recommended for many Illumina protocols.

This metrics provide information of the run and may help troubleshooting issues at the level of library preparation or technical issues with the sequencing instrument.

Secondary analysis: sequence alignment and quantification

FASTQ files are stripped of low-quality sequence reads and/or portions of reads, providing a "clean" FASTQ file. Common tools like FastQC are used to assess per base sequence content and quality scores. This information can be used to decide if and how reads should be trimmed using tools like Cutadapt (M. Martin, 2011) or Trimmomatic (Bolger, Lohse, & Usadel, 2014) to generate the "clean" FASTQ.

If a reference genome already exists, next step is sequence alignment. Using common alignment tools like BWA (H. Li & Durbin, 2009) or Bowtie 2 (Langmead & Salzberg, 2012) reads are mapped onto a pre-assemble genome reference, often provided by consortias such as the Genome Reference Consortium (Church et al., 2011; "Genome Reference Consortium," n.d.). For messenger RNA (mRNA), a splice-aware aligner may be used instead (Dobin et al., 2013).

When a pre-assembled genome is not available, reads generated in a run can be combined into longer contiguous sequences using their overlapping sub-sequences. Ideally, one should use a combination of shorter and longer insert fragments to obtain optimal results. This combination enables detection of structural variants and is necessary for the identification of complex rearrangements (Baker, 2012; Dominguez Del Angel et al., 2018).

Alignment software typically outputs reads, coordinates and additional metadata in Sequence Alignment Map (SAM) format (H. Li et al., 2009), either in text (SAM) or its compressed binary representation (BAM). This file is fed to downstream tools for further analysis. In case of DNA sequencing, genetic variation is usually of interest. Simple single- or multiple-nucleotide variations (SNV and MNV), short insertions and deletions and variant allele fractions (VAF) can be detected by comparing against a reference sequence. Complex structural rearrangements, gene fusions, chromosomal abnormalities,

and copy number variations (CNVs) are other types of mutations which can be detected by sequencing DNA. For RNA, reads are organized into genes and tallied to obtain the quantification at gene or transcript level. It is also possible to reconstruct the full transcript and detect the expressed isoforms with specialized software such as Cufflinks (Trapnell et al., 2012), RSEM (Bo Li & Dewey, 2011), Kallisto (Bray, Pimentel, Melsted, & Pachter, 2016) and Salmon (Patro, Duggal, Love, Irizarry, & Kingsford, 2017).

Tertiary analisys: annotation and interpretation

This step includes the annotation and interpretation of the data produced during the secondary analysis in the context of genes and transcripts. Mutations and structural variants may be annotated using databases for common SNPs. National Institutes of Health sponsors a repository of curated information produced by studies investigating the interaction of genotype and phenotype (Tryka et al., 2014), and also ClinVar (Landrum et al., 2018), a public archive of human genetic variants and interpretations of their significance to disease. Different strategies exist to predict the consequences of mutations. PolyPhen (Ramensky, Bork, & Sunyaev, 2002) combines sequence conservation with structural features to predict the functional impact of protein mutations. MutPred (Biao Li et al., 2009) uses machine learning to model changes of structural features and functional sites between wild-type and mutant sequences, and predict probabilities of gain or loss of protein structure and function. Other databases such as Ensembl (F. J. Martin et al., 2023) provide basic transcript information and prediction of mutational effects on proteins for multiple organisms.

For RNA-seq data, tertiary analysis commonly includes differential expression analysis to identify significant changes in gene expression across multiple libraries. Furthermore, with sequencing mRNA one can identify different splicing events and isoform expression levels between conditions (Conesa et al., 2016). The interpretation of the phenotypical effects of the observed changes is performed in the context of predefined gene sets. Databases such as KEGG (Kanehisa & Goto, 2000), PantherDB (Mi & Thomas, 2009) and Reactome (Gillespie et al., 2022) connect sets of genes with biological pathways. The Gene Ontology Consortium ("The Gene Ontology Resource," 2019) provides a database connecting genes with the different kind of biological functions, the pathways carrying out different biological programs, and the cellular locations where these occur. Other sources of gene sets could be any study describing sets of genes whose relevance was identified in the context of a specific specific disease, drug response or cell type.

Typically, tables and figures summarize the results and help with extracting conclusions from the experiment.

The results can be further integrated with other layers of omics information obtained from different experiments, such as chromatin accessibility, RNA expression, DNA methylation, DNA-protein interaction or spatial organization of chromatin.

Quality control of NGS data

As previously described, it's crucial to check the metrics output by the instrument after the run is finished. Cluster density, reads passing filter, fraction of reads aligned to Illumina's PhiX spike-in control library or distribution of quality scores, all provide useful information to determine whether there was any technical issue that would compromise the number and quality of the output reads. However, these metrics don't provide a link between the quality of the data and the hypothesis being tested with the experiment. For instance, at this point it's not yet possible know whether the sequenced material was DNA or RNA, nor the genomic loci where this comes from, not even if the demultiplexed reads really came from the biological sample the index indicates. Consequently, it's important to incrementally add quality control checkpoints at every stage of the analysis when additional information is available.

Right after rawdata is pre-processed and FASTQ files are available, important information about what has been sequenced is already available. Chemicals used in Illumina sequencing-by-synthesis (SBS) technology degrade the templates and limits the number of sequencing cycles to a few hundred (and thus, the effective read length). It's expected to obtain a drop in quality toward the end of the reads, which may then be trimmed if quality is suboptimal. At this point it's possible to determine the distribution of quality scores across all bases (cycles) of the reads in a FASTQ file, and know if and at which position quality drops below Q30 on average. Additionally, FASTQ files carry important information about sequence composition per position which is crucial for short-read sequencing: are there compositional biases in specific positions? What is the fraction of reads with exactly the same nucleotide sequence (ie. duplicated reads)? Are there over-represented k-mers in the data? Does the GC content distribution match that expected in the organism? Software such as FastQC ("Babraham Bioinformatics - FastQC A Quality Control tool for High Throughput Sequence Data," n.d.) will answer all these questions. These metrics help in determining whether there was any bias in the pool of molecules that were selected for sequencing, or a little diversity in the pool. They are essential to reliably extract conclusions from the data. Depending on the type of experiment and material sequenced, some of these questions may have a different answer. For instance, RNA-seq data is expected to show high levels of read duplication as duplicate reads arise naturally in highly expressed genes (Griffith et al., 2015; Williams, Thomas, Wyman, & Holloway, 2014); or a depletion in the number cytosines at any position in WGBS upon bisulfite treatment (Feil et al., 1994).

Following sequence alignment and quantification, with genomic positions assigned to reads, additional information is available. Given the diversity of biological questions which can be answered with NGS, at this point different quality controls may need to be done. If sequencing DNA for mutation calling, it's important to know the percentage of the genome covered by reads, and the average number of reads covering any position (commonly named *average coverage*). On the other hand, in assays enriching for specific regions of the DNA

Table 3: Metrics of library complexity for ChIP-seq standards.

PBC2	NRF	Bottleneck severity
< 1	< 0.5	Severe
$1 \leq$ PBC2 ≤ 3	$0.5 \leq$ NRF ≤ 0.8	Moderate
$3 \leq$ PBC2 < 10	$0.8 \leq$ NRF < 0.9	Mild
≥ 10	> 0.9	None

such as ChIP-seq, it's essential to assess the initial library complexity and the amount of reads in enriched regions compared to the total number of reads. The ENCODE project (ENCODE Project Consortium, 2012; Landt & Marinov, 2012) defines the *PCR Bottleneck Coefficient* (PBC2, equation (1)), the *Non Redundant Fraction* (NRF, equation (2)) and the *Fraction of reads in peaks* (FRiP, equation (3)) to estimate those metrics and provides some guidelines on how they should be interpreted (table 3).

$$PBC2 = \frac{\text{Number of genomic location where EXACTLY one read maps}}{\text{Number of genomic locations where two reads map uniquely}} \tag{1}$$

$$NRF = \frac{\text{Number of uniquely mapping fragments}}{\text{Total Number of reads}} \tag{2}$$

$$FRiP = \frac{\text{Number of reads in called peak regions}}{\text{all mapped reads}} \tag{3}$$

In RNA-seq it's also possible (and desirable) to estimate the library complexity. This is especially critical when working with single cells or protocols for library preparation starting with little input material. However, using the fraction of duplicate reads as a proxy of library complexity poses some challenges as highly expressed genes are naturally over-sequenced. Therefore, the fraction of duplicate reads is not a useful estimate of library complexity. The next section explores the current scenario with challenges and solutions in depth.

RNA stability also poses some additional challenges for sequencing RNA. Commonly, RNA is either sequenced *in total* after depleting the highly abundant and usually uninteresting ribosomal RNA (rRNA), or after capturing mRNA using the polyadenylated tail. Partially degraded RNA would consist of fragmented molecules with shorter polyadenylated fragments, which in the case of capturing mRNA would represent shorter transcripts, potentially of unequal sizes between samples or conditions especially when prepared in different batches. The reads coming from a transcript will be a function of the length of the transcript and the expression levels of the transcript. Hence optimal and consistent RNA integrity is crucial for extracting reliable conclusions from the experiment. Some popular library preparation protocols may capture the whole body of the transcript. Therefore, it is possible to estimate the average coverage over the transcript body with tools like RSeQC (L. Wang, Wang, & Li, 2012), to see if the whole transcript is captured and not biased toward the 3' end which would indicate partial RNA degradation. In addition, calculating the fraction of reads on different transcript classes will disclose whether the

expected type of RNA molecules were eventually sequenced.

Following tertiary analysis, additional quality controls can be performed. Consistency between replicates within an experiment shall be evaluated with simple techniques such as dimensionality reduction, Pearson correlation, hierarchical clustering or simply calculating the Euclidean distance between pairs of replicates. First, a vector quantifying a set of genomic features such as genes or enriched regions in DNA should be calculated. This could simply be normalized read counts overlapping the feature, or any other form of signal measurement such as log-transformed values of the significance score obtained in peak calling. Dimensionality reduction often employs visual inspection of the first few principal components (typically 2–3) derived from Principal Component Analysis (PCA), a widely used and effective technique. PCA transforms a high-dimensional dataset—such as genomic features of interest—into a new coordinate system, concentrating the majority of the variation in the first few dimensions. A scatterplot of the first two components can then be used to visualize the data, with similar samples clustering together and dissimilar samples appearing farther apart. This approach allows for simultaneous evaluation of multiple replicates, which provide contextual reference for interpretation. However, the method is inherently subjective, and its interpretation depends on the specific experimental context. On the other hand, Pearson correlation calculates an objective metric by describing the cosine of the angle between the two regression lines of the replicates, with values ranging from -1 (perfect anticorrelation) to 1 (perfect correlation). Other methods have been developed specifically to evaluate specific types of experiments. This is the case of the Irreproducible Discovery Rate (IDR) (Q. Li, Brown, Huang, & Bickel, 2011), which evaluates reproducibility of ChIP-seq and ATAC-seq experiments by measuring consistency between peaks sets called on two biological replicates within an experiment.

Additionally, testing the over-representation of significant results in specific chromosomes may be done to see if there's any specific bias which could be explained by eg. different copy numbers. The Tukey mean-difference plot (Cleveland, 1993), commonly known as MA plot, can be used to study the existence of systematic biases in the quantification of features taken in two samples or experimental conditions. The plot visualizes the distribution of the log-transformed ratio of the quantified features against the average quantification, to determine whether a shift systematically occurs toward positive or negative fold-changes.

Notwithstanding, it's important to note at this point that the quality control methods and/or parametrization will depend on the type of experiment and material sequenced.

Library complexity and sources of duplicated reads

With the recent advancements in the chemistry and library preparation protocols, sequencing libraries can be build from tiny amounts of starting material. While the standard Illumina TruSeq library preparation kit requires 10-100 ng of starting RNA ("Illumina Stranded mRNA Prep | A clear view of the coding transcriptome," n.d.), kits suitable

for ultra low input or single cells can go down to 10 pg ("SMART-Seq mRNA LP and SMART-Seq mRNA," n.d.). Commonly, a single mammalian cell only has about 10 pg of total RNA and less than 0.1 pg of mRNA (Huang et al., 2018). Although individual cells can be barcoded first and then pooled, there will be situations in which the amount of starting material would be limited or compromised, even in experiments performed on bulk cells. This means that the cDNA library needs to be amplified before sequencing, typically using PCR-based amplification. Some additional common issues during library preparation, such as adapter ligation bias, random priming not being completely random, degradation, or strong GC bias, may eventually lead to limited diversity of starting molecules (Chepelev, Wei, Tang, & Zhao, 2009). Therefore it's crucial to check that the amount of starting material didn't cause a bottleneck for PCR amplification, leading to high read duplication due to low library complexity (figure 5b). Eventually, the variety of molecules seen after sequencing correlates with the amount of input material.

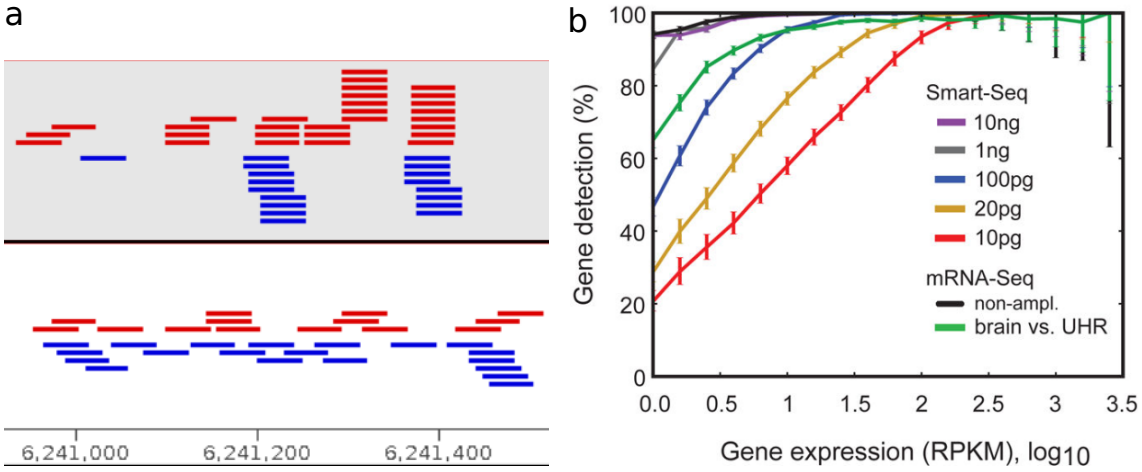


Figure 5: Examples of low and high diversity libraries. (a) reads covering a genomic region from low and high diversity libraries (source: E. Karaulanov); (b) Fraction of detected genes in libraries prepared with different amounts of RNA (Ramsköld et al., 2012).

Duplicated reads observed after sequencing can arise from various sources, making their identification essential for assessing the quality of a sequencing run. Optical duplicates, for instance, may result from overloading the sequencing flow cell or issues with the reagents used during the sequencing process. This happens when a single cluster (ie. a sequenceable molecule on the surface of the flowcell) has been called as two by the instrument's analysis algorithm in non-patterned flowcells, or a molecule occupying two adjacent wells during library preparation in a patterned flowcell. A second type of technical duplicates may arise from amplification during sample preparation (ie. PCR-based), often due to failed RNA/DNA extraction from cells in bulk experiments, limited amount of input material as in single cells, or strong biases (eg. GC-bias or not-so-random priming) which limit the diversity of molecules. A third type of duplication arises naturally in experiments such as sequencing mRNA, where the length of the sequentiable molecule (eg. a transcript) is limited and the number of unique sequentiable fragments which can stem from a single feature is therefore limited too. The rate of natural duplication depends on the feature's

length, its abundance level, and the overall coverage of the sample. While the first two types of duplication don't carry any useful information and reads should be disregarded, the later hides valuable information of the feature's abundance which should be taken into account to derive any conclusion. In the case of mRNA-seq it is common to see top 5% of genes taking more than 50% of all reads, easily surpassing the threshold of 1 read per base-pair of the transcript, at which read duplication is inevitable.

Currently the detection of *harmful* technical duplication depends on global metrics or the evaluation by an expert eye of systematic stacks of identical reads appearing in coverage tracks (figure 5a). Many tools such as Picard ("Broadinstitute/picard," March 28, 2014/2023), RSeQC (L. Wang et al., 2012), Qualimap ("Qualimap," n.d.), FastQC ("Babraham Bioinformatics - FastQC A Quality Control tool for High Throughput Sequence Data," n.d.) or the FASTX toolkit ("FASTX-Toolkit," n.d.) can be used to evaluate global duplication rates. However, global duplication rates do not take into account the effect of gene expression levels which leaves them of limited use for RNA-Seq data and other enrichment-based assays. Identifying by eye stacks of reads remains the most reliable source for identifying technical duplication, although it may be problematic in short+highly abundant features, and lacks of a systematic and interpretable metric to globally evaluate an experiment.

The correction of these artifacts remains an unresolved challenge. In whole-genome DNA sequencing experiments, a conservative approach is often taken, where duplicated reads are discarded. However, in assays that enrich for specific genomic regions, such as ChIP-seq, the limited width of targeted regions can lead to saturation, resulting in underestimating the magnitude of the enrichment in quantitative analyses. Similarly, in RNA-seq, adopting a conservative strategy that discards duplicates regardless of their origin can distort abundance estimates, compromise differential analyses between conditions, and effectively undermine one of the primary applications of RNA-seq. This necessitates the development of alternative analytical methods. Several tools have been proposed to address this issue. RASTA (Baumann & Doerge, 2014) employs hierarchical clustering to estimate true tag abundance by distinguishing genuine reads from incorrectly amplified ones. eXpress (Roberts & Pachter, 2013) uses probabilistic read assignment to transcripts, smoothing read coverage and effectively reducing large stacks of reads in regions where systematic overestimation does not occur. iReckon (Mezlini et al., 2013) incorporates a probabilistic framework that accounts for multiple biological and technical factors, including PCR amplification biases. However, none of these methods is universally applicable, and their effectiveness requires careful evaluation on a case-by-case basis.

Working hypothesis and goals for Chapter 1

The lack of a method which could systematically evaluate the quality of a sequencing experiment in terms of library complexity and molecular bottleneck, especially when working with little input material or single cells, led us to the development of DupRadar. The

method allows the researcher to easily tell apart normal read duplication due to natural over-sequencing of highly expressed genes from PCR clonal artefacts originating from NGS library preparation.

Gene Ontology

Ontologies in molecular biology are a widely used resource for interpretating high-throughput experiments that interrogate thousands of features in parallel. The term *ontology* refers to a relational vocabulary including terms used in a specific domain, along with their definitions and relationships. The Gene Ontology Consortium (Blake & Harris, 2008; "The Gene Ontology Resource," 2019) contributes through the Gene Ontology (GO) project structured vocabularies describing domains of molecular biology for a large number of species in a format readable by both humans and machines, suitable for computational analysis in high-throughput experiments. The organization of terms in a hierarchical manner allows for classification and query at different levels of specificity. The GO project provides 3 orthogonal ontologies:

- **Molecular Function** describes the elemental activity performed by individual gene products, such as enzymatic or structural activities.
- **Biological Process** describes the biological objective in which individual gene products participate, including areas of development, cell communication, physiological processes, or behavior.
- **Cellular Component** describes the location of action within the cell for a gene product.

Other structured vocabularies for biomedical sciences also exist. Disease Ontology (DO) (Schriml et al., 2012) integrates medical vocabularies and biomedical ontologies with human disease. Medical Subject Headings (MeSH) is the U.S. National Library of Medicine (NLM) controlled vocabulary thesaurus used for indexing articles in MEDLINE and PubMed. MeSH terms include subject from NLM databases, organized hierarchically. These terms are associated with genes and structured to optimize text-mining capabilities.

The Gene Ontology is structured hierarchically, meaning that terms are organized in a tree-like structure with more specific terms (child terms) being nested under broader, more general terms (parent terms). This hierarchical structure is intended to capture the relationships between different biological concepts. Enrichment analysis tools often report both specific and parent terms, leading to redundancy in the results (figure 6). For instance, enrichment of a specific biological process term may also imply enrichment of its parent terms. This reporting practice aims to provide a more comprehensive understanding of the biological processes, functions, and cellular components that are overrepresented in a given gene set. However, the hierarchical structure can lead to large and redundant list

of enriched terms, creating an illusion of an overly long list of significant categories, and obscure the relevant biological interpretation.

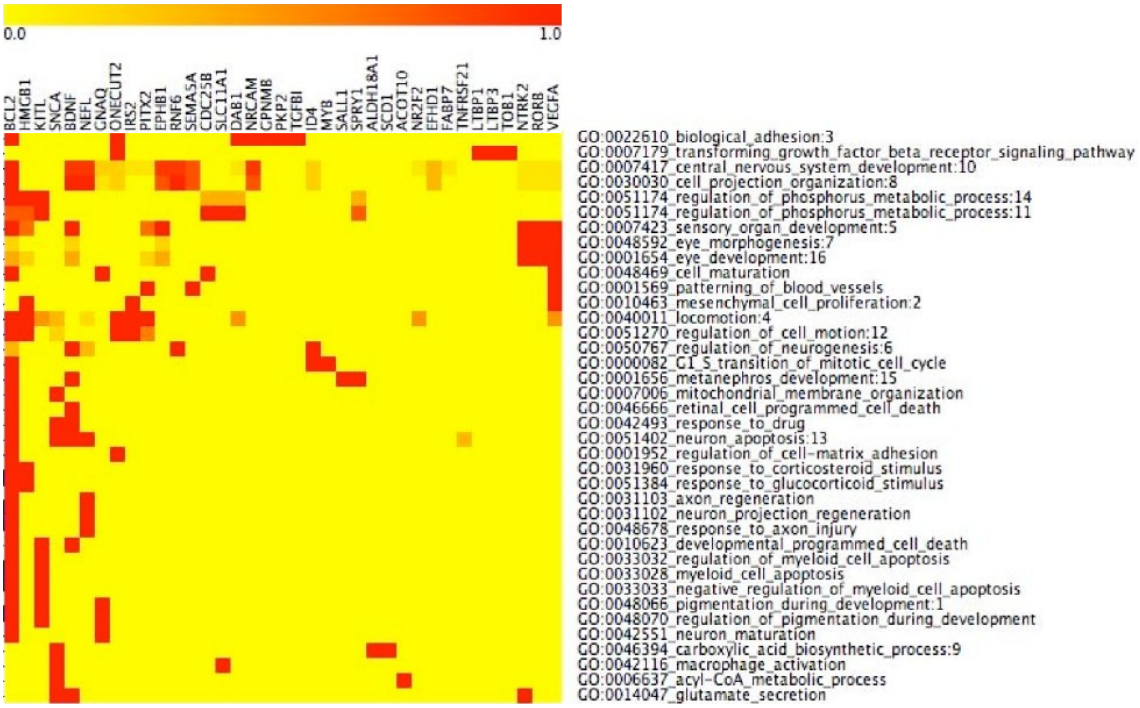


Figure 6: Cluster map of gene - GO categories of an example dataset with genes and GO categories enriched during retinal development. Several GO categories in the map are sufficiently redundant to be grouped together (Zeeberg et al., 2011)

Several methods have been developed in order to trim lists of redundant terms. The Gene Ontology Consortium provides a reduced version of the Gene Ontology that contains a selected number of relevant nodes. One approach to obtain lists free of redundant terms is to use this subset in enrichment analysis tools, allowing for an overall understanding of key biological functions without delving into more specific terms. The main disadvantage of using these limited GO versions is that the results may hide interesting findings represented exclusively by more specific terms which were not included in the annotation. Another approach is to leverage the graph structure of the Gene Ontology, which allows comparing GO terms by assessing the likeness in meaning between them, a concept known as *semantic similarity* (Pesquita, 2017; Pesquita, Faria, Falcão, Lord, & Couto, 2009). This method was originally developed using WordNet, a computationally amenable dictionary/thesaurus, and its application to the Gene Ontology first explored in (Lord, Stevens, Brass, & Goble, 2003). Another approach for defining similarity measures on ontologies involves the use of knowlegde graph embeddings (Kulmanov, Smaili, Gao, & Hoehndorf, 2021; Ristoski & Paulheim, 2016). The concept behind embeddings is to represent ontologies as vectors in a real-valued vector space and utilize a distance measure between vectors as a surrogate metric for similarity.

Measures of semantic similarity have been extensively studied and implemented in many different software packages (Brionne, Juanchich, & Hennequet-Antier, 2019; Supek,

Bošnjak, Škunca, & Šmuc, 2011; G. Yu et al., 2010; Guangchuang Yu, Wang, Han, & He, 2012). Some of these tools also provide methods to visually summarize the relationship between terms in a list, identify redundancy, and determine the importance of a term in the resulting list. However, they are limited in one or several key aspects, which restricts their integration into data analysis pipelines: i) a limited programmatic interface; ii) pre-packaged GO annotations which cannot be overridden; iii) limited visualizations and exploration capabilities. Combining semantic similarity techniques, both existing and customized GO annotations, and robust visualization capabilities within a single tool with programmable access would greatly enhance the biological interpretation of extensive lists of GO terms.

Measures of semantic similarity

Semantic similarity measures can be classified depending on how terms (nodes in the GO graph) are incorporated and the type of information that is used to determine the similarity (Pesquita et al., 2009).

Information Content-based methods

Information Content (IC) based methods are calculated upon the frequencies of two GO terms and that of their closest common ancestor in a specific corpus of GO annotations. The IC of a term t is defined as the negative log probability of the term occurring in a GO corpus, given by (4).

$$IC(x) = -ln(p(x)) \tag{4}$$

$$p(x) = \frac{n_{x'}}{N} | x' \in \{x,\ children\ of\ x\} \tag{5}$$

Where $p(x)$ is the relative frequency of a term x in the GO corpus (5). Thus, an infrequent term contains a higher amount of information.

Some methods based on IC were proposed, initially using Resnik's metric (Resnik, 1999) to quantify semantic similarity between terms in the GO DAG as the information content of the most informative common ancestor (MICA) of these terms (6).

$$sim_{Resnik}(t_1, t_2) = IC(MICA) \tag{6}$$

A drawback of Resnik's approach is that it ignores the information contained in the structure of the ontology and concentrates on the information content of a terms only. These led to the development of other metrics. Jiang and Conrath's (J. J. Jiang & Conrath, 1997) suggested an approach which defines the distance between two terms t_1 and t_2 as the sum of their distances to their MICA (7).

$$sim_{JC}(t_1, t_2) = 1 - (IC(t_1) + IC(t_2) - 2 \cdot IC(MICA)) \tag{7}$$

A different approach to take into account the information of the corpus, Lin's method (Lin, 1998) normalizes the information of the MICA to the information needed to fully describe the two terms (8).

$$sim_{Lin}(t_1, t_2) = \frac{2 \cdot IC(MICA)}{IC(t_1) + IC(t_2)} \tag{8}$$

The Relevance method proposed by Schlicker (Schlicker, Domingues, Rahnenführer, & Lengauer, 2006) is based on Lin's and Resnik's definitions, and improves on Lin's approach by including a term to distinguish between generic and specific terms (9).

$$sim_{Rel}(t_1, t_2) = \frac{2 \cdot IC(MICA)}{IC(t_1) + IC(t_2)} \cdot (1 - p(MICA) \tag{9}$$

Graph-based methods

Yet all IC methods depend on gene annotation to measure the semantic similarity of GO terms. Consequently, different gene annotation may lead to different values for the same pair of GO terms. On the other hand, graph-based methods make use of the topology of the GO graph structure to compute semantic similarity. Each GO term is represented as a tuple describing a directed acyclic graph $DAG_A = (A, T_A, E_A)$, where T_A is the set of GO terms in the graph connecting to term A, and E_A the edges. Wang et al. (J. Z. Wang, Du, Payattakool, Yu, & Chen, 2007) suggested encoding the semantics of a term relative to its location in the GO graph and the semantic relation with its ancestor terms. A GO term A will have a semantic value $SV(A)$ (10) quantifying the semantic contribution of all other terms in DAG_A to term A. $SV(A)$ is defined as the sum of the semantic values for all terms in $DAG_A = (A, T_A, E_A)$ (11). Consequently, semantic similarity between terms A and B, represented by $DAG_A = (A, T_A, E_A)$ and $DAG_B = (B, T_B, E_B)$ respectively, is defined by this formula (12):

$$S_A(t) = \begin{cases} S_A(A) = 1 & \text{if } t = A \\ S_A(t) = max\{w_e \cdot S_A(t') | t' \in children\ of\ (t)\} & \text{otherwise} \end{cases} \tag{10}$$

$$SV(A) = \sum_{r \in T_A} S_A(t) \tag{11}$$

$$S_{GO}(A, B) = \frac{\sum_{t \in T_A \cap T_B} (S_A(t) + S_B(t))}{SV(A) + SV(B)} \tag{12}$$

A different approach to determining the semantic similarity between terms is through the use of embeddings. These embeddings represent nodes in the Gene Ontology graph as vectors in $\mathbb{R}^n$. Onto2Vec (Smaili, Gao, & Hoehndorf, 2018) is a method that generates

embeddings for ontology terms by taking into account all the edges between terms in the GO graph that may contribute to the semantics of the term. These vectors can be utilized to compute similarity between terms using any measure applicable to real-valued vectors, such as the cosine similarity or the Euclidean distance.

Working hypothesis and goals for Chapter 2

Despite the fact that GO redundancy has been extensively studied and approached from different angles, simple tools which integrate current GO annotations for multiple organisms and effective visualizations remain mostly unavailable. REVIGO (Supek et al., 2011) is a popular web tool and service which implements widely used semantic similarity metrics for identifying GO redundancy and effective visualizations. It's simple and straightforward. However, it runs in the **cloud**, lacks a programmatic interface, and cannot use custom or additional GO annotations aside from what the developers made available. ClusterProfiler (Guangchuang Yu et al., 2012) and the GOSemSim (G. Yu et al., 2010) are R packages well integrated within the Bioconductor ecosystem which provide a programmatic interface and access to the last current GO annotations. However, the visualizations provided are limited. Many other tools exist, but are limited in one or several aspects which limit their use in bioinformatic pipelines. This led us to writing the **rrvgo** package, a tool which uniquely combines semantic similarity techniques, access to the last current GO annotations, robust visualization capabilities, in a scientific environment widely used in bioinformatic pipelines such as R.

Intro to CRISPR/Cas9

CRISPR (Clustered Regularly Interspaced Short Palindromic Repeats) shortly after its discovery was considered a revolutionary tool for genome editing. Originated from studies on bacterial immune systems, where it functions as an adaptive defense against bacteriophages (Ishino, Shinagawa, Makino, Amemura, & Nakata, 1987; Mojica, Juez, & Rodr'ıguez-Valera, 1993), the technique allows for the precise correction of endogenous genes without the need to insert additional copies of genes. The development of the CRISPR-Cas9 genome editing technique (Doudna & Charpentier, 2014; Jinek et al., 2012), led by Emmanuelle Charpentier and Jennifer A. Doudna, earned them the Nobel Prize and has since spurred numerous scientific advancements.

A streamlined CRISPR system from Streptococcus pyogenes utilizes the Cas9 protein, which works in conjunction with a short RNA sequence (gRNA) that guides the Cas9 endonuclease to its target (figure 7a). Cas9 contains a recognition domain that scans the genome for sequences complementary to the gRNA, along with two nuclease domains, RuvC and HNH, that cleave both strands of DNA, resulting in double-strand breaks (DSB). Once the DSB is introduced, the cell attempts to repair the DNA via two main pathways (figure 8a): Non-homologous end joining (NHEJ), a low-fidelity repair pathway

that can result in gene disruption due to small insertions or deletions; or homology-directed repair (HDR), a more precise pathway used for gene modification when a template DNA is available. This allows researchers to insert specific sequences or make targeted modifications. Beyond Cas9, there is a broader family of Cas proteins, such as Cas1 and Cas2 in E. coli, and Cas12a (Cpf1, figure 7b) (M.-Y. Yan et al., 2017) and Cas13a (C2c2, figure 7c) (Abudayyeh et al., 2016) in other organisms. These proteins differ from spCas9 in how they bind to and cleave DNA and/or RNA.

CRISPR is lauded as a highly versatile tool not limited to basic genome editing (Nishiga, Liu, Qi, & Wu, 2022). Variants of Cas9, such as dead Cas9 (dCas9) and Cas9 nickase, expand its functionality beyond cutting DNA. dCas9, for instance, can be fused with other functional enzymes, transforming CRISPR into a tool for gene regulation, epigenome modification, and transcriptional control. The Cas13 family is also mentioned for its ability to target RNA, broadening CRISPR's scope of action. CRISPR has wide-ranging applications (Hsu, Lander, & Zhang, 2014):

- Agriculture: It has been employed to engineer crops with enhanced pest resistance and improved nutritional profiles.
- Biofuels: CRISPR is being used to optimize biofuel production.
- Animal Models: Scientists create animal models using CRISPR to study diseases and evaluate drug efficacy.
- Functional genomic screens: Scientists induce systematic knockouts of genes to determine which ones are necessary for a particular cellular function.
- Therapeutics: CRISPR is being tested as a therapeutic tool to treat genetic disorders, including conditions like sickle cell disease —approved for clinical use since late 2023—, beta-thalassemia, and certain cancers.

Additionally, CRISPR has successfully been used in the clinics to edit T-cells for cancer immunotherapy (Ex-vivo editing, CAR-T cell therapy). Intellia Therapeutics' NTLA-2001 trial uses lipid nanoparticles (LNP) to deliver CRISPR components directly into the body (In vivo editing), targeting amyloid deposits caused by a mutation in the TTR gene, a once-and-done therapy showing promising results.

CRISPR is one of several genome-editing technologies, and it stands out for its simplicity, efficiency, and adaptability compared to earlier tools like Zinc Finger Nucleases (ZFNs) and Transcription Activator-Like Effector Nucleases (TALENs). CRISPR-Cas9 relies on a simple guide RNA (sgRNA) to direct the Cas9 nuclease to a specific DNA sequence, inducing double-strand breaks at sites determined by RNA-DNA base pairing, with a requirement for a protospacer adjacent motif (PAM). This design allows for easy programmability and scalability, making CRISPR highly versatile. In contrast, ZFNs use engineered zinc finger domains to recognize specific DNA triplets, which are linked to a FokI nuclease for DSB creation. However, ZFNs require extensive protein engineering for each target sequence, limiting their flexibility and ease of use. Similarly, TALENs operate by recognizing individual DNA bases via TALE domains, but like ZFNs, they necessitate

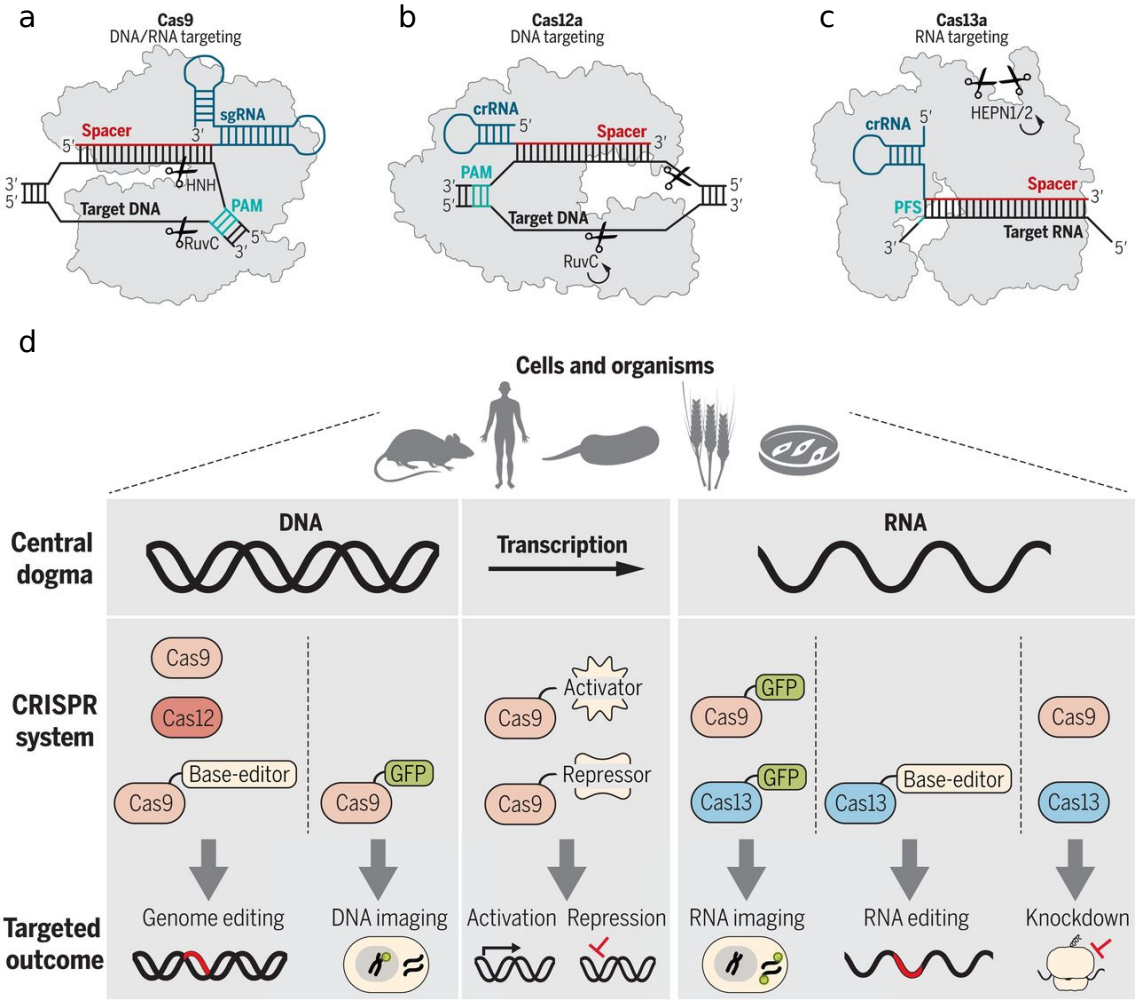


Figure 7: Schematic of CRISPR-Cas systems and applications in genetic manipulation. (a) CRISPR/Cas9 targets double-stranded DNA (dsDNA), with the spacer sequence highlighted in red, sgRNA scaffold in blue, and PAM sequence in turquoise. Scissors indicate the cut site created by the HNH and RuvC domains after correct base-pairing between the DNA and the spacer. (b) CRISPR/Cas12a also targets double-stranded DNA, with the spacer sequence highlighted in red, crRNA scaffold in blue, and PAM sequence in turquoise. Scissors indicate the cut site created by the RuvC domain after correct base-pairing between the DNA and the spacer. (c) CRISPR/Cas13a targets RNA, with the spacer sequence highlighted in red and crRNA scaffold in blue. The HepN RNase nuclease activity is activated after correct base-pairing (scissors). (d) Various applications of CRISPR systems are depicted. Cas9 and Cas12a are utilized to introduce DSB in DNA, while Cas9 and Cas13a interfere with RNA. Engineered versions of Cas proteins with inactivated nuclease domains (dCas9 or dCas13) can be fused to base editors to modify nucleotides in DNA or RNA, to transcriptional activators and repressors for transcriptional regulation, or to a green fluorescent protein (GFP) for imaging. Adapted from (Knott & Doudna, 2018).

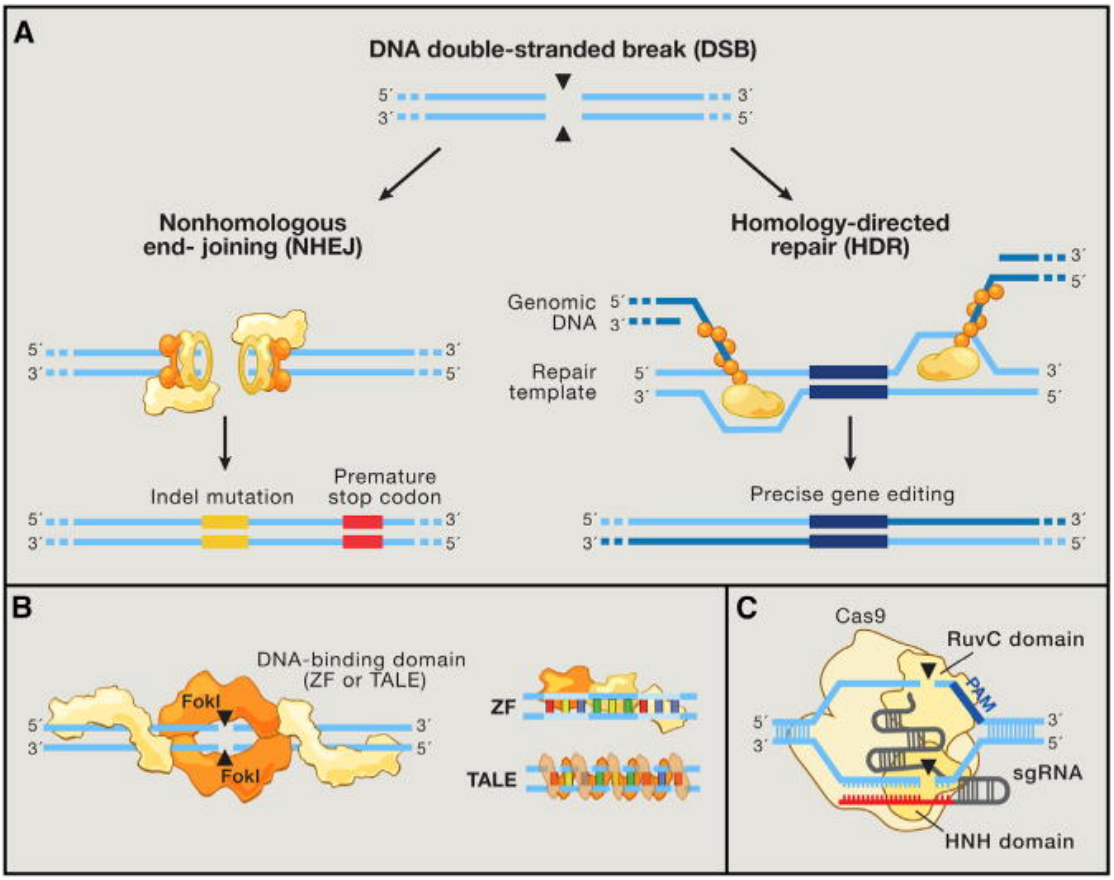


Figure 8: DNA repair pathway are key mediators of gene editing. (a) DNA double-strand breaks are typically repaired by nonhomologous end-joining (NHEJ) or homology-directed repair (HDR). In NHEJ, indels are introduced when the complementary strands undergo end resection and misaligned repair due to micro-homology, eventually leading to frameshift mutations and gene knockout. In HDR, Rad51 proteins bind DSB ends, recruiting accessory factors that direct genomic recombination with homology arms on an exogenous repair template. (b) Zinc finger (ZF) proteins and transcription activator-like effectors (TALEs) are naturally occurring DNA-binding domains which are also used in gene editing. (c) CRISPR/Cas9 targets double-stranded DNA, with the spacer sequence highlighted in red, sgRNA scaffold in grey, and PAM sequence in blue. Arrows indicate the cut site created by the HNH and RuvC domains after correct base-pairing between the DNA and the spacer. Adapted from (Hsu et al., 2014).

complex protein design for each new target. While TALENs offer high specificity due to their single-base recognition, both ZFNs and TALENs are less adaptable and scalable than CRISPR, making them more challenging for high-throughput applications (Gaj, Gersbach, & Barbas, 2013).

Despite its groundbreaking potential, CRISPR technology faces several challenges:

- Off-target effects: Unintended cuts at non-target sites, which may result in harmful mutations.
- Delivery challenges: Efficient and safe delivery to specific cells and tissues is still a major hurdle.
- Toxicity and immunogenicity: The immune system may recognize the Cas proteins as foreign, leading to immune reactions.
- Target site restrictions: Not all DNA sequences are accessible or suitable for CRISPR editing due to PAM sequence limitations.

Additionally, the mechanism that dictates whether Cas9 generates blunt-ended or staggered-ended breaks with overhangs in the DNA is unclear (Zuo & Liu, 2016). Furthermore, studies associate different repair outcome depending on the presence of an overhang, suggesting the necessity of genome wide tools to address these questions (Lemos et al., 2018).

Techniques to study CRISPR-induced DSB

The ability of CRISPR to create targeted DSBs has opened new avenues for gene disruption, correction, and insertion through the cell's natural DNA repair pathways. However, understanding the dynamics and outcomes of these CRISPR-induced DSBs is crucial for optimizing genome editing efficiency, minimizing off-target effects, and ensuring accurate gene modifications. Several advanced molecular and cellular techniques have been developed to detect, quantify, and characterize DSBs, as well as to monitor the repair processes following CRISPR activity. These techniques provide insights into the efficiency of DSB generation, the fidelity of the repair mechanisms, and the occurrence of unwanted off-target effects. In this context, methods such as T7 endonuclease I assays, next-generation sequencing, chromatin immunoprecipitation (ChIP), and various fluorescent reporter systems have become essential tools for researchers aiming to study CRISPR-induced DSBs. Understanding these techniques and their applications is key to refining CRISPR technology for therapeutic and research purposes.

in-vitro

In-vitro methods for studying CRISPR-induced double-strand breaks provide valuable insights into the precision and off-target effects of CRISPR-Cas9 systems by examining cleavage events outside of living cells. CIRCLE-seq (circularization for in vitro reporting of cleavage effects by sequencing) (Tsai et al., 2017) uses circularized DNA exposed

to Cas9-sgRNA complexes to identify off-target sites through sequencing after the DNA is linearized, offering high sensitivity and no requirement for living cells. CHANGE-seq (circularization for high-throughput analysis of nuclease genome-wide effects by sequencing) (Lazzarotto et al., 2020) improves upon previous methods by directly capturing and sequencing Cas9-induced DSBs in a genome-wide manner, reducing background noise and increasing accuracy in detecting off-targets. SITE-seq (selective enrichment and identification of targeted DNA ends by sequencing) (Cameron et al., 2017) employs biotinylated Cas9 to cleave cell-free DNA, enriching for and mapping DSBs at high resolution, with the advantage of not being constrained by chromatin context. Digenome-seq (in vitro Cas9-digested whole-genome sequencing) (Kim et al., 2015) relies on exposing genomic DNA to Cas9-sgRNA complexes in vitro and sequencing the resultant cleaved DNA to identify genome-wide off-target sites. Each of these methods offers high sensitivity for detecting DSBs, but their limitation lies in the lack of cellular context, which may influence the accuracy of off-target predictions in vivo.

in-cellulo

In vivo methods for studying CRISPR-induced double-strand breaks are critical for understanding the behavior of CRISPR-Cas9 in a biological context, where chromatin structure and DNA repair mechanisms play significant roles. DISCOVER-seq (discovery of in situ Cas off-targets and verification by sequencing) (Wienert et al., 2019) tracks the recruitment of endogenous DNA repair factors, such as MRE11, to detect DSBs in live cells, offering real-time insight into repair events with minimal cell manipulation. GUIDE-seq (genome-wide, unbiased identification of DSBs enabled by sequencing) (Tsai et al., 2015) integrates small double-stranded oligonucleotides at DSB sites in live cells, which are then sequenced to map both on-target and off-target effects with high sensitivity. TTISS-seq (tagmentation-based tag integration site sequencing) (Schmid-Burgk et al., 2020) combines tagmentation with the integration of sequencing tags at DSBs, enabling genome-wide detection of CRISPR off-target activity in a high-throughput manner. HTGTS (high-throughput, genome-wide translocation sequencing) (J. Hu et al., 2016) identifies DSBs by detecting chromosomal translocations, allowing the mapping of both on- and off-target breaks across the genome. BLESS (breaks labeling, enrichment on streptavidin and sequencing) (Crosetto et al., 2013) labels DSBs in fixed cells using biotinylation and then enriches and sequences these labeled fragments, providing direct evidence of DSBs in cells or tissues. There's an improved version called BLISS (breaks labeling in situ and sequencing) which doesn't require cell fixation and labels breaks in-situ, preserving the chromatin structure at the expense of a potentially lower sensitivity for low DSB frequencies. These in vivo methods offer more biologically relevant data compared to in vitro approaches, but they can be more technically challenging and may have lower throughput in some cases.

In-silico methods to predict behaviour of Cas9

There are multiple powerful in-silico tools for predicting the outcomes of CRISPR-induced DSB repair, particularly in the context of off-target effects. These methods provide insights into indel types, frequencies, and potential functional consequences, aiding researchers in minimizing off-target mutations and optimizing CRISPR precision before experimental validation. In-silico methods offer a cost-effective and time-efficient approach to narrowing down candidate off-target sites, enabling more focused experimental investigations. Additionally, as these tools are based on increasingly sophisticated algorithms and machine learning techniques, they continue to improve in accuracy, offering researchers valuable insights into guide RNA design and off-target risk assessment.

InDelphi (Shen et al., 2018) is an in-silico model designed to predict insertions and deletions (indels) resulting from CRISPR-Cas9-mediated double-strand breaks. It is based on a machine learning approach trained on large datasets of experimentally induced indels. The model primarily predicts the outcome of non-homologous end joining, the error-prone DNA repair pathway that repairs DSBs. FORECasT (Allen et al., 2019) focuses on both indels and their functional consequences, and provides predictions on the likelihood and type of indels at specific genomic sites and offers insight into the functional impact of these mutations, such as whether they result in a frameshift or non-frameshift mutation. It is particularly useful for gene-editing applications where predicting the functional effects of off-target events is critical. Lindel (Linear Deletion) (W. Chen et al., 2019) is a prediction model that uses a probabilistic framework to estimate the likelihood and types of indels at CRISPR-induced DSB sites. The model is trained on thousands of CRISPR experiments and predicts repair outcomes based on the nucleotide sequence around the DSB. Lindel focuses on small indels (up to 10 base pairs) and is capable of predicting the distribution of deletions and insertions with high accuracy.

In addition to InDelphi, FORECast, and Lindel, several other in-silico tools such as CRISPRoff (Nuñez et al., 2021), Cas-OFFinder (Bae, Park, & Kim, 2014), CRISTA (Abadi, Yan, Amar, & Mayrose, 2017), DeepCRISPR (Chuai et al., 2018), and CCTop (Stemmer, Thumberger, Sol Keyer, Wittbrodt, & Mateo, 2015) offer powerful methods to predict off-target effects. Each of these methods varies in complexity, with some incorporating chromatin data and machine learning algorithms for greater accuracy, while others focus more on sequence similarity. These tools together form a comprehensive suite for predicting off-target effects, crucial for enhancing CRISPR safety and efficiency in both research and therapeutic applications.

Summary

Tables 4, 5, 6 summarize various methods used to detect off-target effects caused by CRISPR. Each method has unique strengths and limitations based on sensitivity, specificity, in vitro versus in vivo conditions, and cost-effectiveness. These tools are critical for optimizing CRISPR technology and ensuring its safety in clinical and research applica-

tions.

Table 7 summarizes various in-silico tools for predicting the outcomes of CRISPR-induced DSB repair.

Working hypothesis and goals for Chapter 3

Despite existing many high-throughput methods to study CRISPR-generated DSB, to this date there's no such method which can determine the scission profile of a DSB genome-wide. Some studies (Lemos et al., 2018; Overbeek et al., 2016) link the presence of overhangs to different repair outcomes, highlighting the importance to characterize the scission profile of a DSB to understand how it is repaired. For this reason, we developed BreakTag, a method for discovering on- and/or off-targets of genome-editing nucleases, in particular CRISPR nucleases such as Cas9, Cas12 and variants thereof in vitro, in cells or in living organisms. BreakTag resolves the structure of the scission profile, enabling the possibility to study and design scission-based gRNA which produce precise, templated and predictable single-nucleotide insertions.

Table 4: In vitro-based methods used for the interrogation of CRISPR off-target effects.

Method	Principle	Advantages	Limitations	References
Digenome-seq	Cas9-sgRNA complexes are incubated with genomic DNA in vitro, and Cas9-induced DSBs are sequenced using whole-genome sequencing (WGS).	High sensitivity, genome-wide detection, no requirement for cell-based experiments.	Requires WGS, time-consuming and expensive, limited by in vitro conditions.	Kim et al., 2015
CIRCLE-seq	Circularized DNA is exposed to Cas9-sgRNA complexes in vitro, and off-target sites are identified through sequencing after linearization.	Does not require NHEJ activity, high sensitivity, no cell lines required.	Limited to pre-determined DNA library, less representative of in vivo conditions.	Tsai et al., 2017
CHANGE-seq	Circularization of genome, cleavage, and sequencing of linearized circles.	High-throughput, low sample input, time-efficient (when used together with a liquid handling platform) and low background.	Can be used only for in vitro nomination of Cas9 off-targets. Does not discriminate Cas9 DSB end structure.	Lazzarotto et al., 2020
SITE-seq	Uses biotinylated, recombinant Cas9 to cut cell-free DNA, and off-target cleavage sites are mapped by sequencing after enrichment of fragments.	High resolution, can identify off-targets independent of chromatin context.	Requires complex biotin-labeled Cas9, in vitro assay not representative of in vivo chromatin.	Cameron et al., 2017

Table 5: In vivo-based methods used for the interrogation of CRISPR off-target effects.

Method	Principle	Advantages	Limitations	References
GUIDE-seq	Detects DSBs by integrating oligonucleotides into breaks in living cells, followed by amplification and sequencing of integration sites.	Highly sensitive, identifies off-targets in live cells, minimal biases.	Requires transfection of double-stranded oligonucleotides, can introduce background noise in some conditions.	Tsai et al., 2015
BLESS/BLISS	Directly labels DSBs in fixed cells using biotinylation followed by enrichment and sequencing of the labeled DNA fragments.	Provides direct evidence of DSBs in fixed cells, can work on primary tissues.	Requires fixation of cells, lower sensitivity compared to other methods, not real-time.	Crosetto et al., 2013 and Yan et al., 2017
DISCOVER-seq	Utilizes the recruitment of DNA repair factors to DSB sites, such as MRE11, to identify off-target activity in live cells.	Detects physiological repair activity, live-cell assay, minimal cell manipulation.	Lower throughput, can miss off-targets where MRE11 recruitment is weak.	Wienert et al., 2019
HTGTS (High-Throughput, Genome-Wide Translocation Sequencing)	Detects chromosomal translocations that result from DSBs, allowing for identification of both on-target and off-target effects.	High sensitivity, genome-wide coverage, can detect large chromosomal rearrangements.	Complex library preparation, time-consuming, may not detect small indels.	Hu et al., 2016
TTISS-seq	Nested PCR for enrichment of integration events	Low sample input, excellent signal/noise ratio and high-throughput.	Misses off-targets repaired by end-joining independent mechanisms, cannot be adapted for base editor off-target discovery and cannot be used in animal or iPSC models. Does not discriminate Cas9 DSB end structure.	Schmid-Burgk et al., 2020

Table 6: Indel detection assays used for the interrogation of CRISPR off-target effects.

Method	Principle	Advantages	Limitations	References
T7 Endonuclease I Assay	Detects mismatches introduced by indels caused by NHEJ at off-target sites. Amplified target sequences are cleaved by T7 endonuclease and analyzed.	Fast, simple, cost-effective, good for detecting indels at known off-target sites.	Low sensitivity, only detects indels, not suitable for genome-wide off-target detection.	Vouillot et al., 2015
Deep Sequencing	Targets specific potential off-target sites by PCR amplification and uses next-generation sequencing (NGS) to detect mutations or indels.	High sensitivity, site-specific off-target detection, good for validation of predicted off-target sites.	Limited to predetermined target sites, expensive if used for genome-wide detection.	Smith et al., 2014

Table 7: In-silico tools for predicting the outcomes of CRISPR-induced DSB repair.

Method	Principle	Advantages	Limitations	References
InDelphi	Predicts the outcome of NHEJ, the types and frequencies of indels at both on-target and off-target sites.	Accurately predicts the type and frequency of indels at CRISPR-targeted sites using machine learning.	Primarily focused on NHEJ, limiting its use in contexts involving other repair pathways.	Shen et al., 2018
FORECasT	Provides predictions on the likelihood and type of indels at specific genomic sites and offers insight into the functional impact of these mutations, such as whether they result in a frameshift or non-frameshift mutation.	rovides highly accurate predictions of small indel distributions at CRISPR sites based on probabilistic modeling.	Restricted to small indels (up to 10 base pairs) and less applicable for larger genomic modifications.	Allen et al., 2018
Lindel	The model is trained on thousands of CRISPR experiments and predicts repair outcomes based on the nucleotide sequence around the DSB..	Predicts both the types of indels and their functional consequences (e.g., frameshift mutations) using deep learning.	Requires large training datasets and focuses on functional impact, which may not be necessary for all applications.	Chen et al., 2019
CRISPRoff	Incorporates information about the epigenetic context to predict where CRISPR might have lower off-target activity due to DNA being less accessible. This makes it particularly useful for predicting functional off-target effects in a cellular context.	Integrates chromatin accessibility data for a more biologically relevant prediction, reducing false positives.	Requires knowledge of chromatin states, which may not be available for all cell types.	Nuñez et al., 2021
Cas-OFFinder	Identifies potential off-target sites in a genome by searching for sequences that are similar to the guide RNA. It allows for flexible mismatch tolerance and indel tolerance at off-target sites.	High flexibility in mismatch tolerance; supports large genomes.	Generates many potential off-targets, some of which may not be biologically relevant due to not considering chromatin context or other factors.	Bae et al., 2014
CCTop (CRISPR-Cas9 Target Online Predictor)	Predicts off-targets by comparing the guide RNA to the entire genome and identifying sequences that match closely. It uses sequence similarity, with an emphasis on the PAM region and the mismatches between the guide RNA and the target.	Simple and user-friendly interface, includes ranking of off-target sites based on mismatch location.	Focuses primarily on sequence similarity without considering repair outcomes or chromatin accessibility.	Stemmer et al., 2015

Table 7: In-silico tools for predicting the outcomes of CRISPR-induced DSB repair.
(continued)

Method	Principle	Advantages	Limitations	References
CRISTA	Machine learning-based prediction tool that combines sequence information and other features, such as chromatin accessibility and DNA binding energy, to predict off-target sites. It generates a score for each potential off-target site, indicating the likelihood of CRISPR cleavage.	Incorporates both sequence context and chromatin features, providing more accurate off-target predictions.	Requires high-quality epigenetic data to maximize accuracy, which may not always be available.	Abadi et al., 2017
DeepCRISPR	Deep learning-based tool for predicting off-target activity. It integrates multiple layers of information, including sequence, structure, and chromatin accessibility, to identify off-targets with high precision. The model is trained on large experimental datasets to improve accuracy.	High precision due to deep learning and integration of various biological parameters.	Computationally intensive and requires significant training data for optimal performance.	Chuai et al., 2018

Chapter 1

dupRadar: a Bioconductor package for the assessment of PCR artifacts in RNA-Seq data

1.1 Preamble

This chapter was published in BMC Bioinformatics on October 2016.

Sayols S, [redacted], [redacted]. *dupRadar: a Bioconductor package for the assessment of PCR artifacts in RNA-Seq data.* BMC Bioinformatics. 2016 Oct 21;17(1):428. doi: 10.1186/s12859-016-1276-2. PMID: 27769170; PMCID: PMC5073875.

The source code of the algorithms implemented for this study are available at:

GitHub: `https://github.com/ssayols/dupRadar`

Bioconductor: `https://www.bioconductor.org/packages/release/bioc/html/dupRadar.html`

Supplementary material is shown in Appendix A or online under "Additional files" via the DOI link above.

1.2 Abstract

Background: PCR clonal artefacts originating from NGS library preparation can affect both genomic as well as RNA-Seq applications when protocols are pushed to their limits. In RNA-Seq however the artifactual reads are not easy to tell apart from normal read duplication due to natural over-sequencing of highly expressed genes. Especially when working with little input material or single cells assessing the fraction of duplicate reads is an important quality control step for NGS data sets. Up to now there are only tools

to calculate the global duplication rates that do not take into account the effect of gene expression levels which leaves them of limited use for RNA-Seq data.

Results: Here we present the tool dupRadar, which provides an easy means to distinguish the fraction of reads originating in natural duplication due to high expression from the fraction induced by artefacts. dupRadar assesses the fraction of duplicate reads per gene dependent on the expression level. Apart from the Bioconductor package dupRadar we provide shell scripts for easy integration into processing pipelines.

Conclusions: The Bioconductor package dupRadar offers straight-forward methods to assess RNA-Seq datasets for quality issues with PCR duplicates. It is aimed towards simple integration into standard analysis pipelines as a default QC metric that is especially useful for low-input and single cell RNA-Seq data sets.

Keywords

RNA-Seq, PCR artefacts, Duplication rate, Single cell RNA-Seq, Bioconductor, Quality control tool.

1.3 Background

1.3.1 Sources of duplicate reads in Next-Generation sequencing

Next Generation Sequencing has become a standard assay for many questions in molecular biology. It involves the preparation of sequencing libraries out of fragments of DNA or RNA molecules and sequencing adapters, PCR amplification and sequencing. The calculation of the fraction of duplicate reads has become a standard step for quality control in NGS experiments, as high duplication rates can hint towards problems in different steps of the NGS library preparation process. In particular, the variety of molecules that can be seen after sequencing correlates with minute amounts of input material ("molecular bottleneck") or too many PCR cycles. This can lead to low library complexity. Furthermore, overloading of a sequencing flow cell may result in optical duplicates or problems with reagents can lead to elevated duplication rates. Duplicate reads can also be caused by a combination of complex genomic loci and insufficient read length or even issues with the reference genome.

In RNA-Seq however it is common to have high overall fractions of duplicate reads not due to technical artifacts. This is known and discussed in the community (e.g. (Griffith et al., 2015; "Should We Remove Duplicated Reads In Rna-Seq ?" n.d.; Williams et al., 2014)) but is still sometimes misunderstood (X. Li, Nair, Wang, & Wang, 2015). Often the top 5% of expressed genes take up more than 50% of all reads in a common RNA-Seq dataset (Tarazona, García-Alcalde, Dopazo, Ferrer, & Conesa, 2011). Read counts for highly expressed genes easily surpass the threshold of 1 read per bp of the exon model, at which read duplication is inevitable. Due to a number of biases in the process of RNA-Seq (Van Dijk, Jaszczyszyn, & Thermes, 2014) read duplication in RNA-Seq starts even

below the 1 read per bp threshold. In RNA-Seq duplication originating from technical artifacts such as described before are confounded with natural read duplication due to highly expressed genes, hence overall duplication rate is not a suitable measure for quality control purposes.

1.3.2 Effects and treatment of PCR duplicates in RNA-Seq data

In assays involving genomic DNA (e.g. resequencing, ChIP-Seq) reads marked as duplicates with tools such as the established picard ("Broadinstitute/picard," March 28, 2014/2023), or the more recent bamUtil dedup ("BamUtil," n.d.) and biobambam (Tischler & Leonard, 2014) are commonly removed before further analyzing the data. In RNA-Seq studies with the aim to quantify expression however the situation is more complex. Duplicate reads also arise naturally in highly expressed genes, hence complete removal of duplicate reads affects estimation of expression levels. Tools such as eXpress (Roberts & Pachter, 2013) attempt to tackle related problems by smoothing the read coverage. However this approach is not applicable to situations in which systematic over-estimation of read counts on a large fraction of genes exists.

1.3.3 Detection of duplicate reads in Next-Generation sequencing

Currently there are many tools available that address the overall duplication rates or read frequencies of NGS data sets ("Babraham Bioinformatics - FastQC A Quality Control tool for High Throughput Sequence Data," n.d.; "Broadinstitute/picard," March 28, 2014/2023; DeLuca et al., 2012; "FASTX-Toolkit," n.d.; Garc'ıa-Alcalde et al., 2012; "Qualimap," n.d.; Schmieder & Edwards, 2011; L. Wang et al., 2012). Commonly, the non-systematic detection of PCR artefacts in RNA-Seq analysis relies on the visual inspection in a genome browser, where problematic data sets show typical stacked reads in loci with low and medium expression.

Here we present dupRadar, a tool to systematically detect anomalous duplication rate profiles and simplify the task of identification of data sets that require further in-depth assessment.

1.4 Implementation

dupRadar relates the duplication rate and length normalized read counts of every gene to model the dependency of this two variables. It requires a BAM file with mapped and duplicate marked reads, and a gene model in GTF format. Internally dupRadar calls the featureCounts function from the RSubread package (Liao, Smyth, & Shi, 2013) several times, to count all and the duplicate marked reads per genes, both uniquely as well as multi-mapping reads. Furthermore dupRadar calculates the per gene duplication rate and reads per kilobase (RPK) as a proxy for relative gene expression. The resulting calculations

are stored in a data frame which can be directly passed on to different visualization functions, which show the dependence of the duplication rate on gene expression. Besides fitting a logistic model to the dependency between duplication rate and RPK, dupRadar estimates the baseline duplication rate for lowly expressed genes which can be used as an indicator for general problems inside a data set.

Additionally, the data frame can be used for further processing of the data in standard read count based differential gene expression tools (Love, Huber, & Anders, 2014; Ritchie et al., 2015; Robinson, McCarthy, & Smyth, 2010), or for other purposes such as the detection of genes that are exclusively covered by multi-mapping reads.

To enable interpretation of the dependency of duplication rate and gene expression, dupRadar currently includes various visualization functions. Beyond that the vignette of the Bioconductor package contains examples for customised plots using dupRadar. For the sake of usability, it includes wrappers for some common tools for duplicate marking in order to streamline the processing of the data sets.

To demonstrate the effect of PCR artefacts also on downstream analysis we perform a simulation study based on the Airway dataset commonly used in Bioconductor courses (Himes et al., 2014) (results in Additional file 1: Figure S1). To obtain a comparable dataset with a high fraction of duplicate reads, we subsampled the reads of the original library to different fractions (50 and 10%), and applied an amplification step to the remaining ones to match again the number of reads in the original library, thus creating simulated libraries with respectively 50 and 90 % of duplicate reads, following a Poisson process to simulate what happens in a PCR..Subsequently we perform differential expression analysis using edgeR (Robinson et al., 2010) for both the original data as well as the datasets with 50 and 90% of artificially added duplicate reads.

1.5 Results and discussion

Recently, RNA-Seq protocols were improved considerably, leading to less technical duplicates and the linked issues. Still in our experience possible problems are worth to be checked for by default, especially if protocols are pushed to or beyond their boundaries or more recent low-input or single cell RNA-Seq protocols are used.

To demonstrate the usage of dupRadar we apply a typical work flow for selected single read RNA-Seq data sets from the study of Marinov et al. (Marinov et al., 2014) ranging from single cells to cell pools to bulk RNA data. We map reads using STAR (Dobin et al., 2013) and mark duplicate reads using BamUtil dedup ("BamUtil," n.d.). Together with the human reference gene annotation GTF included in the iGenomes collection for the UCSC hg19 build ("iGenomes," n.d.), we use the resulting bam files as input for dupRadar's duplication rate calculation function. As an example Table 1.1) contains the entries from a sample of 10 genes out of the full set for the library 13276 (SRR764800). We supply instructions to regenerate the results in the supplement (Additional file 2: Methods

Table 1.1: Example values for a sample of 10 genes from the library 13276

ID	geneLength	allCounts	filteredCounts	dupRate	dupsPerId	RPK
LOC100288069	1371	17	15	0.12	2	12.40
LINC00115	1317	28	28	0.00	0	21.26
LOC643837	9233	281	246	0.12	35	30.43
FAM41C	1706	1	1	0.00	0	0.59
LOC100130417	496	0	0	NA	0	0.00
SAMD11	2554	0	0	NA	0	0.00
NOC2L	2800	329	273	0.17	56	117.50
KLHL17	2564	2	2	0.00	0	0.78
ISG15	666	590	271	0.54	319	885.89
AGRN	7326	3	3	0.00	0	0.41

1, Additional file 3: Methods 2, Additional file 4: Table S1.).

Based on the duplication rates, we generate the main visualizations of dupRadar in Fig. 1.1. The effects of oversequencing libraries of limited complexity in cases of little input material as well as an example for a bulk RNA-Seq dataset without any traces of PCR duplicates. The given plots indicate the duplication rate in relation to the gene expression. Ideally single read RNA-Seq experiments at common read depths are expected to show low duplication rates for lowly expressed genes in the bottom left of the plot, with the duplication rate rising as the expression level approaches the 1 read/bp boundary. Beyond this threshold genes are covered almost completely with reads marked as duplicates due to their high expression levels (e.g. Fig. 1.1c). Data sets based on lower amount of input material show the effects of limited complexity of the library, resulting in higher duplication rates already at lowly expressed genes, leading to the majority of data points being shifted upwards to higher duplication rates also for lowly expressed genes (e.g. Fig. 1.1d). Similar situations can be observed for data sets with actual PCR artifacts. DupRadar does not define fixed thresholds for acceptable data quality on purpose, as PCR duplication rate can be influenced by various parameters. However already low levels of PCR artefacts can have an influence on downstream analysis and interpretation of data.

Although paired-end libraries facilitate the distinction between duplicates due to adding the fragment length as an extra variable to distinguish molecules, the problem is not completely solved. For typical dupRadar plots of paired-end libraries see Additional file 5: Figure S2. The recent introduction of unique molecular identifiers (UMI) during library preparation, allows for exact distinction of technical and biological duplicates and therefore also the removal of technical duplicates (Dobin et al., 2013), which alleviates the described problem on the side of experimental procedures.

To assess the impact of excess PCR amplification on downstream analysis in RNA-Seq studies we simulated data sets with defined amounts of PCR artifacts (Additional file 1: Figure S1 and Additional file 6: Methods 3) based on good quality original data (Himes et al., 2014), and subsequently performed differential expression both on the original data as well as the data with simulated PCR problems. While there is a large overlap of 1199 genes that are differentially expressed in both the good and the bad data, the analysis shows that PCR artefacts introduce both high numbers of false positive (124) and false negative (720) differentially expressed genes.

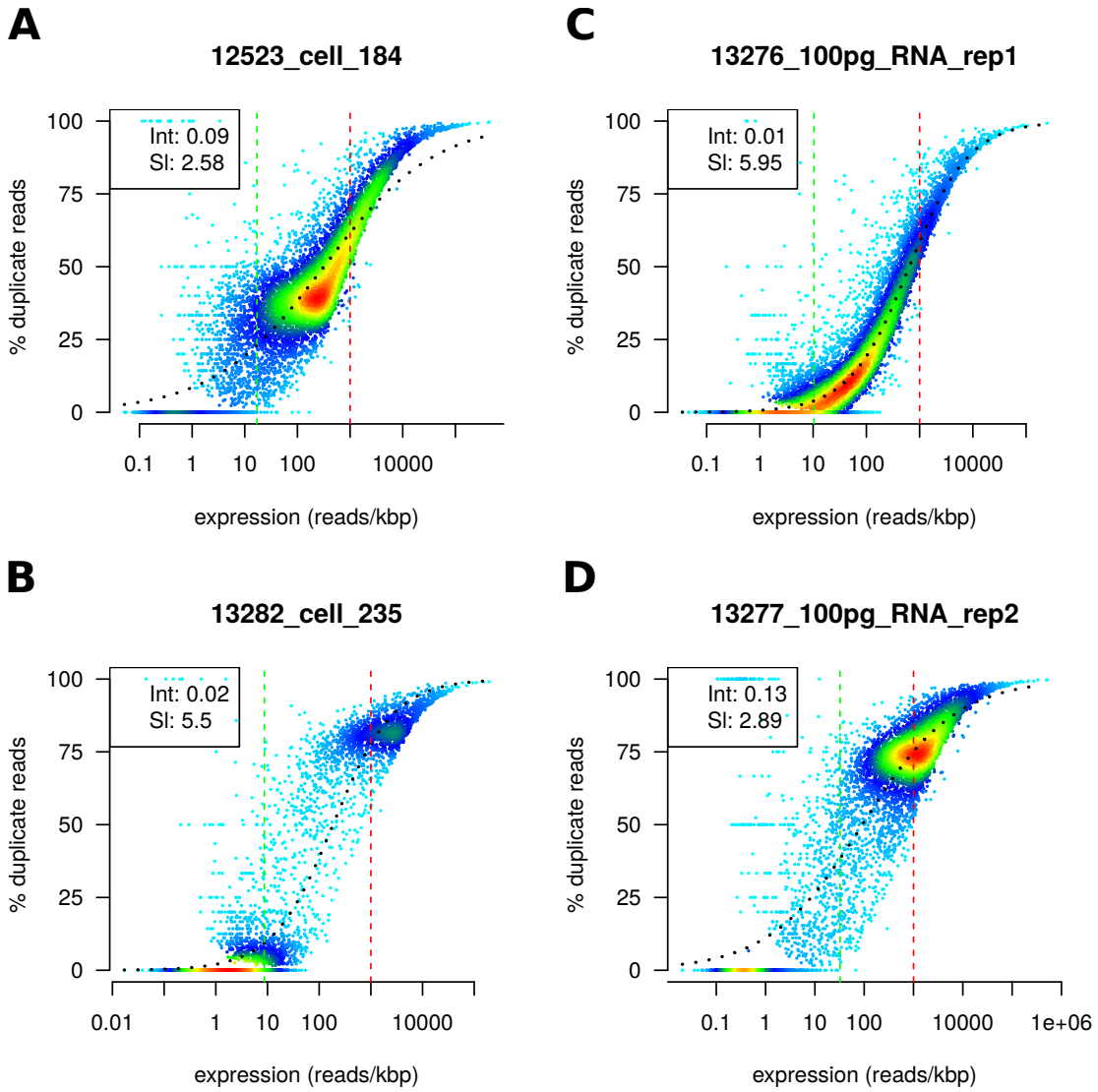


Figure 1.1: Legends shows the intercept and slope of a fitted logit model. (a) single cell experiment with relatively low duplication rates and most of the genes detected; (b) single cell experiment with most of the genes undetected and high duplication rate on the detected ones. c RNA-seq experiment pushing the protocol to only 100 pg of input material, with low duplication rates and relatively good identification of genes. d same RNA-seq experiment, showing over-sequencing due to higher sequencing depth of the library

Choice of the aligner as well as of the reference annotation both influence read mapping, quantification and downstream analyses in RNA-Seq experiments (Engström et al., 2013; S. Zhao & Zhang, 2015). On gene level, differences between aligner and annotation can also be observed in dupRadar results, however globally in our experience the assessment of library quality does not differ depending on the these parameters. We recommend not to make the choice of read mapper or reference annotation dependent on the dupRadar step.

1.6 Conclusions

The Bioconductor package dupRadar offers straightforward methods to assess RNA-Seq datasets for problems with duplicate reads and is aimed towards simple integration into standard analysis pipelines as a default QC metric.

While dupRadar serves as a diagnostics method for PCR duplicates, we regard the issue of correction for these artefacts as yet unsolved, with a potential to extend dupRadar with correction functions. Currently we advise colleagues to treat with caution RNA-seq data strongly affected by technical duplicates and repeat library preparation and sequencing if possible. Furthermore the simulation results suggest that even consistent levels of PCR artifacts over all samples of a project do not cancel out and may lead to wrong conclusions in the downstream analysis of data.

Similar effects comparable to over-sequencing of highly expressed genes are implicated for certain types of enrichment-based assays (e.g. ChIP-Seq of a specific transcription factor with high read-depths). Suitability of dupRadar in this area remains to be explored.

1.7 Declarations

1.7.1 Additional files

Supplementary material is available online under “Additional files” via the DOI link above.

1.7.2 Abbreviations

bp: Base pair; ChIP: Chromatin immunoprecipitation; NGS: Next-generation sequencing; PCR: Polymerase chain reaction; QC: Quality control; RPK: Reads per kilobase; UMI: Unique molecular identifiers

1.7.3 Acknowledgements

We thank the members of the Core Facilities at the Institute for Molecular Biology, especially [redacted], [redacted], [redacted], [redacted] and [redacted], as well as [redacted] from the Computational Biology Group at

Boehringer Ingelheim for discussion, input and proof-reading. We also would like to thank three anonymous reviewers whose suggestions helped to improve our manuscript substantially.

1.7.4 Funding

Publication of this article was funded by Boehringer Ingelheim Pharma GmbH & Co KG.

1.7.5 Authors' contributions

SS and HK conceived of the project. SS, HK designed the software. SS, DS and HK implemented the software. SS and HK tested the software. SS and HK drafted the manuscript. All authors read and approved the final manuscript.

1.7.6 Competing interests

SS reports personal fees from Boehringer Ingelheim Pharma GmbH & Co KG outside of the submitted work. HK reports his directly employed by Boehringer Ingelheim Pharma GmbH. All other authors declare no competing interests.

1.7.7 Consent for publication

Not applicable.

1.7.8 Ethics approval and consent to participate

Not applicable.

1.7.9 Author details

1 Bioinformatics Core Facility, Institute of Molecular Biology, Ackermannweg 4, 55128 Mainz, Germany.

2 Technische Hochschule Bingen, Berlinstraße 109 Bingen am Rhein 55411, Germany.

3 Target Discovery Research, Boehringer Ingelheim Pharma GmbH & Co KG, Birkendorferstraße 67, 88397 Biberach an der Riß, Germany.

Received: 16 January 2016 Accepted: 21 September 2016

Chapter 2

rrvgo: a Bioconductor package for interpreting lists of Gene Ontology terms

2.1 Preamble

This chapter was published in microPublication Biology on April 2023.

Sayols, S (2023). *rrvgo: a Bioconductor package for interpreting lists of Gene Ontology terms.* microPublication Biology. 10.17912/micropub.biology.000811.

The source code of the algorithms implemented for this study are available at:

GitHub: `https://github.com/ssayols/rrvgo`

Bioconductor: `https://www.bioconductor.org/packages/release/bioc/html/rrvgo.html`

Supplementary material is shown in Appendix B.

2.2 Abstract

Gene Ontology (GO) annotation is often used to guide the biological interpretation of high-throughput omics experiments, e.g. by analysing lists of differentially regulated genes for enriched GO terms. Due to the hierarchical nature of GOs, the resulting lists of enriched terms are usually redundant and difficult to summarise and interpret. To facilitate the interpretation of large lists of GO terms, I developed rrvgo, a Bioconductor package that aims at simplifying the redundancy of GO lists by grouping similar terms based on their semantic similarity. rrvgo also provides different visualization options to guide the interpretation of the summarized GO terms. Considering that several software tools have

been developed for this purpose, rrvgo is unique at combining powerful visualizations in a programmatic interface coupled with up-to-date GO gene annotation provided by the Bioconductor project.

2.3 Description

2.3.1 Introduction

Structured vocabularies such as GO ("The Gene Ontology Resource," 2019) are important tools for the biological interpretation of high-throughput omics experiments. Due to the hierarchical nature of GO annotation, lists of enriched GO terms are usually large and redundant. One approach to simplify GO analysis is to use GO Slims (Carbon et al., 2009) representing a subset of the full GO. However, using such limited GO versions may hide interesting findings represented by more specific terms which were excluded. Hence, methods such as semantic similarity may better account for the complex structure of the GO graph and be more effective (Pesquita et al., 2009).

Several online tools to compute semantic similarity between GO terms exist, such as REVIGO (Supek et al., 2011). The accessibility of such tools comes at a price: they usually offer a limited programmatic interface difficult to integrate into pipelines, and provide pre-packaged GO annotations which cannot be overridden. Offline tools also exist, such as clusterProfiler (G. Yu et al., 2010) or ViSEAGO (Brionne et al., 2019) including useful but limited exploration capabilities.

Conveniently, the Bioconductor project (Huber et al., 2015) implements several semantic similarity methods and provides up-to-date GO annotations for a number of model organisms, along with the possibility of preparing custom annotations. I developed rrvgo to integrate in a single package access to the semantic similarity methods and annotations implemented in the Bioconductor project, coupled with highly effective visualizations, providing a one-stop-shop for the interpretation of large lists of GO terms in R.

2.3.2 Implementation

rrvgo requires a list of GO terms, usually identified in an overrespresentation analysis, from any of the three orthogonal taxonomies: Biological Process (BP), Molecular Function (MF) or Cellular Compartment (CC). Each term in the list may optionally include a score (eg. a minus log-transformed p-value). In this case, rrvgo will prefer terms with higher scores to identify the most representative term of a group; otherwise higher-level terms (ie. those comprising more genes) are preferred by default.

rrvgo uses the GOSemSim package (G. Yu et al., 2010) under the hood, which implements methods to compute semantic similarity between pairs of GO terms, and the OrgDb packages of the organisms of interest provided within Bioconductor.

2.3.3 Similarity measures

The application of semantic similarity methods, originally used in Natural Language Processing, to ontological annotation has already been investigated (Lord et al., 2003). Some of these measures are based on the calculation of the term's Information Content (J. J. Jiang & Conrath, 1997; Lin, 1998; Resnik, 1999; Schlicker et al., 2006) or graph-based (J. Z. Wang et al., 2007) and are implemented in the GOSemSim package.

rrvgo uses the similarity between pairs of terms to compute the matrix of dissimilarities. The terms are then clustered using complete linkage, and the cluster is cut at the desired threshold, picking the term with the highest score as the representative of each group.

2.3.4 Organisms supported and creating a custom OrgDb

As of Bioconductor 3.16, there are OrgDb packages available for the most common organisms used in the lab. Consult the OrgDb BiocView for a full list of current OrgDb packages. It is expected that the list fluctuates between versions, but most common species may be very well supported while the project remains healthy.

For organisms not having an OrgDb package in Bioconductor, it is still possible to create custom OrgDb packages using the AnnotationForge package (Carlson & Pag‘es, 2019).

2.3.5 Visualizations

rrvgo provides visualizations of the reduced terms as: (i) scatter plot represented by the first 2 components of a PCoA of the dissimilarity matrix; (ii) space-filling visualization (treemap) of terms grouped by the representative term; (iii) word cloud emphasizing frequent words in GO terms; and (iv) heatmap representation of the similarity matrix. Figure 2.1 A-D.

Alternatively, the results can be interactively explored using the companion shiny app (Figure 2.1 E).

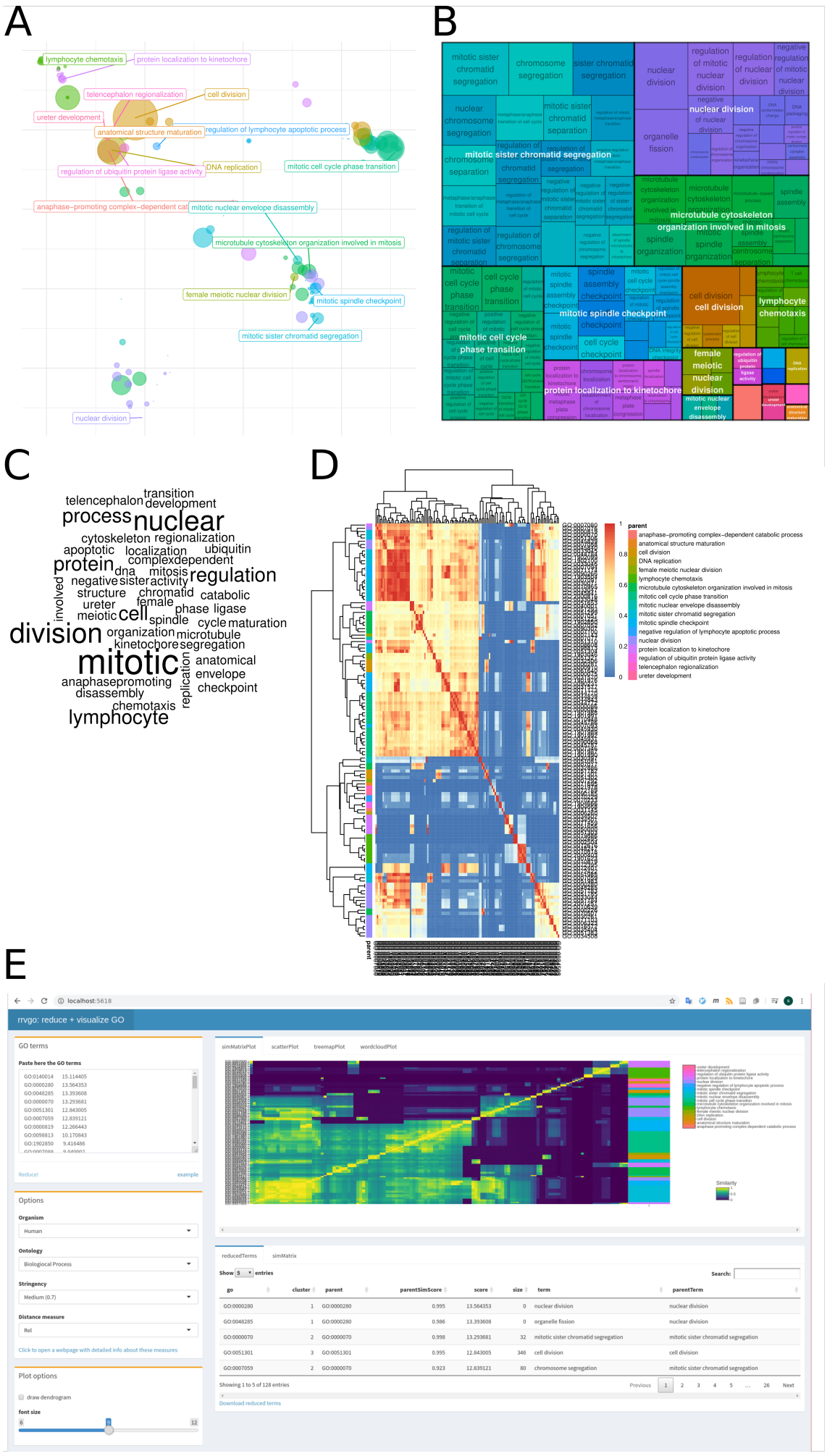

A
lymphocyte chemotaxis
protein localization to kinetochore
telencephalon regionalization
cell division
ureter development
anatomical structure maturation
regulation of lymphocyte apoptotic process
regulation of ubiquitin protein ligase activity
DNA replication
mitotic cell cycle phase transition
anaphase−promoting complex−dependent cat
mitotic nuclear envelope disassembly
microtubule cytoskeleton organization involved in mitosis
female meiotic nuclear division
mitotic spindle checkpoint
mitotic sister chromatid segregation
nuclear division
B
mitotic sister chromatid segregation
nuclear division
organelle fission
nuclear chromosome segregation
chromosome segregation
sister chromatid segregation
mitotic sister chromatid separation
regulation of mitotic nuclear division
negative regulation of nuclear division
regulation of nuclear division
negative regulation of mitotic nuclear division
microtubule cytoskeleton organization involved in mitosis
microtubule cytoskeleton organization in mitosis
spindle organization
mitotic spindle organization
spindle assembly
centrosome separation
mitotic cell cycle phase transition
cell cycle phase transition
mitotic spindle assembly checkpoint
spindle assembly checkpoint
cell cycle checkpoint
mitotic spindle checkpoint
cell division
lymphocyte chemotaxis
mitotic cell cycle phase transition
female meiotic nuclear division
protein localization to kinetochore
mitotic nuclear envelope disassembly
C
transition
telencephalon
development
process
nuclear
cytoskeleton
regionalization
apoptotic
localization
ubiquitin
protein
complex
dependent
dna
mitosis
regulation
involved
negative
sister
activity
structure
chromatid
catabolic
ureter
female
phase
ligase
meiotic
cell
spindle
cycle
maturation
division
organization
microtubule
kinetochore
segregation
mitotic
anatomical
anaphase
promoting
envelope
replication
checkpoint
disassembly
chemotaxis
lymphocyte
D
parent
anaphase−promoting complex−dependent catabolic process
anatomical structure maturation
cell division
DNA replication
female meiotic nuclear division
lymphocyte chemotaxis
microtubule cytoskeleton organization involved in mitosis
mitotic cell cycle phase transition
mitotic nuclear envelope disassembly
mitotic sister chromatid segregation
mitotic spindle checkpoint
negative regulation of lymphocyte apoptotic process
nuclear division
protein localization to kinetochore
regulation of ubiquitin protein ligase activity
telencephalon regionalization
ureter development
E
localhost:5618
rrvgo: reduce + visualize GO
GO terms
Paste here the GO terms
Reduce!
example
Options
Organism
Human
Ontology
Biological Process
Stringency
Medium (0.7)
Distance measure
Rel
Click to open a webpage with detailed info about these measures
Plot options
draw dendrogram
font size
simMatrixPlot
scatterPlot
treemapPlot
wordcloudPlot
reducedTerms
simMatrix
Show 5 entries
Search:
go
cluster
parent
parentSimScore
score
size
term
parentTerm
GO:0000280 1 GO:0000280 0.995 13.564353 0 nuclear division nuclear division
GO:0048285 1 GO:0000280 0.986 13.393608 0 organelle fission nuclear division
GO:0000070 2 GO:0000070 0.998 13.293681 32 mitotic sister chromatid segregation mitotic sister chromatid segregation
GO:0051301 3 GO:0051301 0.995 12.843005 346 cell division cell division
GO:0007059 2 GO:0000070 0.923 12.839121 80 chromosome segregation mitotic sister chromatid segregation
Showing 1 to 5 of 128 entries
Previous
Next
Download reduced terms

Figure 2.1: Different visualizations of the reduced terms provided by rrvgo.(A) scatter plot represented by the first 2 components of a PCoA of the dissimilarity matrix. (B) space-filling visualization (treemap) of terms grouped by the representative term. (C) word cloud emphasizing frequent words in GO terms. (D) heatmap representation of the similarity matrix. (E) Companion Shiny App for interactive visualization of similarity between GO terms.

2.3.6 Conclusion

rrvgo is a Bioconductor package that aims at providing a one-stop-shop for the biological interpretation of large lists of GO terms. It integrates access to semantic similarity methods and visualization in coherent and intuitive manner. This software is heavily influenced by REVIGO, mimicking a good part of its core functionality and some of the visualizations. The strength of rrvgo is its programmatic interface coupled with up-to-date GO gene annotation provided by the Bioconductor project.

2.4 Reagents

rrvgo is available as a Bioconductor package at `http://bioconductor.org/packages/rrvgo/` and released under the GPL-3 License. The version of the software used in this article (rrvgo 1.10.0, Bioconductor 3.16) is also available in the Extended Data Section.

2.5 Declarations

2.5.1 Acknowledgements

I would like to thank the members of the IMB Core Facilities for discussion, input and proof-reading. I also would like to thank [redacted] (California Institute of Technology) for taking the necessary time and effort to review the manuscript.

2.5.2 Extended Data

Description: Source Package. Resource Type: Software. File: rrvgo_1.10.0.tar.gz. DOI: 10.22002/xa9g7-5mm38

Supplementary material is shown in Appendix B.

2.5.3 Funding

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project-ID 393547839 – SFB 1361.

2.5.4 Author Contributions

Sergi Sayols: conceptualization, software, writing - original draft.

Reviewed By: [redacted]

2.5.5 History

Received March 20, 2023 Revision Received April 13, 2023 Accepted April 17, 2023

2.5.6 Copyright

2.5.7 Citation

Sayols, S (2023). *rrvgo: a Bioconductor package for interpreting lists of Gene Ontology terms.* microPublication Biology. 10.17912/micropub.biology.000811

Chapter 3

Linking CRISPR–Cas9 double-strand break profiles to gene editing precision with BreakTag

3.1 Preamble

This chapter was published in Nature Biotechnology on May 2024.

[redacted][1*], Sergi Sayols[1*], [redacted][2], [redacted][1], [redacted][1], [redacted][1,3], [redacted][1,2] (2024). *Linking CRISPR–Cas9 double-strand break profiles to gene editing precision with BreakTag.* Nature Biotechnol (2024). https://doi.org/10.1038/s41587-024-02238-8

[1]Institute of Molecular Biology (IMB); Mainz, Germany

[2]Department of Biology, Medical School, University of Patras; Patras, Greece

[3]Johannes Gutenberg University (JGU); Mainz, Germany

[*]These authors contributed equally to this work

Corresponding author. Email: [redacted]

GitHub: https://github.com/roukoslab/breaktag

GitHub: https://github.com/roukoslab/breakinspectoR

Supplementary figures are shown in Appendix C.

Supplementary tables are available online under "SUPPLEMENTARY DATA" via the DOI link above.

3.2 Abstract

Cas9 can cleave DNA in both blunt and staggered configurations, resulting in distinct editing outcomes, but what dictates the type of Cas9 incisions is largely unknown. In this study, we developed BreakTag, a versatile method for profiling Cas9-induced DNA double-strand breaks (DSBs) and identifying the determinants of Cas9 incisions. Overall, we assessed cleavage by SpCas9 at more than 150,000 endogenous on-target and off-target sites targeted by approximately 3,500 single guide RNAs. We found that approximately 35% of SpCas9 DSBs are staggered, and the type of incision is influenced by DNA:gRNA complementarity and the use of engineered Cas9 variants. A machine learning model shows that Cas9 incision is dependent on the protospacer sequence and that human genetic variation impacts the configuration of Cas9 cuts and the DSB repair outcome. Matched datasets of Cas9 and engineered variant incisions with repair outcomes show that Cas9-mediated staggered breaks are linked with precise, templated and predictable single-nucleotide insertions, demonstrating that a scission-based gRNA design can be used to correct clinically relevant pathogenic single-nucleotide deletions.

3.3 Main

CRISPR–Cas9 has revolutionized genome editing in both basic and applied biomedical research as a means toward programmable, targeted and precise correction of genetic diseases (Cong et al., 2013; Gasiunas, Barrangou, Horvath, & Siksnys, 2012; Jinek et al., 2012; Mali et al., 2013). Although the DNA-targeting specificity of CRISPR–Cas9 has been enhanced by redesigning guide RNAs (gRNAs) and engineering variants with higher fidelity, Cas9 template-free editing in eukaryotic cells has not yet been controlled at the required level for high-precision use in therapeutic applications (J. Y. Wang & Doudna, 2023).

Cas9-mediated DNA editing was initially thought to result in random insertions and deletions (indels); however, mounting evidence indicates that the repair of Cas9-induced DNA breaks is not random but, rather, is strongly dependent on the sequence context of the target site (Chakrabarti et al., 2019; Overbeek et al., 2016; Shen et al., 2018; Taheri-Ghahfarokhi et al., 2018). Large datasets coupling CRISPR–Cas9 target sequences with their respective editing results have been used to develop models for predicting repair outcomes in mammalian cells (Allen et al., 2019; W. Chen et al., 2019; Leenay et al., 2019; Molla & Yang, 2020; Shen et al., 2018). Despite this progress, it is still unclear how Cas9 target sequences mechanistically influence DNA repair outcomes. One possible scenario is that different types of Cas9 incisions are associated with distinct editing outcomes, as shown in individual cases of staggered Cas9-mediated DNA double-strand breaks (DSBs) linked to single-nucleotide insertions (Lemos et al., 2018; Santiago Gisler et al., 2019; Shi et al., 2019; Shou, Li, Liu, & Wu, 2018). Although it is now well accepted that Cas9 can cleave DNA in both blunt and staggered configurations (Jones et al., 2021; Lemos et

al., 2018; Shi et al., 2019; Shou et al., 2018), where, how and at what frequencies these alternative DSB end structures are formed remains unknown. Moreover, the impact of genetic variation on Cas9 scission and editing outcomes has not been investigated—an important gap in knowledge as CRISPR-based therapeutics become increasingly achievable. The scarcity of systematic information on the outcome of Cas9 nuclease function can be attributed mainly to the lack of scalable tools that can simultaneously measure the frequency, location and structure of Cas9-induced DNA breaks.

To address this issue, we developed a next-generation sequencing (NGS)-based methodology, called BreakTag, to comprehensively profile the genome-wide DSB landscape of Cas nucleases along with their end structures at nucleotide resolution. Using BreakTag, we characterized the Cas9 scission at a total dataset of approximately 150,000 endogenous loci targeted by approximately 3,500 single guide RNAs (sgRNAs), and we identified determinants of Cas9 incisions. Furthermore, we investigated the impact of human genetic variation on Cas9 scission profile, and we identified Cas9 variants with biases in cleavage configuration and alternate sequence determinants. Finally, we devised a machine learning model to survey pathogenic single-nucleotide deletions that can be corrected by exploring sequence determinants of staggered cleavage and the predictability of insertions. Our findings establish that the predictability and precision of Cas9-mediated genome editing is mechanistically linked to the Cas9 incision structure and suggest that the flexible cut profile of Cas9, along with engineered nuclease variants with skewed scission profiles, can be harnessed for precise and personalized indel engineering.

3.4 Results

3.4.1 BreakTag systematically profiles genome-wide Cas9 activity

To characterize and identify the determinants of the Cas9 scission profile, we developed BreakTag, an efficient method for unbiased, high-throughput and systematic profiling of Cas9-mediated DSBs. BreakTag is a highly scalable protocol that maps free DSB ends in genomic DNA (gDNA) digested by ribonucleoproteins (RNPs) in vitro in four simple steps: (1) an end repair/A-tailing step prepares the ends for (2) ligation with an adaptor with a unique molecular identifier (UMI) for DSB count and a sample barcode for sample multiplexing, followed by (3) tagmentation with Tn5 transposase and (4) polymerase chain reaction (PCR) amplification of ligated fragments (Fig. 3.1a and Methods). The DSB enrichment step occurs during PCR, yielding a fast (<6h for ready-to-sequence libraries), highly scalable and cost-efficient method for mapping CRISPR nuclease DSBs genome wide. DSB reads start at the cut site, and read directionality is preserved with each side of the break mapping to opposite strands (Fig. 3.1b). Moreover, the end repair step in our experimental procedure enables the enrichment of DSBs containing single-stranded DNA (ssDNA) overhangs, allowing off-target nomination of staggered-cleaving nucleases such as Cas12a with the same protocol (Extended Data Fig. C.1a). We partner BreakTag with

BreakInspectoR, a bioinformatics pipeline for identifying and counting Cas9-induced DSBs in BreakTag data (Extended Data Fig. C.1b,c; see Data, Materials and Code availability sections for links to the code).

To benchmark BreakTag against previously developed tools, we profiled the off-target landscape of 46 sgRNAs (Lazzarotto et al., 2020) (Supplementary Table 1) targeting 12 clinically relevant genes with Streptococcus pyogenes (SpCas9, hereafter 'Cas9'). We observed a wide range of off-targets, identifying sgRNAs with either high specificity or promiscuity (for example, CXCR4 site 2: 10 off-targets; PDCD1 site 12: 9,328 off-targets) (Extended Data Fig. C.1d and Supplementary Table 2). Of note, BreakTag showed excellent reproducibility across different gRNAs commonly used to benchmark off-target mapping tools (Extended Data Fig. C.1e). To benchmark BreakTag, we compared the lists of off-targets nominated by DIGENOME-seq (Kim & Kim, 2018) and CIRCLE-seq (Tsai et al., 2017). BreakTag identified previously characterized off-targets but also sites that were absent in DIGENOME-seq and CIRCLE-seq datasets (Extended Data Fig. C.2a). Furthermore, we identified an excellent correlation between the number of sites nominated by BreakTag and CHANGE-seq, an improved version of CIRCLE-seq (Pearson r=0.8862, P<0.0001) (Extended Data Fig. C.2b). We performed targeted deep sequencing of off-targets nominated by DIGENOME-seq, CIRCLE-seq and BreakTag to validate bona fide Cas9 unintended mutations, and we observed that most sites that showed editing were nominated by all three methods (Extended Data Fig. C.2a). We next tested BreakTag against GUIDE-seq, a sensitive in cellulo method that relies on the incorporation of double-stranded DNA (dsDNA) donor tags at the cut site (Tsai et al., 2015) over 27 matching gRNAs (Lazzarotto et al., 2020). We observed a complete overlap of off-targets nominated with BreakTag and GUIDE-seq in 19 out of 27 tested gRNAs (Extended Data Fig. C.2c). Approximately 85% of all targets nominated by GUIDE-seq were also nominated by BreakTag across all tested gRNAs (Extended Data Fig. C.2c). Of note, we observed an excellent correlation between the number of off-targets nominated per gRNA for the tested methods (r=0.72) (Extended Data Fig. C.2d).

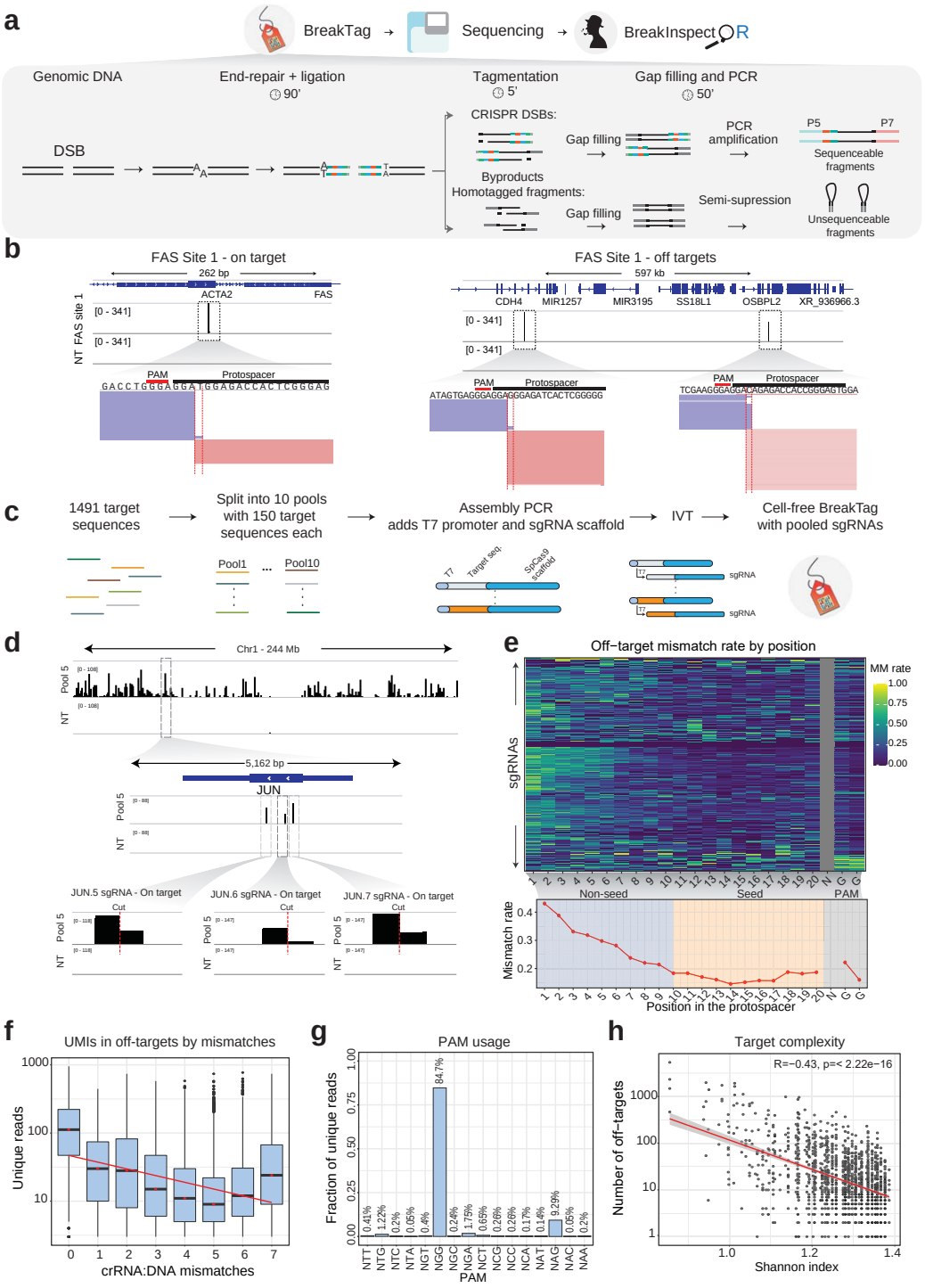


Figure 3.1: BreakTag profiles CRISPR on- and off-target DSBs. **a** Scheme depicting the experimental workflow for BreakTag (see Methods). **b** Representative IGV snapshot showing processed BreakTag data of the on-target DSB of the "FAS site 1" gRNA (left) and two off-target sites (right). Zoomed-in views of the cut site (red dotted lines) and raw mapped reads (blue/pink rectangles) are shown below. NT: nontarget control. gDNA from U2Os cells was used. **c** HiPlex BreakTag strategy. Previously reported genomic Cas9 target sequences 7 were bioinformatically split into 10 pools, each containing 150 sequences. A T7 promoter sequence was added to the 5' end

of each sgRNA protospacer, and a Cas9 sgRNA scaffold sequence at the 3' end by a PCR assembly reaction, which generates a dsDNA template for T7 in vitro transcription (IVT). T7-transcribed sgRNAs were used for BreakTag with Cas9 in gDNA from HepG2 cells. **d** IGV snapshot of chromosome 1, depicting cleaved sites for Pool 5 of the HiPlex1 dataset. Zoomed-in views of on-target DSBs of sgRNAs targeting the JUN gene are shown below. **e** Top: heatmap depicting crRNA:DNA mismatch accumulation along the protospacer of 92,375 off-target sites identified by BreakTag on 1,418 sgRNAs in the HiPlex 1 dataset. Bottom: a plot of the average mismatch rate along the protospacer. **f** Number of unique reads after deduplication using unique molecular identifiers for identified target sites containing 0–7 crRNA:DNA mismatches. **g** Percentage of unique reads for identified target sites containing noncanonical PAM sequences. **h** Correlation between the number of measured off-target cutting events and sequence complexity of the target site measured according to the Shannon index.

To further investigate the determinants of CRISPR–Cas9 off-target activity, we used the scalability of BreakTag to develop HiPlex BreakTag, which takes advantage of high-throughput enzymatic sgRNA synthesis and the pooling of several reactions. We split 1,491 previously described sgRNA sequences targeting human genes (hereafter referred to as the 'HiPlex1' library) (Chakrabarti et al., 2019) into 10 pools (~150 sequences per pool) (Supplementary Table 1) and produced them by T7-mediated in vitro transcription (IVT) (Fig. 3.1c). BreakTag was then performed using as input gDNA digested with the various sgRNA pools. This procedure identified 92,375 on-targets/off-targets (1,418 of the 1,491 on-target sites were cut) (Supplementary Table 3), validating the efficacy of our approach (Fig. 3.1d and Extended Data Fig. C.2e). We used this dataset to investigate the positional effects of incorrect base pairing (mismatches) between the CRISPR RNA (crRNA) and target DNA, complementing previous findings18,19. We observed that protospacer-adjacent motif (PAM)-distal regions were more permissive to incorrect base pairing than the PAM-proximal portion of the protospacer (Fig. 3.1e). In accordance with previous observations showing that mismatches within the seed sequence disrupt R-loop formation and ablate DNA cleavage (Ivanov et al., 2020; Pacesa et al., 2022), target cleavage frequency was inversely correlated with the number of mismatches (Fig. 3.1f)(Lazzarotto et al., 2020). Previous reports showed that Cas9 can use alternative PAM sequences (Jones et al., 2021; Lazzarotto et al., 2020). We identified that 84.7% of the cleaved sites were found next to the canonical PAM NGG, followed by NAG (9.29%) and NGA (1.75%), showing that non-canonical PAMs are used, albeit with lower frequency (Fig. 3.1g). We further identified an inverse correlation between the number of off-targets and the sequence target complexity (measured by the Shannon index; r=-0.43, $P<2x10^{-16}$) (Fig. 3.1h), suggesting that a selection of more complex target sites could be used as a strategy to minimize off-target activity. Taking these findings together, we conclude that BreakTag is a sensitive, fast and scalable methodology for detecting CRISPR–Cas9-induced DSBs and is proficient at identifying the determinants of off-target activity, thus complementing previous efforts (Jones et al., 2021; Lazzarotto et al., 2020).

3.4.2 BreakTag reveals the flexible Cas9 scission profile

A unique advantage of BreakTag is that it allows the original DSB end structure to be retraced, as the filling-in of 5' overhangs and removal of 3' overhangs during BreakTag sample preparation should shift the expected start of the DSB reads, yielding a footprint of the original DSB end structure. To confirm this, we performed BreakTag on gDNA of cells in vitro digested with a panel of restriction enzymes having different cutting structures, and we assessed the read signatures around the expected cut site. We observed that blunt DSBs generated reads that abutted at the expected cut site (Extended Data Fig. C.3a), whereas the use of restriction enzymes that generate 3' or 5' overhangs led to a clear gap or overlap between the DSB reads, respectively, with size corresponding to the length of the expected overhang (Extended Data Fig. C.3b,c). We reasoned that applying the same rationale would enable an investigation of the scission profile of Cas9-induced DSBs. The RuvC domain of Cas9 can cleave the non-target strand at non-canonical positions, generating ssDNA 5' overhangs (Jinek et al., 2012; Jones et al., 2021; Lemos et al., 2018; Shi et al., 2019; Shou et al., 2018). In the scenario of a blunt DSB, both the RuvC and HNH domains cleave the DNA strands between the third and fourth nucleotide upstream of the PAM sequence (positions 18 and 17 of the protospacer, respectively), generating abutting DSB reads aligned at the expected cut site for blunt cuts (Fig. 3.2a and Extended Data Fig. C.3d). If the RuvC domain cleaves the non-target strand upstream of the HNH domain, 5' ssDNA overhangs are generated, and, upon end repair during BreakTag, the PAM-proximal and PAM-distal reads overlap and no longer abut (Fig. 3.2a,b and Extended Data Fig. C.3d). We used this feature of BreakTag to assess the frequency of the different DSB end structures generated by Cas9. To this end, we used a subset of the HiPlex1 dataset with sites containing an NGG PAM, and at least 16 reads at the PAM-proximal side of the DSB, yielding a total of 38,141 on-target/off-target sites. Because the fill-in reaction occurs toward the PAM, the PAM-distal side of the break is expected to map between target positions 17 and 18 regardless of the RuvC cleavage position on the non-target strand (Extended Data Fig. C.3e,f). Therefore, we extended BreakInspectoR to also parse the reads of each DSB into PAM proximal or PAM distal, and we used this feature to calculate the 'blunt rate', defined as the abundance of blunt DSBs profiled at the expected site for a blunt cut (between positions 17 and 18) relative to the total DSBs profiled in a region around [-3, +3] the expected cut site for the PAM-proximal read (Methods). The different sgRNAs self-organized based on their scission profile and preferred overhang length in the expected classes (Fig. 3.2c). Profiling the structure of Cas9-induced DSBs revealed that Cas9 preferentially generates blunt DSBs (61.57%), but a significant portion contains 5' ssDNA overhangs (35.04%) (Fig. 3.2d, left). Interestingly, the presence of mismatches between the crRNA and gDNA influenced the Cas9 scission profile. In the absence of mismatches, 79.78% of the Cas9 DSBs were blunt, whereas approximately 18% of Cas9 DSBs were staggered (Fig. 3.2d, middle). At off-targets, the number of blunt breaks decreased (to 55.89%), whereas the percentage of staggered breaks increased (to ~40%) (Fig. 3.2d, right). The scission profile was target sequence dependent

(Fig. 3.2e), with gRNAs showing nearly completely blunt Cas9 breaks (for example, TAPBP.5) (Fig. 3.2f) and others exhibiting a broader range of Cas9 cuts (for example, SUZ12.6) (Fig. 3.2g). The fraction of blunt/staggered breaks across their target sites was sgRNA dependent. In 15.07% of the sgRNAs tested, Cas9 cut almost exclusively in a blunt configuration (blunt reads>90%), whereas, in 11.77%, Cas9 cut almost exclusively in a staggered fashion (staggered reads>90%) (Fig. 3.2h and Extended Data Fig. C.3g).

In line with our findings indicating that the target sequence and the presence of mismatches influence the Cas9 scission profile, we found that Cas9 blunt rate inversely correlates with the number of identified mismatches (Fig. 3.2i), suggesting that partial complementarity between the crRNA and target site favors more staggered Cas9 cuts. Changes in the blunt rate were higher if mismatches were located at positions 16–20 of the protospacer/target sequence, suggesting that these positions might be important for determining the profile of Cas9 scission (Fig. 3.2j). Given the unique ability of BreakTag to probe the end structure of Cas9 target-dependent proportion of blunt to staggered cuts, we investigated the blunt rate of off-targets nominated by BreakTag alone or shared with CIRCLE-seq and DIGENOME-seq. We observed that BreakTag-exclusive sites showed a higher proportion of staggered reads, suggesting that the end repair step might be beneficial to capture sites with a high proportion of staggered cuts (Extended Data Fig. C.3h).

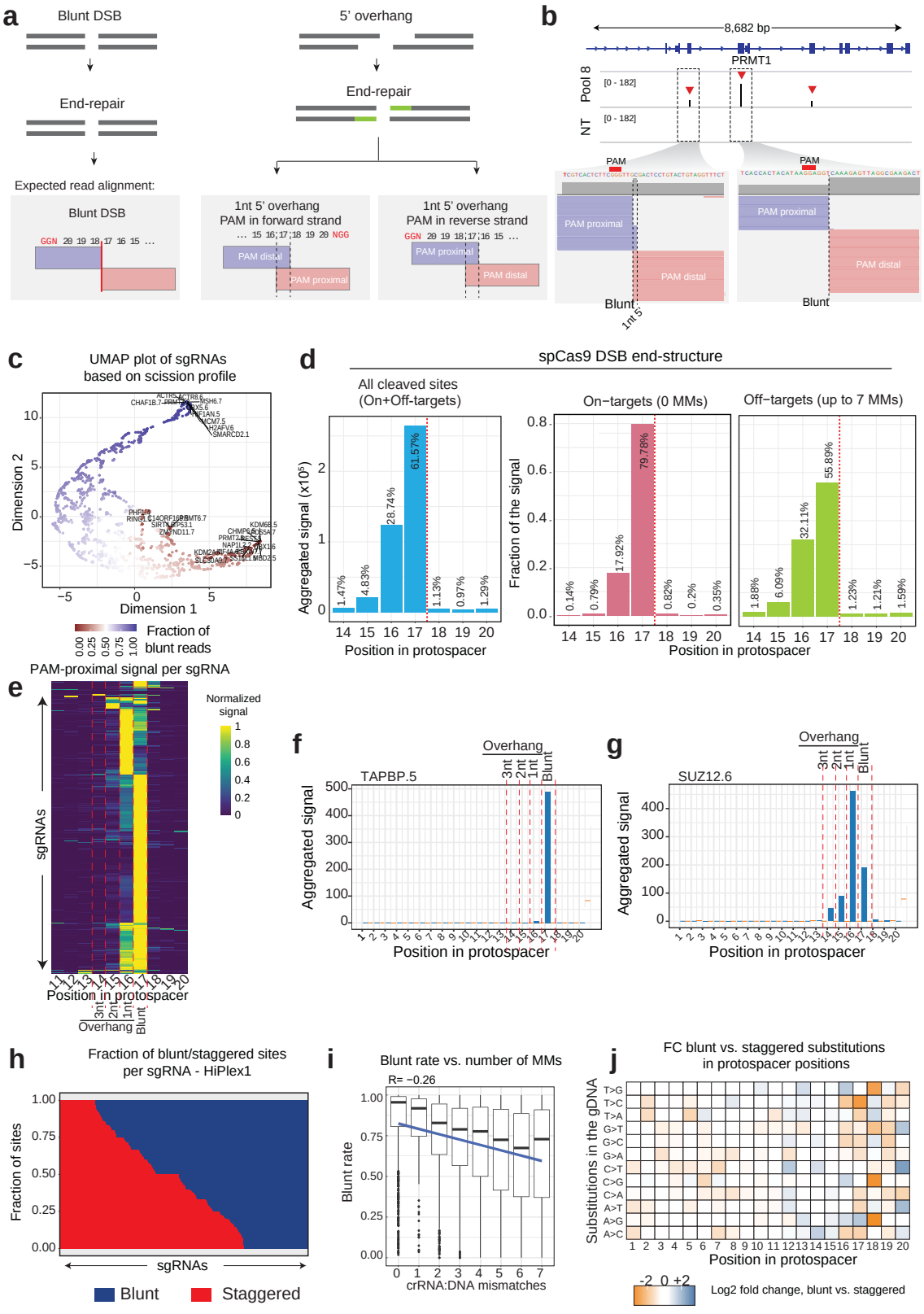


Figure 3.2: High-throughput analysis of Cas9 scission profile. **a** Schematic representation of the different read alignments for 5' overhangs in BreakTag data. Blunt DSB reads start at the same location, but 5' overhangs are shifted due to the end-repair reaction during BreakTag sample

preparation (see also Extended data Fig. 3a–c). **b** Representative IGV snapshot depicting three on-target DSBs for the PRMT1 gene identified by BreakTag on gDNA (HepG2 cells) digested with HiPlex RNPs loaded with pools 1–10 of the HiPlex dataset 1 described in Fig. 1c,d. Examples of sites cut in mostly staggered (left) or blunt (right) configurations are shown below and zoomed-in. **c** UMAP representation on two dimensions of relatedness between sgRNAs based on average scission profile. Dimensions 1 and 2 are representations in a reduced dimensional space (arbitrary units) of the fraction of signal each sgRNA has in positions 14–20 of the protospacer. The color scale represents the fraction of signal at the expected cut site, ranging from 100% (blue, all blunt, no signal outside the expected cut site) to 0% (red, all staggered, all signal outside the expected cut site). The top 25 most blunt and staggered sgRNAs are highlighted in the plot. sgRNA self-organize in this representation based on their scission profile and preferred overhang length. **d** Left: aggregated signal of different DSB end structures for on/off-targets with an "NGG" PAM in the HiPlex1 dataset. The fraction of blunt or staggered DSBs for on-targets (pink) and off-targets with up to 7 mismatches (MM; green) are shown center and right, respectively. Position 17: blunt DSBs; 16–14: 5' overhangs. Dotted line indicates the expected cut site for a blunt DSB. **e** Accumulation of reads mapped onto the PAM-proximal strand (scaled) along the protospacer over 1,418 sgRNAs of the HiPlex1 dataset for all identified targets with an "NGG" PAM. Most sgRNAs accumulate signal at position 17 of the protospacer (corresponding to blunt DSBs), versus approximately 34% at positions 14–16 of the protospacer corresponding to staggered DSBs with 1–3 nt overhangs, respectively. **f**, **g** Representative examples of target sites at which Cas9 cuts preferentially in blunt or staggered configuration. Aggregated BreakTag signal along the protospacer for "TAPB.5" sgRNA on and off-targets (n=3), which preferentially accumulates on position 17 of the protospacer (f). Aggregated BreakTag signal along the protospacer for "SUZ12.6" sgRNA on and off-targets (n=56), which accumulates mostly on position 16 of the protospacer (g). **h** Fraction of blunt (blue) or staggered (red) DSBs for each sgRNA. Each column represents the fraction of blunt or staggered reads for on and off-targets of a given sgRNA. **i** Box plots showing the average blunt rate for sites containing up to seven crRNA:DNA mismatches. **j** Heatmap showing the log2 fold change of frequency of nucleotide substitutions along the protospacer in predominantly blunt sites (blunt raw reads >66%) compared to predominantly staggered sites (blunt raw reads <33%; N=26,802 sites with a BreakTag coverage of at least 16 reads in the PAM-proximal side).

3.4.3 Determinants of Cas9 scission profile mediate precise and predictable indels

To identify important features influencing whether Cas9 cuts in blunt or staggered configuration, we trained an XGBoost regression model using the two-dimensional (2D) one-hot-encoded representation of the correspondence between the 20 nucleotides (nt) of the protospacer and guide sequences as predictors, together with the number of mismatches in the non-seed (positions 1–10) and seed (positions 11–20) parts of the protospacer. The blunt rate for the cleaved loci from our HiPlex1 library dataset was used as the target for this prediction (Extended Data Fig. C.4a). Our model achieved high performance, as measured by the correlation between the predicted and observed blunt rates in the cross-validated sets (r=0.74) (Extended Data Fig. C.4b). The high predictive power of our model allowed us to investigate important positions within the protospacer that determines whether Cas9 cleaves the target DNA in a staggered or blunt manner. We observed

that positions 16–20 (5nt upstream of the PAM) were important for predicting the scission profile, with guanines at positions 17 and 18 having the highest importance (Fig. 3.3a,b and Extended Data Fig. C.4c). We next sought to identify sequence compositions associated with a blunt or staggered cut by interrogating the importance of each base along the protospacer. Strikingly, we identified that a G at position 17 was predictive for a blunt DSB, whereas a G at position 18 was associated with staggered DSBs (Fig. 3.3c and Extended Data Fig. C.4d).

To investigate the effects of 17G and 18G on Cas9 scission with our dataset, we grouped the cleaved sites into 'blunt' (0–33% of PAM-proximal reads mapping outside of position 17: staggered reads), 'middle' (33–66% staggered reads) and 'staggered' (66–100% staggered reads). Cas9 was, in general, more likely to cut blunt at on-target sequences than at off-targets where mismatches are present (ANOVA: $P<2x10^{-16}$) (Fig. 3.2d,i and Extended Data Fig. C.4e). In accordance with the model predictions, Cas9 was more likely to cleave in a blunt configuration at sites with a G at position 17 compared to sites with A, C or T, at both on-targets and off-targets (Pearson's chi-squared test: $P<2x10^{-16}$) (Extended Data Fig. C.4e). In contrast, if a G occupied position 18, Cas9 was more likely to cleave in a staggered configuration than if A, C or T occupied that position (Pearson's chi-squared test: $P<2x10^{-16}$) (Extended Data Fig. C.4e). We further investigated the combination of nucleotides at positions 17 and 18 to determine their preference for either blunt or staggered cuts. Interestingly, the combination of 17T|18G had the most significant impact on promoting staggered cuts, whereas 17G|18C favored blunt breaks (Fig. 3.3d). We conclude that the base composition surrounding the DSB is a strong determinant of the Cas9 scission profile.

Previous evidence supported an association between Cas9 scission and repair outcome (Lemos et al., 2018; Shi et al., 2019; Shou et al., 2018), but the lack of scalable methods to assess scission profiles has precluded a systematic investigation. We deployed our machine learning model to 2,791 genomic gRNA targets, for which the repair outcome was previously characterized (Allen et al., 2019), to predict the blunt rate for each gRNA sequence (Extended Data Fig. C.4f). We then selected the predicted top 700 most blunt and top 700 most staggered sites for HiPlex BreakTag (hereafter referred to as the 'HiPlex2' library) to correlate their Cas9 scission profile with their empirical repair outcome (Supplementary Tables 1 and 4). The predicted blunt rate of this dataset was highly correlated with the actual scission profile obtained by BreakTag, confirming the robustness of our model (Extended Data Fig. C.4g). When interrogating the scission profile as a function of the most common empirically observed indel size for each site, we observed that blunt cuts were equally represented across indel size (Fig. 3.3e). By contrast, a striking enrichment of staggered sites was found at genomic loci that are repaired as single-nucleotide insertions (+1 indels) (Fig. 3.3e). Over 90% of sites with a +1 indel as the most common repair outcome were staggered DSBs, demonstrating a clear association between scission profile and DNA repair (Fig. 3.3f). Staggered breaks generated more precise indels (that is, at a higher frequency) compared to blunt cuts for -1 and +1 indels (Fig. 3.3g). Precise

insertions are desirable repair outcomes in the context of correcting pathogenic alleles and inducing gene knockouts. To understand the effect of sequence on the efficiency of templated insertions, we investigated the number of loci for which the most frequent repair was a templated insertion as a factor of base composition at positions 17 and 18 of the protospacer. If the ssDNA overhang at the cut site is used as a template for repair, we would expect that the most common insertion would be a copy of the overhang sequence. Because most overhangs generated by Cas9 are 1nt long (Fig. 3.2d), we anticipated that position 17 would be duplicated in most cases (Fig. 3.3h). Indeed, the most common nucleotide inserted at staggered sites was a duplication of the base at position 17, indicating that template insertions are a common repair outcome of staggered DSBs (Fig. 3.3i). Target sites with G at position 17 showed a low number of templated insertions, as expected for blunt cuts (Fig. 3.3i,j and Extended Data Fig. C.4h). By contrast, target sites with G in position 18 were more likely to use the nucleotide at position 17 as the template for the single-nucleotide insertions (Fig. 3.3j and Extended Data Fig. C.4i), suggesting that target sequences with a specific nucleotide composition can be selected for precise, predictable and desirable genome editing.

We expanded our scission profile and indel analysis by investigating the most common indel outcome as a function of scission identity in our HiPlex1 dataset (generated in HepG2 gDNA), for which amplicon sequencing data are available (Chakrabarti et al., 2019). Insertions were enriched at staggered-cleaved target sites compared to blunt (Extended Data Fig. C.4j). In line with our previous findings, we observed that 1-nt insertions were highly associated with staggered DSBs (Extended Data Fig. C.4k), with approximately 80% of 1-nt insertions being produced by staggered cuts (Extended Data Fig. C.4l).

To further demonstrate that a pre-selection of target sites with predicted scission profile can be leveraged for increasing insertion precision, we tasked our machine learning trained on SpCas9 HiPlex BreakTag data to predict the blunt rate of Cas9 at various human target sequences, and we grouped them into 'blunt' and 'staggered' groups, showing the highest and lowest blunt rate, respectively (Supplementary Table 10). We then applied a gRNA-target pair cloning strategy10 to assess in parallel the repair outcome of sites predicted to be cut preferably in a blunt or staggered manner. In brief, we designed genomic cassettes of selected target sequences predicted to be cut in blunt or staggered configuration along with its targeting gRNA as pools cloned into lentiviral vectors (Fig. 3.3k and Extended Data Fig. C.5a). Cas9-expressing K562 and HeLa cells were then transduced with the blunt or staggered pool; the gDNA was extracted 7d after transduction; and repair outcomes were assessed via amplicon sequencing (Methods). In accordance with our previous findings, the target sequences predicted to be cleaved in a blunt manner were mostly repaired as deletions, whereas the most common indel for staggered cuts was single-nucleotide insertions (Fig. 3.3l and Extended Data Fig. C.5b). The insertion rate was significantly higher in the staggered pool compared to blunt (Extended Data Fig. C.5c), and approximately 75% of all +1 indels were templated (Extended Data Fig. C.5d). Collectively, these data indicate a strong association between the staggered Cas9 incisions

with repair precision and predictability, highlighting the possibility of using predictions of Cas9 cleavage configurations for more precise and predictable genome editing.

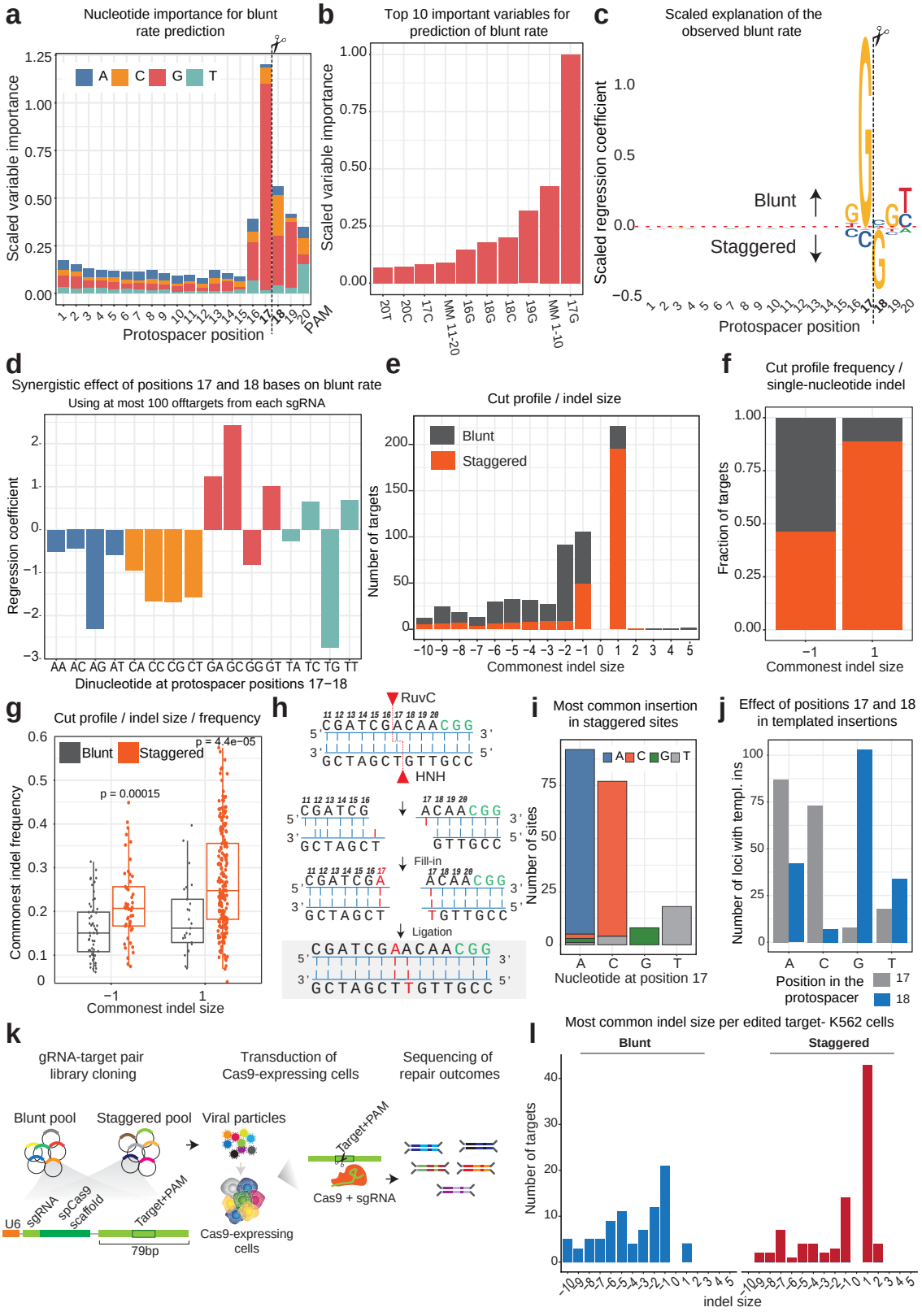


Figure 3.3: Sequence determinants of Cas9 scission profile. **a** Importance of the nucleotide composition and position in the protospacer, as estimated by the XGBoost method. Values on the y axis are scaled to the most important nucleotide+position. The dashed vertical line indicates the cut site for a blunt DSB. **b** Top ten most important variables for the prediction of blunt rate. MM 1–10: mismatches in the non-seed part of the protospacer (positions 1–10); MM 11–20: mismatches in the seed part of the protospacer (positions 11–20). **c** Observed blunt rate explained by the sequence composition of the protospacer. Coefficients of a linear regression model fit to the nucleotide composition independently on each position of the protospacer are shown as letters scaled according to the importance of that nucleotide and position, as estimated by the XGBoost model. The dashed vertical line indicates a cut site for a blunt DSB. **d** The effect of all possible nucleotide combinations in position 17 and 18 in the blunt rate prediction. **e** HiPlex BreakTag (HiPlex2 dataset) was performed to assess the scission of 610 sites in a matched dataset with known repair outcomes (+1 to +5 nt insertions, -1 to -10 nt deletions) [@allenPredictingMutationsGenerated2019]. Using our model, we predicted genome sites with staggered or blunt DSBs (blunt if predicted log2 ratio of blunt vs. staggered BreakTag signal>0; otherwise staggered) from HiPlex2 library to use in this experiment. The same number of blunt and staggered breaks sites were used for the analysis (n=610) **f** Cut profile frequency in single-nucleotide indels shows a significant enrichment of +1 insertions in staggered sites (Fisher's test: odds ratio=8.99, p-value=8.345e-16). Colors represent the fraction of blunt (gray) or staggered (orange) sites showing 1 nt deletions (-1) and/or 1 nt insertions (+1). **g** Frequency of 1 nt deletions or insertions in relation to scission profile (t-test: p-value=0.00015 for -1 deletions, p-value=4.4e-5 for +1 insertions). **h** Scheme depicting how 1 nt 5' overhangs can promote templated repair, leading to 1 nt insertions. **i** Most common insertion at staggered sites according to the type of nucleotide at position 17. **j** Number of loci with templated insertion according to the base composition at positions 17 (gray) or 18 (blue). **k** Schematics of gRNA-target pair experimental design for the blunt and staggered pools. **l** Most common indel size found per edited target in K562-Cas9 cells. A total of 199 gRNA-target pairs (93 staggered and 106 blunt) were used for this analysis after filtering for sites with at least 100 mutated reads and not detected in the experiment performed with cells not expressing Cas9.

3.4.4 Genetic variation impacts Cas9 scission profile and editing outcome

Given the strong dependency of Cas9 scission profile on the sequence context, we surveyed the entire coding human genome for putative Cas9 targets. We used our model to extrapolate the scission profile of every putative Cas9 target in human exons by predicting the blunt rate for over 10 million NGG-endowed sites. Our analysis indicated that 56.58% (5,869,863 of 10,374,276 sites) of putative Cas9 target sites are predicted to be cleaved predominantly in a blunt manner (log2 blunt rate>0; equivalent to >50% blunt breaks) and 43.42% (4,504,413 of 10,374,276 sites) in a staggered configuration (log2 blunt rate<0) (Extended Data Fig. C.6a), with 18.08% of all target sites at human exons (1,875,201 of 10,374,276) to be cleaved in a highly staggered configuration (log2 blunt rate<-2; equivalent to >80% of staggered breaks) (Extended Data Fig. C.6a). Because staggered Cas9-induced DNA breaks are strongly associated with precise and predictable single-nucleotide insertions, our findings suggest that predictable and precise genome editing might be favored by pre-selecting target sites that are predicted to be cleaved in a

staggered configuration.

Single-nucleotide polymorphisms (SNPs) account for most human genetic variation (Auton et al., 2015) and have the potential to affect Cas9 on-target and off-target activity (Cancellieri et al., 2023; Krysler, Cromwell, Tu, Jovel, & Hubbard, 2022; Lazzarotto et al., 2020; Lessard et al., 2017; Scott & Zhang, 2017). However, the impact of human genetic variation on the scission profile of Cas9 has not yet been investigated. To understand how the genetic variation of an individual affects DNA scission by Cas9, we surveyed the 1000 Genomes Project (1000G) database for SNPs at positions 17 and 18 of putative Cas9 targets in exons, and we predicted blunt rates for Cas9 target sites in these different genomes using our machine learning model (Supplementary Table 5). As expected, based on the sequence determinants analysis (Fig. 3.3c), [A/C/T]>G substitutions at position 17 were associated with an increase in the blunt rate (more blunt breaks; 1,964 of 3,086 transitions), whereas G>[A/C/T] substitutions were associated with a decrease (more staggered breaks; 2,385 of 3,448 transitions) (Extended Data Fig. C.6b,d,f). Conversely, at position 18, [A/C/T]>G substitutions were associated with more staggered breaks (1,973 of 2,859) and G>[A/C/T] with more blunt ones (1,569 of 2,679) (Extended Data Fig. C.6c,e,g).

To understand allele-specific changes in the Cas9 scission profile, we leveraged the genomes of seven individuals extensively characterized by the Genome-in-a-Bottle (GIAB) Consortium (Zook et al., 2016, 2019). We first predicted the blunt rate of all loci containing a SNP at positions 17 (n=394,330) or 18 (n=395,368) among GIAB individuals using our machine learning model. Second, we predicted the effect of each base substitution in the Cas9 scission profile by calculating the difference between the predicted blunt rate for reference and alternative alleles. Based on our analysis, we selected 300 sites with a SNP at positions 17 or 18 and the highest predicted difference in blunt rate between the reference and alternative allele, with the goal of identifying SNP-driven changes in the Cas9 scission profile (Fig. 3.4a). Finally, we generated a HiPlex BreakTag dataset of 300 sites with SNPs targeting the reference or mutant allele (hereafter referred to as the 'HiPlex3' library) (Supplementary Table 6). We were able to confirm SNP-driven changes of scission profile predicted by our model in experimental observations. If a SNP was found at position 17 of the target site, an [A/T/C]>G substitution significantly increased the blunt rate, whereas G>[A/T/C] significantly reduced it (Fig. 3.4b–d). Analysis of position 18 revealed a strikingly opposite pattern, with [A/T/C]>G substitutions significantly decreasing the blunt rate and strongly associated with staggered DSBs, whereas G>[A/T/C] changes were significantly associated with blunt breaks (Fig. 3.4e–g).

Following our observation that Cas9 scission profile is a major determinant of repair outcome, we hypothesized that the SNP-driven changes in Cas9 cutting have the potential to change editing outcomes in an allele-specific manner. To test that, we leveraged our gRNA-target pair approach (Fig. 3.4h) to assess the indel outcomes of target sequences with a SNP at position 17 or 18 that displayed differences in scission profile in our BreakTag analysis (Fig. 3.4b–e). As expected by the strong association between the nucleotide type

at positions 17 and 18 with the Cas9 scission profile, we observed changes in the editing outcome depending on the SNP type and position in the protospacer, with insertion rates changing according to the shift in the scission profile promoted by SNPs introducing or removing a G base at position 17 or 18 between the reference and alternative allele (Fig. 3.4i,j). We confirmed these findings by targeting endogenous loci containing a SNP at position 17 or 18 of the protospacer with known scission profiles in lymphoblastoid cell lines from B lymphocytes derived from GIAB donors, and we performed targeted ultra-deep sequencing (~106x) (Extended Data Fig. C.6h–k). As an example, a G>A substitution at position 17, which is associated with a higher proportion of staggered cuts (Fig. 3.4d), led to an increased frequency of +1 indels from 12% to 72% (Fisher's test, $P<2x10^{-16}$) (Extended Data Fig. C.6i), whereas a C>G substitution at position 18, which also favors staggered Cas9 cuts (Fig. 3.4g), greatly increased the frequency of +1 indels from 25% to 75% (Fisher's test, $P<2x10^{-16}$) (Extended Data Fig. C.6j).

Taken together, our data demonstrate that genetic variation directly impacts the Cas9 scission profile along with the editing outcome, highlighting the importance of implementing variant-aware analyses of the Cas9 scission profile for more predictable and precise genome editing.

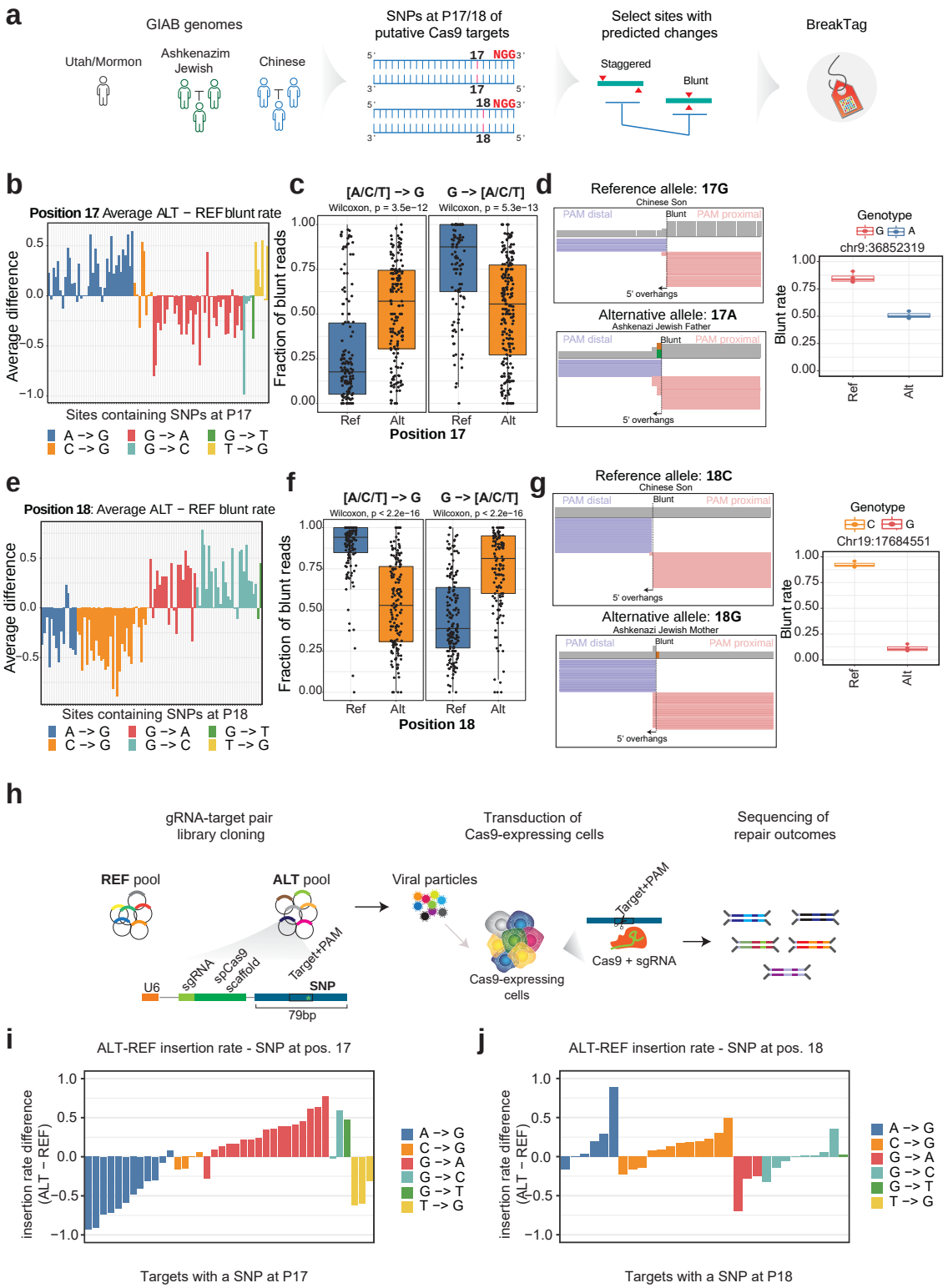


Figure 3.4: Human genetic variation influences Cas9 scission profile and indel outcome. **a**, Experimental design for investigating the role of human genetic variation in Cas9 scission profile. SNP databases curated from individuals of the Genome in a Bottle (GIAB) consortium [@zookExtensiveSequencingSeven2016; @zookOpenResourceAccurately2019] were used to identify Cas9 target sites containing an SNP at position 17 or 18 of a potential protospacer. Using our

model (Extended data Fig.
reffig:breaktagEF4a), the blunt rate was predicted for the reference and alternative allele for each site, and sites with the highest predicted changes for position 17 or 18 were targeted using HiPlex BreakTag (HiPlex library 3) with a total of 600 sgRNAs. **b**, Difference between the average blunt rate of alternative (ALT) and reference (REF) alleles containing an SNP at position 17 of the protospacer. The blunt rate was averaged between individuals with the same genotype. **c**, Fraction of blunt reads out of the total number of reads in the PAM-proximal strand for the target sites containing an SNP in position 17, comparing the reference (blue) and alternative (orange) alleles; left: SNPs mutating into a G; right: SNPs mutating from a G. **d**, Left: a representative IGV snapshot showing BreakTag reads of individuals harboring the reference allele (17G, top), and an SNP (17A, bottom). Right: the blunt rates for the reference and alternative genotypes for that locus. **e**, Difference between the average blunt rate in alternative (ALT) and reference (REF) alleles containing an SNP at position 18. The blunt rate was averaged between individuals with the same genotype. **f**, Fraction of blunt reads over the total number of reads in the PAM-proximal strand for the target sites containing an SNP in position 18, comparing the reference (blue) and alternative (orange) alleles; left: SNPs mutating into a G; right: SNPs mutating from a G. **g**, Left: a representative IGV snapshot showing BreakTag reads for individuals harboring the reference allele (18C, top), and an SNP (18G, bottom). Right: the blunt rates for the reference and alternative genotypes for that locus. **h**, Schematics of gRNA-target pair experimental design for the ALT and REF pools. **i**, Difference in the insertion rate of target sites containing the indicated SNPs at position 17. Targets with at least 100 mutated reads were used for the analysis. **j**, Difference in the insertion rate of target sites containing the indicated SNPs at position 18. Targets with at least 50 mutated reads were used for the analysis.

3.4.5 Engineered Cas9 variants with altered scission profiles

We demonstrated that the protospacer sequence is a major determinant of Cas9 cleavage pattern and the repair outcome, and therefore, pre-selecting target sequence composition can be leveraged for increased staggered cleavage favoring insertions. However, the sequence determinants dictating the Cas9 scission profile limit the number of targets that could be cleaved in a staggered manner, and, therefore, we set out to search for Cas9 variants with altered scission profiles. To this end, we characterized by BreakTag the scission profile of six previously described engineered variants with reduced off-target activity: HiFiCas9 (ref. (Vakulskas et al., 2018)), xCas9 (ref. (J. H. Hu et al., 2018)), SniperCas9 (ref. (Lee et al., 2018)), HypaCas9 (ref. (J. S. Chen et al., 2017)), EvoCas9 (ref. (Casini et al., 2018)) and LZ3Cas9 (ref. (Schmid-Burgk et al., 2020)) (Fig. 3.5a and Extended Data Fig. C.7a).

We performed BreakTag, targeting 150 genomic loci, and calculated the target specificity, the blunt rate and the overlapping off-targets for each variant. The variants displayed different levels of cleavage at on-targets and off-targets compared to SpCas9, with a marked reduction of overall cleavage for xCas9 and EvoCas9 (Extended Data Fig. C.7b,c). Next, we calculated the relative 'Activity' (total on-target reads of variants normalized by total on-target reads of SpCas9) and 'Specificity' (proportion of off-target reads over on-target) of each variant, to investigate if there is a tradeoff between fidelity and over-

all cleavage activity. The variant EvoCas9 had the highest specificity score of all tested variants but displayed an approximately 47% reduction in activity compared to SpCas9 (Extended Data Fig. C.7d). We observed no reduction of SniperCas9 and HypaCas9 on-target activity but a slight increase in specificity of approximately 4% and approximately 12%, respectively (Extended Data Fig. C.7d). Strikingly, the variant LZ3 showed both a higher fidelity (Extended Data Fig. C.7d) and a remarkable reduction of the blunt rate correlation versus SpCas9 (r=0.49) (Fig. 3.5c,d and Extended Data Fig. C.7e,f), along with a skewed distribution toward staggered breaks (Fig. 3.5b–e). We observed that approximately 48% of LZ3 DSB reads accumulated at position 17, reminiscent of blunt DSBs, whereas approximately 47% of breaks displayed 5' overhangs (Fig. 3.5e). Most of the non-blunt breaks were 1-nt 5' overhangs (38.24%), but 2-nt (8.44%) and 3-nt (2.97%) overhangs were also observed (Fig. 3.5e). Of note, the proportion of blunt to staggered breaks was gRNA dependent, indicating that, similar to SpCas9, LZ3's scission profile is target sequence dependent (Extended Data Fig. C.8a). In line with our findings, blunt rate and insertion frequency of SpCas9 and LZ3 were inversely correlated (r=-0.65, $P=7.7x10^{-12}$) (Extended Data Fig. C.8b).

Given the marked reduction in correlation between the blunt rates of LZ3 and SpCas9 (Fig. 3.5c,d), we set out to further characterize the sequence determinants dictating LZ3's scission profile. We applied a XGBoost regression model using the 2D one-hot-encoded representation of the correspondence between the 20nt of the protospacer and guide sequences as predictors, together with the crRNA:DNA mismatches for BreakTag data on LZ3 (Extended Data Fig. C.4a). The model achieved high performance as tested on cross-validated data (Extended Data Fig. C.8c). We next investigated the most important variables and nucleotides along the protospacer for predicting the blunt rate, and, interestingly, a 19G target sequence had a high importance for predicting LZ3 target-specific blunt rate (Extended Data Fig. C.8d,e). Similar to SpCas9, a 17G sequence was predictive of a blunt cut, but a 19G was highly predictive of a staggered DSB (Fig. 3.5f). To assess whether LZ3 could be used as an alternative of Cas9 to generate staggered breaks and produce insertions at target sites where Cas9 cleaves in blunt configuration, we investigated the insertion frequency at staggered DSBs generated by LZ3 but not by SpCas9. We indeed observed that LZ3 can generate higher insertion rates at staggered 19G sites compared to SpCas9 (Extended Data Fig. C.8f), suggesting that a rational engineering of Cas9 variants might be a feasible strategy for introducing high-frequency insertion at target sequences where SpCas9 cleaves in a blunt manner.

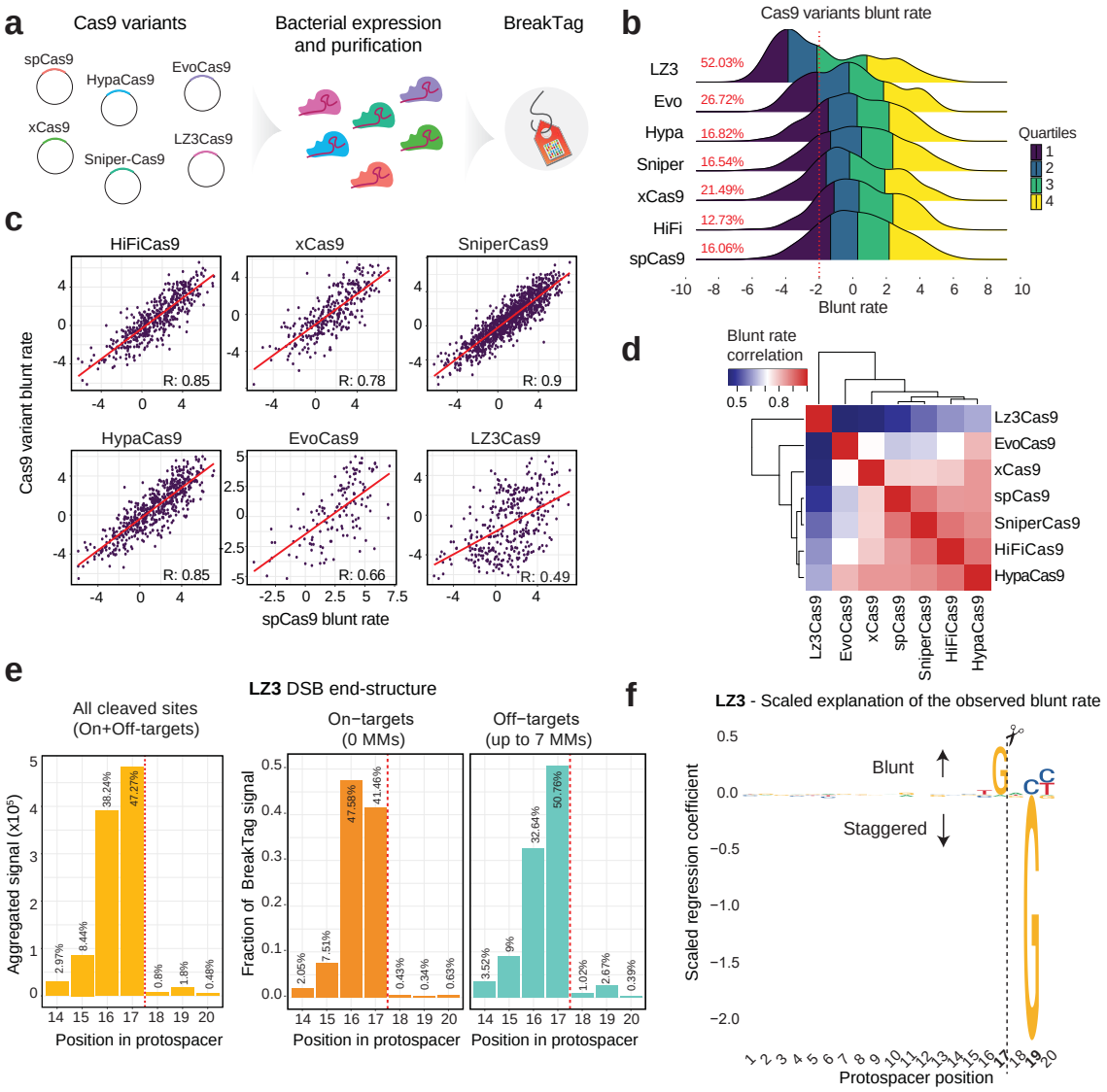


Figure 3.5: Cas9 engineered variants with modulated scission profiles. **a** Schematic of the production of engineered Cas9 variants and characterization of scission profiles. **b** Distribution of blunt rate for tested Cas9 variants for on and off-targets. Colors show quartiles. Dashed line marks log2 rate of -2 (80% staggered DSBs). Sites with at least 8 unique reads on the PAM-proximal side were used for the analysis. The percentage of sites with more than 80% staggered DSBs are shown. **c** Blunt rate correlation between SpCas9 (x axis) and the tested variants (y axis). Each point is a cleaved site (on or off-target with at least 8 unique reads on the PAM-proximal of the break). **d** Matrix depicting blunt rate correlation between the tested variants. **e** Left: aggregated signal of different DSB end structures for on/off-targets in the HiPlex1 library generated with the LZ3 nuclease. The fraction of blunt or staggered DSBs for on-targets (orange) and off-targets with up to 7 mismatches (MM; green) are shown center and right, respectively. Position 17: blunt DSBs; 16–14: 5' overhangs. Dotted line indicates the expected cut site for a blunt DSB. **f** Observed LZ3 blunt rate explained by the sequence composition of the protospacer. Coefficients of a linear regression model fit to the nucleotide composition independently on each position of the protospacer are shown as letters scaled according to the importance of that nucleotide and position, as estimated by the XGBoost model. The dashed vertical line indicates a cut site for a

blunt DSB.

3.4.6 Leveraging scission profile for correction of pathogenic deletions

Given the strong link between scission profile and predictable insertions, we sought to test if a scission-based targeting strategy can be leveraged for correcting pathogenic single-nucleotide deletions. We reasoned that, by exploiting SpCas9 or engineered variant sequence determinants for staggered cleavage, single-nucleotide insertions can be favored, compensating frameshift mutations caused by a pathogenic deletion found in proximity to a PAM sequence. Furthermore, the predictability of insertions (Fig. 3.3i,j) would enable the recovery of the original protein sequence by exploiting codon degeneration.

To estimate how the acquired insights into the scission profiles of Cas9 variants can be leveraged for the correction of pathogenic deletions, we employed our models trained on HiPlex BreakTag data from SpCas9 or LZ3Cas9 to predict the scission profile of 1-nt pathogenic deletions included in the ClinVar database (Fig. 3.6a). Our goal was to assess the potential of inducing 1-nt templated insertions for correcting pathogenic deletions by restoring the frame and maintaining the original amino acid sequence, rescuing protein function (Extended Data Fig. C.9a). In addition to SpCas9, we chose the LZ3Cas9 because it exhibits distinct scission profile sequence determinants that lead to higher insertion rates compared to SpCas9 at 19G loci (Figs. 3.3c and 3.5f and Extended Data Fig. C.8f). From the 31,010 pathogenic single-nucleotide deletions found in exons cataloged in ClinVar, 8,705 were endowed by an NGG PAM and can be targeted by SpCas9 and LZ3 (Fig. 3.6b). A total of 4,999 NGG-endowed alleles were predicted to be restored if a templated insertion takes place, rescuing the healthy protein sequence (Fig. 3.6b). Next, we predicted the blunt rate of gRNAs targeting the candidate deletions for reframing and protein rescue using our model trained on SpCas9 and LZ3 (Supplementary Table 12). We observed that 2,276 alleles were predicted to be cut preferably staggered (blunt rate<0) by SpCas9 and 2,582 by LZ3. From the staggered alleles, 938 were predicted to be cleaved in a highly staggered manner (blunt rate<=-2) by SpCas9 and 1,212 by LZ3, suggesting that templated insertions would be highly favored (Fig. 3.6b). From the highly staggered alleles, we observed that 321 were shared between both nucleases, but most were variant exclusive (607 for Cas9 and 865 for LZ3, in total 1,793 target sites), indicating that different sequence determinants expand the number of target sites that could be cleaved in a highly staggered manner for favoring templated insertions (Fig. c3.6). We confirmed that pre-selection of target sites in which Cas9 induces staggered breaks compared to blunt increases the frequency of templated +1 insertions that could be used to rescue 39 pathogenic single-nucleotide deletions cataloged in ClinVar using the cellular assay used before (Fig. 3.3k). As anticipated, the insertion rate and the frequency of templated insertions over all +1 indels was significantly enriched in the subset of target candidates predicted to be cut highly staggered compared to highly blunt (P=8.6x10-8) (Fig. 3.6d and Extended Data Fig. C.9b,c), demonstrating, as proof of principle, that pre-selection of target sites in which

Cas9 cuts staggered can be used to correct clinically relevant pathogenic deletions. Among those corrected deletions, a single-nucleotide deletion (ClinVar rs2077957264) in exon 1 creates a premature translational stop signal (p.Leu24*) in the TRMU gene, which has been reported to be associated with acute infantile liver failure, and a gRNA targeting the deletion was predicted to be cut in a highly staggered manner (Extended Data Fig. C.9d). Upon targeting this deletion, we observed that most indels were insertions (Extended Data Fig. C.9e), with the vast majority being templated insertions (Extended Data Fig. C.9f). The inserted base would recover the frame and the original amino acid sequence, disrupting the stop codon and recovering the original protein sequence (Extended Data Fig. C.9d,f).

Taken together, our data suggest that predictable and precise gene editing is enhanced by controlling the Cas9 scission profile with three major determinants: sequence-governed rules for gRNA design, accounting for individual genetic variation and leveraging engineered Cas9 variants with differential scission profiles (Fig. 3.6e).

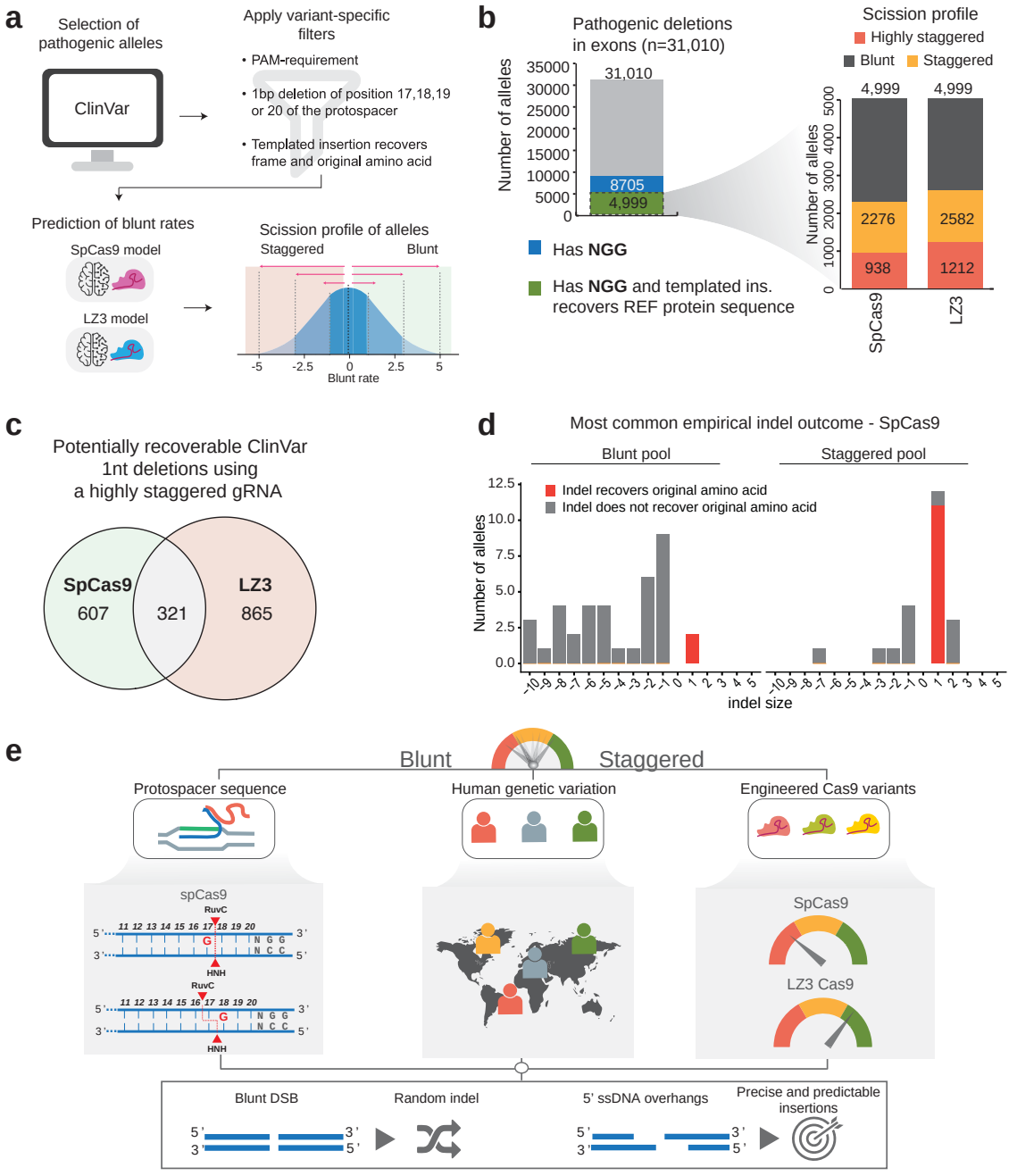


Figure 3.6: Cas9 variants expand the pool of pathogenic alleles amenable for correction. **a** Schematics depicting the workflow for the prediction of scission-aware targeting of pathogenic deletions. **b** Barplot (left) show the number of pathogenic deletions in exons that contain an NGG (blue) or that contain an NGG and a templated insertion recovers the reference protein sequence and frame (green). Horizontal barplots (right) show the predicted scission profile of gRNAs targeting pathogenic deletions with LZ3 or SpCas9. Blunt indicates gRNAs with blunt rate >0, staggered < 0, and highly staggered ≤ −2. **c** Venn diagrams depicting the overlap between pathogenic alleles that are predicted to be cleaved in a highly staggered manner by LZ3 or SpCas9. **d** Most common indel outcome for alleles in the blunt or staggered pool. **e** A model of the determinants of Cas9 scission profile identified using BreakTag. The protospacer sequence, human genetic variation, and engineering Cas9 variants can dictate Cas9 scission profile, which is

strongly associated with precise and predictable genome editing.

3.5 Discussion

We developed and applied BreakTag to survey DSBs generated by Cas9 with over 3,500 sgRNAs in the human genome across different genomic backgrounds. Labeling free DSB ends preserves the directionality of sequencing reads and, coupled with an enzymatic treatment of ssDNA overhangs at the cut site, allows the systematic investigation of the scission profile of Cas9-mediated DNA breaks. BreakTag is a scalable methodology to profile the on-target and off-target Cas9 landscape along with a scission profile. Our work establishes BreakTag as a simple, quick and readily implemented high-throughput tool for assessing CRISPR safety for personalized genome editing, by testing gRNA specificity and scission on gDNA samples. We also report HiPlex BreakTag as a companion approach for targeting thousands of unique loci in a single experiment, enabling systematic analysis of the nuclease activity of CRISPR–Cas genome editors. By combining high-throughput in-house synthesis of sgRNA and targeting several genomic loci in the same pot, we generated robust datasets to probe the determinants of sgRNA specificity and Cas9 cleavage profile preference.

Off-target discovery tools can be grouped into different categories according to the nominating strategy. In cellulo tools, such as GUIDE-seq (Tsai et al., 2015) and TTISS-seq (Schmid-Burgk et al., 2020), are highly sensitive methods that rely on the incorporation of double-stranded oligodeoxynucleotide (dsODN) tags at the cut site. Because the method relies on the co-delivery of the donor sequence with CRISPR to cells, toxicity has been reported in some models, such as induced pluripotent stem cells (Wienert et al., 2019), and delivery of the blunt dsODN requires optimization depending on the experimental model used. However, the excellent signal-to-noise ratio of the method poses a major advantage compared to biochemical assays, providing fewer ‘false positives’ (extensively reviewed in ref. (Atkins et al., 2021)). In vitro tools, such as SITE-seq (Cameron et al., 2017), DIGENOME-seq (Kim et al., 2015), CIRCLE-seq (Tsai et al., 2017) and CHANGE-seq (Lazzarotto et al., 2020), are sensitive approaches for nominating off-targets that rely on the sequencing of DSB ends generated by Cas9 in vitro and provide a list of sites that can be cleaved without chromatin and nuclear architecture present. However, none of the aforementioned methods allows the direct investigation of DSB end structure at scale, preventing a comprehensive scission profile investigation. BreakTag, in contrast, enables the nomination of off-targets for staggered-cleaving nucleases such as Cas12a and allows the parallel investigation of gRNA-specific scission profiles in multiple genomes in the same run, facilitating the study of genetic background-specific changes in scission profiles. One drawback is its relatively higher background compared to in cellulo methods, as it also sequences DSBs generated by intrinsic cell processes (for example, transcription and replication) and mechanical breaks during DNA extraction. These factors can potentially mask extremely low frequency off-targets falling within those regions.

Early studies identified a non-random repair outcome of Cas9-mediated breaks and a dependency on the target site sequence (Chakrabarti et al., 2019; Molla & Yang, 2020; Overbeek et al., 2016; Shen et al., 2018; Taheri-Ghahfarokhi et al., 2018). Evidence using molecular dynamics simulations suggested that binding of two catalytic Mg2+ ions at the RuvC domain could mediate flexible cleavage generating 1-bp 5' overhangs, and biochemical evidence demonstrated that RuvC can cleave the non-target strand at different positions (Jinek et al., 2012; Jones et al., 2021; Shi et al., 2019; Shou et al., 2018; Zuo & Liu, 2016). The flexible cleavage of RuvC was proposed to mediate precise and predictable insertions (Allen et al., 2019; W. Chen et al., 2019; Leenay et al., 2019; Lemos et al., 2018; Santiago Gisler et al., 2019; Shen et al., 2018; Shi et al., 2019; Shou et al., 2018; Taheri-Ghahfarokhi et al., 2018), but the observed frequencies and determinants of staggered DSB ends were never investigated owing to the lack of tools for assessing scission profiles. Using BreakTag, we characterized, to our knowledge for the first time, the relative frequency of, and the factors that determine, the different types of Cas9-induced breaks. We observed that staggered ends represent approximately 35% of SpCas9 on-target and off-target DSBs, and we identified a strongly sgRNA-specific scission profile, highlighting that sequence context plays a role in the positioning of the RuvC domain. Our findings reveal a strong dependence of guanines in the RuvC cleavage site positioning. If guanine occupied position 17, the RuvC domain was more likely to cut between positions 17 and 18, generating a blunt DSB. Conversely, a guanine at position 18 shifted the RuvC cleavage site upstream of the HNH cut, generating staggered DSBs. Using a large matched dataset directly associating Cas9-induced scission profile with the repair outcome and a parallel assessment of repair outcomes of targets predicted to be cut in a blunt or staggered manner, we show that staggered DSBs generate predictable templated insertions with higher precision and that the frequency of templated insertions is increased by targeting sites with a guanine at position 18 for SpCas9. Because single-nucleotide insertions are the most common CRISPR–Cas9 repair outcome (Allen et al., 2019; Chakrabarti et al., 2019; Leenay et al., 2019; Overbeek et al., 2016; Shen et al., 2018; Taheri-Ghahfarokhi et al., 2018), and are valuable for the correction of pathogenic alleles with single-base deletions or gene knockouts, our findings demonstrate that enhancing template-free precise and predictable genome editing is possible by selecting target sites with a staggered cleavage configuration. This is an achievable goal, as modeling the human genome revealed that approximately 18% of potential target sites found in exons are predicted to be cleaved by SpCas9 in a highly staggered configuration. The indel landscape is shaped by different DNA repair pathways influenced by the chromatin environment (Ruben Schep et al., 2021; Xue & Greene, 2021), which might account for the slight deviation in sequence determinants of indels identified by computational predictors trained on repair outcome data (Allen et al., 2019; Chakrabarti et al., 2019; W. Chen et al., 2019; Leenay et al., 2019; Shen et al., 2018; Taheri-Ghahfarokhi et al., 2018) compared to cleavage determinants identified by BreakTag.

Base editors and prime editors allow direct modification of the locus without relying

on a DNA DSB, reducing the likelihood of misrepair that can lead to illegitimate chromosome joining (Anzalone, Koblan, & Liu, 2020). However, base editors are limited to base conversions and cannot induce insertions (Anzalone et al., 2020). Prime editors allow the formation of insertions, deletions and base conversions, but further development is necessary to increase editing efficiencies (Z. Zhao, Shang, Mohanraju, & Geijsen, 2023). Although both prime and base editors bypass the need of a DNA DSB, recent evidence revealed the presence of genotoxic effects associated with this generation of editors, including deleterious deletions and translocations (Fiumara et al., 2024). Cas9 scission profile-based pre-selection of gRNAs for precise insertions is limited to the correction of small deletions but still has a high translational potential as single-nucleotide deletions represent more than 31,000 of pathogenic variants in ClinVar (Fig. 3.6b,c).

Human genetic variation is ubiquitous and was shown to impact Cas9 on-target activity and the off-target landscape (Cancellieri et al., 2023; Krysler et al., 2022; Lazzarotto et al., 2020; Lessard et al., 2017; Scott & Zhang, 2017). In the present study, we identified a central role for genetic variation in genome editing by CRISPR–Cas9 by demonstrating that the presence of SNPs at key positions along the protospacer modulate the indel outcome via changes in the Cas9 cleavage profile. More specifically, we directly demonstrate that SNPs found at positions 17 or 18 of the protospacer alter the SpCas9 scission profile, which dictates genome editing outcome. This notable finding has direct implications for the clinical use of CRISPR–Cas9. Altogether, our findings indicate that personalized genetic variation must be considered at the early stages of designing CRISPR–Cas9 targeting strategies. Furthermore, SNP-driven changes in Cas9 scission profile afford opportunities for precise allele-specific gene editing, and this places BreakTag as an experimental framework for predicting and identifying target sites susceptible to precise and desirable editing.

In a further step, we characterized the scission profile of several Cas9 variants and identified LZ3 as having a skewed distribution in favor of staggered DSBs. LZ3 has been identified as a Cas9 variant exhibiting a distinct insertional profile, with a preference of +1 indels at 19G loci (Schmid-Burgk et al., 2020), further supporting our conclusion that an intrinsic link exists between scission profile and gene editing outcome. LZ3Cas9 contains four mutations—N690C (REC3), G915M (linker 2), N980K (RuvC) and T769J (linker 1)—that confer its higher specificity and/or altered scission profile. Interestingly, another study identified a G915F mutation in an engineered Cas9 variant with an altered scission profile (Shou et al., 2018), indicating that interactions between the linker 2 (L2) domain and the non-target strand might promote a flexible scission. Of note, the residue Gly915 in L2 interacts with position 18 of the non-target strand (F. Jiang & Doudna, 2017); a guanine at position 18 might change the interaction between the non-target strand and Cas9, displacing the RuvC cleavage site. SpCas9 demonstrated a higher incidence of blunt cuts at on-targets compared to off-targets, in line with previous findings on mismatched synthetic substrates for three gRNAs (Jones et al., 2021). Interestingly, we show here that the LZ3 generates a higher proportion of staggered cuts at on-targets compared to off-targets, suggesting that the presence of mismatches can increase or decrease staggered

cleavage in a variant-dependent manner. Taken together, the data-rich BreakTag workflow allows the assessment of variant fidelity, activity and determinants of nuclease scissions within a single assay, providing a platform for a fast, efficient and unbiased discovery of nuclease function.

Finally, we demonstrated how templated insertions can be explored for the correction of pathogenic single-nucleotide deletions. We leveraged flexible scission profile determinants of SpCas9 and LZ3 to predict pathogenic alleles amenable for precise corrective gene editing via predictable insertions. We envision that future development of engineered Cas9 variants with increased fidelity, alternate sequence determinants for staggered cleavage and decreased PAM requirements would expand the collection of sites amenable to precise gene editing.

In summary, we characterized the Cas9 endonuclease scission profile and established that the sequence of CRISPR–Cas9 target sites, human genetic variation and alternative Cas9 variants are three principal influencers of Cas9 cleavage pattern and, therefore, of gene editing outcomes. Our work illuminates the fundamental properties of Cas9 nuclease activity and lays the foundation for harnessing the flexible scission profile of Cas9 and engineered variants for precise, predictable and personalized genome editing.

3.6 Methods

3.6.1 Cell culture and genomic DNA extraction

Human osteosarcoma U2OS cells (American Type Culture Collection (ATCC)), human embryonic kidney cells (HEK293, ATCC) and HepG2 cells (a gift from Julian König's laboratory) were cultured in DMEM (Gibco, 41965062) supplemented with 10% FBS (PAN-Biotech, P40-37500), 100Uml^{-1} penicillin–streptomycin and 2mM L-glutamine. K562-Cas9 cells (GeneCopoeia, SL552) were cultured in RPMI1640 medium (Gibco, 11875093) supplemented with 10% FBS (PAN-Biotech, P40-37500), 100Uml^{-1} penicillin–streptomycin and 2mM L-glutamine and kept under selection with hygromycin. HeLa Kyoto cells were infected with viral particles from LentiCas9-Blast (Addgene, 5292), and stable clones expressing Cas9 were maintained in DMEM supplemented with 10% FBS, 100Uml^{-1} penicillin–streptomycin, 2mM L-glutamine and 7μgml^{-1} blasticidin. Immortalized B cells from GIAB donors Chinese son (GM24631, Coriell), Chinese father (GM24694, Coriell), Chinese mother (GM24695, Coriell), Ashkenazi Jewish son (GM24385, Coriell) and Ashkenazi Jewish mother (GM24143, Coriell) were maintained in RPMI 1640 medium (Gibco, 11875093) supplemented with 15% FBS (PAN-Biotech, P40-37500), 100Uml^{-1} penicillin–streptomycin and 2mM L-glutamine. All cell lines were maintained in a humidified incubator at 37°C supplemented with 5% CO2.

The gDNA of cells was extracted using a Qiagen Blood & Tissue Kit (Qiagen, 69506) following the manufacturer's instructions and eluted in nuclease-free water.

gDNA of GIAB (Zook et al., 2016, 2019) individuals was purchased from Coriell:

Table 3.1: Oligos for Tn5 loading

Step	Temperature	Time
1	95°C	5 min
2	65°C	-0.1°C/s
3	65°C	5 min
4	4°C	-0.1°C/s
5	4°C	Hold

Table 3.2: Oligos for BreakTag linker preparation

Step	Temperature	Time
1	95°C	5 min
2	Cool to 25°C	-0.1°C/s
3	25°C	Hold

female Utah/Mormon (NA12878), Ashkenazi Jewish son (NA24385), Ashkenazi Jewish father (NA24149), Ashkenazi Jewish mother (NA24143), Chinese son (NA24631), Chinese father (NA24694) and Chinese mother (NA24695).

3.6.2 Expression and purification of homemade Tn5

Expression and purification of hyperactive Tn5 (E54K, L372P) were performed as described previously (Hennig et al., 2018) with the following modifications: Tn5 was expressed as an N-terminal His_6-GST fusion followed by a 3C protease cleavage site. GSH affinity purification was used to capture the fusion protein and it was subsequently cleaved using recombinant 3C protease.

3.6.3 Tn5 loading and BreakTag linker preparation

Tn5-B adapter was prepared by mixing 100 μM Tn5ME-B (Illumina FC-121-1031) and 100 μM Tn5MErev (Picelli et al., 2014) (Supplementary Table 7) resuspended in annealing buffer (50 mM NaCl, 40 mM Tris, pH 8) at a 1:1 ratio. The oligos were annealed in a thermocycler programmed as follows:

Homemade Tn5 was loaded with pre-annealed Tn5 B adapter for 1 hour at room temperature with agitation (300 rpm) in a thermoshaker.

The BreakTag linker was prepared by combining 10 μM BreakTag_fwd and 10 μM BreakTag_rev oligos (Supplementary Table 7) in T4 polynucleotide kinase buffer (NEB M0201S). The oligos were annealed in a thermocycler programmed as follows:

3.6.4 In vitro digestion of gDNA with Cas9 ribonucleoproteins

RNPs were assembled by mixing Cas9 and sgRNA at equimolar ratios in NEB 3.1 buffer (NEB, B72030), followed by incubation at 37°C for 10min. For HiPlex BreakTag, pools were mixed with the nuclease at a 2:1 ratio. An input of 500ng of gDNA was mixed with each RNP at a final concentration of 90nM and incubated at 37°C for 1h in a thermocy-

cler with the lid set at 37°C. The reaction was terminated by adding RNase A (Thermo Fisher Scientific, 10753721) and proteinase K (NEB, P8107) at final concentrations of $0.8\mu g\mu l^{-1}$ and $0.2\mu g\mu l^{-1}$, respectively, at 37°C for 20min, followed by incubation at 55°C for 20min. Nuclease-digested gDNA was purified with DNA AMPure XP beads (1.2x volumes, Beckman Coulter, A63881).

3.6.5 HiPlex sgRNA production

Sequences for HiPlex1 (ref. (Chakrabarti et al., 2019)) and HiPlex2 (ref. (Allen et al., 2019)) pools (Supplementary Table 1) were bioinformatically split into 10 pools. Each pool contained 150 gRNAs for HiPlex1 and 140 gRNAs for HiPlex2, modified as follows: the last nucleotide at the 5' end of the gRNA sequence (position 20) was replaced with a G for efficient T7 transcription. A T7 promoter sequence 5'-GGATCCTAATACGACTCACTATAG-3' was added at the 5' end of the protospacer, and a SpCas9 scaffold sequence 5'-GTTTTAGAGCTAGAA-3' was added at the 3' end. The sequences were ordered as DNA oPools (Integrated DNA Technologies (IDT)) and reconstituted in nuclease-free water at 100μM. In-house production of sgRNAs was performed using the HighYield T7 sgRNA Synthesis Kit (SpCas9) (Jena Bioscience, RNT-105) following the manufacturer's instructions. In brief, each pool (1μM) was used for an assembly PCR reaction using three primers: T7fwd_sRNA: 5'-GGATCCTAATACGACTCACTATAG-3', T7rev_sgRNA: 5'-AAAAAAGCACCGACTCGG-3' and SpCas9_scaffold: 5'-AAAAAAGCACCGACTCGGTGCCACTTTTTCAAGTTGATAAC-GGACTAGCCTTATTTTAACTTGCTATTTCTAGCTCTAAAAC-3'. To increase complexity and avoid PCR bias, we performed three separate PCR reactions for each pool, which were then combined before IVT. The expected size of the assembled DNA template was confirmed on an agarose gel and used directly for T7 IVT. Three IVT reactions per pool were performed for increased yield and were incubated for 90min at 37°C. IVT products were purified using 2x volumes of Agencourt RNAClean XP magnetic beads (Beckman Coulter, A66514) and resuspended in nuclease-free water. RNA concentration was estimated using Qubit RNA Broad Range (Invitrogen, Q10211).

3.6.6 BreakTag procedure and sequencing

DNA DSB ends of nuclease-digested gDNA were repaired and 3' adenylated using the NEBNext Ultra II End Repair/dA-Tailing Module (NEB, E7546) according to the manufacturer's instructions with the following modification: the total volume of the reaction was halved by using half the volume of the reagents. Labeling of DSB ends by ligation with the BreakTag linker was performed using the NEBNext Ultra II Ligation Module (NEB, E7595) according to the manufacturer's instructions with the following modifications: the total volume of the reaction was halved by using half the volume of the reagents, and the USER enzyme digestion step was omitted. The BreakTag linker was used at a final concentration of 50nM per sample. Labeled DNA was size selected two times us-

Table 3.3: BreakTag procedure and sequencing

Step	Temperature	Time
1	72°C	5 min
2	98°C	30 s
3	98°C	10 s
4	63°C	30 s
5	72°C	60 s
6	72°C	5 min
7	12°C	Hold

ing 0.7x volumes of DNA AMPure XP beads (Beckman Coulter, A63987) and eluted in nuclease-free water. Tagmentation with in-house Tn5 was performed in freshly prepared 10mM Tris-HCl (pH 7.5) buffer containing 10mM MgCl2 and 25% N,N-dimethylformamide (DMF, Sigma-Aldrich, 227056). Tagmentation reactions were assembled using 100–200ng of DSB-labeled DNA as input. Single-handle hyperactive Tn5 was used at a final concentration of 1.25ngμl^{-1} per reaction. Tn5 was loaded with the Tn5ME-B oligonucleotide for 1h at room temperature (Supplementary Table 7). The tagmentation mix was then incubated at 55°C for 5min in a pre-heated thermocycler followed by termination with 0.2% SDS at room temperature for 5min. Libraries were amplified with NEBNext Ultra II Q5 Master Mix (NEB, M0544) in a thermocycler programmed as follows:

Amplified and barcoded samples were size selected by performing two consecutive 0.5x volume right-tail + 0.35x volume left-tail size (final volume 0.85x) selections using DNA AMPure XP beads (Beckman Coulter, A63987). Libraries were quantified using a Qubit dsDNA High Sensitivity Assay Kit or a sparQ Universal Library Quant Kit (QuantaBio, 95210-100), and fragment size distribution was assessed on a Bioanalyzer High Sensitivity DNA chip. Libraries were pooled and sequenced on a NextSeq 500/550 platform with NextSeq 500/550 High Output Kit v2 chemistry for SE 1x75bp sequencing or NovaSeq PE 2x150bp with a 15% PhiX spike-in.

3.6.7 BreakTag data analysis with BreakInspectoR

Initial pre-processing was done in a Linux cluster using the BreakTag NGSpipe2go pipeline (`https://github.com/roukoslab/breaktag`). The pipeline processes raw reads as they are output by the sequencer and generates a BED file with coordinates containing DSBs. Raw reads (single-end or paired-end) were first scanned, and those not containing the expected 8-nt UMI followed by the 8-nt sample barcode in the 5' end of read 1 were discarded. Valid reads were aligned to the human reference genome version hg38 downloaded from UCSC with timestamp of 15 January 2014, 21:14, using the 'mem' command in BWA (version 0.7.17-r1188) (H. Li, 2013) with a seed length of 19 and default scoring/penalty values for mismatches, gaps and read clipping. Reads mapped with a minimum quality score Q=60 were retained to ensure that we worked only with uniquely mapping reads. A final de-duplication step was performed in which spatial consecutive reads mapping within a window of 30nt, and their UMIs differing by up to two mismatches, were considered close PCR duplicates, and only one was kept. The resulting reads were aggregated per position

and reported as a BED file.

Subsequent analysis was done using the BreakInspectoR package in R (`https://github.com/roukoslab/breakinspectoR`), which performs a guided search toward putative on-targets/off-targets. Starting from the previously generated BED files, BreakInspectoR identifies stacks of read ends near a PAM as candidate loci for containing a DSB, and it calculates a P value and a false discovery rate for each site identified, considering also the signal found in a non-targeted library. For HiPlex libraries, this process was sequentially repeated for all sgRNAs included in the pool. BreakInspectoR may identify ambiguous targets for sgRNAs in the pool that are separated by a Hamming distance of seven substitutions or less. Any ambiguous targets were removed from the list of all targets for a HiPlex library as necessary. The identification of sites required the function 'breakinspectoR()' to search for stacks of at least three read ends at a distance of 3nt from an 'NGG' PAM, which is preceded by a protospacer sequence that differs by seven mismatches at most from the sgRNA sequence. Only breaks identified in standard chromosomes were retained. For the 'PAM usage' analysis (Fig. 3.1g), we called 'breakinspectoR()' with the same parameters but allowing any PAM ('NNN'). RNA and DNA bulges in the off-targets nominated with BreakInspectoR were not excluded from the analysis.

3.6.8 Blunt rate estimation

For each site identified by BreakInspectoR, we analyzed the scission profile using the 'scission_profile_analysis()' function. This function analyzes the signal in the PAM-proximal side and returns a table in the form of a 'data.frame' attached as metadata columns of a 'GRanges' object (Lawrence et al., 2013). The table extends the coordinates of the original DSB with the signal found around the position at which the enzyme is expected to cut, a P value and a false discovery rate that assess the significance of the signal found outside the expected cut site compared to the non-target library and the classification of a site according to its preference for forming blunt or staggered breaks. We performed the analysis by using the function to look in a region between [-3, +3] nucleotides upstream/downstream of the expected cut site; for Cas9, this was 3nt upstream (toward the 5' end) from the PAM. To avoid sites that could mislead the analysis, we focused only on sites with an 'NGG' PAM, for which, in principle, expected cut sites are readily identified. Finally, from the table generated by 'scission_profile_analysis()', we could calculate the blunt rate for a site. We did this in two ways: (1) as a fraction of the signal found in the expected cut site (PAM 3nt upstream—that is, position 17 of the protospacer) and the total amount of signal in the region [-3, +3] around the cut site and (2) as a log2 ratio of the signal in the expected cut site versus the signal in the region [-3, +3] around the cut site after excluding the signal in the cut site.

3.6.9 Machine learning model for the prediction of blunt rates

We trained a machine learning model to predict scission profiles using the XGBoost flavor of the Gradient Boosting Machine algorithm implemented in the H2O.ai framework (Extended Data Fig. C.4a). The software was installed in the Bioconductor R container release version 3.15 (ref. (Huber et al., 2015)) (bioconductor/bioconductor_docker:RELEASE_3_15). We tuned the hyperparameters of the algorithm to use 1,000 trees of unlimited depth, DART as the booster algorithm (Rashmi & Gilad-Bachrach, 2015) and five folds for K-fold cross-validation with automatic fold assignment of instances.

Because the number and scission profiles of the identified targets differ greatly among sgRNA constructs, we used only a subset of the total identified targets as training instances. We selected only highly covered sites with at least 16 raw reads in the PAM-proximal side and accounted for specific biases. We limited the number of targets selected per sgRNA to 100 to avoid biases toward highly promiscuous sgRNA sequences and additionally sampled staggered targets with a probability K-1, where K is the ratio between the number of staggered (blunt reads<20%) and blunt (blunt reads>80%) targets for a specific sgRNA, to pick more from the pool of staggered targets and compensate for their under-representation in the total set of identified targets. This resulted in a final set of 18,759 'instances' in the training set.

The 'response' variable to be predicted was the log2 ratio between the number of raw reads mapped in the PAM-proximal side exactly at position 17 of the protospacer (the expected cut site) and the sum of raw reads mapped in the PAM-proximal side found in positions 14–16 and 18–20 of the protospacer. A pseudocount was added to both the denominator and numerator of this fraction to avoid a division by 0.

We reflected in the 'predictor' variables both the on-target/off-target protospacer sequence and the actual gRNA sequence, along with the mismatches between the two. We performed one-hot encoding by constructing a 4x4 matrix for each of the 20 positions of the protospacer, each row representing one of the possible nucleotides (A, C, G, T) to occupy that position in the targeted protospacer, and in each column the same for the sgRNA sequence. The matrix was filled with '0' with the exception of the cell representing the nucleotide in the protospacer (row) and the sgRNA (column) for that position, which would contain '1'. Each matrix was converted into a vector of length 16 by concatenating the column vectors, and, finally, the 20 vectors were concatenated into one large vector of length 320 with the final representation of the one-hot encoding. In addition, we included an additional predictor variable representing the number of mismatches between the targeted protospacer and the sgRNA sequence in the first 10 positions of the protospacer and a second variable representing the mismatches in the last 10 positions of the protospacer. In total, we used 322 variables to represent each training instance. Sequence motifs related to the scission profile were produced with the ggseqlogo package in R (Wagih, 2017).

3.6.10 Selection of SNP-containing sites in Genome in a Bottle genomes for HiPlex BreakTag

We downloaded the VCF file containing the single-nucleotide variants (SNVs) called in GIAB (Zook et al., 2016) (Supplementary Table 9). We filtered the files to retain SNPs only and retrieved the 20bp of sequence context around those sites. We retained two subsets of 394,585 and 395,392 putative CRISPR–Cas9 target sites that contain an 'NGG' PAM preceded by a protospacer containing at positions 17 or 18 (respectively) a SNP found in at least one of the GIAB samples. We then used the reduced machine learning model, which uses only the last 10 positions of the protospacer, to predict the expected blunt rate of those putative target sites for the reference allele sequence targeted with an sgRNA matching the reference sequence and also for the mutated allele targeted with an sgRNA containing the mutation. The top 150 sites with the lowest blunt rates (75 in sense and 75 in antisense strands) and targets with the highest predicted changes were selected for HiPlex BreakTag sgRNA pool generation. For greater statistical power, we selected sites for which the alternative allele is found in three or four donors.

3.6.11 Genome in a Bottle SNP analysis

We used the 'scission_profile_analysis()' function in BreakInspectoR to obtain the scission profile of the 300 sites picked from the previously selected SNP-containing sites in GIAB genomes. We calculated the blunt rate as the fraction of the BreakTag signal in the expected cut site (position 17 of the protospacer) with respect to the total signal in the region [-3, +3] around the cut site, obtaining an approximation for the number of blunt breaks compared to the total number of breaks as captured by BreakTag. For the visualizations comparing the blunt rate and the genotype, we selected highly covered sites with at least 16 raw reads in the PAM-proximal side and reference and alternative genotype information in at least one sample for each genotype.

3.6.12 1000 Genomes database SNP analysis

The full set of biallelic SNVs and indels called by Lowy-Gallego et al. (Lowy-Gallego et al., 2019) from phase three of the 1000 Genomes Project was downloaded from the EBI's FTP server (`http://ftp.1000genomes.ebi.ac.uk/vol1/ftp/data_collections/1000_genomes_project/release/20190312_biallelic_SNV_and_INDEL/ALL.wgs.shapeit2_integrated_snvindels_v2a.GRCh38.27022019.sites.vcf.gz`) with the timestamp of 12 March 2019, 16:06. We further processed the file to keep only the SNPs that were called in at least 10% of the samples used in this call set (n=5,248). The positions of the SNPs were cross-referenced with a table of all 11,431,163 putative CRISPR–Cas9 targets on exons annotated in the Ensembl version 98 database (Cunningham et al., 2021) that have an NGG PAM. We shortlisted two subsets of 18,961 and 18,883 putative target sites with a SNP at positions 17 or 18 (respectively) of the protospacer sequence. We

then used the reduced machine learning model, which uses only the last 10 positions of the protospacer, to predict the expected blunt rate of those putative target sites for the reference allele sequence targeted with an sgRNA matching the reference sequence and also for the mutated allele targeted with an sgRNA containing the mutation.

3.6.13 Prediction of blunt rates of gRNAs targeting pathogenic deletions

The full set of variants annotated in ClinVar as of April 2023, comprising a total of 2,122,310 variants, was downloaded from the National Institutes of Health FTP server (`https://ftp.ncbi.nih.gov/pub/clinvar/vcf_GRCh38/clinvar.vcf.gz`). Only variants that were 1-nt deletions, located in standard chromosomes, overlapping an exon annotated in TxDb.Hsapiens.UCSC.hg38.knownGene (data package made from resources at UCSC on 16:50:30+0000, Thursday, 7 April 2022) and annotated in ClinVar as 'Pathogenic' or 'Likely_pathogenic', were considered (31,010 variants). We focused on a subset of 8,705 deletions that had an NGG motif directly adjacent to them in either strand and up to 4nt upstream. Those sites were candidates for being cut by Cas9 in a staggered manner, which could potentially induce a templated +1 insertion as the repair outcome, correcting the frameshift in the pathogenic allele and potentially recovering the original protein sequence. We calculated that a total of 4,999 of those deletions would recover the original protein sequence with a templated +1 insertion. Next, we designed 'in silico' the gRNA sequences that would target the regions containing the deletions, and we estimated the blunt rate using the previously described XGBoost models for SpCas9 and LZ3 trained with the HiPlex library. Those sites predicted to be cut in a highly staggered manner (log2 blunt rate<-2) in which a templated insertion would recover the original protein were finally reported as pathogenic variants being potentially treated with a CRISPR–Cas9 therapy.

3.6.14 Construction of gRNA-target pair lentiviral libraries

Using our XGBoost models for SpCas9, we predicted the blunt rate of human genome sites and selected 150 sites predicted to be cut mostly blunt and 150 sites predicted to be cut mostly staggered. For the 'ALT' and 'REF' libraries, all gRNAs used in the HiPlex3 dataset were used. The cloning strategy of gRNA-target pair lentiviral libraries was adapted from Allen et al. (Allen et al., 2019). In brief, a scaffoldless lentiviral expression vector, pKLV2-U6(BbsI)-PKGpuro2ABFP-W, was generated by removing the improved gRNA SpCas9 scaffold from pKLV2-U6gRNA5(BbsI)-PKGpuro2ABFP-W34 (gift from Kosuke Yusa, Addgene plasmid no. 67974). The deletion was generated by amplifying two fragments encompassing the 5' end of the AmpR cassette to U6 promoter and PGK promoter of the 3' end of the AmpR cassette, followed by Gibson assembly. The empty vector was transformed into Stabl3 chemically competent cells; single colonies were picked; and scaffold deletion was confirmed via Sanger sequencing.

For the library cloning step, we generated a 170-nt oligonucleotide pool (IDT) encoding the gRNA and a portion of the allele sequence containing 79 nucleotides with the target sequence + PAM in the center for the four individual libraries (Extended Data Fig. C.5a). The oligonucleotide was amplified with primers compatible with the scaffold used, and a Gibson assembly was used to fuse the amplified pool to a 193-nt Ultramer duplex (IDT) encoding the improved version of the gRNA scaffold and a spacer sequence (Allen et al., 2019). Three separated Gibson assembly reactions were performed per pool at a 1:1 molar ratio, followed by an incubation for 1h at 50°C, and subsequently pooled for column-based purification (Monarch PCR & DNA Cleanup Kit, NEB, T1030S), and removal of linear DNA was achieved by treating the samples with Plasmid-Safe ATP-Dependent DNAse (Epicentre). The intermediate circular insert and scaffoldless vector were linearized with a FastDigest BpiI (IIs class) kit (Thermo Fisher Scientific, FD1014) for 30min and ligated in triplicates per pool (T4 DNA ligase, NEB, M0202). The replicates were pooled and transformed in Stabl3 chemically competent cells.

3.6.15 Transduction of gRNA-target lentiviral pools

For lentiviral packaging of gRNA-target libraries, the gRNA-target libraries were independently co-transfected with the two packaging plasmids, and the supernatants were pooled and concentrated 50–100-fold. Packaging and transduction were performed as described previously (Papapetrou & Sadelain, 2011). In brief, we produced the viruses by co-transfection of 293T cells with each of the four library pools and two helper plasmids, psPAx2 and pMD2.g, encoding the VSV-G envelope and the lentiviral gag-pol genes, respectively. We harvested the lentiviral vector-containing supernatant twice, at approximately 42h and 66h after transfection, and concentrated it by using Lenti-X Concentrator (Takara, 631232). We plated 300,000 cells in a well of a six-well plate and transduced with the vector supernatants and $4\mu gml^{-1}$ polybrene in a total volume of 2ml. After 48h, the transduced cells were removed from the six-well plate, and one fifth of the cells were tested for BFP expression by flow cytometry (BD Canto), whereas the rest were plated in 10-cm2 tissue culture dishes for selection with puromycin ($1\mu gml^{-1}$). Cells were kept under puromycin selection for 5d. On the last day, cells were collected and tested for BFP expression, and gDNA was isolated using the Qiagen Blood & Tissue Kit (Qiagen, 69506).

3.6.16 gRNA-target pair amplicon sequencing library preparation

The region containing the gRNA sequence and 79-nt portion of the allele was amplified using the Fwd_pool and Rev_pool primers (Supplementary Table 13) with NEBNext Ultra II Q5 Master Mix (NEB, M0544) with the following program: 98°C for 60s, 24 loops of 98°C for 10s and 72°C for 30s, followed by a final extension at 72°C for 2min. The PCR product was purified using 0.9x volumes of DNA AMPure XP beads (Beckman Coulter, A63987) and eluted in nuclease-free water. The cleanup product was used for a second PCR round with indexed primers (Supplementary Table 13) with the following

conditions: 98°C for 60s, 13 loops of 98°C for 10s, 67°C for 10s and 72°C for 20s, followed by a final extension at 72°C for 2min. The indexed libraries were pooled, and the band corresponding to the amplicon size (464bp) was excised from a 2% agarose gel, purified and sequenced in paired-end mode (2x150bp) in a NextSeq 2000 sequencer with 40% PhiX spike-in.

3.6.17 Analysis of gRNA-target repair outcomes

The first read in pair was used solely to estimate the abundance of each gRNA, as it reads into the gRNA portion of the construct. The second pair that reads into the target sequence was reverse complemented with the fastx_toolkit (`http://hannonlab.cshl.edu/fastx_toolkit`) and stripped from the first 57 bases and kept only the immediate 79nt using Trimmotatic (Bolger et al., 2014) with options SE HEADCROP:57 CROP:79, which would keep only the 79-nt-long portion of the read containing the actual amplicon of the targeted sequence. Processed reads from technical replicates were merged in a single FASTQ file, and indels were called using CRISPResso2 (ref. (Clement et al., 2019)) in pooled mode (CRISPRessoPooled), restricting the analysis to regions with at least 100 aligned reads and ignoring substitutions other than indels. gRNAs with detected activity in wild-type (WT) cells not expressing Cas9 that had been reported in the CRISPResso2 analysis with at least 100 edited reads were excluded from the analysis. For the rest, we extracted from the CRISPResso2 analysis output the length of the indel, the frequency of the most common +1 insertion over all edited sequences and the inserted nucleotide.

3.6.18 Nucleofection of RNP complexes into lymphoblastoid cells

For the preparation of RNP complexes, sgRNAs targeting SNP-containing loci (Supplementary Table 8) were generated in-house using the HighYield T7 sgRNA Synthesis Kit (SpCas9) (Jena Bioscience, RNT-105). Two hundred picomolar sgRNA was mixed with 100pM Alt-R S.p. Cas9-GFP V3 (IDT, 10008100) and incubated at room temperature for 10min. A total of 5x105 cells per reaction were resuspended in SF Cell Line 4D-Nucleofector solution (Lonza, V4XC-2032) and nucleofected in a 4D-NucleoFector system using the pulse code DN-100. Nucleofected cells were transferred to a plate containing culture medium and kept in a humidified incubator at 37°C supplemented with 5% CO2 for 3d before gDNA was extracted for indel analysis.

3.6.19 Amplicon sequencing and editing analysis using CRISPResso2

The gDNA of lymphoblastoid cells nucleofected with RNPs was extracted 3d after CRISPR delivery. Approximately 100ng of gDNA from each sample was used for locus amplification using the primers listed in Supplementary Table 8. Amplicon libraries were generated as described previously (Yau & Rana, 2018) with the following modifications: a first round of amplification using NEBNext Ultra II Q5 Master Mix (M0544) was performed with 33

cycles. The amplified DNA was purified using a 1x volume of DNA AMPure XP beads (Beckman Coulter, A63987), and the entire purified product was used for a second round of PCR with primers containing p5 and p7 sequences for Illumina sequencing (Supplementary Table 8). Amplicons were pooled and sequenced in a MiniSeq sequencer in single-read mode and 150 cycles.

Indel analysis was performed in a local Linux cluster using CRISPresso2 in pooled format (Clement et al., 2019) using the following parameters: *–amplicon_min_alignment_score 50 –quantification_window_size 10 –quantification_window_center -3 –exclude_bp_from_left 0 –exclude_bp_from_right 0 –ignore_substitutions –plot_window_size 20 –min_frequency_alleles_around_cut_to_plot 0.*

3.6.20 Cas9 variant cloning, expression and purification

The pET-Cas9-NLS-6xHis expression vectors for Cas9 variants were generated by using Gibson assembly. As a PCR template for the expression vector backbone, pET WT Cas9-NLS-6xHis was used (Zuris et al., 2015) (Addgene plasmid no. 62933). The PCR templates for the Cas9 variants were pX165-LZ3 Cas9 (Addgene plasmid no. 140561), pX165-evoCas9 (Addgene plasmid no. 140569), pX165-xCas9 (Addgene plasmid no. 140568), pX165-HypaCas9 (Addgene plasmid no. 140567) and pX165-SniperCas9 (Addgene plasmid no. 140560).

The pET expression vectors were transformed into Escherichia coli BL21 (DE3) CodonPlus (Agilent) and grown at 37°C and 140r.p.m. until an optical density at 600nm (OD600) value of 0.5 was achieved. Cultures were cooled to 18°C on ice, and protein expression was induced using IPTG at a final concentration of 0.5mM and incubated for a further 21h at 18°C and 140r.p.m. Cells were harvested by centrifugation (4,000g, 15min), resuspended in ice-cold lysis buffer (30mM Tris-HCl, 500mM NaCl, 10mM imidazole, 1mM MgCl2, 1mM TCEP, 5% glycerol, 1x complete protease inhibitor, 100Uml^{-1} benzonase, pH 8.0) and lysed by high-pressure homogenization at 28kpsi (Constant Systems CF1 Cell Disruptor). Cells were cleared by centrifugation (40,000g, 30min, 4°C), and the cleared lysate was applied to a HisTrap FF 5-ml column (Cytiva), using an automated chromatography system (Bio-Rad, NGC Quest Plus; used for all chromatography steps). The column was washed with 20 CV wash buffer (30mM Tris-HCl, 500mM NaCl, 10mM imidazole, 5% glycerol), and the Cas9 variants were eluted from the Ni–NTA column by applying a linear gradient of 10–500mM imidazole (containing 30mM Tris-HCl, 500mM NaCl, 5% glycerol). The eluted proteins were diluted 1:10 in a low-salt buffer (25mM Na–HEPES, pH 7.2, 100mM NaCl, 5% glycerol), applied to a HiTrap Heparin 5-ml column (Cytiva) and eluted by applying a linear NaCl gradient from 100mM to 1,000mM. Elution fractions containing the Cas9 variants were pooled and concentrated using Amicon Ultra-15 spin concentrators (Merck). Concentrated proteins were applied to a gel filtration column (Superdex 200 16/60pg, Cytica, 40mM Na–HEPES, pH 7.4, 400mM NaCl, 10% glycerol). Peak fractions

containing the Cas9 variants were pooled, concentrated to 6.4gL^{-1} and diluted 1:2 with 86% glycerol to a final concentration of 3.2gL^{-1} (20μM). HiFiCas9 was purchased from IDT (no. 1081060).

3.7 Data availability

All genomics data produced in this study have been deposited in the Gene Expression Omnibus under accession number GSE223772. Source data are provided with this paper.

3.8 Code availability

The BreakInspectoR pipeline and relevant bioinformatics pipelines used in this study can be found at `https://github.com/roukoslab/breaktag` and at `https://github.com/roukoslab/breakinspectoR`.

3.9 Acknowledgements

We thank [redacted] for critically reading the manuscript. The plasmids pX165-LZ3 Cas9 (Addgene, 140561), pX165-evoCas9 (Addgene, 140569), pX165-xCas9 (Addgene, 140568), pX165-HypaCas9 (Addgene, 140567) and pX165-Sniper-Cas9 (Addgene, 140560) were kind gifts from [redacted]. We thank [redacted], [redacted], [redacted] and the [redacted] laboratory for sharing the amplicon sequencing raw data on high-fidelity Cas9 variants. We thank [redacted] and [redacted] for sharing instrumentation and [redacted] for organizational and logistic support. Support by the IMB Genomics Core Facility and the use of its NextSeq 500 (INST 247/870-1 FUGG) is gratefully acknowledged. The Roukos laboratory is supported by funding from the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation; project IDs 393547839–SFB 1361, 402733153-SPP 2202 and 455784893, to V.R); the DFG Major Research Instrumentation Program (INST 247/845-1 FUGG, to V.R); the Fondation Santé; MEDICUS grants ($\phi\kappa$ 81969, to V.R); and the Hellenic Foundation for Research and Innovation (HFRI, $\epsilon\lambda\iota\delta\epsilon\kappa$, 14925, to V.R).

3.10 Author Information

These authors contributed equally: [redacted], Sergi Sayols.

3.10.1 Authors and Affiliations

Institute of Molecular Biology (IMB), Mainz, Germany

[redacted], Sergi Sayols, [redacted], [redacted], [redacted] & [redacted]

Department of Biology, Medical School, University of Patras, Patras, Greece

[redacted] & [redacted]

Johannes Gutenberg University (JGU), Mainz, Germany

[redacted]

3.10.2 Contributions

G.M.C.L., S.S. and V.R. conceived and designed the study. G.M.C.L. designed BreakTag and performed all experiments. S.S. wrote BreakInspectoR and trained the machine learning model. G.M.C.L. and S.S. performed the bioinformatics analyses. A.G.K. performed the transduction of gRNA-target lentiviral pools in the Cas9-expressing cells and library preparation for amplicon sequencing. V.R. supervised the study. M.M. and S.H. cloned and produced the recombinant engineered Cas9 variants, and P.B. provided expertise. G.M.C.L. and V.R. wrote the manuscript, with input from all authors.

3.10.3 Corresponding author

Correspondence to Vassilis Roukos.

3.11 Ethics declarations

3.11.1 Competing interests

G.M.C.L., S.S., and V.R. are inventors in a pending patent application related to BreakTag and BreakInspectoR. The remaining authors declare no conflict of interest.

3.12 Peer review

3.12.1 Peer review information

Nature Biotechnology thanks the anonymous reviewers for their contribution to the peer review of this work.

3.13 Additional information

3.14 Rights and permissions

Discussion and outlook

Over the past decade, Next Generation Sequencing has emerged as a key technology in Molecular Biology which enabled researchers to address a variety of questions in a genome-wide and cost-effective manner. This dissertation presents 3 different bioinformatics methods implemented as R packages, covering orthogonal areas in the application of high-throughput sequencing of nucleic acids: i) quality control of low-input and single-cell sequencing of RNA, ii) functional interpretation of gene sets, and iii) genome-wide detection of CRISPR offtargets and double-strand break structure.

The decision to implement these methods as R packages aligns with the longstanding tradition of using R (R Core Team, 2021) —the language and platform— within the biological sciences, a practice later adopted by the bioinformatics field. This long and successful marriage has evolved into a solid and mature platform supported by a strong ecosystem led by leading experts in the field, establishing R —along with Python in some areas such as in single-cell or spatial-omics— as the *de facto* standard for *omics* data analysis. A notable example is the Bioconductor project (Huber et al., 2015), which extends R by providing software, annotation and experiment packages tailored for bioinformatics applications. Leveraging the popularity of Bioconductor and its open structure for contributions, I made the tools *rrvgo* and *dupRadar* available as part of the project, extending their accessibility to a broader audience.

While the chapters presented in this dissertation address the research hypothesis in detail and achieve the defined goals, this work has also generated new, compelling questions for further development. The context of these results, along with their limitations and potential future directions, are discussed in the following chapters.

Chapter 1: dupRadar

The article *dupRadar: a Bioconductor package for the assessment of PCR artifacts in RNA-Seq data* introduces *dupRadar*, a tool designed to assess the presence and impact of PCR duplicates in RNA-Seq datasets. PCR duplicates, especially in low-input and single-cell RNA-Seq experiments, can confound the biological interpretation of sequencing data by artificially inflating read counts for highly expressed genes. *dupRadar* addresses this by modeling the relationship between gene expression levels and duplication rates, providing insights into whether observed duplication is due to technical artifacts or biological

processes. The tool integrates easily into standard RNA-Seq analysis pipelines, offering visualizations and metrics to ensure data quality. Overall, *dupRadar* offers a straightforward solution for monitoring and interpreting PCR duplication in RNA-Seq experiments, ensuring the reliability of downstream analyses.

For a common experimental setup, read duplication is to be expected and deduplication purely based on read sequence will be detrimental (Parekh, Ziegenhain, Vieth, Enard, & Hellmann, 2016). Genomic regions of high signal (*i.e.* highly expressed genes in an RNA-seq experiment) are expected to generate natural duplication, due to the fact that fragments from those abundant regions are more likely to be sampled for sequencing. The sampling bottleneck becomes a major problem for modern library preparation protocols which require very little starting material. One specific example of limited abundance of input material is with single-cell protocols. However, the use of unique molecular identifiers (UMIs) enable differentiation between unwanted PCR duplicates originating from a single molecule and biologically significant identical reads originating from distinct molecules (Fu, Wu, Beane, Zamore, & Weng, 2018). While the use of UMIs have become ubiquitous in single-cell protocols, it is not the case for bulk sequencing of cells and other protocols for low-input which require extensive monitoring of PCR duplicates upon sequencing in order to extract the right conclusions from the data.

While the article examines the impact of unwanted PCR duplicates specifically within the context of RNA-seq, the findings are applicable to other NGS assays (Dozmorov et al., 2015; Tian et al., 2019). Enrichment-based assays, such as ChIP-seq—where only a targeted fraction of the genome undergoes amplification—may encounter similar challenges as RNA-seq, particularly in regions with high signal density. In these cases, the limited number of unique read start positions can lead to signal saturation, resulting in an underestimation of the true signal in these regions. Currently, the use of UMIs in enrichment-based protocols remains uncommon, with duplicate removal systematically integrated into data analysis pipelines (Hitz et al., n.d.; Sims, Sudbery, Ilott, Heger, & Ponting, 2014). One possible solution to address the limit of read start positions is to use paired-end sequencing, where both ends of each fragment are sequenced. This approach reduces the likelihood of finding two identical start and ends positions. We encourage researchers to further investigate this issue, particularly in low-input protocols involving narrow regions of enrichment.

While *dupRadar* serves primarily as a diagnostic tool, the correction of these artefacts remains unresolved. Several methods to probabilistically identify clonal duplication have been explored (Baumann & Doerge, 2014; Mezlini et al., 2013; Roberts & Pachter, 2013), however it is not clear whether those artefacts get systematically removed without creating any further artefacts. Their use is not widespread either. Future improvements could include in *dupRadar* methods for correcting these artefacts. We currently recommend that researchers systematically screen for technical duplicates in RNA-seq and enrichment-based assays, utilize UMIs or paired-end sequencing whenever feasible, and consider repeating library preparation and sequencing if a high number of technical

duplicates are detected (although we acknowledge this may not always be an option for rare low input samples).

Chapter 2: rrvgo

The article *rrvgo: a Bioconductor package for interpreting lists of Gene Ontology terms* introduces *rrvgo*, a tool developed to address the challenge of interpreting large, redundant lists of Gene Ontology (GO) terms often generated in high-throughput *omics* studies.

Ontologies, through controlled and structured vocabularies, have enabled researchers to interpret results in a genome-wide context. Gene Ontology (The Gene Ontology Consortium et al., 2023) is a widely adopted tool for this purpose; however, its extensive number of terms can make it challenging to apply in certain contexts. Traditional methods, such as GO Slims (Carbon et al., 2009; Mi, Muruganujan, Ebert, Huang, & Thomas, 2019; The Gene Ontology Consortium et al., 2023), are particularly effective for providing a high-level summary of the functions and biological processes in a given organism. These methods reduce term redundancy, offering a high-level summary of an area of biology based on selected GO terms; however, they may omit important, specific terms. Despite the existence of systematic frameworks (Davis, Sehgal, & Ragan, 2010) for developing domain-specific subsets of ontology terms, creating useful, non-redundant, yet comprehensive subsets remains a challenge that requires careful expert evaluation.

Another widely used technique applied with significant success is the use of semantic similarity to objectively evaluate the functional similarity of a set of terms (Lord et al., 2003; Pesquita, 2017; Pesquita et al., 2009). Various approaches leverage the use of the graph-structured GO ontology (Pesquita et al., 2009), either measuring the information contained within the *edges* of the graph, the *nodes*, or a hybrid approach. Calculating the *distance* between two terms by counting the number of edges is a measure which is simple to calculate and intuitive to interpret; yet it relies on the assumptions that i) nodes and edges are uniformly distributed in the ontology, and ii) all edges represent the same distance anywhere in the graph—assumptions that the GO structure does not fulfill. In contrast, node-based information measures address these limitations but are sensitive to annotation imbalances, as some terms are inherently better annotated in areas of high scientific interest. *rrvgo* leverages semantic similarity measures implemented within Bioconductor (Huber et al., 2015; G. Yu et al., 2010), which correspond to the most widely used in research.

Various software tools differ in how they evaluate semantic similarity metrics to identify redundant terms and select the most representative term. *REVIGO* (Supek et al., 2011) sequentially traverses a list of terms and scores, dynamically setting thresholds to determine redundancy based on the distribution of other terms, ultimately selecting the most significant term according to its associated score. In contrast, *clusterProfiler* (Guangchuang Yu et al., 2012) and *GOSemSim* (G. Yu et al., 2010) consider terms redun-

dant if their semantic similarity exceeds a specified *cutoff*; these tools then either select the most informative term based on precomputed information content or allow users to define a selection function based on custom criteria. In contrast, *rrvgo* groups similar terms in a context-sensitive manner by applying hierarchical clustering with complete linkage to the semantic similarity matrix, then partitions clusters according to a researcher-defined redundancy threshold, selecting each cluster's representative term based on a provided significance score. This method has the advantage that it is especially simple to visualize and interpret using heatmaps augmented with clustered rows and columns. Neither *REVIGO* nor *rrvgo* prioritize higher-level or lower-level GO terms as cluster representatives –instead, the user-supplied score, such as a *p-value* or odds ratio, are used to guide the selection.

Extending the effective visualizations provided by *REVIGO*, *rrvgo* offers static and interactive plots summarizing the semantic similarity between terms as a heatmap, treemap, tag clouds or a scatterplot using the first 2-dimensions calculated by multidimensional scaling. These visualizations have been studied in detail and proven to be highly effective to identify terms in close proximity which may be redundant (Reijnders & Waterhouse, 2021; Supek & Škunca, 2017). Other useful visualizations are also widely used in the field, such as Upset plots comparing the overlap of terms between clusters (Brionne et al., 2019) or graph-based (Merico, Gfeller, & Bader, 2009) in *Cytoscape* (Shannon et al., 2003) plugins *EnrichmentMap* (Merico, Isserlin, & Bader, 2011) and *BINGO* (Maere, Heymans, & Kuiper, 2005).

rrvgo provides access to GO annotation, methods to identify similarity between terms and visualizations within a single software package, though each of these components offers potential for future extension. In terms on annotation, support could be expanded to include other widely used collections of gene sets for overrepresentation analysis, such as the Molecular Signatures Database (MSigDB) hallmark gene set collection (Liberzon et al., 2015), the Kyoto Encyclopedia of Genes and Genomes (KEGG) (Kanehisa & Goto, 2000), and Reactome (Milacic et al., 2024). Methodologically, enhancements should aim to integrate the three orthogonal GO domains—Biological Process (BP), Molecular Function (MF), and Cellular Component (CC)—into a unified framework for similarity calculation, clustering, and visualization, as these are currently analyzed separately. Finally, in terms of visualization, incorporating additional formats, such as *upset* or *sunburst* plots, would offer specific interpretative benefits and further broaden the range of visualization options within *rrvgo*.

Chapter 3: BreakTag

The article *Linking CRISPR–Cas9 double-strand break profiles to gene editing precision with BreakTag* presents BreakTag, a novel technique that enables scalable measurement of the frequency, location, and structure of Cas9-induced DNA breaks, addressing a significant gap in knowledge as CRISPR-based therapeutics advance. In the context of

other high-throughput techniques, BreakTag joins a suite of methods designed to study CRISPR double-strand breaks. These include GUIDE-seq (Tsai et al., 2015), CIRCLE-seq (Tsai et al., 2017), DIG-seq (Kim & Kim, 2018), CHANGE-seq (Lazzarotto et al., 2020),BLESS/BLISS [(Crosetto et al., 2013; W. X. Yan et al., 2017), and TTIS-seq (Schmid-Burgk et al., 2020), among others. While these techniques have expanded our understanding of CRISPR off-target effects and genome-wide activity, BreakTag is unique at resolving break profiles and linking them to editing precision.

In cellulo tools such as GUIDE-seq and TTISS-seq rely on the incorporation of a double-stranded oligodeoxynucleotide at the cut site, delivered simultaneously with the CRISPR components. In contrast, BreakTag will capture all DSBs occurring naturally within the cell, including those arising during transcription and replication, or during DNA extraction. This distinction offers a significant advantage for in cellulo tools, which are specifically designed to capture only CRISPR-induced DSBs. Consequently, BreakTag generally achieves a lower signal-to-noise ratio, with potentially more false positives and diminished power to detect low-frequency off-target sites (Atkins et al., 2021).

The primary novelty introduced by BreakTag is its capability to systematically interrogate DSB end structures across a broad set of Cas9 cleavage sites, marking a first in the field. Previous studies provided evidence of the presence of blunt and staggered DSB end structures resulting from the flexible cleavage activity of the RuvC domain (Allen et al., 2019; W. Chen et al., 2019; Leenay et al., 2019; Lemos et al., 2018; Santiago Gisler et al., 2019; Shen et al., 2018; Shi et al., 2019; Shou et al., 2018; Taheri-Ghahfarokhi et al., 2018), while others identified non-random repair outcomes influenced by target site sequences (Chakrabarti et al., 2019; Molla & Yang, 2020; Overbeek et al., 2016; Shen et al., 2018; Taheri-Ghahfarokhi et al., 2018). This study bridges the flexible cleavage activity of the RuvC domain to non-random repair outcomes through the structural analysis of DSB ends. Our findings reveal specific sequence determinants within the protospacer that influence DSB end structures: for SpCas9, a Guanine at position 17 is strongly associated with the formation of blunt ends, while a Guanine at position 18 tends to favor the formation of 3' overhangs due to misalignment between the RuvC and HNH nuclease domains. Other nucleotides in these positions exhibit less influence over break structure. Additionally, a marked preference for single-nucleotide insertions in staggered DSB ends was observed, the most frequent CRISPR/Cas9 repair outcomes (Allen et al., 2019; Chakrabarti et al., 2019; Leenay et al., 2019; Overbeek et al., 2016; Shen et al., 2018; Taheri-Ghahfarokhi et al., 2018). This insight opens new possibilities for precision target design to control repair outcomes predictably, with potential applications in personalized medicine and the correction of monogenic disorders caused by single-nucleotide deletions.

To analyze BreakTag data we developed the *BreakTag* pipeline and the *breakinspectoR* R package. The pipeline is implemented in the `Bpipe` platform (Sadedin, Pope, & Oshlack, 2012) for running data analytic workflows. Several alternative platforms forunning scientific workflows exist, including Ruffus (Goodstadt, 2010), Snakemake (Köster & Rahmann, 2012), Nextflow (Di Tommaso et al., 2017), Common Workflow Language

(CWL) (Crusoe et al., 2022), Galaxy (The Galaxy Community et al., 2024), KNIME (Berthold et al., 2009) and others. While these platforms support essential features like flow control, branching, checkpoints and restart, and interact with typical HPC resource managers such as SLURM (Yoo, Jette, & Grondona, 2003), they all have unique features which make them suitable in specific situations. For example, Galaxy and KNIME offer user-friendly GUI interfaces, ideal for non-technical users writing and running pipelines in complex environments; CWL is widely used in projects like the 1000 Genomes Project Consortium due to its specificity for pipeline execution, while Ruffus and Snakemake provide Python extensions to run pipelines by building a dependency tree very similar to a Makefile; Nextflow has emerged as a very popular platform in the field of bioinformatics offering advanced features for workload containerization and cloud support. The choice of Bpipe was motivated because it is a simple but mature platform which has been extensively used in our research group and is well understood. The platform has several advantages which make it suitable for running serious scientific workflows in an HPC environment or in the cloud: it is programmed with the Groovy programming language —giving it full access to the Java Virtual Machine (JVM)—, is specifically designed to run in an HPC environment —though it has support for major cloud providers—, interacts well with SLURM, generates intermediate scripts which are easy to debug —something that GUI-based frameworks make it harder by hiding all complexity of intermediate scripts—, and offer all desired features of a workflow manager in a simpler framework which offers higher amounts of control and customization. The BreakTag pipeline consists of eight key steps: (1) raw data preparation, including the filtering of non-barcoded reads through a custom Perl tool (chosen for its efficiency in text stream processing and Regular Expression handling); (2) quality control of NGS raw data using FASTQC ("Babraham Bioinformatics - FastQC A Quality Control tool for High Throughput Sequence Data," n.d.); (3) genome alignment using BWA-MEM (H. Li, 2013); (4) removal of duplicate reads based on mapping coordinates and UMIs through a custom Python script; (5) DSB counting from read ends; (6) library-specific statistics collection; (7) recording of tools and versions used in analysis; and (8) a MultiQC (Ewels, Magnusson, Lundin, & Käller, 2016) report summarizing the experiment's key quality and analytical parameters. This pipeline provides an efficient, comprehensive approach to BreakTag data analysis, with robust quality control and reporting functions.

The *breakinspectoR* R package is specifically designed for the detection of CRISPR off-targets and analysis of DSB end structures. The tool operates in two primary steps. In the initial step, users input an sgRNA sequence, a PAM sequence, the expected cut site position of the chosen CRISPR nuclease (e.g., typically located -3bp upstream of the PAM for Cas9), and a reference genome. Based on this input, the software identifies loci where CRISPR nuclease binding is associated with DSB levels significantly enriched above background levels. For background estimation, *breakinspectoR* supports the use of a control library treated with an inactive nuclease, which should only capture endogenous breaks occurring naturally or as artifacts of DNA extraction. The package provides q-values

adjusted for multiple testing to assess the significance of DSB enrichment and calculates the empirical False Discovery Rate (eFDR) by shuffling signals within the target library to establish a null distribution and estimate false positives. In the second analysis phase, *breakinspectoR* conducts a detailed evaluation of DSB end structures at all loci identified in the first step. It calculates the number of DSB ends detected in the neighboring positions of these loci and provides multiple-testing corrected q-values to assess the probability of non-blunt cuts, indicating possible misalignment of the Cas9 HNH and RuvC domains. The package also includes visualization tools to explore results across the genome, providing insights into the distribution and magnitude of DSBs, fidelity to the sgRNA sequence, off-target frequency, sequence composition, and blunt versus staggered DSB ratios. While other tools, such as BLENDER used in DISCOVER-seq (Wienert et al., 2019), also report DSBs from NGS libraries, they lack specific functionality for analyzing and visualizing DSB end structures, making them impractical for BreakTag data analysis. In contrast, *breakinspectoR* is optimized for BreakTag experimental setups, which typically include a non-target control library. Additionally, the software is designed to leverage HPC environments for highly parallel processing and large-memory, making it efficient for rapid processing of BreakTag libraries.

In a further step, we characterized the sequence determinants that have an influence on the DSB end structure. Several machine learning (ML) and deep learning (DL) techniques exist which can obtain the importance of the predictor variables (Grinsztajn, Oyallon, & Varoquaux, 2022; Ye, Liu, Cai, Zhou, & Zhan, 2024), especially powerful and relatively simple are those based on a random ensemble of trees (Breiman, 2001; T. Chen & Guestrin, 2016) which will naturally obtain it. Using a combination of simple regression coefficients and the importance of the predictor variables obtained with XGBoost, our observations indicate that staggered ends constitute approximately 35% of both on-target and off-target DSBs generated by SpCas9, with a distinct sgRNA-specific cleavage profile suggesting that sequence context significantly influences RuvC domain positioning. The results demonstrate a role of guanine residues within the RuvC cleavage site: when guanine occupied position 17, the RuvC domain exhibited a higher likelihood of cleavage between positions 17 and 18, resulting in a blunt DSB, while a guanine at position 18 shifted the RuvC cleavage site upstream of the HNH cut and generated staggered DSBs.

The development of BreakTag required novel methodologies and software for the systematic identification of DSBs genome-wide from NGS data, incorporating ML techniques to derive additional insights from these data. This study demonstrates BreakTag's utility in identifying on-target and off-target sites, as well as characterizing the DSB end structures of SpCas9 and other Cas9 variants across three comprehensive CRISPR sgRNA libraries (named HiPlex 1, 2, and 3 in the article). We further established the association between staggered DSBs and single-nucleotide insertions, emphasizing the relevance of genetic context for applications of CRISPR in personalized medicine, with potential therapeutic applications for monogenic diseases caused by single-nucleotide deletions that result in non-functional proteins due to a frameshift. BreakTag's capabilities extend to

characterizing new Cas9 variants or other Cas family endonucleases, advancing the potential for controlled and predictable genome edits in clinical and personalized medicine applications. Future research should focus on evaluating Cas9 variants with respect to their activity, fidelity, and DSB end structures, as well as exploring the potential of other CRISPR nucleases such as Cas12 and Cas13 for generating precise, predictable repair outcomes.

General conclusion

In conclusion, this work introduces three novel methodologies to facilitate quality control, interpretation, and analysis of NGS data, applicable across various high-throughput molecular biology experiments. These methods integrate statistical approaches such as clustering, model fitting, and the generation of null distributions for p-value estimation, alongside the use of machine learning techniques and advanced computational practices, including parallel programming and containerization. Overall, this thesis exemplifies a synthesis of expertise from molecular biology, mathematics, and computer science, underscoring an interdisciplinary approach to advancing genomic data analysis.

Appendix A

DupRadar vignette

Using the dupRadar package

Sergi Sayols, Holger Klein

2024-12-02

Contents

1 Introduction to dupRadar

RNA-Seq experiments are a common strategy nowadays to quantify the gene expression levels in cells. Due to its higher sensitivity compared to arrays and the ability to discover novel features, makes them the choice for most modern experiments.

In RNA-Seq - as in all other NGS applications - quality control is essential. Current NGS workflows usually involve PCR steps at some point, which involves the danger of PCR artefacts and over-amplification of reads. For common DNA-based assays PCR duplicates are often simply removed before further analysis and their overall fraction or the read multiplicity taken as quality metrics. In RNA-Seq however, apart from the technical PCR duplicates, there is a strong source for biological read duplicates: for a given gene length and sequencing depth there exists an expression level beyond which it is not possible to add place more reads inside a gene locus without placing a second read exactly on the position of an already existing read. For this reason the overall duplication rate is not a useful measure for RNA-Seq.

As in the NGS/RNA-Seq QC ecosystem there were no suitable tools to address this question, we set out to develop a new tool. The *dupRadar* package gives an insight into the duplication problem by graphically relating the gene expression level and the duplication rate present on it. Thus, failed experiments can be easily identified at a glance.

Note: by now RNA-Seq protocols have matured so that for bulk RNA protocols data rarely suffer from technical duplicates. With newer low-input or single cell RNA-Seq protocols technical duplicates possible problems are worth to be checked for by default, especially if protocols are pushed to or beyond their boundaries. Paired-end libraries make the distinction between duplicates due to highly expressed genes and PCR duplicates easier, but the problem itself is not completely solved, especially at higher sequencing depths.

2 Getting started using dupRadar

2.1 Preparing your data

Previous to running the duplication rate analysis, the BAM file with your mapped reads has to be duplicate marked with a like Picard, or the faster BamUtil dedup BioBamBam. The *dupRadar* package only works with duplicate marked BAM files.

If you do not have/want a duplication marking step in your default pipeline, the *dupRadar* package includes, for your convenience, wrappers to properly call some of these tools from your R session. Note that you still have to supply the path of the dupmarker installation though:

```
library(dupRadar)
```

Now, simply call the wrapper:

Using the dupRadar package

```
# call the duplicate marker and analyze the reads
bamDuprm <- markDuplicates(dupremover="bamutil",
                           bam="test.bam",
                           path="/opt/bamUtil-master/bin",
                           rminput=FALSE)
```

Simply specify which tool to use, the path where this tool is installed, and the input bam file to be analyzed. After marking duplicates, it's safe to remove to original BAM file in order to save space.

The *dupRadar* package currently comes with support for:

- Picard MarkDuplicates
- BamUtil

After the BAM file is marked for duplicates, dupRadar is ready to analyze how the duplication rate is related with the estimated gene expression levels.

2.2 A GTF file

Unless there is any specific reason, dupRadar can use the same GTF file that will be later used to count the reads falling on features.

A valid GTF file can be obtained from UCSC, Ensembl, the iGenomes or other projects.

Note that the resulting duplication rate plots depend on the GTF annotation used. GTF files from the gencode projects result in a less clear picture of duplication rates, as there are many more features and feature types annotated, which overlap heavily as well. In some cases creating the plots only using subsets of gencode annotation files (e.g. just protein coding genes) serve the QC purpose of this tool better.

2.3 AnnotationHub as a source of GTF files

The Bioconductor *AnnotationHub* package provides an alternative approach to obtain annotation in GTF format from entities such as Ensembl, UCSC, ENCODE, and 1000 Genomes projects.

This is partly outlined in the AnnotationHub 'HOWTO' vignette section "Ensembl GTF and FASTA files for TxDb gene models and sequence queries"; for the Takifugu example, the downloaded GTF file is available from the cache.

```
if(suppressWarnings(require(AnnotationHub))) {
  ah = AnnotationHub()
  query(ah, c("ensembl", "80", "Takifugu", "gtf"))    # discovery
  cache(ah["AH47101"])                                # retrieve
}
```

3 dupRate demo data

In the package we include two precomputed duplication matrices for two RNASeq experiments used as examples of a good and a failed (in terms of high redundancy of reads) experiments. The experiments come from the ENCODE project, as a source of a wide variety of protocols, library types and sequencing facilities.

Load the example dataset with:

```
attach(dupRadar_examples)
```

4 The duplication rate analysis

The analysis requires some info about the sequencing run and library preparation protocol:

- The strandess information about the reads: is the sequencing strand specific? if so, are the reads reversely sequenced?
- Are the reads paired, or single?

Due to its phenomenal performance, internally we use the featureCounts() function from the Bioconductor *Rsubread* package, which also supports multiple threads.

```
# The call parameters:
bamDuprm <- "test_duprm.bam" # the duplicate marked bam file
gtf      <- "genes.gtf" # the gene model
stranded <- 2           # '0' (unstranded), '1' (stranded) and '2' (reverse)
paired   <- FALSE       # is the library paired end?
threads  <- 4           # number of threads to be used

# Duplication rate analysis
dm <- analyzeDuprates(bamDuprm,gtf,stranded,paired,threads)
```

The duplication matrix contains read counts in different scenarios and RPK and RPKM values for every gene.

ID	geneLength	allCounts	filteredCounts	dupRate	dupsPerId	RPK
Xkr4	3634	2	2	0.0000	0	0.5503
Rp1	9747	0	0	NaN	0	0.0000
Sox17	3130	0	0	NaN	0	0.0000
Mrpl15	4203	419	258	0.3842	161	99.6906
Lypla1	2433	1069	562	0.4742	507	439.3752
Tcea1	2847	1822	696	0.6180	1126	639.9719

Using the dupRadar package

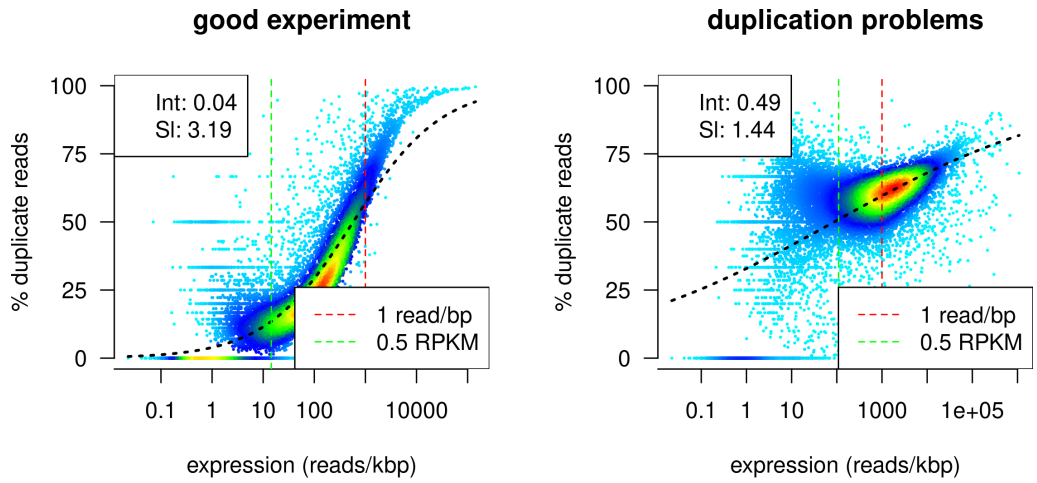
5 Plotting and interpretation

The number of reads per base assigned to a gene in an ideal RNA-Seq data set is expected to be proportional to the abundance of its transcripts in the sample. For lowly expressed genes we expect read duplication to happen rarely by chance, while for highly expressed genes - depending on the total sequencing depth - we expect read duplication to happen often.

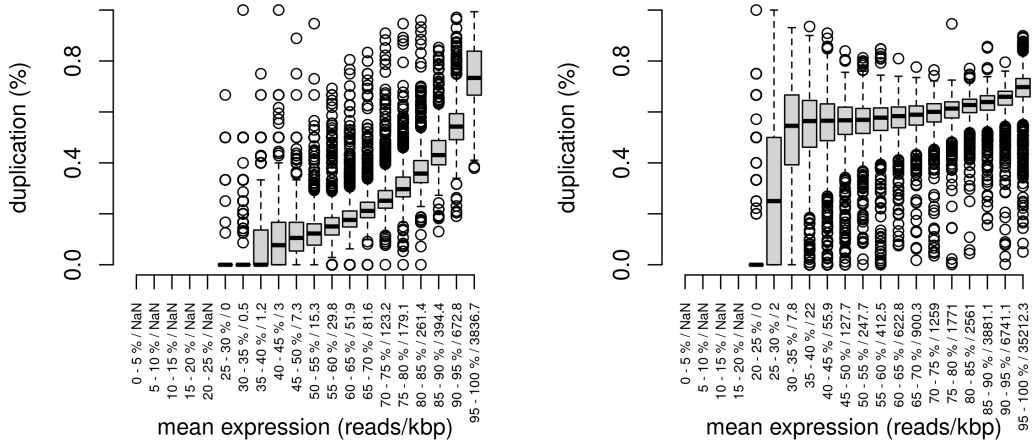
A good way to learn if a dataset is following this trend is by relating the normalized number of counts per gene (RPK, as a quantification of the gene expression) and the fraction represented by duplicated reads.

The *dupRadar* offers several functions to assess this relationship. The most prominent may be the 2d density scatter plot:

```
## make a duprate plot (blue cloud)
par(mfrow=c(1,2))
duprateExpDensPlot(DupMat=dm)     # a good looking plot
title("good experiment")
duprateExpDensPlot(DupMat=dm.bad) # a dataset with duplication problems
title("duplication problems")
```




```
## duprate boxplot
duprateExpBoxplot(DupMat=dm)      # a good looking plot
duprateExpBoxplot(DupMat=dm.bad)  # a dataset with duplication problems
```

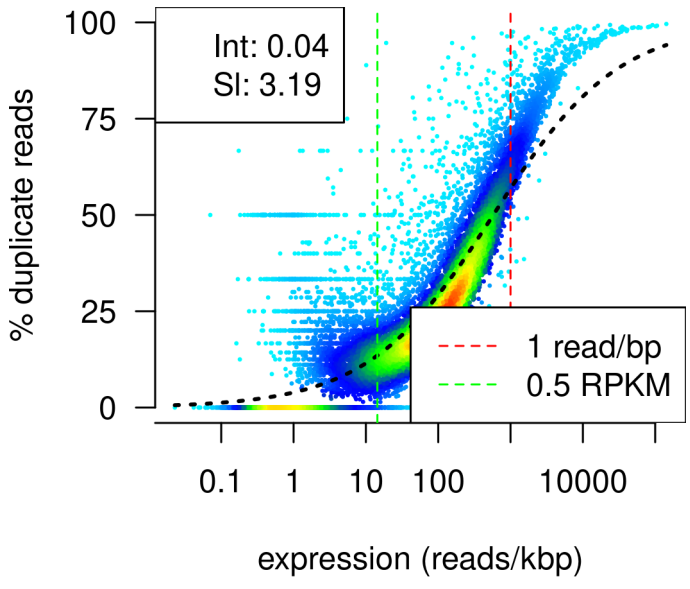


The *duprateExpDensPlot* has helper lines at the threshold of 1 read/bp and at 0.5 RPKM. Moreover by default a generalized linear model is fit to the data and overplotted (see also below).

5.1 Fitting a model into the data

To summarize the relationship between duplication rates and gene expression, we propose fitting a logistic regression curve onto the data. With the coefficients of the fitted model, one can get an idea of the initial duplication rate at low read counts (Intercept) and the progression of the duplication rate along with the progression of the read counts (Slope).

```
duprateExpDensPlot(DupMat=dm)
```




```
# or, just to get the fitted model without plot
fit <- duprateExpFit(DupMat=dm)
cat("duprate at low read counts: ",fit$intercept,"\n",
```

Using the dupRadar package

```
    "progression of the duplication rate: ",fit$slope,fill=TRUE)
## duprate at low read counts:  0.04075061
##  progression of the duplication rate:
## 3.186793
```

Our main use case for that function is the condensation of the plots into quality metrics that can be used for automatic flagging of possibly problematic samples in large experiments or aggregation with other quality metrics in large tables to analyse interdependencies.

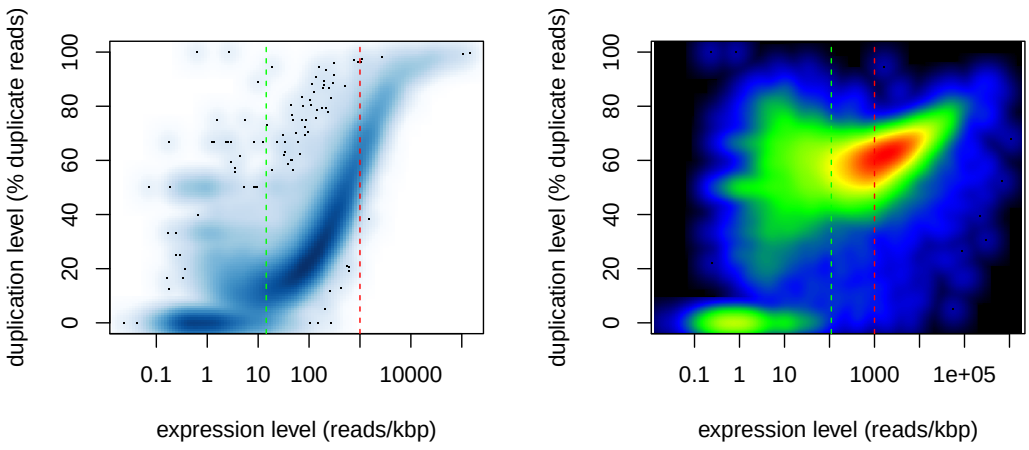
The *duprateExpBoxplot* plot shows the range of the duplication rates at 5% bins (default) along the distribution of RPK gene counts. The x-axis displays the quantile of the RPK distribution, and the average RPK of the genes contained in this quantile.

Individual genes can be identified in the plot:

```
## INTERACTIVE: identify single points on screen (name="ID" column of dm)
duprateExpPlot(DupMat=dm)    # a good looking plot
duprateExpIdentify(DupMat=dm)
```

One can also call the function *duprateExpPlot* to get smooth color density representation of the same data:

```
par(mfrow=c(1,2))
cols <- colorRampPalette(c("black","blue","green","yellow","red"))
duprateExpPlot(DupMat=dm,addLegend=FALSE)
duprateExpPlot(DupMat=dm.bad,addLegend=FALSE,nrpoints=10,nbin=500,
               colramp=cols)
```



Any further parm sent to the *duprateExpPlot* is also sent to the *smoothScatter* function.

5.2 Comparing the fitted parameters to other datasets

The parameters of the fitted model may mean very little (or just nothing) for many, unless there's other data to compare with. We provide the pre-computed duplication matrices for all the RNA-Seq experiments publicly available from the ENCODE project, for human and mouse.

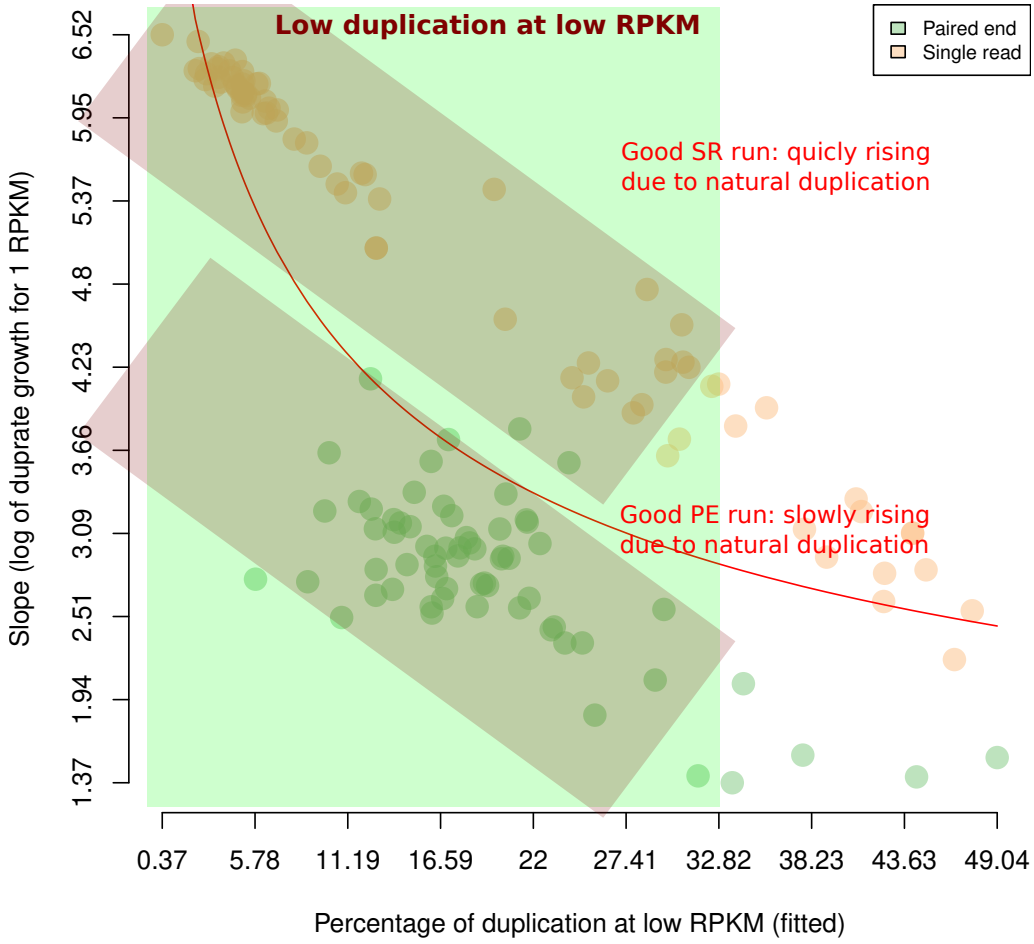


Figure 1: dupRadar analysis of duplicated reads for several ENCODE experiments.

With the experience from the ENCODE datasets, we expect from single read experiments little duplication at low RPKM (low **intercept**) rapidly rising because of natural duplication (high **slope**). In contrast, paired-end experiments have a more mild rising of the natural duplication (low **slope**) due to having higher diversity of reads pairs since pairs with same start may still have different end.

The common denominator for both designs is the importance of having a low intercept, suggesting that duplication rate at lowly expressed genes may serve as a quality measure.

All the pre-computed duplication matrices are available in the dupRadar Github site.

Using the dupRadar package

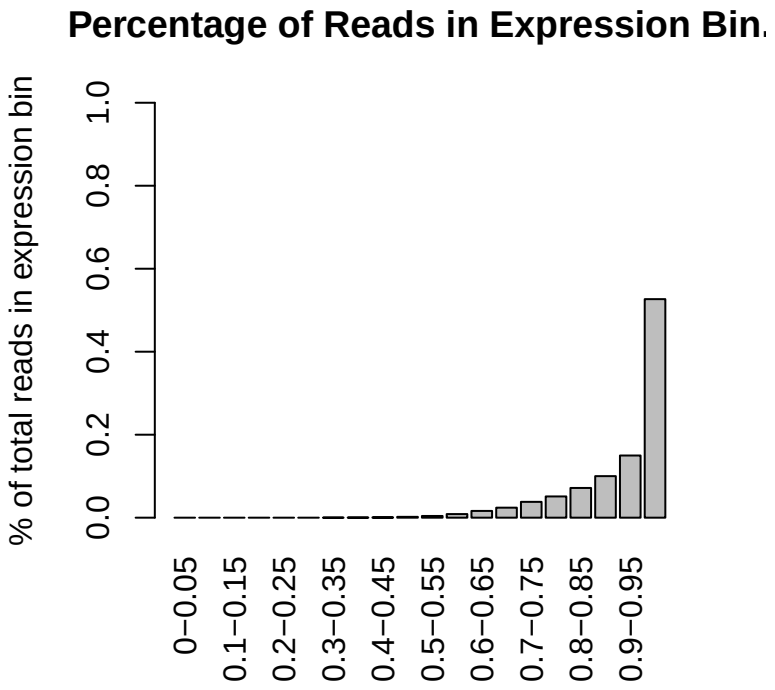
5.3 Other plots

CAVEAT: Sometimes in discussions duplicate reads (i.e. two physically different reads are mapped to the exact same position) and multi-mapping reads (i.e. a single read is mapped to two or more locations in the genome) are mixed up. *dupRadar*'s main focus are PCR duplicates in RNA-Seq. However internally we keep track of unique mappers and multimappers, and we use both in some of the examples, to illustrate use cases of our package beyond the main aim. Multi-mapping reads are completely independent of PCR duplicates.

Apart from the plots relating RPK and duplication rate, the *dupRadar* package provides some other useful plots to extract information about the gene expression levels.

An interesting quality metric are the fraction of reads taken up by groups of genes binned by 5% expression levels.

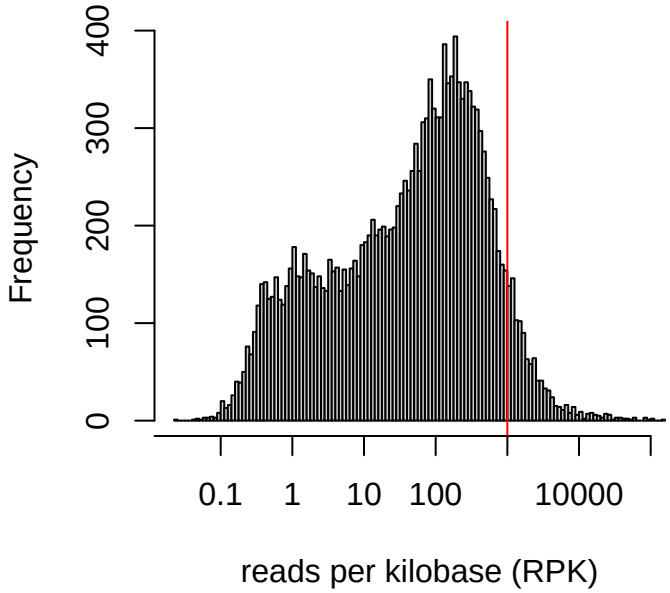
```
readcountExpBoxplot(DupMat=dm)
```



In the example we see that the 5% of highest expressed genes in our sample data set take up around 60% of all reads.

The distribution of RPK values per genes can be plotted with:

```
expressionHist(DupMat=dm)
```



This would help in identifying skewed distributions with unusual amount of lowly expressed genes, or to detect no consensus between replicates.

6 Other information deduced from the data

The duplication rate matrix calculated by the function *analyzeDuprates()* contains some useful information about the sequencing experiment, that can be used to assess the quality of the data.

6.1 Fraction of multimappers per gene

Analogous to per gene duplication rate, the fraction of mutimappers can be easily calculated fom the duplication matrix.

Taking the counts from the column *allCountsMulti*, and substracting form it the counts from the column *allCounts*, one can get the total number of multihits. Thus, the fraction of multihits per gene can be calculating then dividing by *allCountsMulti*.

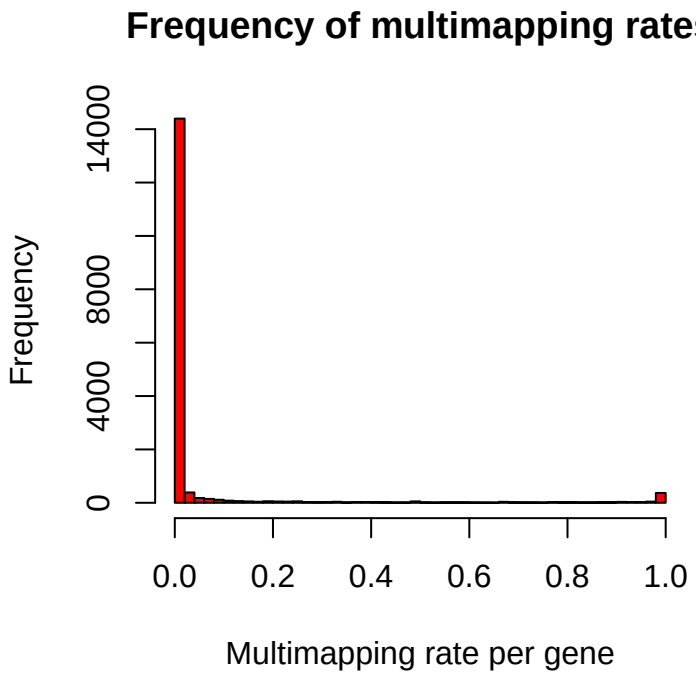
```
# calculate the fraction of multimappers per gene
dm$mhRate <- (dm$allCountsMulti - dm$allCounts) / dm$allCountsMulti
# how many genes are exclusively covered by multimappers
sum(dm$mhRate == 1, na.rm=TRUE)
## [1] 295

# and how many have an RPKM (including multimappers) > 5
sum(dm$mhRate==1 & dm$RPKMMulti > 5, na.rm=TRUE)
## [1] 8

# and which are they?
```

Using the dupRadar package

```
dm[dm$mhRate==1 & dm$RPKMMulti > 5, "ID"]
## [1] GPR89C       LOC728643    LOC606724    TBC1D3C      TBC1D3H
## [6] MIR650       LOC100133050 HIST1H4J
## 23228 Levels: 1/2-SBSRNA4 A1BG A1BG-AS1 A1CF A2LD1 A2M A2ML1 A2MP1 ... ZZZ3
```

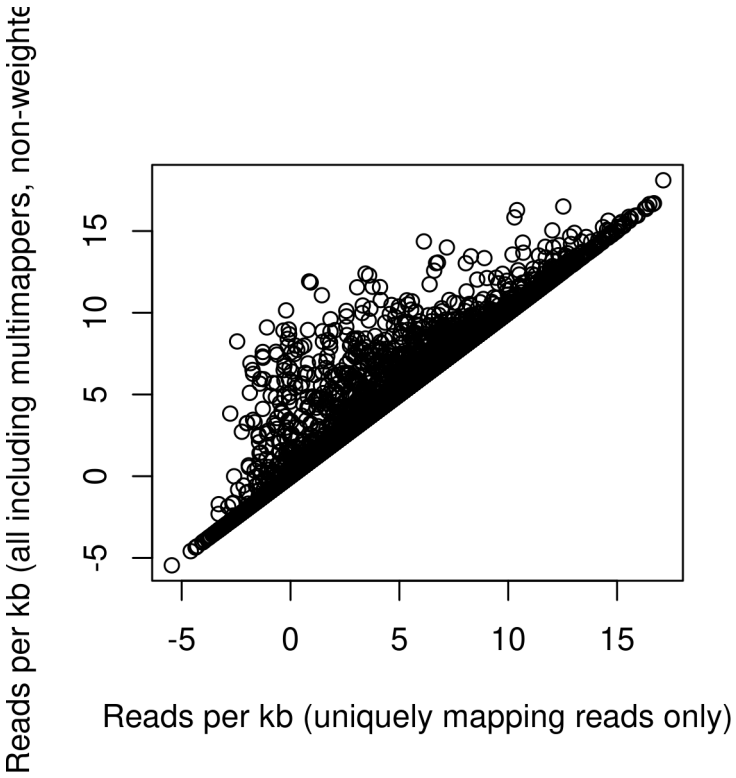
We can also generate an overall picture about less extreme cases:

```
hist(dm$mhRate,
     breaks=50,
     col="red",
     main="Frequency of multimapping rates",
     xlab="Multimapping rate per gene",
     ylab="Frequency")
```



Also the direct comparison of reads per kilobase between uniquely and multimappers is possible.

```
# comparison of multi-mapping RPK and uniquely-mapping RPK
plot(log2(dm$RPK),
     log2(dm$RPKMulti),
     xlab="Reads per kb (uniquely mapping reads only)",
     ylab="Reads per kb (all including multimappers, non-weighted)"
)
```



Use with identify() to annotate interesting cases interactively.

```
identify(log2(dm$RPK),
         log2(dm$RPKMulti),
   labels=dm$ID)
```

6.2 Connection between possible PCR artefacts and GC content

In some cases we wondered about influence of GC content on PCR artefacts. An easy way to check using our *dupRadar* package in conjunction with *biomaRt* is demonstrated in the following. For simplicity we use our demo data here in which we *do not* see a big influence.

```
library(dupRadar)
library(biomaRt)

## for detailed explanations on biomaRt, please see the respective
## vignette

## set up biomart connection for mouse (needs internet connection)
ensm <- useMart("ensembl")
ensm <- useDataset("mmusculus_gene_ensembl", mart=ensm)

## get a table which has the gene GC content for the IDs that have been
## used to generate the table (depends on the GTF annotation that you
```

Using the dupRadar package

```
## use)
tr <- getBM(attributes=c("mgi_symbol", "percentage_gc_content"),
            values=TRUE, mart=ensm)

## create a GC vector with IDs as element names
mgi.gc <- tr$percentage_gc_content
names(mgi.gc) <- tr$mgi_symbol

## using dm demo duplication matrix that comes with the package
## add GC content to our demo data and keep only subset for which we can
## retrieve data
keep <- dm$ID %in% tr$mgi_symbol
dm.gc <- dm[keep,]
dm.gc$gc <- mgi.gc[dm.gc$ID]

## check distribution of annotated gene GC content (in %)
boxplot(dm.gc$gc, main="Gene GC content", ylab="% GC")
```

Now we can compare the dependence of duplication rate on expression level independently for below and above median GC genes (and to mention again, in this data set we don't have a big difference).

```
par(mfrow=c(1,2))

## below median GC genes
duprateExpDensPlot(dm.gc[dm.gc$gc<=45,], main="below median GC genes")

## above median GC genes
duprateExpDensPlot(dm.gc[dm.gc$gc>=45,], main="above median GC genes")
```

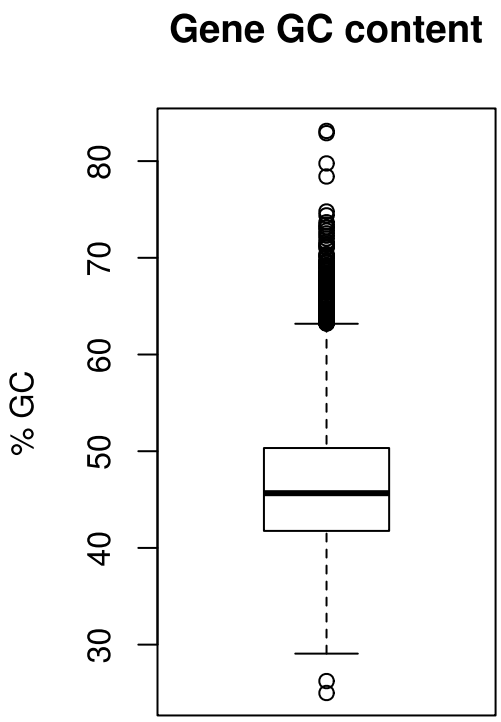


Figure 2: Percentage GC content of genes analyzed.

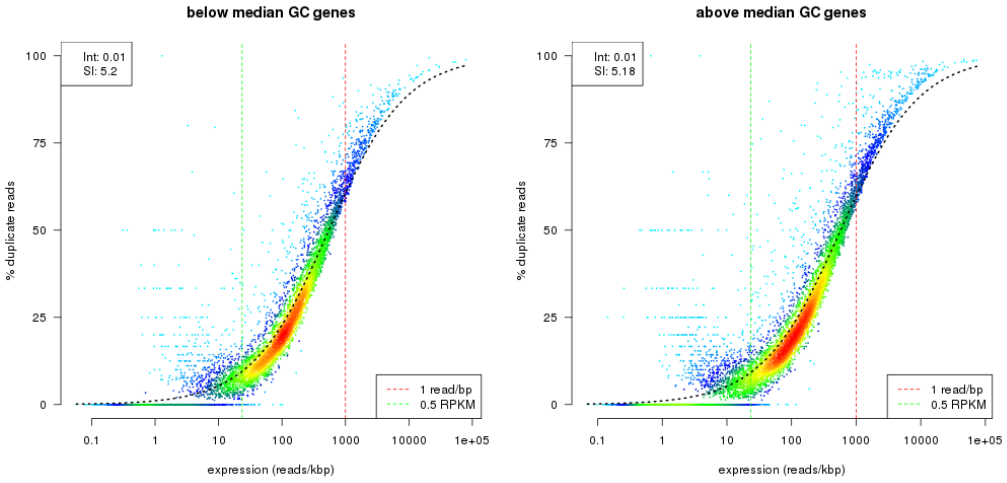


Figure 3: Dependence of duplication rates and GC content.

Using the dupRadar package

7 Conclusion

The *dupRadar* package provides a framework for the analysis of duplicate rates in RNAseq datasets. In addition, it includes a set of convenient wrappers to correctly call some common tools used for marking PCR duplicated reads in the BAM file. It's shipped as Bioconductor package in order to offer a common framework for all the tools involved, and simplify its use.

8 Including dupRadar into pipelines

To include dupRadar as a single step in an RNA-Seq pipeline, integration into a short R-script can be done like in the following:

```
#!/usr/bin/env Rscript

########################################
##
## dupRadar shell script
## call dupRadar R package from the shell for
## easy integration into pipelines
##
## Holger Klein & Sergi Sayols
##
## https://github.com/ssayols/dupRadar
##
## input:
## - _duplicate marked_ bam file
## - gtf file
## - parameters for duplication counting routine:
##   stranded, paired, outdir, threads.
##
########################################

library(dupRadar)

####################
##
## get name patterns from command line
##
args   <- commandArgs(TRUE)

## the bam file to analyse
bam <- args[1]
## usually, same GTF file as used in htseq-count
gtf <- gsub("gtf=","",args[2])
## no|yes|reverse
```

```
stranded <- gsub("stranded=","",args[3])
## is a paired end experiment
paired   <- gsub("paired=","",args[4])
## output directory
outdir   <- gsub("outdir=","",args[5])
## number of threads to be used
threads  <- as.integer(gsub("threads=","",args[6]))

if(length(args) != 6) {
  stop (paste0("Usage: ./dupRadar.sh <file.bam> <genes.gtf> ",
               "<stranded=[no|yes|reverse]> paired=[yes|no] ",
               "outdir=./ threads=1"))
}

if(!file.exists(bam)) {
  stop(paste("File",bam,"does NOT exist"))
}

if(!file.exists(gtf)) {
  stop(paste("File",gtf,"does NOT exist"))
}

if(!file.exists(outdir)) {
  stop(paste("Dir",outdir,"does NOT exist"))
}

if(is.na(stranded) | !(grepl("no|yes|reverse",stranded))) {
  stop("Stranded has to be no|yes|reverse")
}

if(is.na(paired) | !(grepl("no|yes",paired))) {
  stop("Paired has to be no|yes")
}

if(is.na(threads)) {
  stop("Threads has to be an integer number")
}

stranded <- if(stranded == "no") 0 else if(stranded == "yes") 1 else 2

## end command line parsing
##
#########################################

#########################################
```

Using the dupRadar package

```
##
## analyze duprates and create plots
##
cat("Processing file ", bam, " with GTF ", gtf, "\n")

## calculate duplication rate matrix
dm <- analyzeDuprates(bam,
                      gtf,
                      stranded,
                      (paired == "yes"),
                      threads)

## produce plots

## duprate vs. expression smooth scatter
png(file=paste0(outdir,"/",gsub("(.*)\\.[^.]+","\\1",basename(bam)),
                "_dupRadar_drescatter.png"),
    width=1000, height=1000)
duprateExpDensPlot(dm, main=basename(bam))
dev.off()

## expression histogram
png(file=paste0(outdir,"/",gsub("(.*)\\.[^.]+","\\1",basename(bam)),
                "_dupRadar_ehist.png"),
    width=1000, height=1000)
expressionHist(dm)
dev.off()

## duprate vs. expression boxplot
png(file=paste0(outdir,"/",gsub("(.*)\\.[^.]+","\\1",basename(bam)),
                "_dupRadar_drebp.png"),
    width=1000, height=1000)
par(mar=c(10,4,4,2)+.1)
duprateExpBoxplot(dm, main=basename(bam))
dev.off()
```

8.1 Citing dupRadar

Please consider citing dupRadar if used in support of your own research:

```
citation("dupRadar")
## To cite package 'dupRadar' in publications use:
##
##   Sergi Sayols, Denise Scherzinger and Holger Klein (2016): dupRadar: a
##   Bioconductor package for the assessment of PCR artifacts in RNA-Seq
```

```
##   data. BMC Bioinformatics, 17:428, doi:10.1186/s12859-016-1276-2
##
## A BibTeX entry for LaTeX users is
##
##   @Article{,
##     title = {dupRadar: a Bioconductor package for the assessment of PCR artifacts in RNA-Se
##     author = {Sergi Sayols and Denise Scherzinger and Holger Klein},
##     year = {2016},
##     journal = {BMC Bioinformatics},
##     doi = {10.1186/s12859-016-1276-2},
##     url = {http://dx.doi.org/10.1186/s12859-016-1276-2},
##     volume = {17},
##     issue = {1},
##     pages = {428},
##   }
```

8.2 Reporting problems or bugs

If you run into problems using dupRadar, the Bioconductor Support site is a good first place to ask for help. If you think there is a bug or an unreported feature, you can report it using the dupRadar Github site.

9 Session info

The following package and versions were used in the production of this vignette.

```
## R version 4.4.2 (2024-10-31)
## Platform: x86_64-pc-linux-gnu
## Running under: Debian GNU/Linux trixie/sid
##
## Matrix products: default
## BLAS:   /usr/lib/x86_64-linux-gnu/atlas/libblas.so.3.10.3
## LAPACK: /usr/lib/x86_64-linux-gnu/atlas/liblapack.so.3.10.3;  LAPACK version 3.12.0
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## time zone: Europe/Madrid
## tzcode source: system (glibc)
```

Using the dupRadar package

```
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] dupRadar_1.34.0  knitr_1.49       BiocStyle_2.32.1
##
## loaded via a namespace (and not attached):
##  [1] Rsubread_2.18.0     digest_0.6.37       codetools_0.2-20
##  [4] bookdown_0.41       fastmap_1.2.0       Matrix_1.7-1
##  [7] xfun_0.49           lattice_0.22-6      magrittr_2.0.3
## [10] KernSmooth_2.23-24  parallel_4.4.2      htmltools_0.5.8.1
## [13] rmarkdown_2.29      tinytex_0.54        cli_3.6.3
## [16] grid_4.4.2          compiler_4.4.2      rstudioapi_0.17.1
## [19] tools_4.4.2         evaluate_1.0.1      Rcpp_1.0.13-1
## [22] magick_2.8.5        yaml_2.3.10         BiocManager_1.30.25
## [25] rlang_1.1.4
```

Appendix B

rrvgo vignette

Using the rrvgo package

Sergi Sayols

2024-12-02

Contents

1 Introduction to rrvgo

Gene Ontologies (GO) are often used to guide the interpretation of high-throughput omics experiments, with lists of differentially regulated genes being summarized into sets of genes with a common functional representation. Due to the hierachical nature of Gene Ontologies, the resulting lists of enriched sets are usually redundant and difficult to interpret.

`rrvgo` aims at simplifying the redundance of GO sets by grouping similar terms based on their semantic similarity. It also provides some plots to help with interpreting the summarized terms.

This software is heavily influenced by REVIGO. It mimics a good part of its core functionality, and even some of the outputs are similar. Without aims to compete, `rrvgo` tries to offer a programatic interface using available annotation databases and semantic similarity methods implemented in the Bioconductor project.

2 Using rrvgo

2.1 Getting started

Starting with a list of genes of interest (eg. coming from a differential expression analysis), apply any method for the identification of eneriched GO terms (see GOStats or GSEA).

`rrvgo` does not care about genes, but GO terms. The input is a vector of enriched GO terms, along with (recommended, but not mandatory) a vector of scores. If scores are not provided, `rrvgo` takes the GO term (set) size as a score, thus favoring *broader* terms.

2.2 Calculating the similarity matrix and reducing GO terms

First step is to get the similarity matrix between terms. The function `calculateSimMatrix` takes a list of GO terms for which the semantic simlarity is to be calculated, an `OrgDb` object for an organism, the ontology of interest and the method to calculate the similarity scores.

```
library(rrvgo)
go_analysis <- read.delim(system.file("extdata/example.txt",
                                      package="rrvgo"))
simMatrix <- calculateSimMatrix(go_analysis$ID,
                                orgdb="org.Hs.eg.db",
                                ont="BP",
                                method="Rel")
```

The `semdata` parameter (see `?calculateSimMatrix`) is not mandatory as it is calculated on demand. If the function needs to run several times with the same organism, it's advisable to save the `GOSemSim::godata(orgdb, ont=ont)` object, in order to reuse it between calls and speedup the calculation of the similarity matrix.

Using the rrvgo package

From the similarity matrix one can group terms based on similarity. `rrvgo` provides the `reduceSimMatrix` function for that. It takes as arguments i) the similarity matrix, ii) an optional *named* vector of scores associated to each GO term, iii) a similarity threshold used for grouping terms, and iv) an orgdb object.

```
scores <- setNames(-log10(go_analysis$qvalue), go_analysis$ID)
reducedTerms <- reduceSimMatrix(simMatrix,
                                scores,
                                threshold=0.7,
                                orgdb="org.Hs.eg.db")
```

`reduceSimMatrix` groups terms which are at least within a similarity below `threshold`, and selects as the group representative the term with the higher score within the group. In case the vector of scores is not available, `reduceSimMatrix` can either use the *uniqueness* of a term (default), or the GO term *size*. In the case of *size*, `rrvgo` will fetch the GO term size from the `OrgDb` object and use it as the score, thus favoring broader terms. **Please note that scores are interpreted in the direction that higher are better**, therefore if you use p-values as scores, minus log-transform them before.

NOTE: `rrvgo` uses the similarity between pairs of terms to compute a distance matrix, defined as `(1-simMatrix)`. The terms are then hierarchically clustered using complete linkage, and the tree is cut at the desired threshold, picking the term with the highest score as the representative of each group.
Therefore, higher thresholds lead to fewer groups, and the threshold should be read as the minimum similarity between group representatives.

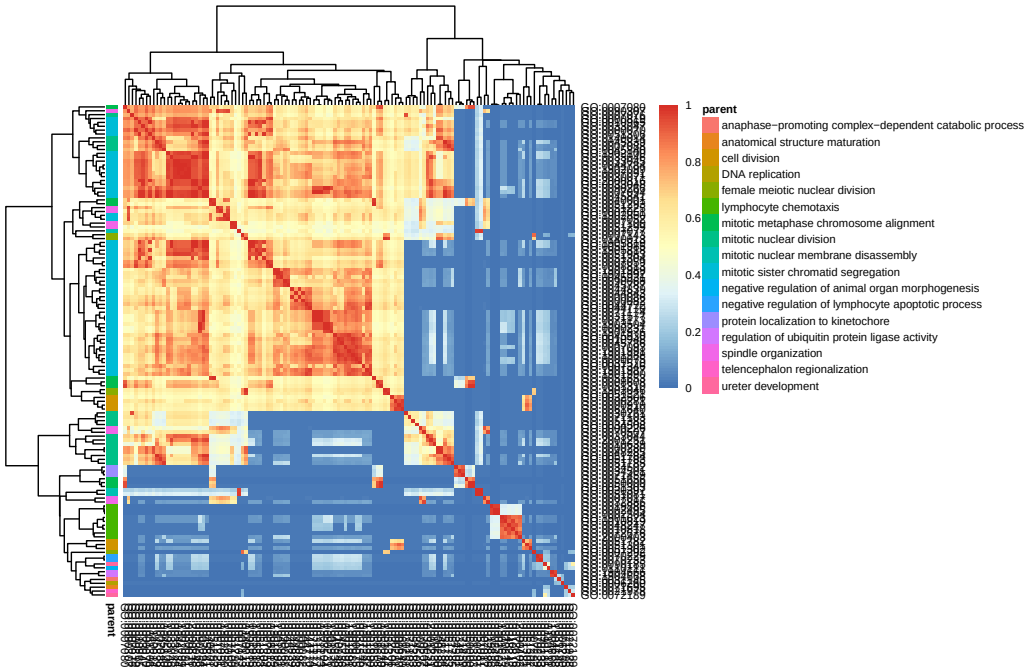
2.3 Plotting and interpretation

`rrvgo` provides several methods for plotting and interpreting the results.

2.3.1 Similarity matrix heatmap

Plot similarity matrix as a heatmap, with clustering of columns of rows turned on by default (thus arranging together similar terms).

```
heatmapPlot(simMatrix,
            reducedTerms,
            annotateParent=TRUE,
            annotationLabel="parentTerm",
            fontsize=6)
```



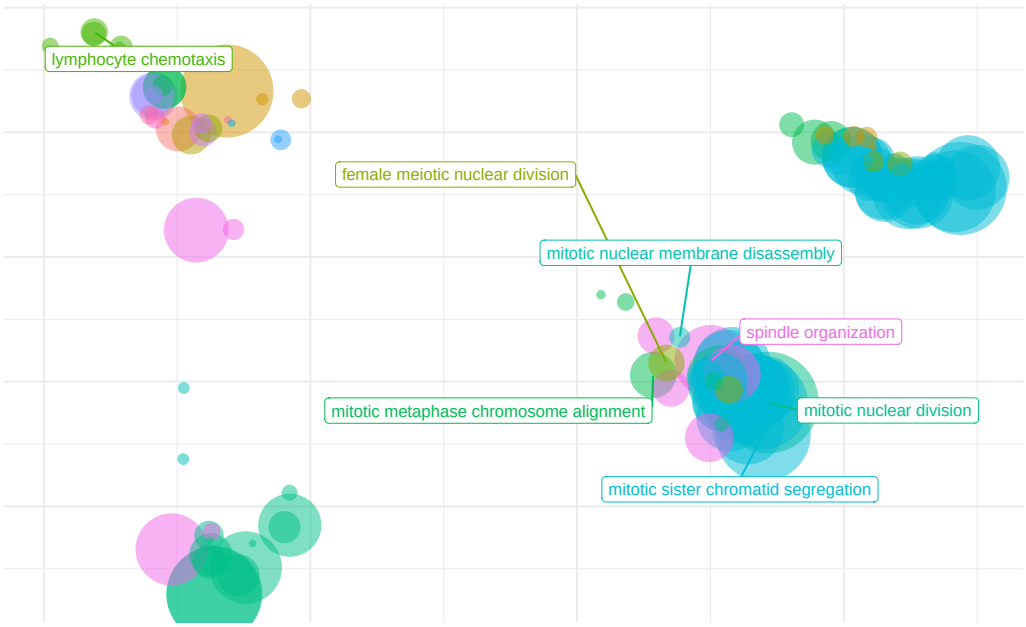
The function internally uses `pheatmap`, and further parameters can be passed to this function.

2.3.2 Scatter plot depicting groups and distance between terms

Plot GO terms as scattered points. Distances between points represent the similarity between terms, and axes are the first 2 components of applying a PCoA to the (di)similarity matrix. Size of the point represents the provided scores or, in its absence, the number of genes the GO term contains.

```
scatterPlot(simMatrix, reducedTerms)
```

Using the rrvgo package



2.3.3 Treemap plot

Treemaps are space-filling visualization of hierarchical structures. The terms are grouped (colored) based on their parent, and the space used by the term is proportional to the score. Treemaps can help with the interpretation of the summarized results and also comparing differents sets of GO terms.

```
treemapPlot(reducedTerms)
```

The function internally uses `treemap`, and further parameters can be passed to this function.

2.3.4 Word cloud

Word clouds are visualizations which reproduce a text putting emphasis to words which appear frequently in a text. They can help to identify processes and functions that happen more commonly in a set of enriched GO terms, as well as comparing between different sets.

```
wordcloudPlot(reducedTerms, min.freq=1, colors="black")
```

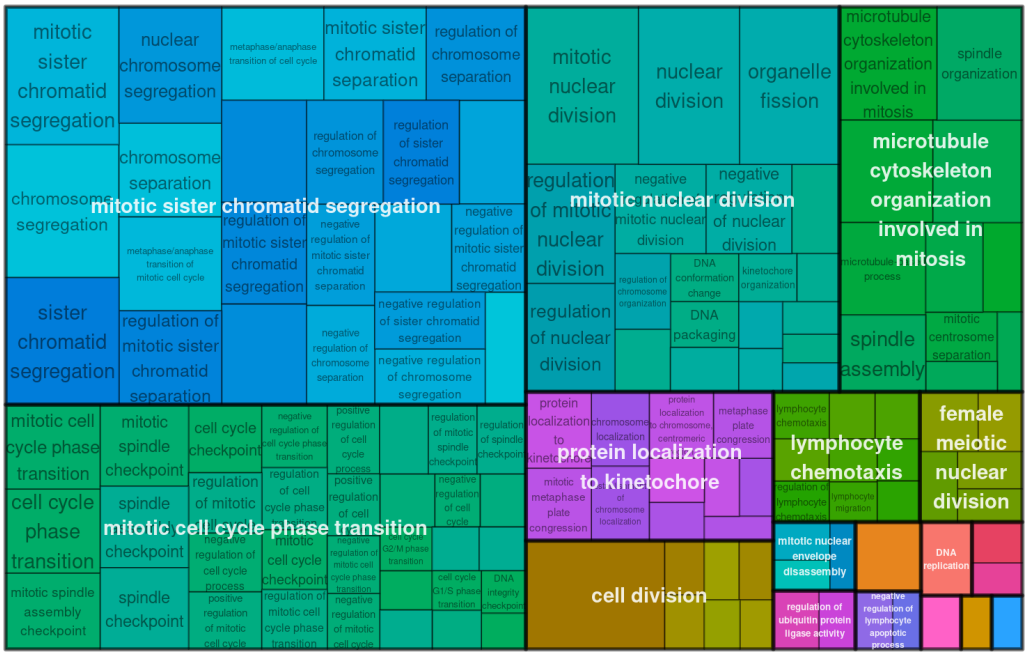
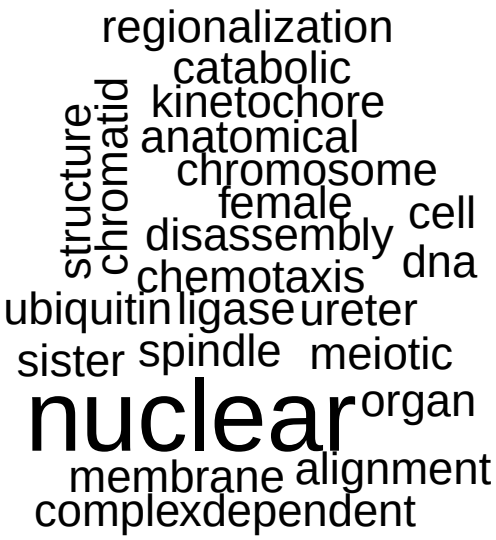


Figure 1: treemap



The function internally uses `wordcloud`, and further parameters can be passed to this function.

2.4 Shiny app

To make the software more accessible to a non-technical audience, `rrvgo` packages a shiny app which can be accessed calling the `shiny_rrvgo()` function from the R console.

```
rrvgo::shiny_rrvgo()
```

The app offers *interactive* access to the plots and tables calculated by `rrvgo`.

Using the rrvgo package

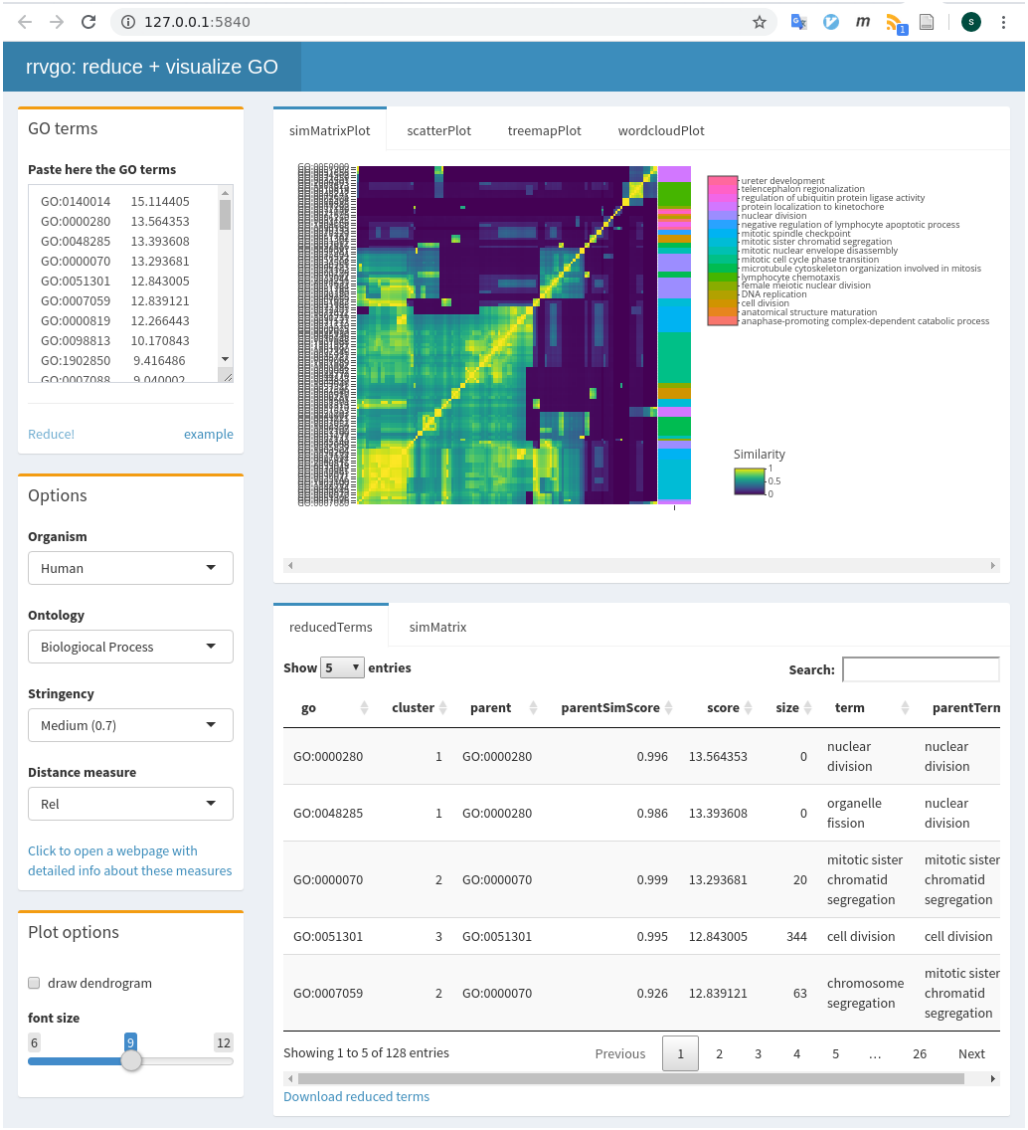


Figure 2: shiny_app

3 Currently supported

3.1 Similarity methods

All similarity measures available are those implemented in the GOSemSim package, namely the Resnik, Lin, Relevance, Jiang and Wang methods. See the Semantic Similarity Measurement Based on GO section from the GOSeSim documentation for more details.

3.2 Organisms

Bioconductor current provides `OrgDb` objects for 20 species provided by the following packages:

Package	Organism
org.Ag.eg.db	Anopheles
org.At.tair.db	Arabidopsis
org.Bt.eg.db	Bovine
org.Ce.eg.db	Worm
org.Cf.eg.db	Canine
org.Dm.eg.db	Fly
org.Dr.eg.db	Zebrafish
org.EcK12.eg.db	E coli strain K12
org.EcSakai.eg.db	E coli strain Sakai
org.Gg.eg.db	Chicken
org.Hs.eg.db	Human
org.Mm.eg.db	Mouse
org.Mmu.eg.db	Rhesus
org.Mxanthus.db	Myxococcus xanthus DK 1622
org.Pf.plasmo.db	Malaria
org.Pt.eg.db	Chimp
org.Rn.eg.db	Rat
org.Sc.sgd.db	Yeast
org.Ss.eg.db	Pig
org.Xl.eg.db	Xenopus

If the organism is not supported in Bioconductor, you can still build your own `OrgDb` object usign the `AnnotationForge` package and rendering the necessary data for semantic similarity using the `GOSemSim` package with:

```
my_new_fancy_orgdb_object <- 'org.Zz.eg.db'
hsGO <- GOSemSim::godata(my_new_fancy_orgdb_object, ont="MF")
```

3.3 Gene Ontologies

One of *Biologiocal Process* (BP), *Molecular Function* (MF) or *Cellular Compartment* (CC).

4 Demo data

Taken as is from the DOSE package, which was derived from the R package breastCancerMAINZ. It contains 200 samples with breast cancer at different grades (I, II and III). The dataset basically contains log2 ratios of the geometric means of grade III vs. grade I samples (34 vs. 29 repectively).

5 Citing rrvgo

Please consider citing rrvgo if used in support of your own research:

Using the rrvgo package

```
citation("rrvgo")
## To cite package 'rrvgo' in publications use:
##
##   Sayols, S (2023). rrvgo: a Bioconductor package for interpreting
##   lists of Gene Ontology terms. microPublication Biology.
##   10.17912/micropub.biology.000811
##
## A BibTeX entry for LaTeX users is
##
##   @Article{,
##     title = {rrvgo: a Bioconductor package to reduce and visualize Gene Ontology terms},
##     author = {Sergi Sayols},
##     year = {2023},
##     journal = {microPublication Biology},
##     doi = {10.17912/micropub.biology.000811},
##     url = {https://www.micropublication.org/journals/biology/micropub-biology-000811},
##   }
```

5.1 Reporting problems or bugs

If you run into problems using rrvgo, the Bioconductor Support site is a good first place to ask for help. If you think there is a bug or an unreported feature, you can report it using the rrvgo github site.

5.2 Session info

The following package and versions were used in the production of this vignette.

```
## R version 4.4.2 (2024-10-31)
## Platform: x86_64-pc-linux-gnu
## Running under: Debian GNU/Linux trixie/sid
##
## Matrix products: default
## BLAS:   /usr/lib/x86_64-linux-gnu/atlas/libblas.so.3.10.3
## LAPACK: /usr/lib/x86_64-linux-gnu/atlas/liblapack.so.3.10.3;  LAPACK version 3.12.0
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## time zone: Europe/Madrid
```

```
## tzcode source: system (glibc)
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] rrvgo_1.16.0     knitr_1.49       BiocStyle_2.32.1
##
## loaded via a namespace (and not attached):
##  [1] tidyselect_1.2.1        gridBase_0.4-7          farver_2.1.2
##  [4] dplyr_1.1.4             blob_1.2.4              R.utils_2.12.3
##  [7] Biostrings_2.72.1       fastmap_1.2.0           treemap_2.4-4
## [10] promises_1.3.0          digest_0.6.37           mime_0.12
## [13] lifecycle_1.0.4         NLP_0.3-2               KEGGREST_1.44.1
## [16] RSQLite_2.3.8           magrittr_2.0.3          compiler_4.4.2
## [19] rlang_1.1.4             tools_4.4.2             wordcloud_2.6
## [22] igraph_2.1.1            utf8_1.2.4              yaml_2.3.10
## [25] data.table_1.16.2       labeling_0.4.3          askpass_1.2.1
## [28] bit_4.5.0               reticulate_1.40.0       xml2_1.3.6
## [31] RColorBrewer_1.1-3      withr_3.0.2             BiocGenerics_0.50.0
## [34] R.oo_1.27.0             grid_4.4.2              stats4_4.4.2
## [37] fansi_1.0.6             GOSemSim_2.30.2         xtable_1.8-4
## [40] tm_0.7-15               colorspace_2.1-1        GO.db_3.19.1
## [43] ggplot2_3.5.1           scales_1.3.0            tinytex_0.54
## [46] cli_3.6.3               rmarkdown_2.29          crayon_1.5.3
## [49] generics_0.1.3          umap_0.2.10.0           rstudioapi_0.17.1
## [52] RSpectra_0.16-2         httr_1.4.7              DBI_1.2.3
## [55] cachem_1.1.0            zlibbioc_1.50.0         parallel_4.4.2
## [58] AnnotationDbi_1.66.0    BiocManager_1.30.25     XVector_0.44.0
## [61] yulab.utils_0.1.8       vctrs_0.6.5             Matrix_1.7-1
## [64] jsonlite_1.8.9          slam_0.1-55             bookdown_0.41
## [67] IRanges_2.38.1          S4Vectors_0.42.1        bit64_4.5.2
## [70] ggrepel_0.9.6           glue_1.8.0              gtable_0.3.6
## [73] later_1.3.2             GenomeInfoDb_1.40.1     UCSC.utils_1.0.0
## [76] munsell_0.5.1           tibble_3.2.1            pillar_1.9.0
## [79] rappdirs_0.3.3          htmltools_0.5.8.1       openssl_2.2.2
## [82] GenomeInfoDbData_1.2.12 R6_2.5.1                httr2_1.0.6
## [85] lattice_0.22-6          evaluate_1.0.1          shiny_1.9.1
## [88] Biobase_2.64.0          R.methodsS3_1.8.2       png_0.1-8
## [91] pheatmap_1.0.12         memoise_2.0.1           httpuv_1.6.15
## [94] Rcpp_1.0.13-1           org.Hs.eg.db_3.19.1     xfun_0.49
## [97] fs_1.6.5                pkgconfig_2.0.3
```

Appendix C

BreakTag

C.1 Extended data

C.1.1 Supplementary note 1: BreakTag DNA double-strand break amplification strategy

BreakTag is a highly scalable four-step protocol that maps free DSB ends in gDNA digested in vitro to RNPs. Ready-to-sequence libraries are achieved in less than 6 hours with minimal hands-on time. The method is performed in multi-well plates with the use of a multichannel pipette, and automation is simple. The procedure starts with a blunting step, in which 5' overhangs are filled-in and 3' overhangs are resected followed by A-tailing where a single adenine is added to the 3' end of the DSB prior to labeling. Processed ends are then ligated with a customized BreakTag linker. The linker contains a PCR handle, the sequencing primer binding site (mosaic end, ME) a unique molecular identifier (UMI) for removal of PCR duplicates, and a sample barcode. The sample barcode, which is embedded in the linker, allows an extra layer of barcoding and increases the throughput, such that samples can be pooled and further processed in the same tube if necessary. After ligation with the BreakTag linker, the gDNA is tagmented with a single-handle Tn5 containing a second PCR handle, which cuts the DNA randomly and inserts an adapter into the 5' end of the fragment 51. Ligation of DSB ends with the BreakTag linker followed by tagmentation with single-handle Tn5 generates two populations of fragments, one termed "homotagged", in which both ends of the fragment contain the same sequence added during tagmentation, and a second "heterotagged" population, in which fragments contain the BreakTag linker at one end and the tagmentation linker at the other. The latter are amenable for exponential amplification as they contain two distinct PCR handles for primers that introduce functional p5 and p7 sequences. Homotagged fragments <1kb are not exponentially amplified and do not cluster during sequencing with Illumina sequencers (Fig. 1a).

C.1.2 Extended Figures

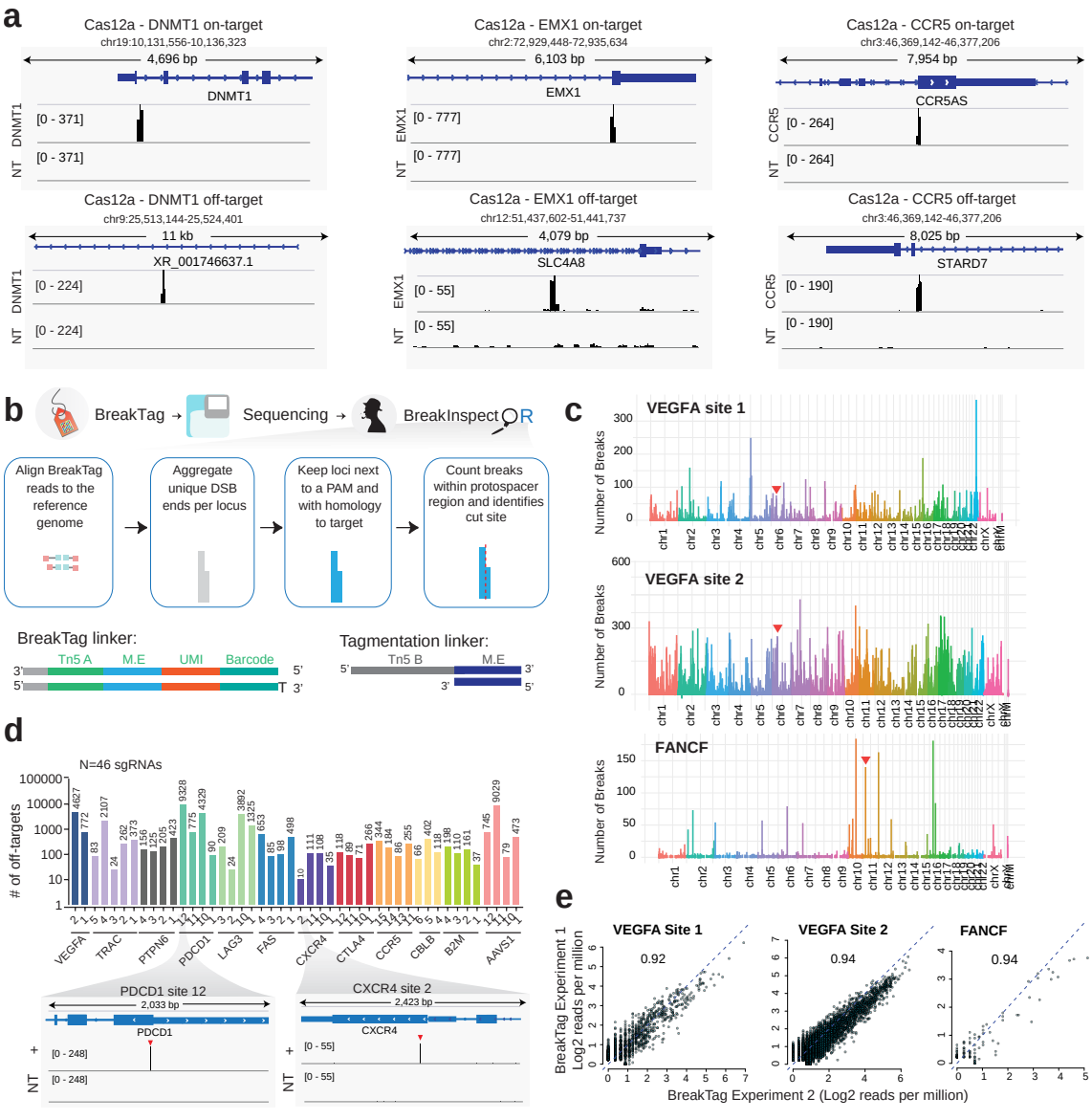


Figure C.1: BreakTag and BreakInspector allow high-throughput, genome-wide assessment of Cas9 and Cas12a on- and off-targets. (a) IGV snapshots of Cas12a on targets and representative off-targets of 3 gRNAs. (b) Schematics of BreakInspectoR analysis workflow. (c) Manhattan plots showing off-targets nominated for 'VEGFA site 1', 'VEGFA site 2' and 'FANCF'. Red arrowheads indicate on-target sequences. BreakTag was performed in gDNA from U2OS cells. (d) Number of off-targets mapped by BreakTag in gDNA of U2OS cells digested with Cas9 and 46 different clinically relevant gRNAs. Representative IGV snapshots of the on-target region of 'PDCD1 site 12' and 'CXCR4 site 2' are shown below. Off-targets were called using a low threshold of at least 3 reads and up to 7 mismatches. (e) Correlation between two independent BreakTag runs for three sgRNAs commonly used in the benchmarking of off-target-nominating tools.

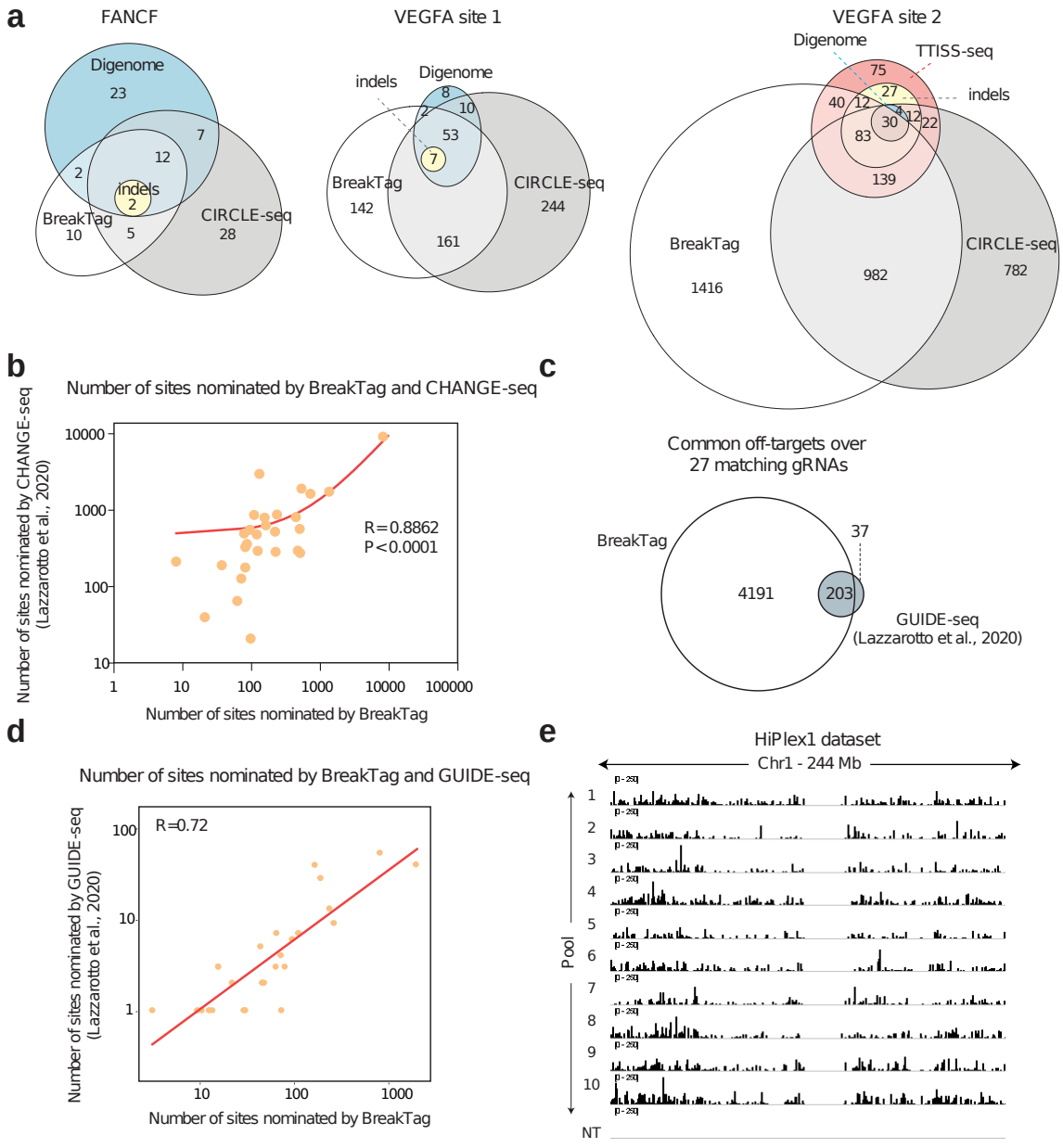


Figure C.2: Benchmarking of BreakTag against other off-target nominating tools. (a) Venn diagrams showing the overlap between sites nominated by BreakTag, DIGENOME-seq, and CIRCLE-seq. Off-targets were selected for validation using targeted deep sequencing. TTISS-seq was used to generate a refined list of in cellulo VEGFA site 2 off-targets due to its high promiscuity. A minimum of 8 reads and a maximum of 6 mismatches was used for BreakTag off-targets in order to match public available data's thresholds. (b) Correlation between number of off-targets nominated by CHANGE-seq and BreakTag over 44 gRNAs arbitrarily selected from the CHANGE-seq dataset. (c) Common off-target sites identified by GUIDE-seq (data produced in Lazzarotto 2020) and BreakTag over matching 27 gRNAs. For GUIDE-seq only targets supported by at least 8 reads, up to 6 mismatches between crRNA:DNA and an NGG PAM were considered; for BreakTag targets supported by at least 8 reads, up to 6 mismatches and a FDR<1% were considered. (d) Correlation between the number of off-targets nominated by BreakTag and GUIDE-seq data. (e) IGV snapshot of chromosome 1 of HepG2 cells digested with Pools 1–10 from the HiPlex1 library.

Each bar represents a cleaved site. NT: nontarget control.

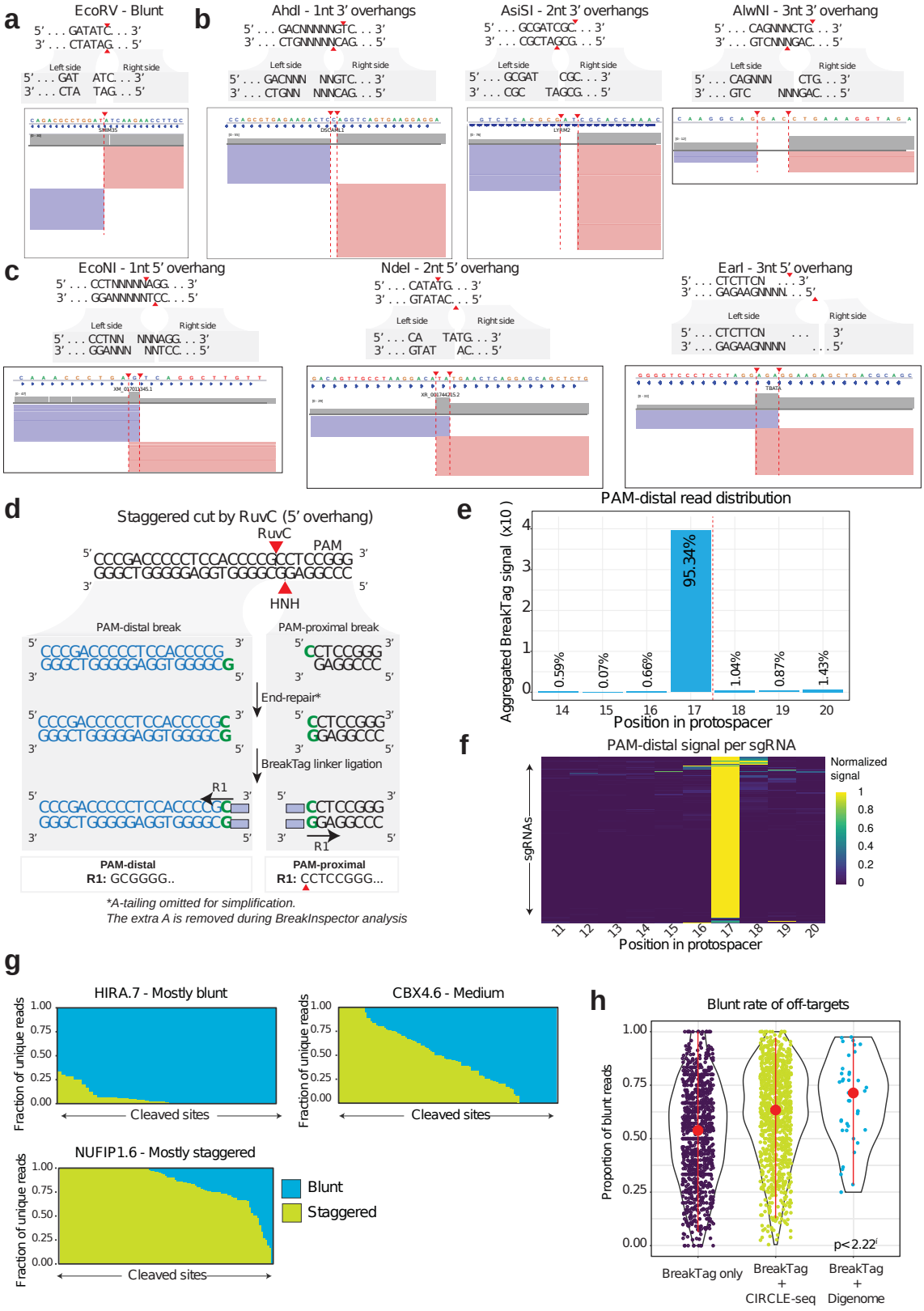


Figure C.3: BreakTag allows profiling of Cas9 scission. (a) gDNA of HEK293 cells was in vitro

digested with a panel of restriction enzymes that generate blunt DSBs, (b) 1–3 nt long 3' ssDNA overhangs, or (c) 1–3 nt long 5' ssDNA overhangs at the cut site, and BreakTag was performed. IGV snapshots show raw mapped reads for a representative target site for each enzyme. Arrowheads indicate the start of DSB reads. (d) Scheme depicting a staggered DSB with a 1 nt 5' overhang. PAM-proximal side of the break starts 1 nt upstream (16|17) of the expected site for a blunt cut. (e) Read distribution of the PAM-distal read along the protospacer. Because of the direction of the reaction to fill-in 5' overhangs during end repair, PAM-distal reads map to position 17 (cut site from the HNH domain) for both blunt and staggered reads. (f) PAM-distal signal distribution along the protospacer for each sgRNA used in the HiPlex 1 data set. (g) Fraction of BreakTag reads accumulating on position 17 (blue) suggestive of a blunt incision, or in other positions of the protospacer (green) indicative of a staggered cut, for three sgRNAs. Each column represents a cleaved site including on and off-targets. (h) Blunt rate of off-targets nominated exclusively by BreakTag or shared with CIRCLE-seq or Digenome-seq. The line range in red characterizes the sample using the median (Q2) - depicted with a point - and the range between percentiles 0.025 and 0.975 (n=4,375 sites, two-sided ANOVA test comparing means, P-value<2.22e-16).

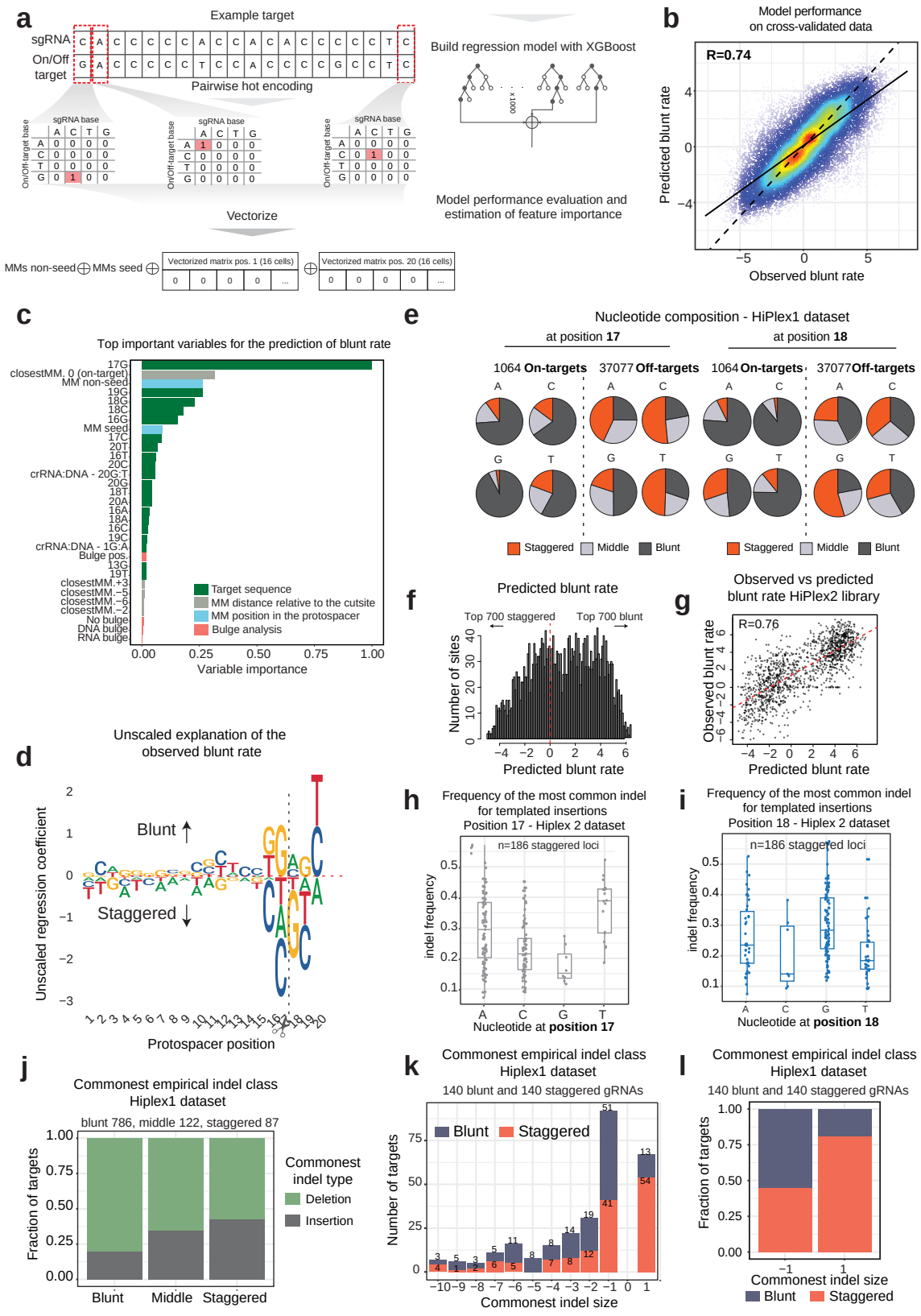


Figure C.4: Determinants of Cas9 scission profile. (a) Schematics of XGBoost method trained on BreakTag data. Training set consisted of a balanced set of 18,759 on and off-targets with a coverage of at least 16 reads in the PAM-proximal strand. (b) Model performance evaluation

using cross-validated data (Ten rounds of cross-validation). Panel shows the correspondence between expected (predicted) and observed log2 ratio of reads indicating a blunt or a staggered cut. (c) Scaled feature importance estimated by XGBoost. (d) Unscaled sequence explanation of the observed blunt rate using at most 100 off-targets identified by BreakTag for each sgRNA of the HiPlex1 library. (e) The effect of each base at positions 17 (left) and 18 (right) in the scission profile for on and off-targets in the HiPlex1 library for sites with at least 16 reads in the PAM-proximal strand. (f) Distribution of the predicted blunt rate for 2,791 gRNAs. (g) Correlation between predicted blunt rate by our model and observed blunt rate using BreakTag for top 700 staggered and top 700 blunt gRNAs identified. (h) Frequency of the most common indel for templated insertions as a function of nucleotide at position 17 for all staggered-cleaved loci with a+1 indel as the main repair outcome (n=186). Box plots show the lower (Q1) median (Q2) and upper quartile (Q3) with whiskers extending up to 1.5 times the interquartile range (IQR=Q3-Q1) from the box edges. (i) Frequency of the most common indel for template insertions as a function of nucleotide at position 18 for 186 staggered loci with templated insertions. (j) Fraction of targets where the most common repair outcome was a deletion (green) or insertion (gray). Cuts were grouped into 'blunt' (66-100% of blunt reads) 'middle' (33-66% of blunt reads) and 'staggered' (0-33% of blunt reads). Publicly available amplicon sequencing data was used. (k) Most common indel size as a function of scission profile. Cuts were grouped into 'blunt' (>=50% of blunt reads) and 'staggered' (<50% of blunt reads). (l) Proportion of sites where the most common outcome was -1 (1nt deletion) or +1 (1nt insertion) as a function of scission profile.

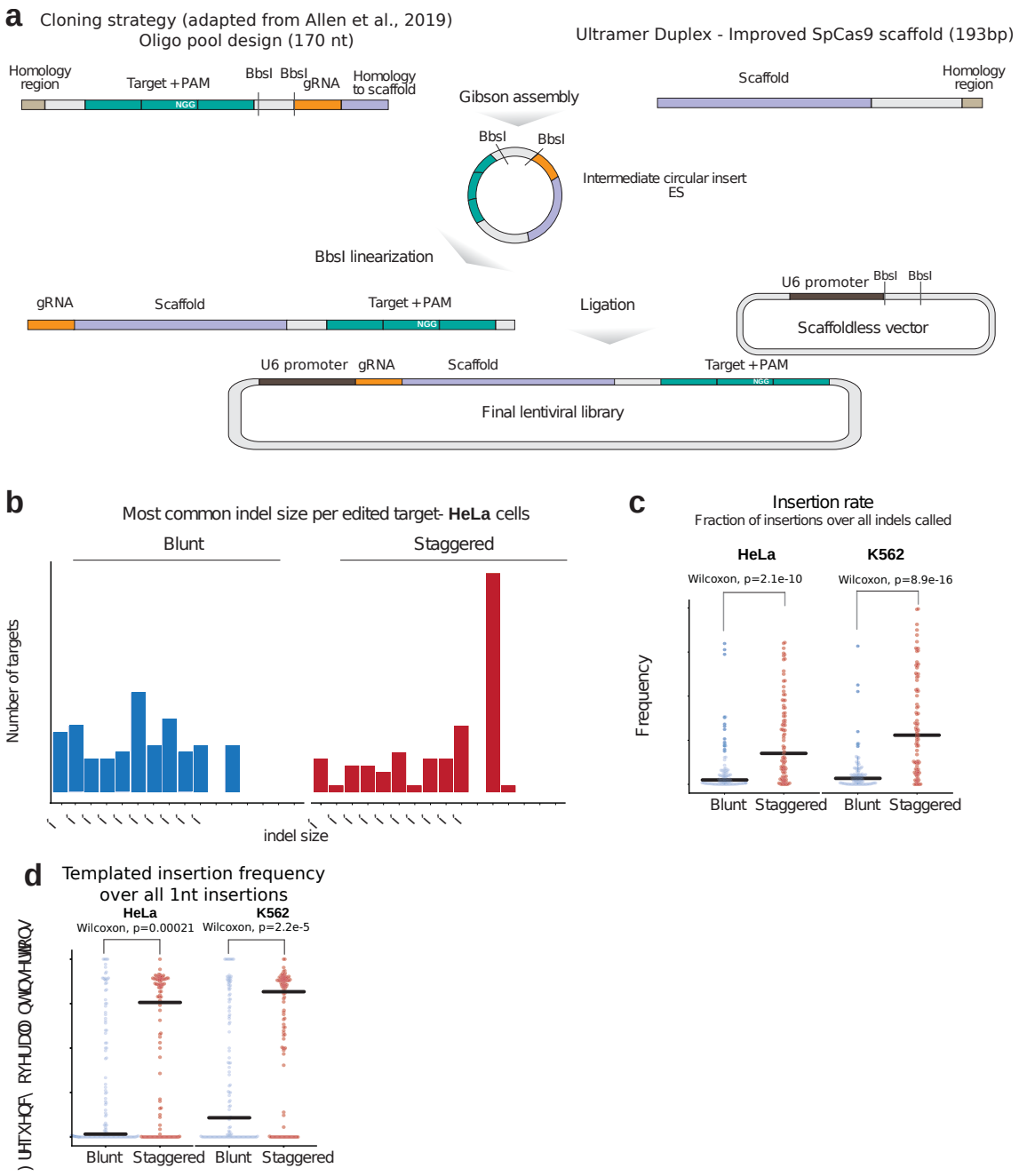


Figure C.5: Parallel assessment of indel outcomes of target sequences predicted to be cut preferably in a blunt or staggered manner. (a) Schematics of the strategy used to clone gRNA-target pairs into a lentiviral vector (adapted from Allen 2019). Briefly, we designed the 79nt portion of the pathogenic allele carrying the deletion and PAM and its gRNA and ordered it in a Pool format. We performed a Gibson assembly reaction with an Ultramer Duplex containing a portion of the improved SpCas9 scaffold. The intermediate circular insert was linearized and ligated into a scaffoldless pKLV2-U6(BbsI)-PKGpuro2ABFP-W (addgene #67974). (b) Most common indel size found per edited target in HeLa-Cas9. A total of 200 gRNA-target pairs (91 staggered and 109 blunt) were used for this analysis after filtering for sites with at least 100 mutated reads and not detected in the experiment performed with cells not expressing Cas9. (c) Insertion rate of target sequences predicted to be cleaved preferably in a blunt or staggered manner. Insertion

rate was calculated as the fraction of insertion over all indels called. Horizontal lines represent the median values. A two-sided Wilcoxon test was performed to assess the significance of the differences observed between the mean signed ranks of the two conditions being compared (HeLa blunt vs. HeLa staggered P-value 2.1e-10; K562 blunt vs. K562 staggered P-value 8.9e-16; n=399 independent Cas9-induced cutsites). (d) Frequency of templated insertions over all +1 indels. Insertions were considered as templated when the inserted base is the same nucleotide found in position 17 of the protospacer. Horizontal lines represent the median values. A two-sided Wilcoxon test was performed to assess the significance of the differences observed between the mean signed ranks of the two conditions being compared (HeLa blunt vs. HeLa staggered P-value 0.00021; K562 blunt vs. K562 staggered P-value 2.2e-5; n=399 independent Cas9-induced cutsites).

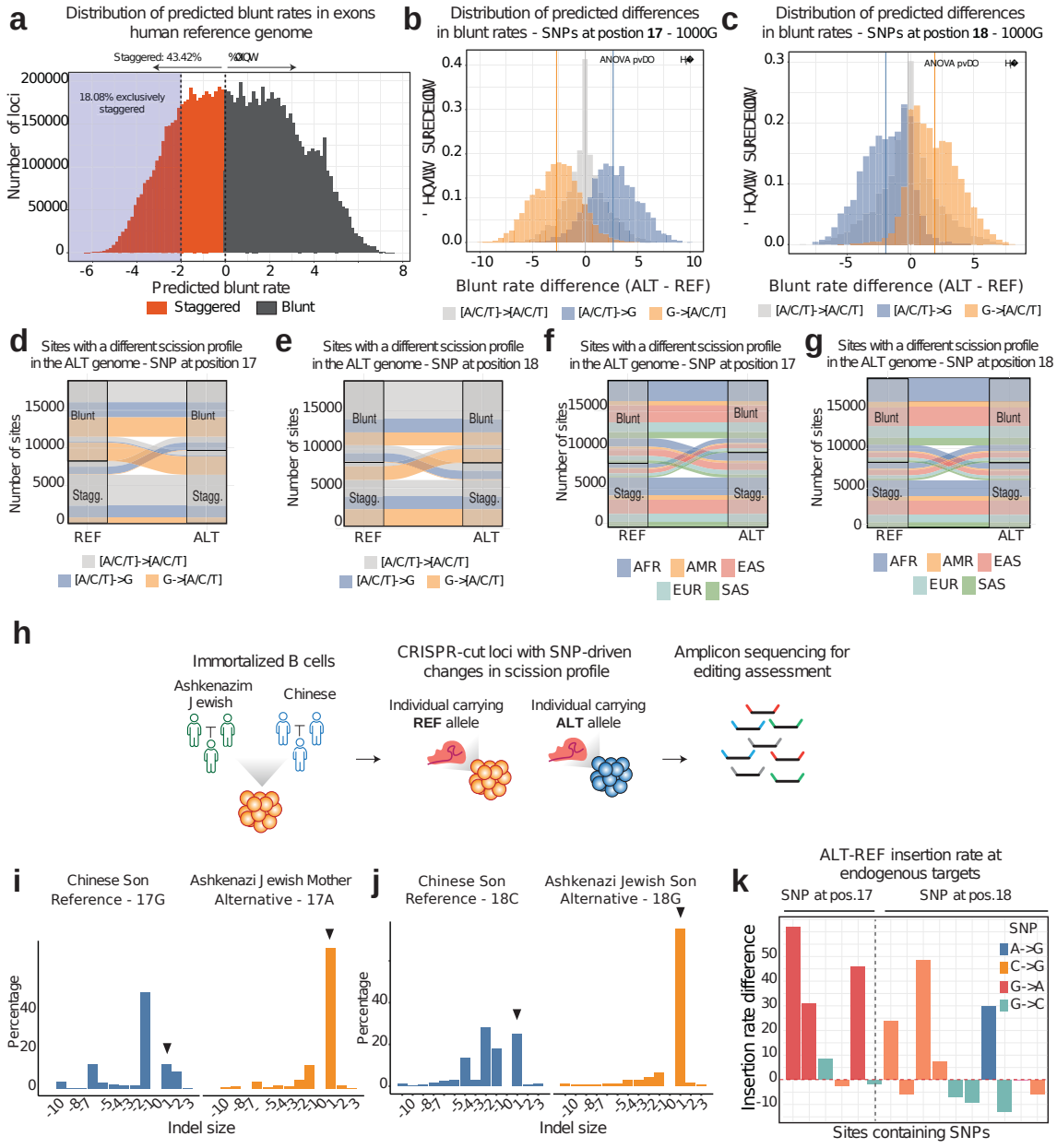


Figure C.6: Predicting changes in scission profile driven by SNPs at key positions along the protospacer. (a) Prediction of the blunt rate of every putative Cas9 target site found within

exons in the human genome. Dashed lines mark thresholds at log2 rates of 0 (50% blunt DSBs, gray distribution; 50% staggered DSBs, orange distribution) and -2 (80% staggered DSBs, orange distribution). (b) Distribution of predicted changes in blunt rates for SNPs found at position 17 for the 1000G dataset. (two-sided ANOVA test comparing means, P-value<2.2e-16). (c) Distribution of predicted changes in blunt rates for SNPs found at position 18 for the 1000G dataset. (two-sided ANOVA test comparing means, P-value<2.2e-16) (d, e) Sankey diagrams showing transitions between scission profile classes for SNPs found at positions 17 (d) and 18 (e). The colors indicate genotype. Blunt threshold is log2 rate > 0, otherwise staggered. (f, g(Superpopulation-resolved Sankey diagrams showing predicted SNP-driven transitions between scission profile classes for positions 17 (f) and 18 (g). AFR: African; AMR: American; EAS: East Asian; EUR: European; SAS: South Asian. (h) Schematics of the experimental design for targeting the REF and ALT allele-containing GIAB donor B cells. (i) Indel size distribution of the targeted locus containing an SNP at position 17 as shown in panel G. Indels of sizes between -10 and +3 were used for this analysis. Arrow heads indicate +1 indels. (j) Indel size distribution of a locus containing an SNP at position 18 as shown in panel J. Indels of sizes between -10 and +3 were used for this analysis. Arrow heads indicate +1 indels. (k) Difference in the insertion rate of target sites containing the indicated SNPs at position 17 or 18. Positive values indicate an increase in the insertion rate in the ALT allele, and negative values indicate a decrease in the insertion rate in ALT allele compared to REF.

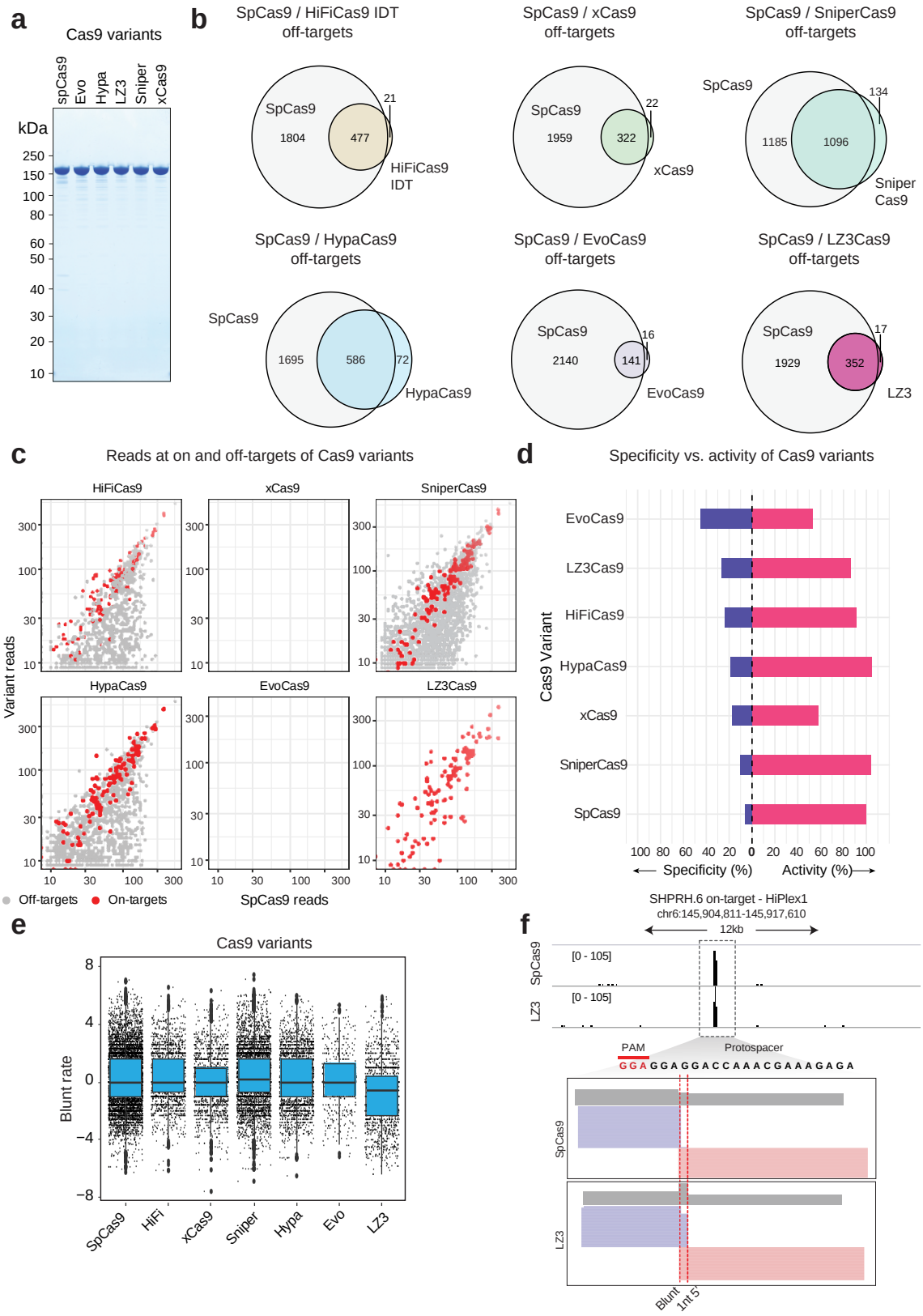


Figure C.7: Cas9 variant specificity, activity and blunt rate analysis as measured by BreakTag. (a) Coomassie Blue staining of recombinant Cas9 variants used here. (b) Venn diagrams showing common cleaved sites mapped with BreakTag between SpCas9 and the tested Cas9 vari-

ant. Off-targets with at least 8 reads were used for this analysis. (c) Reads at on and off-targets (up to 7 mismatches) for SpCas9 (x axis) and variants (y axis). Red dots indicate on-target signal and gray dots indicate off-targets. Off-targets with at least 8 reads were used for this analysis. (d) Specificity (left direction) and activity (right direction) of tested Cas9 variants as calculated with BreakTag readout. Activity is reported in relation to SpCas9. (e) Distribution of blunt rate for each Cas9 variant identified by BreakTag. Each point is a cleaved site (on-target or off-target). Blunt rate was calculated over 2 technical replicates. Boxes characterize the sample using the lower quartile (Q1) median (Q2) and upper quartile (Q3)—and the interquartile range (IQR=Q3-Q1), and whiskers extend to the most extreme data point that is no more than 1.5 × IQR from the edge of the box. (f) IGV snapshot showing an example of differential scission profile for the on-target sequence of SHPRH.6 sgRNA (HiPlex1 library).

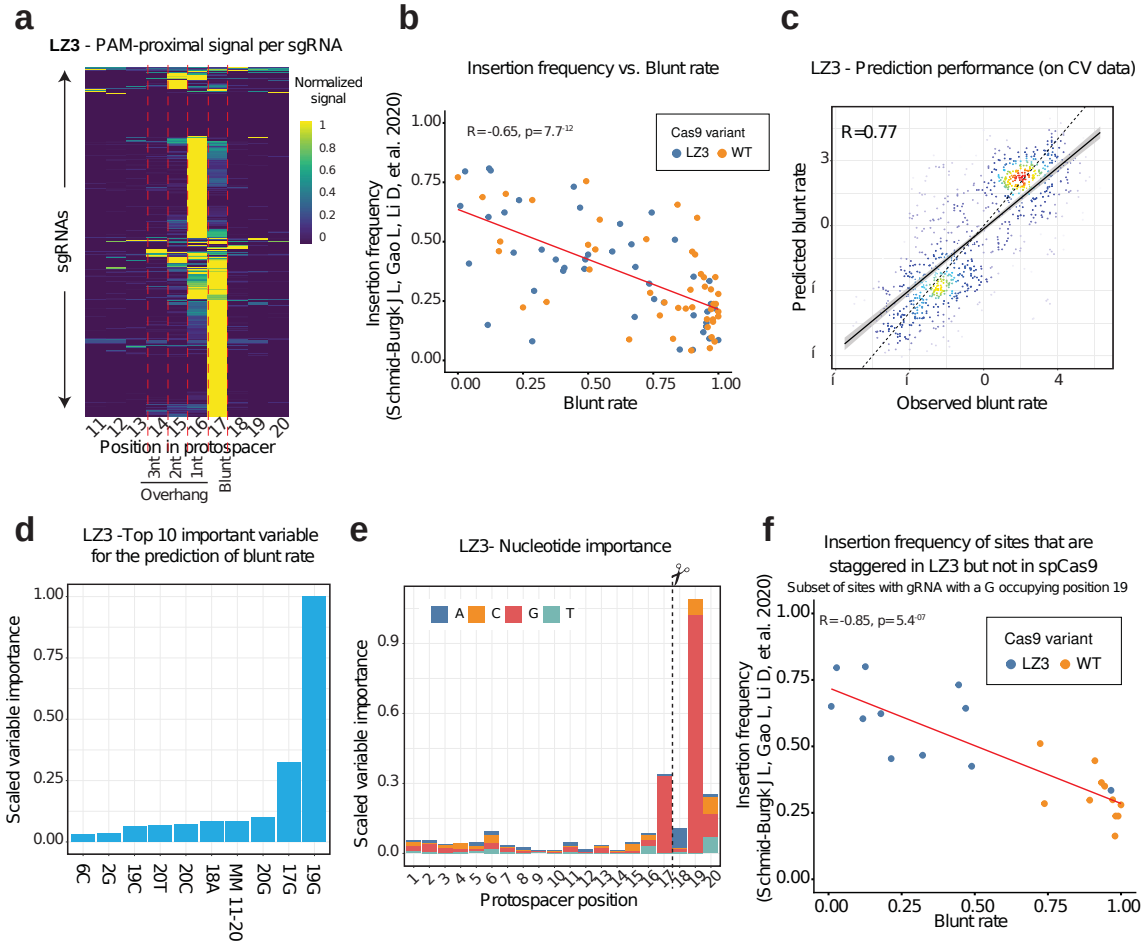


Figure C.8: Characterization of the sequence determinants of the LZ3 flexible scission profile. (a) Accumulation of reads mapped onto the PAM-proximal strand (scaled) along the protospacer over 4,543 sgRNAs of the HiPlex1 library generated with the LZ3 nuclease for all identified targets with an 'NGG' PAM. (b) Correlation between insertion frequency and blunt rate calculated with BreakTag for 95 gRNAs for each Cas9 variant. (c) Model performance evaluation using cross-validated (CV) data. This panel shows the correspondence between expected (predicted) and observed log2 ratio of reads indicating a blunt or a staggered cut. (Pearson correlation R=-0.65, P-value=7.7e-12). (Pearson correlation R=0.77). Dotted line represents perfect correlation (R=1); error bands represent the 95% confidence interval around the linear model fit. (d)

Top ten most important variables for the prediction of LZ3 blunt rate. MM 11–20: mismatches in the seed part of the protospacer (positions 11–20). (e) Top ten most important variables for the prediction of LZ3 blunt rate. MM 11–20: mismatches in the seed part of the protospacer (positions 11–20). (f)Correlation between insertion frequency and blunt rate of the subset of 22 sites where a G occupied position 19 of the protospacer that are staggered when LZ3 was used but blunt when SpCas9 was used. (Pearson correlation R=-0.85, P-value=5.4e-7).

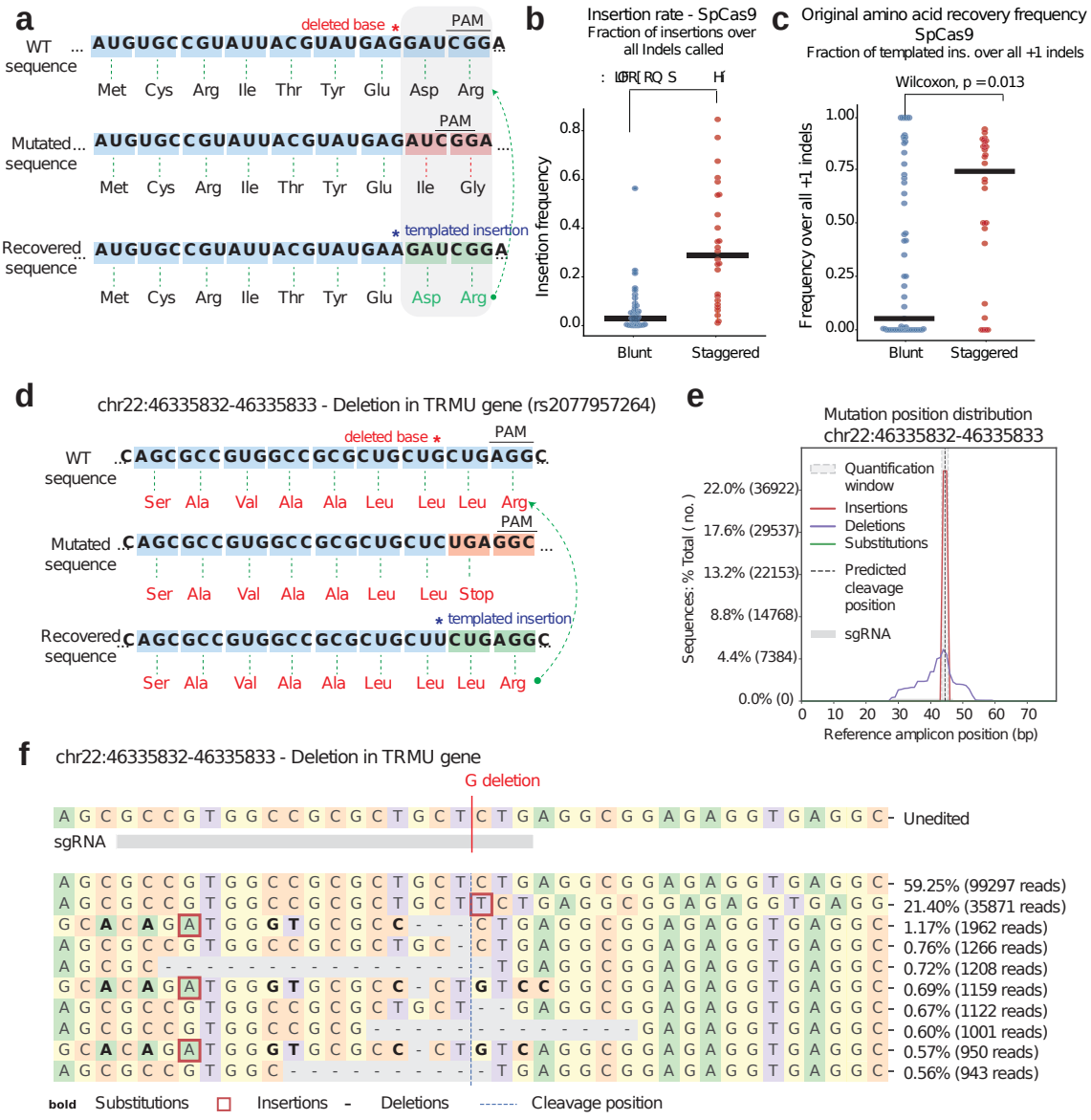


Figure C.9: Investigation of indel outcomes at targeted pathogenic single-nucleotide deletions. (a) Example of 1nt deletion generating a frameshift mutation, and a templated insertion rescuing the frame and original amino acid sequence. (b) Insertion rate of pathogenic 1nt deletions predicted to be cleaved in a blunt or staggered manner. Horizontal lines represent the median values. A two-sided Wilcoxon test was performed to assess the significance of the differences observed between the mean signed ranks of the two conditions being compared (blunt vs. staggered P-value 8.6e-8; n=145 independent Cas9-induced cutsites). (c) Rate of original protein sequence recovery, as measured by the frequency of templated insertions (i.e, duplication of the base found

at position 17 of the protospacer) over all +1 indels. Horizontal lines represent the median values. A two-sided Wilcoxon test was performed to assess the significance of the differences observed between the mean signed ranks of the two conditions being compared (blunt vs. staggered P-value 0.013; n=145 independent Cas9-induced cutsites). (d) Example of a pathogenic allele in the staggered pool. The 1nt deletion generates a stop codon in the TRMU gene, but the correct ORF is recovered upon templated +1 insertion. (e) CRISPResso2 output of the mutation outcome type distribution of the TRMU 1nt deletion depicted in Extended Data Fig. 9d. f, Table depicting the top 10 repair outcomes after targeting the 1nt deletion in the TRMU gene with SpCas9.

C.2 Companion software: the BreakTag pipeline and breakinspectoR package

BreakTag pipeline

Here we provide the tools to perform paired end or single read BreakTag raw data processing. The pipeline is also valid for (s)BLISS data. As input files you may use either gzipped fastq-files (.fastq.gz) or mapped read data (.bam files). In case of paired end reads, corresponding fastq files should be named using *.R1.fastq.gz* and *.R2.fastq.gz* suffixes. Some steps of the pipeline are based on the blissNP pipeline developed at the BriCo lab for BLISS data.

Overall description

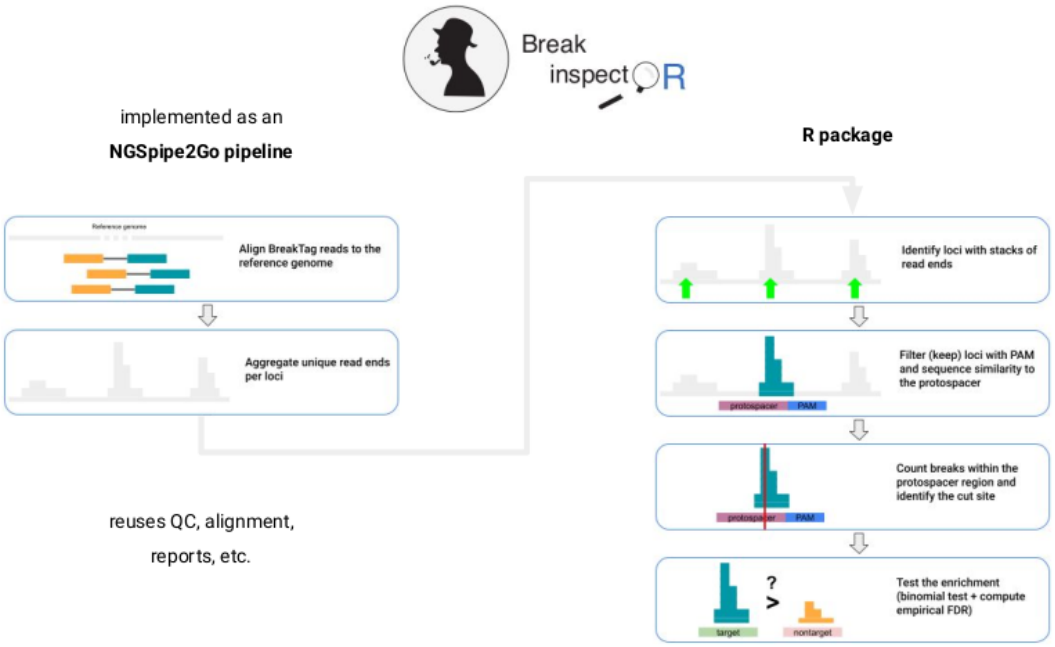


Figure 1: breakinspectoR workflow

Initial preprocessing

Initial preprocessing is typically done in a linux cluster using the Breaktag pipeline. It includes the following steps:

1. scanning for reads (single- or paired-end) containing the expected 8-nt UMI followed by the 8-nt sample barcode in the 5' end of read 1.
2. alignment of reads to reference genome with BWA, with a seed length of 19 and default scoring/penalty values for mismatches, gaps and read clipping.

3. reads mapped with a minimum quality score Q (defaults to Q=60) are retained.
4. close spatial consecutive reads within a window of 30 nucleotides and UMI differing with up to 2 mismatches are considered PCR duplicates and only one is kept.

The resulting reads are aggregated per position and reported as a BED file.

breakinspectoR analysis

This R package implements the identification of CRISPR (currently, Cas9 only) targets and estimation of the scission profile. Additionally it provides several plotting functions to graphically summarize the results.

Installing the pipeline

Docker installation

This repository contains a `Dockerfile` which can be used to create a Docker image with all dependencies. This is the preferred and easiest way to have all the dependencies satisfied and run the pipeline.

Manual installation

Install and make available in the path the following dependencies: - Bpipe - Bedtools - BWA - FastQC - MultiQC - Samtools - Several Unix standard tools (perl5, python3, awk, etc.)

If you're running the pipeline in a cluster, you probably want to edit the tools config file and tell the pipeline how these tools are loaded (added in $PATH).

Running the pipeline

Tools are expected to be in the PATH. From the root of the folder where you clone the breaktag pipeline:

- edit the parameters file:
  `breaktag/pipelines/breaktag/essential.vars.groovy`
- edit the targets file:
  `breaktag/pipelines/breaktag/targets.txt`
- softlink these 2 files to the root folder:
  ```
  ln -s breaktag/pipelines/breaktag/essential.vars.groovy .
  ln -s breaktag/pipelines/breaktag/targets.txt .
  ```
- run the pipeline with this command (eg. from within the docker container):

```
bpipe run -n256 \
  breaktag/pipelines/breaktag/breaktag.pipeline.groovy \
  ./rawdata/*.fastq.gz
```

Pipeline-specific parameter settings (files you need to setup in order to run the pipeline):

- `targets.txt`: tab-separated txt-file giving information about the analysed samples. The following columns are required
  - name: sample name. Experiment ID found in fastq filename: expID_R1.fastq.gz
  - pattern: UMI+barcode pattern file used in the linker
- essential.vars.groovy: essential parameters describing the experiment
  - ESSENTIAL_PROJECT: root folder of the analysis
  - ESSENTIAL_SAMPLE_PREFIX: sample name prefix to be trimmed in the results
  - ESSENTIAL_THREADS: number of threads for parallel tasks
  - ESSENTIAL_BWA_REF: path to bwa indexed reference genome
  - ESSENTIAL_PAIRED: either paired end ("yes") or single read ("no") design
  - ESSENTIAL_QUALITY: minimum mapping quality desired

breaktag ESSENTIAL VARIABLES

Important parameters are included in this file. They're distributed in several sections.

General parameters

```
ESSENTIAL_PROJECT="/project/folder"
ESSENTIAL_SAMPLE_PREFIX=""
ESSENTIAL_THREADS=16
```

Mapping parameters

```
ESSENTIAL_BWA_REF="/ref/index/bwa/hg38.fa" // BWA index of reference genome
ESSENTIAL_PAIRED="no"     // paired end design
ESSENTIAL_QUALITY=60      // min mapping quality of reads to be kept
                          // Defaults to 60 (discard multimappers and
                          // low quality mapping)
```

Other You probably don't need to touch these lines:

```
// further optional pipeline stages to include
RUN_IN_PAIRED_END_MODE=(ESSENTIAL_PAIRED == "yes")

// project folders
PROJECT=ESSENTIAL_PROJECT
LOGS=PROJECT + "/logs"
MAPPED=PROJECT + "/mapped"
QC=PROJECT + "/qc"
RAWDATA=PROJECT + "/rawdata"
REPORTS=PROJECT + "/reports"
RESULTS=PROJECT + "/results"
TMP=PROJECT + "/tmp"
TRACKS=PROJECT + "/tracks"
TARGETS=PROJECT + "/targets.txt"
```

More fine-grained tunning of the tools called by the pipeline can be controled from the `.header` files in the `breaktag/modules` folder.

Targets file

A tab-separated file with the filenames (excluding the .fastq.gz extension) and the breaktag barcode and the position of the UMI within the read.

```
name        pattern                 umi
FANCF_rep1  ^(........)(CTCACACGT)  $1
FANCF_rep2  ^(........)(CTCACACGT)  $1
NT_rep1     ^(........)(CTCACACGT)  $1
NT_rep2     ^(........)(CTCACACGT)  $1
```

bluntPred

For your set of gRNAs, you may want to run the prediction of blunt rates of Streptococcus pyogenes Cas9 (SpCas9) using the XGBoost model trained with HiPlex1 data.

Dependencies

The main dependency is H2O.

Remove any previously installed H2O packages for R.

```
if ("package:h2o" %in% search()) { detach("package:h2o", unload=TRUE) }
if ("h2o" %in% rownames(installed.packages())) { remove.packages("h2o") }
```

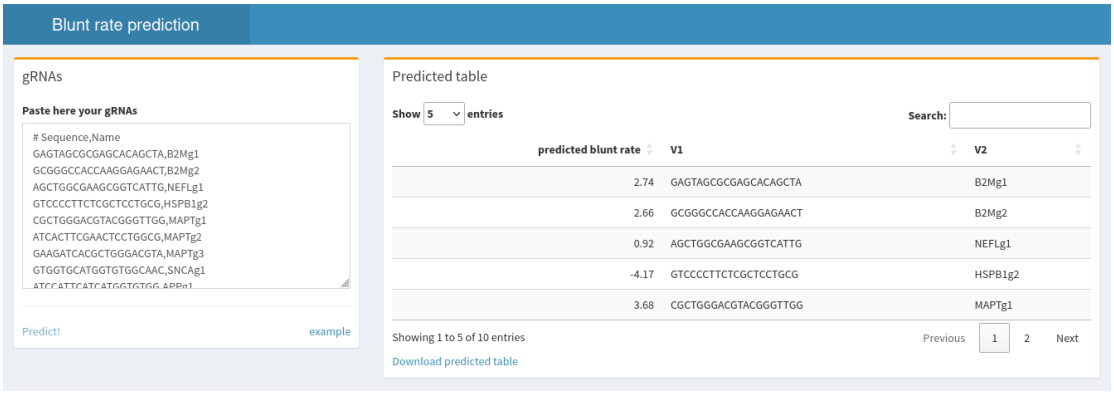


Figure 2: bluntPred

Download packages that H2O depends on.

```
install.packages(c("RCurl","jsonlite", "devtools"))
```

Download and install the H2O package for R. The models were trained on H2O version 3.36.1.2, therefore specifically install this version.

```
install.packages(
  "h2o",
  type="source",
  repos="http://h2o-release.s3.amazonaws.com/h2o/rel-zumbo/2/R"
)
```

Test the H2O installation with:

```
library(h2o)
localH2O = h2o.init()
demo(h2o.kmeans)
```

Now, it's all set to install the package.

The package resides in GitHub only. You will probably need `devtools` for that (`install.packages("devtools")`).

```
devtools::install_github("roukoslab/bluntPred")
```

Run

Open the web app in your R console:

```
bluntPred::shiny_bluntPred()
```

Paste a list of gRNAs targets and click on `Predict`. The list can actually be a table with <tab> or <comma> separated fields. The gRNA sequence is expected to be in the *first* column.

NOTE: Only the seed portion of the protospacer (this is, the last 10 nucloetides of the target sequence) are used for the prediction in this model.

Cite

If you find this tool useful and use it in your research, please cite our publication:

Longo, Sayols et al., Linking CRISPR–Cas9 double-strand break profiles to gene editing precision with BreakTag. Nat. Biotechnol. 2024. DOI: https://doi.org/10.1038/s41587-024-02238-8

breakinspectoR

A companion R package to the BreakTag protocol for the identification of CRISPR offtargets. breakinspectoR is an R package which performs a guided search toward putative on-/off-targets.

Overall description

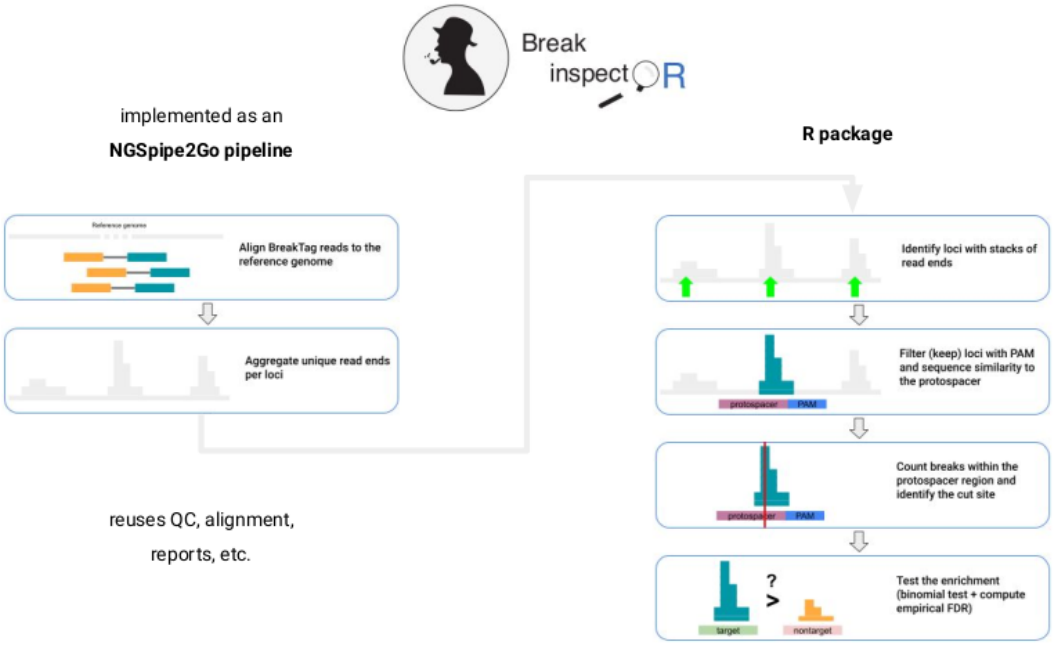


Figure 1: breakinspectoR workflow

Initial preprocessing

Initial preprocessing is typically done in a linux cluster using the Breaktag pipeline. It includes the following steps:

1. scanning for reads (single- or paired-end) containing the expected 8-nt UMI followed by the 8-nt sample barcode in the 5' end of read 1.
2. alignment of reads to reference genome with BWA, with a seed length of 19 and default scoring/penalty values for mismatches, gaps and read clipping.
3. reads mapped with a minimum quality score Q (defaults to Q=60) are retained.
4. close spatial consecutive reads within a window of 30 nucleotides and UMI differing with up to 2 mismatches are considered PCR duplicates and only

one is kept.

The resulting reads are aggregated per position and reported as a BED file.

breakinspectoR analysis

The analysis path consists of the following steps:

1. from the previously generated BED file, identify stacks of read ends as the candidate loci of being CRISPR-edited.
2. obtain the sequence context of the candidate loci, and keep only those which are at most `N` nucleotides upstream from a PAM and contain up to M mismatches with regard to the gDNA guide sequence. Defaults are the canonical N=3, M=7, PAM="NGG".
3. count number of reads (== signal or DSBreaks) in the targeted and the non-targeted control library.
4. test if the enrichment of reads we see in the targeted library is significant compared to the non-targeted control library. Here breakinspectoR will perform a binomial test with the following criteria:
    1. preconditions:
        1. A = number of breaks within the region in the target library
        2. B = total number of breaks in the whole target library
        3. C = number of breaks within the region in the nontarget library
        4. D = total number of breaks in the whole nontarget library
    2. the binomial model for calculating the p-Value is:
        1. number of trials = A+1
        2. number of successes = B+1
        3. estimated success probability in each trial = (C+1) / (D+1)
    3. the enrichment is then calculated as: ((A+1) / (B+1)) / ((C+1) / (D+1))
    4. note that we add a pseudocount to avoid dividing by 0
    5. q-values and local False Discovery Rate values are estimated for FDR control.
5. additionally, breakinspectoR implements a complimentary [and elaborated] method to estimate the false discovery of targets. To summarize, breakinspectoR reshuffles the signal in the target library using several multinomially distributed random number vectors sampled with equal probabilities to the signal in the originally detected offtargets. Then, breakinspectoR analysis is done in the reshuffled target library vs. the non-target library, and an FDR is estimated comparing the signal of each offtarget called in the original target library to the targets detected in the reshuffled target library (where no targets were expected to be called).

6. breakinspectoR includes several handy visualizations to further analyze and summarize the on-/off-targets detected. Some of these functions include the analysis of fidelity of the gDNA, sequence composition of target regions, frequency of mismatches per position of the protospacer, or the genomic distribution of the targeted regions.

Installation

Open R and install directly from Github with `devtools` (install the package `devtools` if you haven't, yet):

```
devtools::install_github("roukoslab/breakinspectoR")
```

Example usage

This is a simple example using the demo data for human chr6 included with the package. The experiment identifies offtargets generated by the VEGFA site 2 sgRNA with CRISPR/Cas9.

Call the breakinspectoR analysis to find offtargets enriched in the targeted library compared to the non-targeted. We'll stick to the default 7 mismatches allowed to the guide, with the expected cut site 3 bp away from the PAM (Cas9).

```
target_file     <- system.file("extdata/vegfa.chr6.bed.gz",
                               package="breakinspectoR")
non_target_file <- system.file("extdata/nontarget.chr6.bed.gz",
                               package="breakinspectoR")
guide           <- "GACCCCCTCCACCCCGCCTC"
PAM             <- c(canonical="NGG", "NAG")
bsgenome        <- "BSgenome.Hsapiens.UCSC.hg38"

offtargets <- breakinspectoR(
    target        =target_file,
    nontarget     =non_target_file,
    guide         =guide,
    PAM           =PAM,
    bsgenome      =bsgenome,
    cutsiteFromPAM=3,
    verbose       =FALSE
)
```

The analysis will take few seconds to run. Afterwards we have a comprehensive table with few hundred enriched offtarget loci, which we can summarize using the accompanying plotting functions:

```
plot_position_cutsite(offtargets, guide=guide, pam=PAM["canonical"])

plot_mismatch_freq(offtargets)

plot_offtargets_by_pam(offtargets)

plot_sequence_composition(offtargets, guide=guide, pam=PAM["canonical"])

plot_guide_fidelity(offtargets, guide=guide, pam=PAM["canonical"])

manhattan_plot(offtargets, bsgenome=bsgenome)
```

Input files

Input BED files describing coordinates and number of DSB are expected for the "target" and "non-target" libraries. It's possible to run breakinspectoR without the non-target library, nevertheless it is advised to include such experiment to calculate a p-value and control the false discovery rate.

```
chr6  148074  148075  .  2  +
chr6  148093  148094  .  1  -
chr6  148240  148241  .  1  -
chr6  148503  148504  .  1  +
chr6  148636  148637  .  1  -
chr6  148697  148698  .  1  -
chr6  149009  149010  .  1  -
chr6  149363  149364  .  1  +
chr6  150252  150253  .  2  +
chr6  150263  150264  .  1  +
```

These files are typically created with the Breaktag pipeline, although any conforming BED file is accepted in breakinspectoR.

BTmotif

Additionally, you may want to run the companion shiny app to derive Cas9 sequence determinants from BreakInspectoR output.

It uses XGBoost and the provided sequence (usually, a protospacer) to predict which nucleotides and positions are important to predict any numerical outcome (eg. the blunt rate, Cas9 activity, etc.).

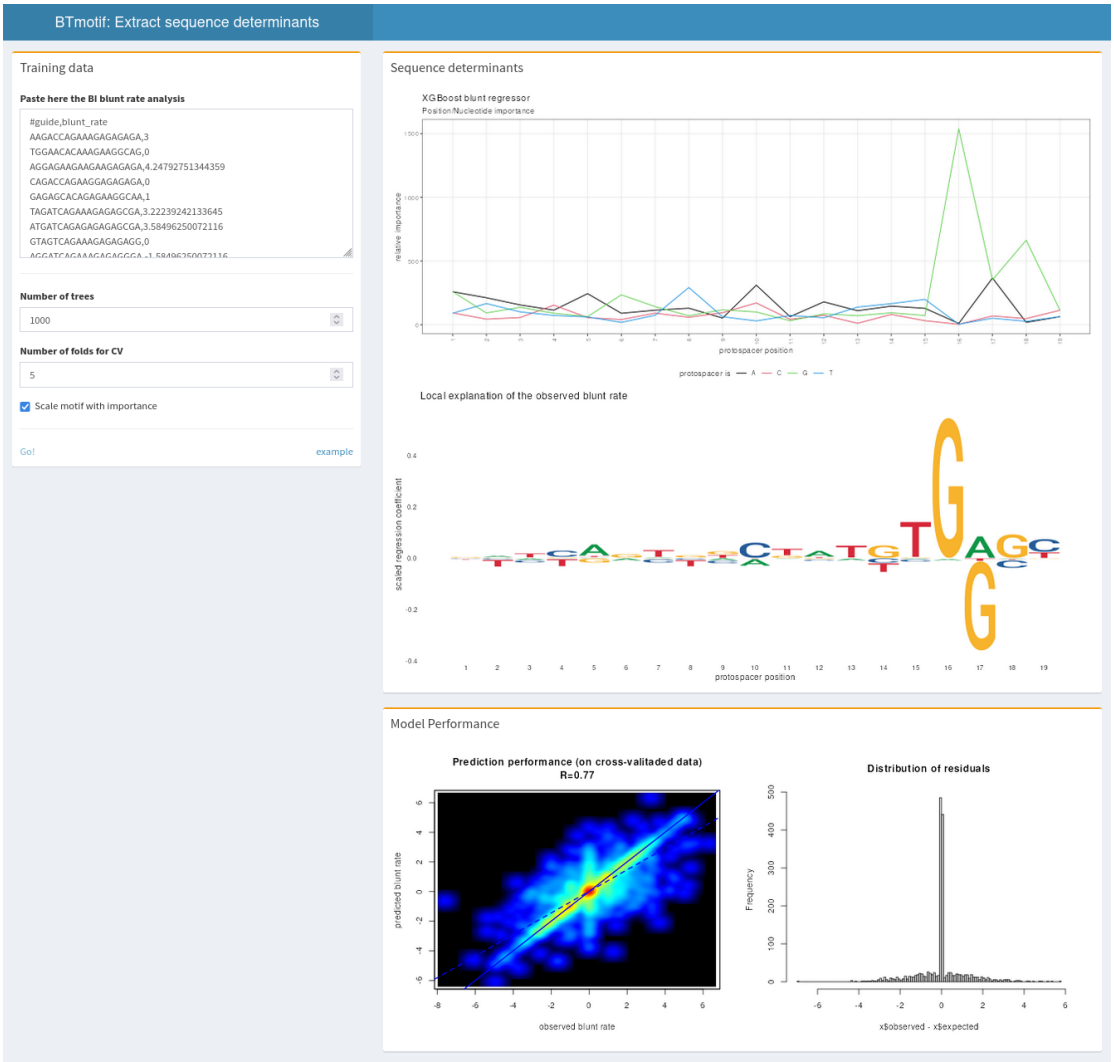


Figure 2: BTmotif

Dependencies

The main dependency is H2O, which can be installed from CRAN. The app has been tested with H2O version 3.36.1.2.

```
install.packages(
  "h2o",
  type="source",
  repos="http://h2o-release.s3.amazonaws.com/h2o/rel-zumbo/2/R"
)
```

Test the H2O installation with:

```
library(h2o)
localH2O = h2o.init()
demo(h2o.kmeans)
```

You'll need a couple of packages to run the web app:

```
install.packages("shiny")
install.packages("shinydashboard")
```

To generate the motifs, you'll also need `ggplot2` and `ggseqlogo`:

```
install.packages("ggplot2")
devtools::install_github("omarwagih/ggseqlogo")
```

Run

Open the web app in your R console:

```
breakinspectoR::shiny_BTmotif()
```

Paste a table of targets and click on `Go!`. Or check the `Example` data. The list can actually be a table with <tab> or <comma> separated fields. The columns are expected to be in this order: protospacer_sequence | blunt_rate.

bluntPred

For your set of gRNAs, you may want to run the prediction of blunt rates of Streptococcus pyogenes Cas9 (SpCas9) using the XGBoost model trained with HiPlex1 data.

Dependencies

The main dependency is H2O.

Remove any previously installed H2O packages for R.

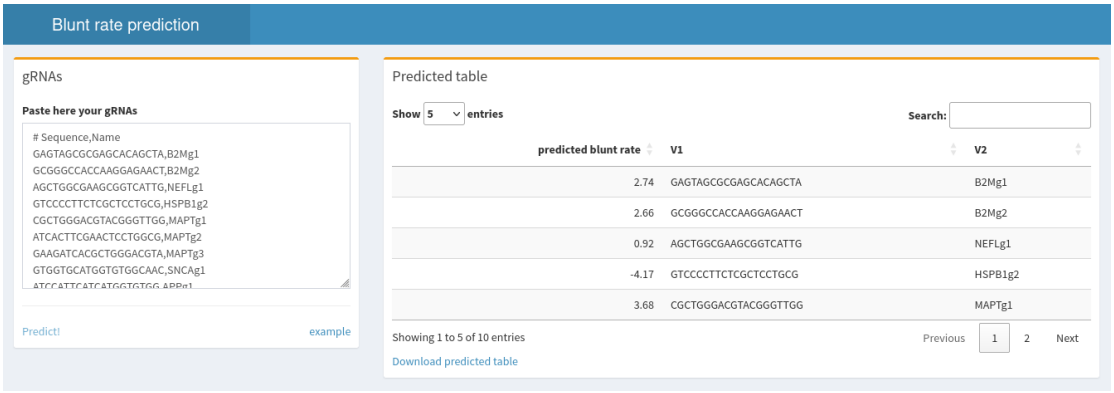


Figure 3: bluntPred

```
if ("package:h2o" %in% search()) { detach("package:h2o", unload=TRUE) }
if ("h2o" %in% rownames(installed.packages())) { remove.packages("h2o") }
```

Download packages that H2O depends on.

```
install.packages(c("RCurl","jsonlite", "devtools"))
```

Download and install the H2O package for R. The models were trained on H2O version 3.36.1.2, therefore specifically install this version.

```
install.packages(
  "h2o",
  type="source",
  repos="http://h2o-release.s3.amazonaws.com/h2o/rel-zumbo/2/R"
)
```

Test the H2O installation with:

```
library(h2o)
localH2O = h2o.init()
demo(h2o.kmeans)
```

Now, it's all set to install the package.

The package resides in GitHub only. You will probably need `devtools` for that (`install.packages("devtools")`).

```
devtools::install_github("roukoslab/bluntPred")
```

Run

Open the web app in your R console:

```
bluntPred::shiny_bluntPred()
```

Paste a list of gRNAs targets and click on `Predict`. The list can actually be a table with <tab> or <comma> separated fields. The gRNA sequence is expected to be in the *first* column.

NOTE: Only the seed portion of the protospacer (this is, the last 10 nucloetides of the target sequence) are used for the prediction in this model.

Cite

If you find this tool useful and use it in your research, please cite our publication:

Longo, Sayols et al., Linking CRISPR–Cas9 double-strand break profiles to gene editing precision with BreakTag. Nat. Biotechnol. 2024. DOI: https://doi.org/10.1038/s41587-024-02238-8

References

Abadi, S., Yan, W. X., Amar, D., & Mayrose, I. (2017). A machine learning approach for predicting CRISPR-Cas9 cleavage efficiencies and patterns underlying its mechanism of action. *PLOS Computational Biology*, *13*(10), e1005807. http://doi.org/10.1371/journal.pcbi.1005807

Abudayyeh, O. O., Gootenberg, J. S., Konermann, S., Joung, J., Slaymaker, I. M., Cox, D. B. T., ... Zhang, F. (2016). C2c2 is a single-component programmable RNA-guided RNA-targeting CRISPR effector. *Science (New York, N.Y.)*, *353*(6299), aaf5573. http://doi.org/10.1126/science.aaf5573

Allen, F., Crepaldi, L., Alsinet, C., Strong, A. J., Kleshchevnikov, V., De Angeli, P., ... Parts, L. (2019). Predicting the mutations generated by repair of Cas9-induced double-strand breaks. *Nature Biotechnology*, *37*(1), 64–72. http://doi.org/10.1038/nbt.4317

Anzalone, A. V., Koblan, L. W., & Liu, D. R. (2020). Genome editing with CRISPR-Cas nucleases, base editors, transposases and prime editors. *Nature Biotechnology*, *38*(7), 824–844. http://doi.org/10.1038/s41587-020-0561-9

Atkins, A., Chung, C.-H., Allen, A. G., Dampier, W., Gurrola, T. E., Sariyer, I. K., ... Wigdahl, B. (2021). Off-Target Analysis in Gene Editing and Applications for Clinical Translation of CRISPR/Cas9 in HIV-1 Therapy. *Frontiers in Genome Editing*, *3*, 673022. http://doi.org/10.3389/fgeed.2021.673022

Auton, A., Abecasis, G. R., Altshuler, D. M., Durbin, R. M., Abecasis, G. R., Bentley, D. R., ... family=Angel. (2015). A global reference for human genetic variation. *Nature*, *526*(7571), 68–74. http://doi.org/10.1038/nature15393

Babraham Bioinformatics - FastQC A Quality Control tool for High Throughput Sequence Data. (n.d.). Retrieved November 16, 2023, from `https://www.bioinformatics.babraham.ac.uk/projects/fastqc/`

Bae, S., Park, J., & Kim, J.-S. (2014). Cas-OFFinder: A fast and versatile algorithm that searches for potential off-target sites of Cas9 RNA-guided endonucleases. *Bioinformatics (Oxford, England)*, *30*(10), 1473–1475. http://doi.org/10.1093/bioinformatics/btu048

Bahcall, O. G. (2018). UK Biobank - a new era in genomic medicine. *Nature Reviews Genetics*, *19*(12), 737–737. http://doi.org/10.1038/s41576-018-0065-3

Baker, M. (2012). De novo genome assembly: What every biologist should know. *Nature Methods*, *9*(4, 4), 333–337. http://doi.org/10.1038/nmeth.1935

BamUtil: Dedup - Genome Analysis Wiki. (n.d.). Retrieved November 16, 2023, from `https://genome.sph.umich.edu/wiki/BamUtil:_dedup`

Barua, S., Bandopadhyay, S., Biswas, S., & Gupta, P. (2022). What Is Next-Generation Sequencing and Why do we Need it? *Frontiers for Young Minds*, *10*, 746502. http://doi.org/10.3389/frym.2022.746502

Baumann, D. D., & Doerge, R. W. (2014). Robust adjustment of sequence tag abundance. *Bioinformatics*, *30*(5), 601–605. http://doi.org/10.1093/bioinformatics/btt575

Bcl2fastq2 Conversion Software v2.20 Software Guide (15051736). (n.d.).

Berthold, M. R., Cebron, N., Dill, F., Gabriel, T. R., Kötter, T., Meinl, T., ... Wiswedel, B. (2009). KNIME - the Konstanz information miner: Version 2.0 and beyond. *SIGKDD Explor. Newsl.*, *11*(1), 26–31. http://doi.org/10.1145/1656274.1656280

Blake, J. A., & Harris, M. A. (2008). The Gene Ontology (GO) Project: Structured Vocabularies for Molecular Biology and Their Application to Genome and Expression Analysis. *Current Protocols in Bioinformatics*, *23*(1), 7.2.1–7.2.9. http://doi.org/10.1002/0471250953.bi0702s23

Bolger, A. M., Lohse, M., & Usadel, B. (2014). Trimmomatic: A flexible trimmer for Illumina sequence data. *Bioinformatics*, *30*(15), 2114–2120. http://doi.org/10.1093/bioinformatics/btu170

Bray, N. L., Pimentel, H., Melsted, P., & Pachter, L. (2016). Near-optimal probabilistic RNA-seq quantification. *Nature Biotechnology*, *34*(5), 525–527. http://doi.org/10.1038/nbt.3519

Breiman, L. (2001). Random Forests. *Machine Learning*, *45*(1), 5–32. http://doi.org/10.1023/A:1010933404324

Brionne, A., Juanchich, A., & Hennequet-Antier, C. (2019). ViSEAGO: A Bioconductor package for clustering biological functions using Gene Ontology and semantic similarity. *BioData Mining*, *12*(1), 16. http://doi.org/10.1186/s13040-019-0204-1

Broadinstitute/picard. (2023, November 15). Broad Institute. Retrieved from `https://github.com/broadinstitute/picard` (Original work published March 28, 2014)

Buenrostro, J. D., Giresi, P. G., Zaba, L. C., Chang, H. Y., & Greenleaf, W. J. (2013). Transposition of native chromatin for fast and sensitive epigenomic profiling of open chromatin, DNA-binding proteins and nucleosome position. *Nature Methods*, *10*(12), 1213–8. http://doi.org/10.1038/nmeth.2688

Cameron, P., Fuller, C. K., Donohoue, P. D., Jones, B. N., Thompson, M. S., Carter, M. M., ... May, A. P. (2017). Mapping the genomic landscape of CRISPR–Cas9 cleavage. *Nature Methods*, *14*(6), 600–606. http://doi.org/10.1038/nmeth.4284

Cancellieri, S., Zeng, J., Lin, L. Y., Tognon, M., Nguyen, M. A., Lin, J., ... Pinello, L. (2023). Human genetic diversity alters off-target outcomes of therapeutic gene editing. *Nature Genetics*, *55*(1), 34–43. http://doi.org/10.1038/s41588-022-01257-y

Carbon, S., Ireland, A., Mungall, C. J., Shu, S., Marshall, B., Lewis, S., ... the Web Presence Working Group. (2009). AmiGO: Online access to ontology and annotation data. *Bioinformatics*, *25*(2), 288–289. http://doi.org/10.1093/bioinformatics/btn615

Carlson, M., & Pag'es, H. (2019). *AnnotationForge: Tools for building SQLite-based annotation data packages.*

Carneiro, M. O., Russ, C., Ross, M. G., Gabriel, S. B., Nusbaum, C., & DePristo, M. A. (2012). Pacific biosciences sequencing technology for genotyping and variation discovery in human data. *BMC Genomics*, *13*, 375. http://doi.org/10.1186/1471-2164-13-375

Casini, A., Olivieri, M., Petris, G., Montagna, C., Reginato, G., Maule, G., ... Cereseto, A. (2018). A highly specific SpCas9 variant is identified by in vivo screening in yeast. *Nature Biotechnology*, *36*(3), 265–271. http://doi.org/10.1038/nbt.4066

Chakrabarti, A. M., Henser-Brownhill, T., Monserrat, J., Poetsch, A. R., Luscombe, N. M., & Scaffidi, P. (2019). Target-Specific Precision of CRISPR-Mediated Genome Editing. *Molecular Cell*, *73*(4), 699–713.e6. http://doi.org/10.1016/j.molcel.2018.11.031

Chen, J. S., Dagdas, Y. S., Kleinstiver, B. P., Welch, M. M., Sousa, A. A., Harrington, L. B., ... Doudna, J. A. (2017). Enhanced proofreading governs CRISPR-Cas9 targeting accuracy. *Nature*, *550*(7676), 407–410. http://doi.org/10.1038/nature24268

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794). New York, NY, USA: Association for Computing Machinery. http://doi.org/10.1145/2939672.2939785

Chen, W., McKenna, A., Schreiber, J., Haeussler, M., Yin, Y., Agarwal, V., ... Shendure, J. (2019). Massively parallel profiling and predictive modeling of the outcomes of CRISPR/Cas9-mediated double-strand break repair. *Nucleic Acids Research*, *47*(15), 7989–8003. http://doi.org/10.1093/nar/gkz487

Chepelev, I., Wei, G., Tang, Q., & Zhao, K. (2009). Detection of single nucleotide variations in expressed exons of the human genome using RNA-Seq. *Nucleic Acids Research*, *37*(16), e106. http://doi.org/10.1093/nar/gkp507

Chuai, G., Ma, H., Yan, J., Chen, M., Hong, N., Xue, D., ... Liu, Q. (2018). DeepCRISPR: Optimized CRISPR guide RNA design by deep learning. *Genome Biology*, *19*(1), 80. http://doi.org/10.1186/s13059-018-1459-4

Church, D. M., Schneider, V. A., Graves, T., Auger, K., Cunningham, F., Bouk, N., ... Hubbard, T. (2011). Modernizing reference genome assemblies. *PLoS Biology*, *9*(7), e1001091. http://doi.org/10.1371/journal.pbio.1001091

Clarke, J., Wu, H.-C., Jayasinghe, L., Patel, A., Reid, S., & Bayley, H. (2009). Continuous base identification for single-molecule nanopore DNA sequencing. *Nature Nanotechnology*, *4*(4), 265–270. http://doi.org/10.1038/nnano.2009.12

Clement, K., Rees, H., Canver, M. C., Gehrke, J. M., Farouni, R., Hsu, J. Y., ... Pinello, L. (2019). CRISPResso2 provides accurate and rapid genome editing sequence analysis. *Nature Biotechnology*, *37*(3), 224–226. http://doi.org/10.1038/s41587-019-0032-3

Cleveland, W. S. (1993). *Visualizing data.* Summit (N.J.): Hobart Press.

Cock, P. J. A., Fields, C. J., Goto, N., Heuer, M. L., & Rice, P. M. (2010). The Sanger FASTQ file format for sequences with quality scores, and the

Solexa/Illumina FASTQ variants. *Nucleic Acids Research*, *38*(6), 1767–1771. http://doi.org/10.1093/nar/gkp1137

Conesa, A., Madrigal, P., Tarazona, S., Gomez-Cabrero, D., Cervera, A., McPherson, A., ... Mortazavi, A. (2016). A survey of best practices for RNA-seq data analysis. *Genome Biology*, *17*(1), 13. http://doi.org/10.1186/s13059-016-0881-8

Cong, L., Ran, F. A., Cox, D., Lin, S., Barretto, R., Habib, N., ... Zhang, F. (2013). Multiplex Genome Engineering Using CRISPR/Cas Systems. *Science (New York, N.Y.)*, *339*(6121), 819. http://doi.org/10.1126/science.1231143

Crosetto, N., Mitra, A., Silva, M. J., Bienko, M., Dojer, N., Wang, Q., ... Dikic, I. (2013). Nucleotide-resolution DNA double-strand break mapping by next-generation sequencing. *Nature Methods*, *10*(4), 361–365. http://doi.org/10.1038/nmeth.2408

Crusoe, M. R., Abeln, S., Iosup, A., Amstutz, P., Chilton, J., Tijani'c, N., ... Community, T. C. (2022). Methods included: Standardizing computational reuse and portability with the Common Workflow Language. *Communications of the ACM*, *65*(6), 54–63. http://doi.org/10.1145/3486897

Cunningham, F., Allen, J. E., Allen, J., Alvarez-Jarreta, J., Amode, M. R., Armean, I. M., ... Flicek, P. (2021). Ensembl 2022. *Nucleic Acids Research*, *50*(D1), D988. http://doi.org/10.1093/nar/gkab1049

Davis, M. J., Sehgal, M. S. B., & Ragan, M. A. (2010). Automatic, context-specific generation of Gene Ontology slims. *BMC Bioinformatics*, *11*(1), 498. http://doi.org/10.1186/1471-2105-11-498

DeLuca, D. S., Levin, J. Z., Sivachenko, A., Fennell, T., Nazaire, M.-D., Williams, C., ... Getz, G. (2012). RNA-SeQC: RNA-seq metrics for quality control and process optimization. *Bioinformatics (Oxford, England)*, *28*(11), 1530–1532. http://doi.org/10.1093/bioinformatics/bts196

Di Tommaso, P., Chatzou, M., Floden, E. W., Barja, P. P., Palumbo, E., & Notredame, C. (2017). Nextflow enables reproducible computational workflows. *Nature Biotechnology*, *35*(4), 316–319. http://doi.org/10.1038/nbt.3820

Dobin, A., Davis, C. A., Schlesinger, F., Drenkow, J., Zaleski, C., Jha, S., ... Gingeras, T. R. (2013). STAR: Ultrafast universal RNA-seq aligner. *Bioinformatics*, *29*(1), 15–21. http://doi.org/10.1093/bioinformatics/bts635

Dominguez Del Angel, V., Hjerde, E., Sterck, L., Capella-Gutierrez, S., Notredame, C., Vinnere Pettersson, O., ... Lantz, H. (2018). Ten steps to get started in Genome Assembly and Annotation. *F1000Research*, *7*, ELIXIR–148. http://doi.org/10.12688/f1000research.13598.1

Doudna, J. A., & Charpentier, E. (2014). The new frontier of genome engineering with CRISPR-Cas9. *Science*, *346*(6213), 1258096. http://doi.org/10.1126/science.1258096

Dozmorov, M. G., Adrianto, I., Giles, C. B., Glass, E., Glenn, S. B., Montgomery, C., ... Wren, J. D. (2015). Detrimental effects of duplicate reads and low complexity regions on RNA- and ChIP-seq data. *BMC Bioinformatics*, *16*(13), S10. http://doi.org/10.1186/1471-2105-16-S13-S10

Dudley, J. T., & Butte, A. J. (2009). A quick guide for developing effective bioinformatics programming skills. *PLoS Computational Biology*, *5*(12), e1000589. http://doi.org/10.1371/journal.pcbi.1000589

Durbin, R. M., Altshuler, D., Durbin, R. M., Abecasis, G. R., Bentley, D. R., Chakravarti, A., ... The Translational Genomics Research Institute. (2010). A map of human genome variation from population-scale sequencing. *Nature*, *467*(7319, 7319), 1061–1073. http://doi.org/10.1038/nature09534

Eid, J., Fehr, A., Gray, J., Luong, K., Lyle, J., Otto, G., ... Turner, S. (2009). Real-time DNA sequencing from single polymerase molecules. *Science (New York, N.Y.)*, *323*(5910), 133–138. http://doi.org/10.1126/science.1162986

Eisenstein, M. (2012). Oxford Nanopore announcement sets sequencing sector abuzz. *Nature Biotechnology*, *30*(4), 295–296. http://doi.org/10.1038/nbt0412-295

ENCODE Project Consortium. (2012). An integrated encyclopedia of DNA elements in the human genome. *Nature*, *489*(7414), 57–74. http://doi.org/10.1038/nature11247

Engström, P. G., Steijger, T., Sipos, B., Grant, G. R., Kahles, A., Rätsch, G., ... Bertone, P. (2013). Systematic evaluation of spliced alignment programs for RNA-seq data. *Nature Methods*, *10*(12), 1185–1191. http://doi.org/10.1038/nmeth.2722

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

FASTX-Toolkit. (n.d.). Retrieved November 16, 2023, from `http://hannonlab.cshl.edu/fastx_toolkit/`

Feil, R., Charlton, J., Bird, A. P., Walter, J., & Reik, W. (1994). Methylation analysis on individual chromosomes: Improved protocol for bisulphite genomic sequencing. *Nucleic Acids Research*, *22*(4), 695–696. Retrieved from `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC307863/`

Fiumara, M., Ferrari, S., Omer-Javed, A., Beretta, S., Albano, L., Canarutto, D., ... Naldini, L. (2024). Genotoxic effects of base and prime editing in human hematopoietic stem cells. *Nature Biotechnology*, *42*(6), 877–891. http://doi.org/10.1038/s41587-023-01915-4

Fu, Y., Wu, P.-H., Beane, T., Zamore, P. D., & Weng, Z. (2018). Elimination of PCR duplicates in RNA-seq and small RNA-seq using unique molecular identifiers. *BMC Genomics*, *19*(1), 531. http://doi.org/10.1186/s12864-018-4933-1

Gaj, T., Gersbach, C. A., & Barbas, C. F. (2013). ZFN, TALEN and CRISPR/Cas-based methods for genome engineering. *Trends in Biotechnology*, *31*(7), 397–405. http://doi.org/10.1016/j.tibtech.2013.04.004

Garc'ıa-Alcalde, F., Okonechnikov, K., Carbonell, J., Cruz, L. M., Götz, S., Tarazona, S., ... Conesa, A. (2012). Qualimap: Evaluating next-generation sequencing alignment data. *Bioinformatics*, *28*(20), 2678–2679. http://doi.org/10.1093/bioinformatics/bts503

Gasiunas, G., Barrangou, R., Horvath, P., & Siksnys, V. (2012). Cas9–crRNA ri-

bonucleoprotein complex mediates specific DNA cleavage for adaptive immunity in bacteria. *Proceedings of the National Academy of Sciences*, *109*(39), E2579–E2586. http://doi.org/10.1073/pnas.1208507109

Genome Reference Consortium. (n.d.). Retrieved November 29, 2023, from `https://www.ncbi.nlm.nih.gov/grc`

Gillespie, M., Jassal, B., Stephan, R., Milacic, M., Rothfels, K., Senff-Ribeiro, A., … D'Eustachio, P. (2022). The reactome pathway knowledgebase 2022. *Nucleic Acids Research*, *50*(D1), D687–D692. http://doi.org/10.1093/nar/gkab1028

Goodstadt, L. (2010). Ruffus: A lightweight Python library for computational pipelines. *Bioinformatics*, *26*(21), 2778–2779. http://doi.org/10.1093/bioinformatics/btq524

Goodwin, S., Gurtowski, J., Ethe-Sayers, S., Deshpande, P., Schatz, M. C., & McCombie, W. R. (2015). Oxford Nanopore sequencing, hybrid error correction, and de novo assembly of a eukaryotic genome. *Genome Research*, *25*(11), 1750–1756. http://doi.org/10.1101/gr.191395.115

Griffith, M., Walker, J. R., Spies, N. C., Ainscough, B. J., & Griffith, O. L. (2015). Informatics for RNA Sequencing: A Web Resource for Analysis on the Cloud. *PLOS Computational Biology*, *11*(8), e1004393. http://doi.org/10.1371/journal.pcbi.1004393

Grinsztajn, L., Oyallon, E., & Varoquaux, G. (2022, July 18). Why do tree-based models still outperform deep learning on tabular data? http://doi.org/10.48550/arXiv.2207.08815

Hennig, B. P., Velten, L., Racke, I., Tu, C. S., Thoms, M., Rybin, V., … Steinmetz, L. M. (2018). Large-Scale Low-Cost NGS Library Preparation Using a Robust Tn5 Purification and Tagmentation Protocol. *G3 (Bethesda, Md.)*, *8*(1), 79–89. http://doi.org/10.1534/g3.117.300257

Himes, B. E., Jiang, X., Wagner, P., Hu, R., Wang, Q., Klanderman, B., … Lu, Q. (2014). RNA-Seq Transcriptome Profiling Identifies CRISPLD2 as a Glucocorticoid Responsive Gene that Modulates Cytokine Function in Airway Smooth Muscle Cells. *PLoS ONE*, *9*(6), e99625. http://doi.org/10.1371/journal.pone.0099625

Hitz, B. C., Lee, J.-W., Jolanki, O., Kagda, M. S., Graham, K., Sud, P., … Michael, J. (n.d.). The ENCODE Uniform Analysis Pipelines.

Hsu, P. D., Lander, E. S., & Zhang, F. (2014). Development and Applications of CRISPR-Cas9 for Genome Engineering. *Cell*, *157*(6), 1262–1278. http://doi.org/10.1016/j.cell.2014.05.010

Hu, J. H., Miller, S. M., Geurts, M. H., Tang, W., Chen, L., Sun, N., … Liu, D. R. (2018). Evolved Cas9 variants with broad PAM compatibility and high DNA specificity. *Nature*, *556*(7699), 57–63. http://doi.org/10.1038/nature26155

Hu, J., Meyers, R. M., Dong, J., Panchakshari, R. A., Alt, F. W., & Frock, R. L. (2016). Detecting DNA double-stranded breaks in mammalian genomes by linear amplification–mediated high-throughput genome-wide translocation sequencing. *Nature Protocols*, *11*(5, 5), 853–871. http://doi.org/10.1038/nprot.2016.043

Huang, X., Li, X., Qin, P., Zhu, Y., Xu, S., & Chen, J. (2018). Technical Ad-

vances in Single-Cell RNA Sequencing and Applications in Normal and Malignant Hematopoiesis. *Frontiers in Oncology*, *8*. Retrieved from https://www.frontiersin.org/articles/10.3389/fonc.2018.00582

Huber, W., Carey, V. J., Gentleman, R., Anders, S., Carlson, M., Carvalho, B. S., ... Morgan, M. (2015). Orchestrating high-throughput genomic analysis with Bioconductor. *Nature Methods*, *12*(2), 115–121. http://doi.org/10.1038/nmeth.3252

iGenomes. (n.d.). Retrieved October 25, 2024, from https://support.illumina.com/sequencing/sequencing_software/igenome.html

Illumina. (2017). DNA Sequencing Methods Collection: An overview of recent DNA-seq publications featuring Illumina® technology, 146.

Illumina Stranded mRNA Prep | A clear view of the coding transcriptome. (n.d.). Retrieved December 1, 2023, from https://emea.illumina.com/products/by-type/sequencing-kits/library-prep-kits/stranded-mrna-prep.html

Ishino, Y., Shinagawa, H., Makino, K., Amemura, M., & Nakata, A. (1987). Nucleotide sequence of the iap gene, responsible for alkaline phosphatase isozyme conversion in Escherichia coli, and identification of the gene product. *Journal of Bacteriology*, *169*(12), 5429–5433. Retrieved from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC213968/

Ivanov, I. E., Wright, A. V., Cofsky, J. C., Aris, K. D. P., Doudna, J. A., & Bryant, Z. (2020). Cas9 interrogates DNA in discrete steps modulated by mismatches and supercoiling. *Proceedings of the National Academy of Sciences of the United States of America*, *117*(11), 5853–5860. http://doi.org/10.1073/pnas.1913445117

Jiang, F., & Doudna, J. A. (2017). CRISPR–Cas9 Structures and Mechanisms. *Annual Review of Biophysics*, *46*(1), 505–529. http://doi.org/10.1146/annurev-biophys-062215-010822

Jiang, J. J., & Conrath, D. W. (1997). Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. In *Proceedings of the 10th Research on Computational Linguistics International Conference* (pp. 19–33). Taipei, Taiwan: The Association for Computational Linguistics and Chinese Language Processing (ACLCLP). Retrieved from https://www.aclweb.org/anthology/O97-1002

Jinek, M., Chylinski, K., Fonfara, I., Hauer, M., Doudna, J. A., & Charpentier, E. (2012). A Programmable Dual-RNA–Guided DNA Endonuclease in Adaptive Bacterial Immunity. *Science*, *337*(6096), 816–821. http://doi.org/10.1126/science.1225829

Jones, S. K., Hawkins, J. A., Johnson, N. V., Jung, C., Hu, K., Rybarski, J. R., ... Finkelstein, I. J. (2021). Massively parallel kinetic profiling of natural and engineered CRISPR nucleases. *Nature Biotechnology*, *39*(1), 84–93. http://doi.org/10.1038/s41587-020-0646-5

Ju, J., Kim, D. H., Bi, L., Meng, Q., Bai, X., Li, Z., ... Turro, N. J. (2006). Four-color DNA sequencing by synthesis using cleavable fluorescent nucleotide reversible terminators. *Proceedings of the National Academy of Sciences of the United States of America*, *103*(52), 19635–19640. http://doi.org/10.1073/pnas.0609513103

Kanehisa, M., & Goto, S. (2000). KEGG: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Research*, *28*(1), 27–30. http://doi.org/10.1093/nar/28.1.27

Kim, D., Bae, S., Park, J., Kim, E., Kim, S., Yu, H. R., ... Kim, J.-S. (2015). Digenome-seq: Genome-wide profiling of CRISPR-Cas9 off-target effects in human cells. *Nature Methods*, *12*(3, 3), 237–243. http://doi.org/10.1038/nmeth.3284

Kim, D., & Kim, J.-S. (2018). DIG-seq: A genome-wide CRISPR off-target profiling method using chromatin DNA. *Genome Research*, *28*(12), 1894–1900. http://doi.org/10.1101/gr.236620.118

Knott, G. J., & Doudna, J. A. (2018). CRISPR-Cas guides the future of genetic engineering. *Science*, *361*(6405), 866–869. http://doi.org/10.1126/science.aat5011

Köster, J., & Rahmann, S. (2012). Snakemake–a scalable bioinformatics workflow engine. *Bioinformatics (Oxford, England)*, *28*(19), 2520–2. http://doi.org/10.1093/bioinformatics/bts480

Krysler, A. R., Cromwell, C. R., Tu, T., Jovel, J., & Hubbard, B. P. (2022). Guide RNAs containing universal bases enable Cas9/Cas12a recognition of polymorphic sequences. *Nature Communications*, *13*(1), 1617. http://doi.org/10.1038/s41467-022-29202-x

Kulmanov, M., Smaili, F. Z., Gao, X., & Hoehndorf, R. (2021). Semantic similarity and machine learning with ontologies. *Briefings in Bioinformatics*, *22*(4), bbaa199. http://doi.org/10.1093/bib/bbaa199

Lander, E. S., Linton, L. M., Birren, B., Nusbaum, C., Zody, M. C., Baldwin, J., ... The Wellcome Trust: (2001). Initial sequencing and analysis of the human genome. *Nature*, *409*(6822, 6822), 860–921. http://doi.org/10.1038/35057062

Landrum, M. J., Lee, J. M., Benson, M., Brown, G. R., Chao, C., Chitipiralla, S., ... Maglott, D. R. (2018). ClinVar: Improving access to variant interpretations and supporting evidence. *Nucleic Acids Research*, *46*(D1), D1062–D1067. http://doi.org/10.1093/nar/gkx1153

Landt, S., & Marinov, G. (2012). ChIP-seq guidelines and practices of the ENCODE and modENCODE consortia. *Genome ...*, 1813–1831. http://doi.org/10.1101/gr.136184.111.

Langmead, B., & Salzberg, S. L. (2012). Fast gapped-read alignment with Bowtie 2. *Nature Methods*, *9*(4), 357–359. http://doi.org/10.1038/nmeth.1923

Lawrence, M., Huber, W., Pag'es, H., Aboyoun, P., Carlson, M., Gentleman, R., ... Carey, V. J. (2013). Software for computing and annotating genomic ranges. *PLoS Computational Biology*, *9*(8), e1003118. http://doi.org/10.1371/journal.pcbi.1003118

Lazzarotto, C. R., Malinin, N. L., Li, Y., Zhang, R., Yang, Y., Lee, G., ... Tsai, S. Q. (2020). CHANGE-seq reveals genetic and epigenetic effects on CRISPR–Cas9 genome-wide activity. *Nature Biotechnology*, *38*(11, 11), 1317–1327. http://doi.org/10.1038/s41587-020-0555-7

Lee, J. K., Jeong, E., Lee, J., Jung, M., Shin, E., Kim, Y.-H., ... Kim, J.-S. (2018). Directed evolution of CRISPR-Cas9 to increase its specificity. *Nature Communications*, *9*(1), 3048. http://doi.org/10.1038/s41467-018-05477-x

Leenay, R. T., Aghazadeh, A., Hiatt, J., Tse, D., Roth, T. L., Apathy, R., ... Zou, J. (2019). Large dataset enables prediction of repair after CRISPR-Cas9 editing in primary T cells. *Nature Biotechnology*, *37*(9), 1034. http://doi.org/10.1038/s41587-019-0203-2

Lemos, B. R., Kaplan, A. C., Bae, J. E., Ferrazzoli, A. E., Kuo, J., Anand, R. P., ... Haber, J. E. (2018). CRISPR/Cas9 cleavages in budding yeast reveal templated insertions and strand-specific insertion/deletion profiles. *Proceedings of the National Academy of Sciences of the United States of America*, *115*(9), E2040–E2047. http://doi.org/10.1073/pnas.1716855115

Lessard, S., Francioli, L., Alfoldi, J., Tardif, J.-C., Ellinor, P. T., MacArthur, D. G., ... Canver, M. C. (2017). Human genetic variation alters CRISPR-Cas9 on- and off-targeting specificity at therapeutically implicated loci. *Proceedings of the National Academy of Sciences of the United States of America*, *114*(52), E11257–E11266. http://doi.org/10.1073/pnas.1714640114

Li, Bo, & Dewey, C. N. (2011). RSEM: Accurate transcript quantification from RNA-Seq data with or without a reference genome. *BMC Bioinformatics*, *12*(1), 323. http://doi.org/10.1186/1471-2105-12-323

Li, Biao, Krishnan, V. G., Mort, M. E., Xin, F., Kamati, K. K., Cooper, D. N., ... Radivojac, P. (2009). Automated inference of molecular mechanisms of disease from amino acid substitutions. *Bioinformatics*, *25*(21), 2744–2750. http://doi.org/10.1093/bioinformatics/btp528

Li, H. (2013, May 26). Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. http://doi.org/10.48550/arXiv.1303.3997

Li, H., & Durbin, R. (2009). Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics (Oxford, England)*, *25*(14), 1754–60. http://doi.org/10.1093/bioinformatics/btp324

Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., ... Durbin, R. (2009). The Sequence Alignment/Map format and SAMtools. *Bioinformatics*, *25*(16), 2078–2079. http://doi.org/10.1093/bioinformatics/btp352

Li, Q., Brown, J. B., Huang, H., & Bickel, P. J. (2011). Measuring reproducibility of high-throughput experiments. *The Annals of Applied Statistics*, *5*(3), 1752–1779. http://doi.org/10.1214/11-AOAS466

Li, X., Nair, A., Wang, S., & Wang, L. (2015). Quality Control of RNA-Seq Experiments. In E. Picardi (Ed.), *RNA Bioinformatics* (Vol. 1269, pp. 137–146). New York, NY: Springer New York. http://doi.org/10.1007/978-1-4939-2291-8_8

Liao, Y., Smyth, G. K., & Shi, W. (2013). The Subread aligner: Fast, accurate and scalable read mapping by seed-and-vote. *Nucleic Acids Research*, *41*(10), e108–e108. http://doi.org/10.1093/nar/gkt214

Liberzon, A., Birger, C., Thorvaldsd'ottir, H., Ghandi, M., Mesirov, J. P., & Tamayo, P. (2015). The Molecular Signatures Database (MSigDB) hallmark gene set collection. *Cell Systems*, *1*(6), 417. http://doi.org/10.1016/j.cels.2015.12.004

Lin, D. (1998). An Information-Theoretic Definition of Similarity, 9.

Lord, P. W., Stevens, R. D., Brass, A., & Goble, C. A. (2003). Investigating semantic similarity measures across the Gene Ontology: The relationship between sequence and annotation. *Bioinformatics*, *19*(10), 1275–1283. http://doi.org/10.1093/bioinformatics/btg153

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

Lowy-Gallego, E., Fairley, S., Zheng-Bradley, X., Ruffier, M., Clarke, L., Flicek, P., & Consortium, T. 1000. G. P. (2019). Variant calling on the GRCh38 assembly with the data from phase three of the 1000 Genomes Project. *Wellcome Open Research*, *4*, 50. http://doi.org/10.12688/wellcomeopenres.15126.2

Ma, A., McDermaid, A., Xu, J., Chang, Y., & Ma, Q. (2020). Integrative Methods and Practical Challenges for Single-cell Multi-omics. *Trends in Biotechnology*, *38*(9), 1007–1022. http://doi.org/10.1016/j.tibtech.2020.02.013

Maere, S., Heymans, K., & Kuiper, M. (2005). BiNGO: A Cytoscape plugin to assess overrepresentation of Gene Ontology categories in Biological Networks. *Bioinformatics*, *21*(16), 3448–3449. http://doi.org/10.1093/bioinformatics/bti551

Mali, P., Yang, L., Esvelt, K. M., Aach, J., Guell, M., DiCarlo, J. E., ... Church, G. M. (2013). RNA-Guided Human Genome Engineering via Cas9. *Science*, *339*(6121), 823–826. http://doi.org/10.1126/science.1232033

Margulies, M., Egholm, M., Altman, W. E., Attiya, S., Bader, J. S., Bemben, L. A., ... Rothberg, J. M. (2005). Genome sequencing in microfabricated high-density picolitre reactors. *Nature*, *437*(7057, 7057), 376–380. http://doi.org/10.1038/nature03959

Marinov, G. K., Williams, B. A., McCue, K., Schroth, G. P., Gertz, J., Myers, R. M., & Wold, B. J. (2014). From single-cell to cell-pool transcriptomes: Stochasticity in gene expression and RNA splicing. *Genome Research*, *24*(3), 496–510. http://doi.org/10.1101/gr.161034.113

Marioni, J. C., Mason, C. E., Mane, S. M., Stephens, M., & Gilad, Y. (2008). RNA-seq: An assessment of technical reproducibility and comparison with gene expression arrays. *Genome Research*, *18*(9), 1509–1517. http://doi.org/10.1101/gr.079558.108

Martin, F. J., Amode, M. R., Aneja, A., Austine-Orimoloye, O., Azov, A. G., Barnes, I., ... Flicek, P. (2023). Ensembl 2023. *Nucleic Acids Research*, *51*(D1), D933–D941. http://doi.org/10.1093/nar/gkac958

Martin, M. (2011). Cutadapt removes adapter sequences from high-throughput sequencing reads. *EMBnet.journal*, *17*, 10–12. Retrieved from `http://journal.embnet.org/index.php/embnetjournal/article/view/200`

Maxam, A., & Gilbert, W. (1977, February 1). A new method for sequencing DNA. http://doi.org/10.1073/pnas.74.2.560

Merico, D., Gfeller, D., & Bader, G. D. (2009). How to visually interpret biological data using networks. *Nature Biotechnology*, *27*(10), 921–924.

http://doi.org/10.1038/nbt.1567

Merico, D., Isserlin, R., & Bader, G. D. (2011). Visualizing Gene-Set Enrichment Results Using the Cytoscape Plug-in Enrichment Map. In G. Cagney & A. Emili (Eds.), *Network Biology: Methods and Applications* (pp. 257–277). Totowa, NJ: Humana Press. http://doi.org/10.1007/978-1-61779-276-2_12

Method of the Year 2019: Single-cell multimodal omics. (2020). *Nature Methods*, *17*(1), 1. http://doi.org/10.1038/s41592-019-0703-5

Meyer, K. D., Saletore, Y., Zumbo, P., Elemento, O., Mason, C. E., & Jaffrey, S. R. (2012). Comprehensive Analysis of mRNA Methylation Reveals Enrichment in 3' UTRs and Near Stop Codons. *Cell*, *149*(7), 1635–1646. http://doi.org/10.1016/j.cell.2012.05.003

Mezlini, A. M., Smith, E. J. M., Fiume, M., Buske, O., Savich, G. L., Shah, S., … Brudno, M. (2013). iReckon: Simultaneous isoform discovery and abundance estimation from RNA-seq data. *Genome Research*, *23*(3), 519–529. http://doi.org/10.1101/gr.142232.112

Mi, H., Muruganujan, A., Ebert, D., Huang, X., & Thomas, P. D. (2019). PANTHER version 14: More genomes, a new PANTHER GO-slim and improvements in enrichment analysis tools. *Nucleic Acids Research*, *47*(D1), D419–D426. http://doi.org/10.1093/nar/gky1038

Mi, H., & Thomas, P. (2009). PANTHER Pathway: An ontology-based pathway database coupled with data analysis tools. *Methods in Molecular Biology (Clifton, N.J.)*, *563*, 123–140. http://doi.org/10.1007/978-1-60761-175-2_7

Milacic, M., Beavers, D., Conley, P., Gong, C., Gillespie, M., Griss, J., … D'Eustachio, P. (2024). The Reactome Pathway Knowledgebase 2024. *Nucleic Acids Research*, *52*(D1), D672–D678. http://doi.org/10.1093/nar/gkad1025

Mojica, F. J., Juez, G., & Rodr'ıguez-Valera, F. (1993). Transcription at different salinities of Haloferax mediterranei sequences adjacent to partially modified PstI sites. *Molecular Microbiology*, *9*(3), 613–621. http://doi.org/10.1111/j.1365-2958.1993.tb01721.x

Molla, K. A., & Yang, Y. (2020). Predicting CRISPR/Cas9-Induced Mutations for Precise Genome Editing. *Trends in Biotechnology*, *38*(2), 136–141. http://doi.org/10.1016/j.tibtech.2019.08.002

Mortazavi, A., Williams, B. A., McCue, K., Schaeffer, L., & Wold, B. (2008). Mapping and quantifying mammalian transcriptomes by RNA-Seq. *Nature Methods*, *5*(7, 7), 621–628. http://doi.org/10.1038/nmeth.1226

Nishiga, M., Liu, C., Qi, L. S., & Wu, J. C. (2022). The use of new CRISPR tools in cardiovascular research and medicine. *Nature Reviews. Cardiology*, *19*(8), 505–521. http://doi.org/10.1038/s41569-021-00669-3

Nuñez, J. K., Chen, J., Pommier, G. C., Cogan, J. Z., Replogle, J. M., Adriaens, C., … Weissman, J. S. (2021). Genome-wide programmable transcriptional memory by CRISPR-based epigenome editing. *Cell*, *184*(9), 2503–2519.e17. http://doi.org/10.1016/j.cell.2021.03.025

Nurk, S., Koren, S., Rhie, A., Rautiainen, M., Bzikadze, A. V., Mikheenko, A., …

Phillippy, A. M. (2022). The complete sequence of a human genome. *Science (New York, N.Y.)*, *376*(6588), 44–53. http://doi.org/10.1126/science.abj6987

Overbeek, M. van, Capurso, D., Carter, M. M., Thompson, M. S., Frias, E., Russ, C., … May, A. P. (2016). DNA Repair Profiling Reveals Non-random Outcomes at Cas9-Mediated Breaks. *Molecular Cell*, *63*(4), 633–646. http://doi.org/10.1016/j.molcel.2016.06.037

Pacesa, M., Lin, C.-H., Cl'ery, A., Saha, A., Arantes, P. R., Bargsten, K., … Jinek, M. (2022). Structural basis for Cas9 off-target activity. *Cell*, *185*(22), 4067–4081.e21. http://doi.org/10.1016/j.cell.2022.09.026

Packer, J., & Trapnell, C. (2018). Single-Cell Multi-omics: An Engine for New Quantitative Models of Gene Regulation. *Trends in Genetics: TIG*, *34*(9), 653–665. http://doi.org/10.1016/j.tig.2018.06.001

Papapetrou, E. P., & Sadelain, M. (2011). Generation of transgene-free human induced pluripotent stem cells with an excisable single polycistronic vector. *Nature Protocols*, *6*(9), 1251–1273. http://doi.org/10.1038/nprot.2011.374

Parekh, S., Ziegenhain, C., Vieth, B., Enard, W., & Hellmann, I. (2016). The impact of amplification on differential expression analyses by RNA-seq. *Scientific Reports*, *6*(1), 25533. http://doi.org/10.1038/srep25533

Park, P. J. (2009). ChIP-seq: Advantages and challenges of a maturing technology. *Nature Reviews. Genetics*, *10*(10), 669–680. http://doi.org/10.1038/nrg2641

Patro, R., Duggal, G., Love, M. I., Irizarry, R. A., & Kingsford, C. (2017). Salmon provides fast and bias-aware quantification of transcript expression. *Nature Methods*, *14*(4, 4), 417–419. http://doi.org/10.1038/nmeth.4197

Pesquita, C. (2017). Semantic Similarity in the Gene Ontology. In C. Dessimoz & N. Škunca (Eds.), *The Gene Ontology Handbook* (pp. 161–173). New York, NY: Springer. http://doi.org/10.1007/978-1-4939-3743-1_12

Pesquita, C., Faria, D., Falcão, A. O., Lord, P., & Couto, F. M. (2009). Semantic Similarity in Biomedical Ontologies. *PLoS Computational Biology*, *5*(7). http://doi.org/10.1371/journal.pcbi.1000443

Picelli, S., Björklund, A. K., Reinius, B., Sagasser, S., Winberg, G., & Sandberg, R. (2014). Tn5 transposase and tagmentation procedures for massively scaled sequencing projects. *Genome Research*, *24*(12), 2033–2040. http://doi.org/10.1101/gr.177881.114

Qualimap: Evaluating next-generation sequencing alignment data | Bioinformatics | Oxford Academic. (n.d.). Retrieved November 16, 2023, from `https://academic.oup.com/bioinformatics/article/28/20/2678/206551`

R Core Team. (2021). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from `https://www.R-project.org/`

Ramensky, V., Bork, P., & Sunyaev, S. (2002). Human non-synonymous SNPs: Server and survey. *Nucleic Acids Research*, *30*(17), 3894–3900. Retrieved from `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC137415/`

Ramsköld, D., Luo, S., Wang, Y.-C., Li, R., Deng, Q., Faridani, O. R., … Sandberg, R. (2012). Full-length mRNA-Seq from single-cell levels of RNA and individual circulating tumor cells. *Nature Biotechnology*, *30*(8), 777–782. http://doi.org/10.1038/nbt.2282

Rashmi, K. V., & Gilad-Bachrach, R. (2015, May 7). DART: Dropouts meet Multiple Additive Regression Trees. http://doi.org/10.48550/arXiv.1505.01866

Reijnders, M. J. M. F., & Waterhouse, R. M. (2021). Summary Visualizations of Gene Ontology Terms With GO-Figure! *Frontiers in Bioinformatics*, *1*. http://doi.org/10.3389/fbinf.2021.638255

Resnik, P. (1999). Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language. *Journal of Artificial Intelligence Research*, *11*, 95–130. http://doi.org/10.1613/jair.514

Ristoski, P., & Paulheim, H. (2016). RDF2Vec: RDF graph embeddings for data mining. In *The semantic web – ISWC 2016: 15th international semantic web conference, kobe, japan, october 17–21, 2016, proceedings, part I* (pp. 498–514). Berlin, Heidelberg: Springer-Verlag. http://doi.org/10.1007/978-3-319-46523-4_30

Ritchie, M. E., Phipson, B., Wu, D., Hu, Y., Law, C. W., Shi, W., & Smyth, G. K. (2015). Limma powers differential expression analyses for RNA-sequencing and microarray studies, 1–25.

Roberts, A., & Pachter, L. (2013). Streaming fragment assignment for real-time analysis of sequencing experiments. *Nature Methods*, *10*(1, 1), 71–73. http://doi.org/10.1038/nmeth.2251

Robinson, M. D., McCarthy, D. J., & Smyth, G. K. (2010). edgeR : A Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, *26*(1), 139–140. http://doi.org/10.1093/bioinformatics/btp616

Rothberg, J. M., Hinz, W., Rearick, T. M., Schultz, J., Mileski, W., Davey, M., … Bustillo, J. (2011). An integrated semiconductor device enabling non-optical genome sequencing. *Nature*, *475*(7356), 348–352. http://doi.org/10.1038/nature10242

Ruben Schep, Eva K. Brinkman, Christ Leemans, Xabier Vergara, Robin H. van der Weide, Ben Morris, … Bas van Steensel. (2021). Impact of chromatin context on Cas9-induced DNA double-strand break repair pathway balance. *Molecular Cell*, *81*(10), 2216. http://doi.org/10.1016/j.molcel.2021.03.032

Sadedin, S. P., Pope, B., & Oshlack, A. (2012). Bpipe: A tool for running and managing bioinformatics pipelines. *Bioinformatics*, *28*(11), 1525–1526. http://doi.org/10.1093/bioinformatics/bts167

Sanger, F., & Coulson, A. R. (1975). A rapid method for determining sequences in DNA by primed synthesis with DNA polymerase. *Journal of Molecular Biology*, *94*(3), 441–448. http://doi.org/10.1016/0022-2836(75)90213-2

Santiago Gisler, Joana P Goncalves, Waseem Akhtar, Johann de Jong, Alexey V. Pindyurin, Lodewyk F. A. Wessels, & Maarten van Lohuizen. (2019). Multiplexed Cas9 targeting reveals genomic location effects and gRNA-based staggered breaks influencing mutation efficiency. *Nature Communications*, *10*(1), 1598.

http://doi.org/10.1038/s41467-019-09551-w

Schlicker, A., Domingues, F. S., Rahnenführer, J., & Lengauer, T. (2006). A new measure for functional similarity of gene products based on Gene Ontology. *BMC Bioinformatics*, *7*(1), 302. http://doi.org/10.1186/1471-2105-7-302

Schmid-Burgk, J. L., Gao, L., Li, D., Gardner, Z., Strecker, J., Lash, B., & Zhang, F. (2020). Highly Parallel Profiling of Cas9 Variant Specificity. *Molecular Cell*, *78*(4), 794–800.e8. http://doi.org/10.1016/j.molcel.2020.02.023

Schmieder, R., & Edwards, R. (2011). Quality control and preprocessing of metagenomic datasets. *Bioinformatics*, *27*(6), 863–864. http://doi.org/10.1093/bioinformatics/btr026

Schriml, L. M., Arze, C., Nadendla, S., Chang, Y.-W. W., Mazaitis, M., Felix, V., … Kibbe, W. A. (2012). Disease Ontology: A backbone for disease semantic integration. *Nucleic Acids Research*, *40*(D1), D940–D946. http://doi.org/10.1093/nar/gkr972

Scott, D. A., & Zhang, F. (2017). Implications of human genetic variation in CRISPR-based therapeutic genome editing. *Nature Medicine*, *23*(9), 1095–1101. http://doi.org/10.1038/nm.4377

Shade, A., & Teal, T. K. (2015). Computing Workflows for Biologists: A Roadmap. *PLoS Biology*, *13*(11), e1002303. http://doi.org/10.1371/journal.pbio.1002303

Shannon, P., Markiel, A., Ozier, O., Baliga, N. S., Wang, J. T., Ramage, D., … Ideker, T. (2003). Cytoscape: A Software Environment for Integrated Models of Biomolecular Interaction Networks. *Genome Research*, *13*(11), 2498. http://doi.org/10.1101/gr.1239303

Shen, M. W., Arbab, M., Hsu, J. Y., Worstell, D., Culbertson, S. J., Krabbe, O., … Sherwood, R. I. (2018). Predictable and precise template-free CRISPR editing of pathogenic variants. *Nature*, *563*(7733), 646–651. http://doi.org/10.1038/s41586-018-0686-x

Shi, X., Shou, J., Mehryar, M. M., Li, J., Wang, L., Zhang, M., … Wu, Q. (2019). Cas9 has no exonuclease activity resulting in staggered cleavage with overhangs and predictable di- and tri-nucleotide CRISPR insertions without template donor. *Cell Discovery*, *5*, 53. http://doi.org/10.1038/s41421-019-0120-z

Shou, J., Li, J., Liu, Y., & Wu, Q. (2018). Precise and Predictable CRISPR Chromosomal Rearrangements Reveal Principles of Cas9-Mediated Nucleotide Insertion. *Molecular Cell*, *71*(4), 498–509.e4. http://doi.org/10.1016/j.molcel.2018.06.021

Should We Remove Duplicated Reads In Rna-Seq ? (n.d.). Retrieved November 16, 2023, from `https://www.biostars.org/p/55648/`

Sims, D., Sudbery, I., Ilott, N. E., Heger, A., & Ponting, C. P. (2014). Sequencing depth and coverage: Key considerations in genomic analyses. *Nature Reviews Genetics*, *15*(2), 121–132. http://doi.org/10.1038/nrg3642

Smaili, F. Z., Gao, X., & Hoehndorf, R. (2018). Onto2Vec: Joint vector-based representation of biological entities and their ontology-based annotations. *Bioinformatics*, *34*(13), i52–i60. http://doi.org/10.1093/bioinformatics/bty259

SMART-Seq mRNA LP and SMART-Seq mRNA. (n.d.). Retrieved December 1, 2023, from https://www.takarabio.com/products/next-generation-sequencing/rna-seq/ultra-low-input-rna-seq/smart-seq-mrna-lp-and-smart-seq-mrna

Solomon, M. J., Larsen, P. L., & Varshavsky, A. (1988). Mapping protein-DNA interactions in vivo with formaldehyde: Evidence that histone H4 is retained on a highly transcribed gene. *Cell*, *53*(6), 937–947. http://doi.org/10.1016/s0092-8674(88)90469-2

Stemmer, M., Thumberger, T., Sol Keyer, M. del, Wittbrodt, J., & Mateo, J. L. (2015). CCTop: An Intuitive, Flexible and Reliable CRISPR/Cas9 Target Prediction Tool. *PLOS ONE*, *10*(4), e0124633. http://doi.org/10.1371/journal.pone.0124633

Stuart, T., & Satija, R. (2019). Integrative single-cell analysis. *Nature Reviews. Genetics*, *20*(5), 257–272. http://doi.org/10.1038/s41576-019-0093-7

Supek, F., Bošnjak, M., Škunca, N., & Šmuc, T. (2011). REVIGO Summarizes and Visualizes Long Lists of Gene Ontology Terms. *PLOS ONE*, *6*(7), e21800. http://doi.org/10.1371/journal.pone.0021800

Supek, F., & Škunca, N. (2017). Visualizing GO Annotations. *Methods in Molecular Biology (Clifton, N.J.)*, *1446*, 207–220. http://doi.org/10.1007/978-1-4939-3743-1_15

Taheri-Ghahfarokhi, A., Taylor, B. J. M., Nitsch, R., Lundin, A., Cavallo, A.-L., Madeyski-Bengtson, K., … Maresca, M. (2018). Decoding non-random mutational signatures at Cas9 targeted sites. *Nucleic Acids Research*, *46*(16), 8417. http://doi.org/10.1093/nar/gky653

Tang, F., Barbacioru, C., Wang, Y., Nordman, E., Lee, C., Xu, N., … Surani, M. A. (2009). mRNA-Seq whole-transcriptome analysis of a single cell. *Nature Methods*, *6*(5), 377–382. http://doi.org/10.1038/nmeth.1315

Tarazona, S., García-Alcalde, F., Dopazo, J., Ferrer, A., & Conesa, A. (2011). Differential expression in RNA-seq: A matter of depth. *Genome Research*, *21*(12), 2213–2223. http://doi.org/10.1101/gr.124321.111

The Galaxy Community, Abueg, L. A. L., Afgan, E., Allart, O., Awan, A. H., Bacon, W. A., … Zoabi, R. (2024). The Galaxy platform for accessible, reproducible, and collaborative data analyses: 2024 update. *Nucleic Acids Research*, *52*(W1), W83–W94. http://doi.org/10.1093/nar/gkae410

The Gene Ontology Consortium, Aleksander, S. A., Balhoff, J., Carbon, S., Cherry, J. M., Drabkin, H. J., … Westerfield, M. (2023). The Gene Ontology knowledgebase in 2023. *GENETICS*, *224*(1), iyad031. http://doi.org/10.1093/genetics/iyad031

The Gene Ontology Resource: 20 years and still GOing strong. (2019). *Nucleic Acids Research*, *47*, D330–D338. http://doi.org/10.1093/nar/gky1055

Tian, S., Peng, S., Kalmbach, M., Gaonkar, K. S., Bhagwate, A., Ding, W., … Slager, S. L. (2019). Identification of factors associated with duplicate rate in ChIP-seq data. *PLOS ONE*, *14*(4), e0214723. http://doi.org/10.1371/journal.pone.0214723

Tischler, G., & Leonard, S. (2014). Biobambam: Tools for read pair collation based algorithms on BAM files. *Source Code for Biology and Medicine*, *9*, 13.

http://doi.org/10.1186/1751-0473-9-13

Trapnell, C., Roberts, A., Goff, L., Pertea, G., Kim, D., Kelley, D. R., ... Pachter, L. (2012). Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and Cufflinks. *Nature Protocols*, *7*(3), 562–78. http://doi.org/10.1038/nprot.2012.016

Tryka, K. A., Hao, L., Sturcke, A., Jin, Y., Wang, Z. Y., Ziyabari, L., ... Feolo, M. (2014). NCBI's Database of Genotypes and Phenotypes: dbGaP. *Nucleic Acids Research*, *42*, D975–D979. http://doi.org/10.1093/nar/gkt1211

Tsai, S. Q., Nguyen, N. T., Malagon-Lopez, J., Topkar, V. V., Aryee, M. J., & Joung, J. K. (2017). CIRCLE-seq: A highly sensitive in vitro screen for genome-wide CRISPR–Cas9 nuclease off-targets. *Nature Methods*, *14*(6, 6), 607–614. http://doi.org/10.1038/nmeth.4278

Tsai, S. Q., Zheng, Z., Nguyen, N. T., Liebers, M., Topkar, V. V., Thapar, V., ... Joung, J. K. (2015). GUIDE-seq enables genome-wide profiling of off-target cleavage by CRISPR-Cas nucleases. *Nature Biotechnology*, *33*(2), 187–197. http://doi.org/10.1038/nbt.3117

Vakulskas, C. A., Dever, D. P., Rettig, G. R., Turk, R., Jacobi, A. M., Collingwood, M. A., ... Behlke, M. A. (2018). A high-fidelity Cas9 mutant delivered as a ribonucleoprotein complex enables efficient gene editing in human hematopoietic stem and progenitor cells. *Nature Medicine*, *24*(8), 1216–1224. http://doi.org/10.1038/s41591-018-0137-0

Van Dijk, E. L., Jaszczyszyn, Y., & Thermes, C. (2014). Library preparation methods for next-generation sequencing: Tone down the bias. *Experimental Cell Research*, *322*(1), 12–20. http://doi.org/10.1016/j.yexcr.2014.01.008

Venter, J. C., Adams, M. D., Myers, E. W., Li, P. W., Mural, R. J., Sutton, G. G., ... Zhu, X. (2001). The sequence of the human genome. *Science (New York, N.Y.)*, *291*(5507), 1304–1351. http://doi.org/10.1126/science.1058040

Wagih, O. (2017). Ggseqlogo: A versatile R package for drawing sequence logos. *Bioinformatics (Oxford, England)*, *33*(22), 3645–3647. http://doi.org/10.1093/bioinformatics/btx469

Wang, J. Y., & Doudna, J. A. (2023). CRISPR technology: A decade of genome editing is only the beginning. *Science*, *379*(6629), eadd8643. http://doi.org/10.1126/science.add8643

Wang, J. Z., Du, Z., Payattakool, R., Yu, P. S., & Chen, C.-F. (2007). A new method to measure the semantic similarity of GO terms. *Bioinformatics*, *23*(10), 1274–1281. http://doi.org/10.1093/bioinformatics/btm087

Wang, L., Wang, S., & Li, W. (2012). RSeQC: Quality control of RNA-seq experiments. *Bioinformatics (Oxford, England)*, *28*(16), 2184–2185. http://doi.org/10.1093/bioinformatics/bts356

Wetterstrand, K. A. (2021, November). The Cost of Sequencing a Human Genome. Retrieved November 28, 2023, from `https://www.genome.gov/about-genomics/fact-sheets/Sequencing-Human-Genome-cost`

Wienert, B., Wyman, S. K., Richardson, C. D., Yeh, C. D., Akcakaya, P., Porritt, M. J., ... Corn, J. E. (2019). Unbiased detection of CRISPR off-targets in vivo using DISCOVER-Seq. *Science*, *364*(6437), 286–289. http://doi.org/10.1126/science.aav9023

Williams, A. G., Thomas, S., Wyman, S. K., & Holloway, A. K. (2014). RNA-seq Data: Challenges in and Recommendations for Experimental Design and Analysis. *Current Protocols in Human Genetics*, *83*(1). http://doi.org/10.1002/0471142905.hg1113s83

Xue, C., & Greene, E. C. (2021). DNA Repair Pathway Choices in CRISPR-Cas9-Mediated Genome Editing. *Trends in Genetics: TIG*, *37*(7), 639–656. http://doi.org/10.1016/j.tig.2021.02.008

Yan, M.-Y., Yan, H.-Q., Ren, G.-X., Zhao, J.-P., Guo, X.-P., & Sun, Y.-C. (2017). CRISPR-Cas12a-Assisted Recombineering in Bacteria. *Applied and Environmental Microbiology*, *83*(17), e00947–17. http://doi.org/10.1128/AEM.00947-17

Yan, W. X., Mirzazadeh, R., Garnerone, S., Scott, D., Schneider, M. W., Kallas, T., ... Crosetto, N. (2017). BLISS is a versatile and quantitative method for genome-wide profiling of DNA double-strand breaks. *Nature Communications*, *8*(1), 15058. http://doi.org/10.1038/ncomms15058

Yau, E. H., & Rana, T. M. (2018). Next-Generation Sequencing of Genome-Wide CRISPR Screens. *Methods in Molecular Biology (Clifton, N.J.)*, *1712*, 203–216. http://doi.org/10.1007/978-1-4939-7514-3_13

Ye, H.-J., Liu, S.-Y., Cai, H.-R., Zhou, Q.-L., & Zhan, D.-C. (2024, July 1). A Closer Look at Deep Learning on Tabular Data. http://doi.org/10.48550/arXiv.2407.00956

Yoo, A. B., Jette, M. A., & Grondona, M. (2003). SLURM: Simple Linux Utility for Resource Management. In D. Feitelson, L. Rudolph, & U. Schwiegelshohn (Eds.), *Job Scheduling Strategies for Parallel Processing* (pp. 44–60). Berlin, Heidelberg: Springer Berlin Heidelberg.

Yu, G., Li, F., Qin, Y., Bo, X., Wu, Y., & Wang, S. (2010). GOSemSim: An R package for measuring semantic similarity among GO terms and gene products. *Bioinformatics*, *26*(7), 976–978. http://doi.org/10.1093/bioinformatics/btq064

Yu, Guangchuang, Wang, L.-G., Han, Y., & He, Q.-Y. (2012). clusterProfiler: An R Package for Comparing Biological Themes Among Gene Clusters. *OMICS: A Journal of Integrative Biology*, *16*(5), 284–287. http://doi.org/10.1089/omi.2011.0118

Zeeberg, B. R., Liu, H., Kahn, A. B., Ehler, M., Rajapakse, V. N., Bonner, R. F., ... Pommier, Y. G. (2011). RedundancyMiner: De-replication of redundant GO categories in microarray and proteomics analysis. *BMC Bioinformatics*, *12*(1), 52. http://doi.org/10.1186/1471-2105-12-52

Zhao, J., Ohsumi, T. K., Kung, J. T., Ogawa, Y., Grau, D. J., Sarma, K., ... Lee, J. T. (2010). Genome-wide identification of polycomb-associated RNAs by RIP-seq. *Molecular Cell*, *40*(6), 939–953. http://doi.org/10.1016/j.molcel.2010.12.011

Zhao, S., & Zhang, B. (2015). A comprehensive evaluation of ensembl, RefSeq, and UCSC annotations in the context of RNA-seq read mapping and gene quantification. *BMC*

Genomics, *16*(1), 97. http://doi.org/10.1186/s12864-015-1308-8

Zhao, Z., Shang, P., Mohanraju, P., & Geijsen, N. (2023). Prime editing: Advances and therapeutic applications. *Trends in Biotechnology*, *41*(8), 1000–1012. http://doi.org/10.1016/j.tibtech.2023.03.004

Zhou, W., Li, W., Chen, J., Zhou, Y., Wei, Z., & Gong, L. (2021). Microbial diversity in full-scale water supply systems through sequencing technology: A review. *RSC Advances*, *11*(41), 25484–25496. http://doi.org/10.1039/D1RA03680G

Zook, J. M., Catoe, D., McDaniel, J., Vang, L., Spies, N., Sidow, A., ... Salit, M. (2016). Extensive sequencing of seven human genomes to characterize benchmark reference materials. *Scientific Data*, *3*, 160025. http://doi.org/10.1038/sdata.2016.25

Zook, J. M., McDaniel, J., Olson, N. D., Wagner, J., Parikh, H., Heaton, H., ... Salit, M. (2019). An open resource for accurately benchmarking small variant and reference calls. *Nature Biotechnology*, *37*(5), 561–566. http://doi.org/10.1038/s41587-019-0074-6

Zuo, Z., & Liu, J. (2016). Cas9-catalyzed DNA Cleavage Generates Staggered Ends: Evidence from Molecular Dynamics Simulations. *Scientific Reports*, *6*(1), 37584. http://doi.org/10.1038/srep37584

Zuris, J. A., Thompson, D. B., Shu, Y., Guilinger, J. P., Bessen, J. L., Hu, J. H., ... Liu, D. R. (2015). Cationic lipid-mediated delivery of proteins enables efficient protein-based genome editing in vitro and in vivo. *Nature Biotechnology*, *33*(1), 73–80. http://doi.org/10.1038/nbt.3081