Universitätsklinikum Hamburg-Eppendorf

Zentrum für Molekulare Neurobiologie Hamburg

Institut für Medizinische Systembiologie

Development of interactive software and AI-based algorithms for the analysis of biomedical data

Dissertation

zur Erlangung des Doktorgrades Dr. rer. biol. hum. / PhD an der Medizinischen Fakultät der Universität Hamburg.

vorgelegt von:

Daniel Sumner Magruder

Betreuer: Prof. Dr. Stefan Bonn

Hamburg 2021

Angenommen von der
Medizinischen Fakultät der Universität Hamburg am: 22.09.2021

Veröffentlicht mit Genehmigung der Medizinischen Fakultät der Universität Hamburg.

Prüfungsausschuss, der/die Vorsitzende:
Prof. Dr. Stefan Bonn

Prüfungsausschuss, zweite/r Gutachter/in:
Prof. Dr. Med. Christian Kubisch

Prüfungsausschuss, dritte/r Gutachter/in:
Prof. Dr. Thomas Oertner

Contents

List of Figures

List of Tables

1 Synopsis / Synopse

1.1 Introduction

Academia is paradoxically in both a golden age and a dark age. Never before has the presses for publications been running so rapidly [1, 2, 3]. Yet whilst the increase of research may initially seem ideal, this influx of scholarly capital has several downsides. Foremost, reproducibility is already an underserved and struggling aspect of academia [4, 5, 6, 7, 3, 8, 9, 10]. Coupled with big-data projects requiring state-of-the-art systems to super computers, some papers are untenable for such undertakings. Additionally, researchers already pressed under the "publish or perish" system can not keep pace with the outcoming novel developments [4]. Such may lead to duplicity, while nevertheless circumventing reproducibility studies [9]. Further, researchers unable to stay atop of the increase of relevant papers may result in their own work failing to incorporate and benefit from others contributions. Worse still is that those researchers with needed and novel developments may be totally overlooked amidst the deluge of other relevant or topical papers. While academia may never before have been so well funded and fueled, mere publication may be insufficient for relevancy [10]. Thus a greater importance of a manuscript's contents is how immediately and effortlessly such contributions can be accessed, used, and/or otherwise incorporated into one's own work. In other words the accessibility and reusability of a novel development may carry more weight than the impact-factor of the journal it was published in.

Consider a novel algorithm that is published along side its invocation in a repository. If the reader of said paper desired to try and reproduce the paper's results or apply the algorithm to their own work, they must navigate to the third-party site, download the requisite files, do any set-up / installations required (as well as those for dependencies), all before even getting to attempt running or using the algorithm. This process, whilst seemingly mundane to an software developer, can be riddled with unseen encumbrances especially for someone who is less versed in IT. Does the user have proper rights to download and install the algorithm? Is the user's system compatible with the provided implementation of the algorithm? Is the author's implementation of the algorithm integratable with the researcher's pipeline? For example, is the algorithm implemented in a proprietary language such as MatLab or in a more niche language like C#, whereas the researcher might use a more commonplace programming language like Python. Does the user's system have the requisite specifications to utilize the algorithm? For example, does the algorithm make heavy use (16+ Gbs) of RAM or does it require a powerful GPU? All of these factors and more are at play when considering incorporating novel research findings into one's own work.

A putative solution for the reader is author dependent. Rather than solely supplying an implementation of their work a user-friendly, well documented, fully feature progressive web application (PWA) can be deployed alongside the manuscript. By having the author additionally undertake the efforts to develop their contributions into a web application, many of the aforementioned hurdles of merely testing said contributions are removed for the reader. Foremost, a deployed PWA exists on the internet. While browsers such as Safari, Opera, Firefox, Chrome, etc may render aspects of the site differently, the PWA is generally browser agnostic. Further, web browsers are not tied a computer's operating system, thereby making deployed websites ideal for accessibility. For the same reason PWAs are not barred regarding device type (desktop, laptop, tablet, phone), as they can be accessed via said devices

internet browser [11]. Thus a web application immediately reduces the barrier to entry for accessibility to an author's work. Additionally, by having the author undertake responsibility for deploying their contributions in the form of an application, system requirements (e.g. a powerful GPU) can be handled by the author not the reader. Further still, administrative rights for downloading and installing third-party software are rendered moot as the deployed website is accessible to everyone via an internet connect and a browser. Therefore, if an author wishes to increase the ease of access to and reuse of their work an PWA alongside the manuscript and implementation can increase awareness and adoption of their contributions. Not to mention, it allows for an author to increase their work's impact by leveraging search engine optimization (SEO).

Unfortunately website development is a non trivial undertaking for the uninitiated. Technically, one could produce and deploy a static website built solely off of HTML. However HTML alone is insufficient for interactivity and the requisite features for allowing a user to upload / specify input and utilize a novel algorithm or tool. For such capabilities JaveScript (JS) is required which, if placed between an HTML script tag may work, still leaves the computational needs on the client rather than the developer. In such instances a backend is required to handle computationally demanding requests, that the results of which are returned to the client's frontend. Already, the list of things for the author to both learn and do to increase their paper's relevancy is ballooning. Thus need for the capacity of boilerplate applications capable of handling novel algorithms and tools is readily apparent and addressed in the methods section 1.2.

So far emphasis has been given to (1) the need for and value of increased accessibility to academia as a whole, (2) the putative solution to this need via PWAs, and (3) the costs from the viewpoint of the researchers who may choose to employ (2) it may be worthwhile to stress the value of (2) via the benefits to some who might use them. Foremost with the increase of broadband access (spurred in part

for economic reasons), mobile access to the internet also rises [12, 13, 14, 15]. While easy to overlook, there are many scenarios in which access to a desktop computer or server rack is utterly impractical. "Bulk" in general is unfavored, which may drive the adoption of wearable devices and increase in tablet devices. Therefore the development of a PWA, accessible via phone or tablet, would allow for individuals like medical doctors to utilize state-of-the-art software whilst tending to patients. While the centralization of computing resources is not a novel concept - most universities have some shared system(s) - not everyone is comfortable (or OS-dependent able) to secure shell connect to utilize these resources fully. Programming, although certainly useful, is not a requisite skill for many biologists.

Along this line of reasoning, the advent of the bioinformatician itself is testament to (1) the increasing complexity of biological experimentation understanding of which is needed to correct for bias, (2) the rapid expansion of artificial intelligence and layman-obtuse statistical techniques, and (3) the need for individuals to straddle two disciplines to further scientific research. For example, Single-Cell sequencing continues to both improve and increase in popularity, leading to large and complex datasets [16, 17, 18, 19, 20]. For such high-dimensional data, computationally expensive dimensional reduction techniques such as t-SNE have become both favored and utilized in post-stream analysis [21, 22, 23, 24, 25, 26, 27, 28]. Consequentially it may become the case where one's academic independence for the analysis, visualization, and exploration their own data is untenable. Thus the encapsulation of bioinfomatic analyses, tools, pipelines, etc into PWAs that anyone can leverage may also usher in new insights via the restoration of researcher independence to freely query their own work.

1.2 Methods

1.2.1 Libraries and Packages

An overview of libraries and packages utilized in the development of tools, applications, and novel methods can be found in table 1.1. Given both their reuse and number, two examples (frxt and d3sm) are highlighted in sections 1.2.1.1 and 1.2.1.2 respectively.

1.2.1.1 FRXT and vFRXT

A recurrent problem faced when developing a bioinformatic tool, be it an integrative platform or a hyper specific suite, is competing with user expectations for ease of use. If a tool returns a non-singular result, suddenly both search and filter functionality are expected and required. Compared to search engines like Google, seemingly everything is more convoluted. Part of what makes Google's search engine extremely intuitive is its "free text" input; in other words, users simply type what they wish to know as they wish to know it. While recreating Google's success for each and every application developed is beyond most research teams' means, the necessity of reusable, plug-and-play search and filter functionality has not gone unnoticed. To capitalizes on such a need companies (e.g. Algolia, Searchify, Elastic, Yext, Hawksearch, Clerk.io, etc) offering search engines for niche databases or web assets have abound. To this end a typescripted library for free text search and filtering (FRXT) and a Vue component-based library offering accessibility thereof (vFRXT) were developed.

FRXT focuses on a few key features, namely:

1. sufficiently-free user input from which logical filters might be extracted,
2. the application of conjunctive normal form (CNF) logical filtering of specified user requests, and
3. multisort (also known as TimSort) of SQL like data provided in the JSON

Name	Language	Applications	Purpose
frxt	TypeScript	EXT	free text search and filter functionality for SQL-esque data.
vfrxt	TypeScript	EXT	Reusable Vue components for frxt functionality.
v-focal	JaveScript	BED.AI, EXT	Focus user attention towards regions of the web application.
apoll	JavaScript	KNIT, SEA, BED.AI	polling an API until results received.
d3sm	JavaScript	Oasis2.0, SEA, EXT	extension of d3 utilizing closures for plot types.
tagahead	JavaScript	SEA	typeahead selection for multiple tags.
ankr	JavaScript	EXT	(re)-positioning of overlay elements that snap into place.
vue-ankr	JavaScript	EXT	Vue wrapper of of ankr functionality.
sfo	Python	BED.AI, scGANs	Extension of the File Observer from the Sacred library.
neumf	Python	EXT	Neural Network based matrix factorization.
mag	Python	BED.AI, KNIT, EXT	Magazine of utilities for machine learning across the NumPy, SciKit Learn, SciPy, and TensorFlow including multilabel metrics, network architectures, weight pruning and more.
ntai	Python	BED.AI, EXT	Extracting, encoding, and decoding FASTA sequences into tensors.
sil	Python	BED.AI, KNIT, EXT	Shared memory status indicator for parallel processing.
lrng	Cython	BED.AI	Labeled range manager for comparing BED files.
parpar	Python	BED.AI, EXT	Parallel parser for large files.
bedpy	Python	BED.AI, EXT	Classes corresponding to reading and manipulating BED files and entries thereof.
ksp	Python	KNIT	lightweight k-shortest paths package.
cnf	Python	EXT	Conjunctive Normal Form filtering functionality..
fio	Python	BED.AI, EXT	Feature input / output augmentation for TensorFlow serving API.
GRAND	Python	EXT	Graphs as Nested Dictionaries data type coupled with graph drawing functionality.

Table 1.1: Analytical, visual, user interaction / experience, and utility libraries and packages developed for and used in the contents of this cumulative thesis and external projects. Applications external to this dissertation are collectively refereed to as EXT. **Analytical**: sfo, neumf, mag, ntai, bedpy, ksp, cnf, and GRAND. **Visual**: d3sm, mag, and GRAND. **User interaction / experience**: vfrxt, v-focal, tagahead, and vue-ankr. **Utility**: frxt, apoll, ankr, ntai, sil, lrng, parpar, cnf, and fio.

format.

What does it mean for user input to be sufficiently-free? If reproducing Google's success on many, much smaller applications is untenable, then a sufficiently-free user inquiry is a user's input with a set of restraints. For FRXT the constraint is posed in the form of the user specifying following in order:

1. logic (optional, defaults to logical and),
2. field (the property to search upon),
3. function (optional, defaults to the identity function),
4. conditional (the comparison to be made for the value of the property in a database's record and user's request), and
5. value (what to compare the value of a record's property to, to see if it should be returned).

Provided text satisfying this constraint, it is sufficient to determine a logical filter, e.g. "the adjusted p-value is less than 0.005."

When chained together these logical filters provide users full access to any query they may wish to ask e.g. "the adjusted p-value is less than 0.005 or log2 fold change is greater than 10 and replicates is equal to five."

Naturally, such syntax is verbose. Therefore FRXT allows for logical filters to be entered all at once or in succession. Additionally, FRXT is based on text tokenization. Therefore phrases like "greater than or equal to" can be also entered as $\geq$ and still be acknowledged. The tokenization, known functions which are applicable (e.g. length of a property should its value be a list), etc are all configurable. Consequentially, once familiar with this paradigm, logically expressive yet user friendly search and filter functionality becomes readily available. Coupling the results of FRXT with data visualization greatly facilitates understanding of visually noisy charts (see figure 1.1).

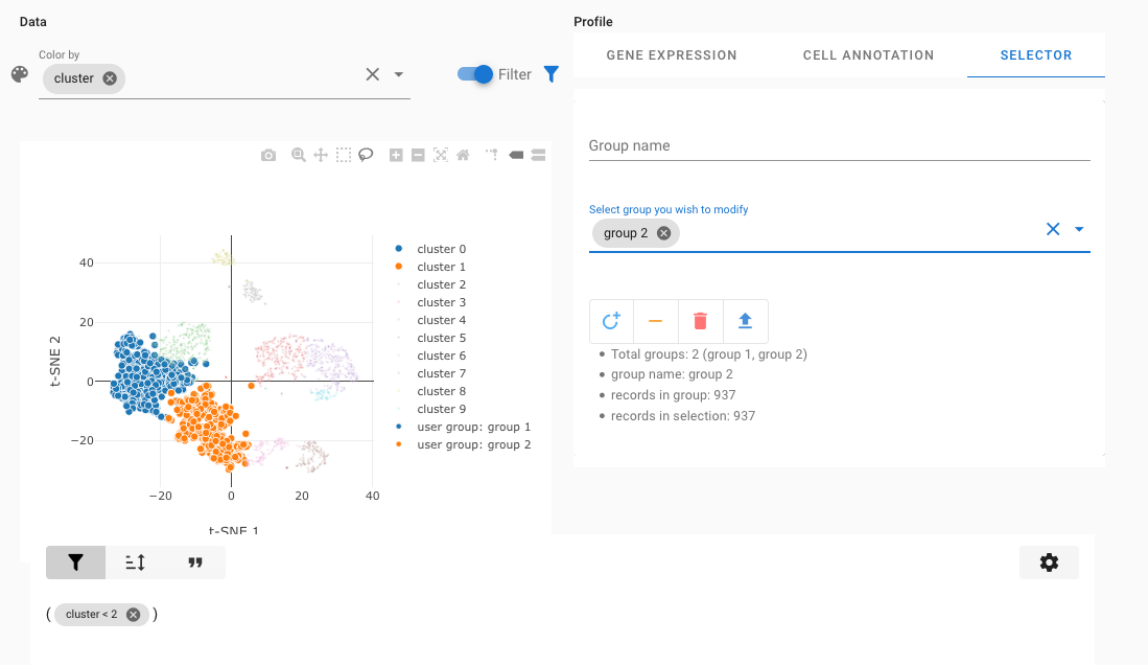
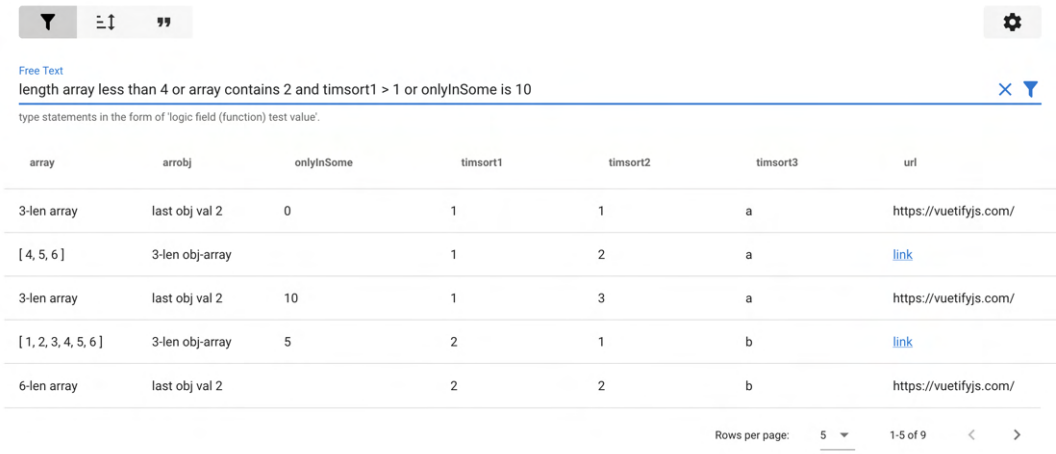


Figure 1.1: vFRXT used in conjunction with data visualization. Left: t-SNE of single cell experiment where cells are colored by cluster identification. Bottom: vFRXT search and filter functionality with the logical filter "(and) cluster is less than 2"

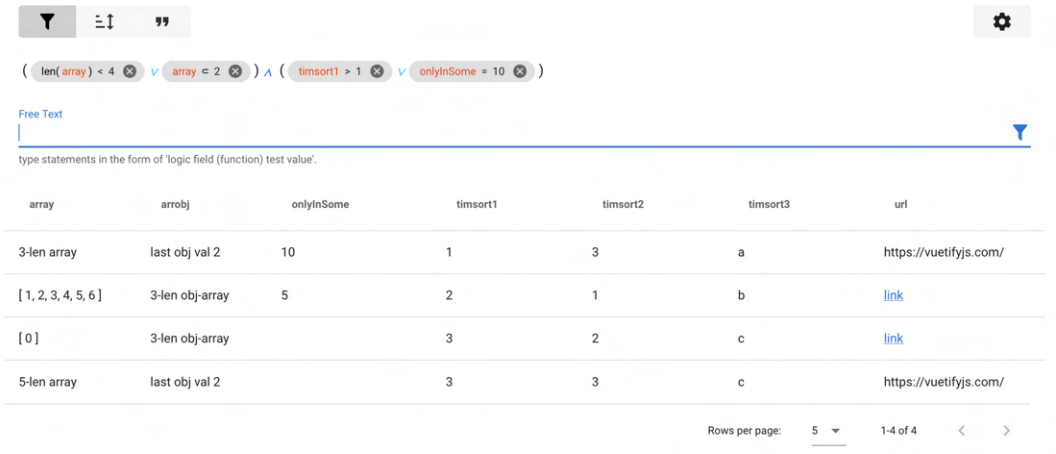
The vFRXT library wraps the FRXT functionality into a Vue component library for easy integration into web applications. Although every component is separate, practically the most prominently used component is the records table which contains all of the FRXT library's functionality. The basic table of figure 1.2 is produced with the following code from listing 1.1.

Listing 1.1: Single File Component file for the table in figure 1.2. A logical complete search engine is provided with minimal configuration.

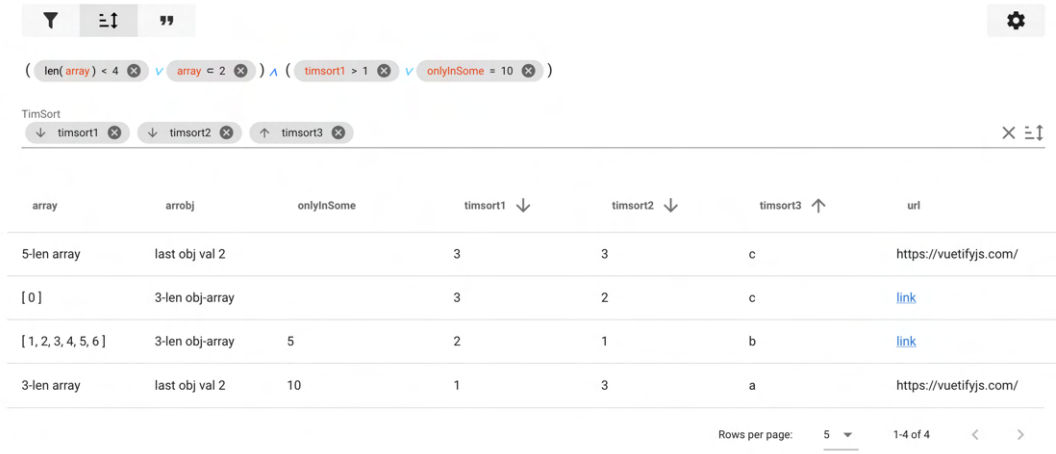
```
<body>
  <records-table :json="json">
  </records-table>
</body>
<script>
const json = {
  // ...
```



(a) vFRXT raw user input



(b) vFRXT logical filters



(c) TimSort

Figure 1.2: Example of vFRXT handling user input.

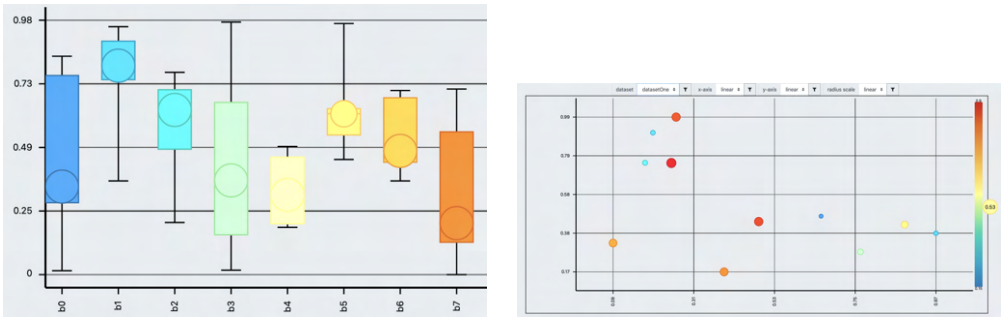
```
}
</script>
```

While having reliable, free, logically complete, plug-and-play search and filter functionality is readily applicable, care should be noted when concerning the data FRXT can search on i.e. "SQL-esque." SQL is a database language for relational database tables and a constraint of SQL is that each record (row in the table) has a unique identifier and that each property / field in a record (column of the table) exists in each row. SQL is strict in that the data types of a field in a record can be are limited (e.g. char, smallint, decafloat, blob, etc). While JSON is a valid data type for SQL, and an array of JSON objects can be a valid SQL constructed type, FRXT is not a database language. Rather, it relies on the premise that the input data is similar to that of a SQL record table. Of note is that FRXT is often used client side. Therefore the passing of large JSON objects to client for FRXT to then sort and filter upon may be undesirable.

1.2.1.2 d3sm

Data Driven Documents (D3) - available at https://d3js.org/ - is a lightweight JavaScript, closure based library for bringing data to life. As the specifics of interactivity can change vastly from project to project, even if the underlying data is similar, it is not uncommon for such visualization code to be more "one-off." d3sm (https://-sumneuron.gitlab.io/d3sm) provides some of the fundamental chart types such as bar, box-and-whisker, violin, scatter, etc (see figure 1.3). Additionally d3sm covers some niche charts like UpSet plots (see figure 1.4). All charts have "baked-in' functionality such as tooltips, opacity changes, etc (see figure 1.5).

d3sm is not a high level library. Rather it is intermediary, providing reusable building blocks. This middleware level of functionality is apparent in listing 1.2, where a general chart object is set up. Thereafter, various chart components are filled in to build the desired chart e.g. axes, a legend, the lasso widget, etc. This approach



(a) A box-and-whisker chart produced by d3sm.

(b) An interactive scatter chart produced by d3sm with dynamic axes and lasso selection configured.

Figure 1.3: Example of d3sm core chart types.

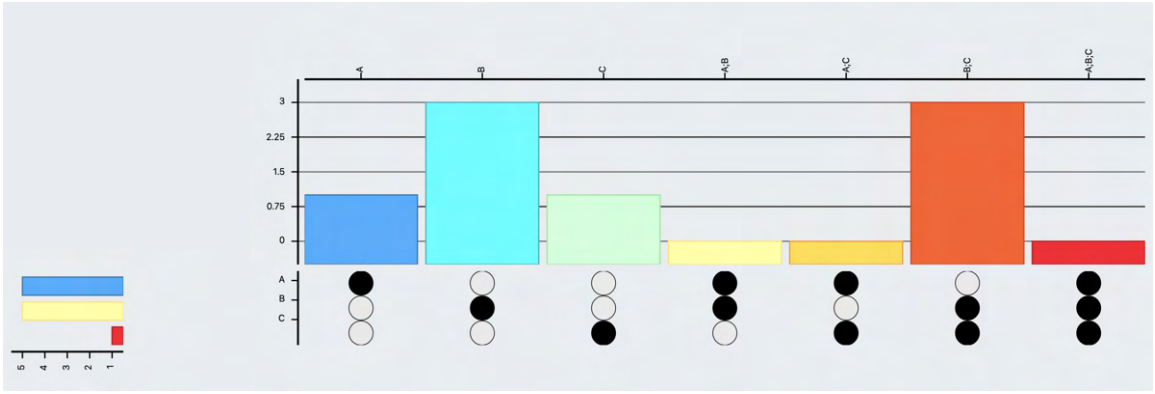
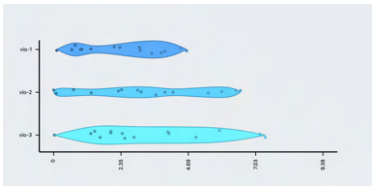


Figure 1.4: Example of one of d3sm's more niche chart types: an UpSet chart. The circular indicator grid assists in navigating this reinterpretation of the venn diagram. Vertically there is one circle per set, and those marked with a darker color specify the exclusive region for the intersection of sets e.g. one marked circle are unique elements to a set while all marked circles are the elements shared by all sets. The bar chart above the indicator grid reveals the cardinality of the specified venn diagram region. The bar chart left of the indicator grid specifies the cardinality of a given set.

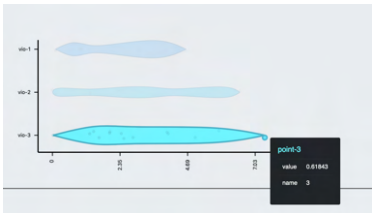
balances reuse with highly niche nature of interactive visualizations, therefore it is included in both SEA (section 1.3.1.3) and Oasis2.0 (section 1.3.1.4).

Listing 1.2: Code that produces the resizable scatter chart in figure 1.3.

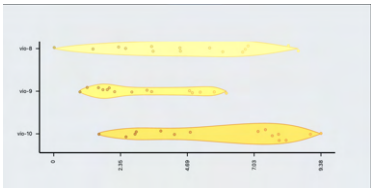
```
function scatterPlot( selection ) {
  var data, namespace, chart, xAxis, yAxis,
  dataSelect, legend, lasso, lassoWidget

  // define setters / getters
  plot.data = function(_){return arguments.length ? (data = _, plot) : data; };
  // ...

  function plot() {
```



(a) A violin plot produced by d3sm using kernel smoothing for the density and points are jittered within the domain.



(b) Tooltips are automatically produced both for the points inside a violin as well as for the violin itself.



(c) Scrolling over the interactive region reveals more of the plot to help ensure that the entire chart can fit within the monitor without being shrunk to unusable proportions.

Figure 1.5: Example of d3sm baked in features for a simple violin plot

```
  // set selections, data and scales
    var selections = d3sm.utils.misc.setupStandardChartContainers(
      selection, namespace, ...
    ),
    currentKeys, xScale, yScale, rScale, currentData

    // configure chart, axes, legend and lasso
    chart = d3sm.charts.scatter(chartSelection)
    //.setOptionOne(...).setOptionTwo(...)...
    chart()

    yAxis = d3sm.axis(yAxisSelection)//.setOptionOne(...)...
    yAxis()

    xAxis = d3sm.axis(xAxisSelection)//.setOptionOne(...)...
    xAxis()

    legend = d3sm.legends.numeric(legendSelect)//.setOptionOne(...)...
    legend()

    lasso = d3sm.aux.lasso( )
    //.setOptionOne(...)...
    lasso()
  }
  return plot
}
```

1.2.2 Reusable Schema

As stated in the introduction, development of a reusable template PWA for novel algorithms and tools are a vested interest for researchers. To that end a generalized deployment schema was produced. For the algorithm / tool in question, a trained TensorFlow model was selected, but is readily swappable according to the developer's

needs. The selection of a trained TensorFlow model is motivated by the facts that 1. TensorFlow is widely popular neural network library, thus this schema is ready-made for users thereof 2. State-of-the-art neural networks for biological purposes are hindered by accessibility (e.g. the user's computer limitations, the user's IT knowledge for downloading and running a trained model) 3. Such tools may become cornerstones in other and / or larger pipelines for which a production solution is needed. Indeed, this schema was promptly applied to both SCADEN (section 1.3.1.2) and BED.AI (section 4.1).

This generalized deployment schema has three core constituent parts:

1. the frontend: what the user interacts with,
2. the tool: what is being published, and
3. the backend: middleware for scheduling and routing frontend requests to the tool and the results from the tool back to the frontend.

1.2.2.1 Frontend

Frontend, or the client side of the application, is comprised generally of three main languages HTML (the content), Cascading Style Sheets (aka CSS, the style), and JS (the interactivity). While still feasible to produce a frontend using only these three programming languages without a single dependency, due to browser-dependent rendering of CSS and varying supported features, at the very least CSS frameworks like Bootstrap, Tailwind, Burma, are employed to ensure consistency. Many of these frameworks (e.g. Bootstrap) have extended to include small amounts of JavaScript as well for some ready made components (e.g. expansion panels). This terminology, components, are now used to describe self-contained aspects of the frontend. For example a button might be a component which contains the HTML (button tag), the CSS for styling the button, and the JS for what happens when the user clicks the button. While all of the code (HTML, CSS, JS) for such a button could exist sprawled

across a large HTML file, the practice of developing components as standalone parts of the frontend was formalized in 2013 when React was first released. Since then, component-based web development dominates the frontend landscape with alternative frameworks such as Angular, Svelte, and Vue learning from their predecessors' shortcomings. As both development and testing of components is easier to do, just as Bootstrap CSS frameworks exists, now too are component-based libraries built on top of them readily found. Collectively, component-based frameworks and their optional styled-component libraries make frontend development decidedly more streamlined than before. Further still, frameworks built atop these component-based frameworks such as Nextjs and Nuxtjs make deployment of component-based websites significantly easier. For example, applications built with Nuxtjs can be PWA compliant out of the box. To this end the frontend component of this reusable schema is a NGINX deployed Vue application powered by Nuxtjs making use of the Vuetify - a Material Design based - component library.

1.2.2.2 Backend

A core requisite met by decoupling the frontend from the backend is allowing the developer to deploy on a system capable of meeting their users' needs. What functionally is needed is a scheduler that accepts user requests, enqueues them, and has them processed by the tool before returning the results to the user. Such requirements describe a task-based API. Task-based APIs have a several of benefits, including:

1. tasks can be enqueued from a light backend hosted by a provider, while the tasks themselves are run elsewhere,
2. allows for easy scaling by increasing workers,
3. tasks get around timeout issues (for larger / longer processess),
4. tasks allow for hooks to be added (before queued, queued, etc), and
5. allows submitted task to be immediately checked for proper input.

For these reasons the backend component of this schema is a lightweight Flask API with a queuing system provided by Redis to ensure tasks are properly submitted.

1.2.2.3 Tool

As aforementioned this schema is setup around a trained TensorFlow neural network. However, the task-based API is capable of accepting any function, thereby making this schema readily adaptable to the author's needs.

Together these three aspects of the schema - frontend, backend, and tool - are dockerized for easier deployment and scaling. Since the frontend is built on a component-based library, as an author continues to develop their own custom components to suite their tastes, they are readily transferable to another application.

1.3 Results

1.3.1 Interactive Web Applications

1.3.1.1 KNIT

Understanding gene regulation networks, how one gene's expression impacts another's, is a prominent field of research for bioinformaticians due to its complexity. Foremost it demands the requisite knowledge of graph theory. Additionally, due to the number of genes, analytical tools and methods thereof require care for handling very large graphs [29, 30, 31, 32, 33, 34, 35, 36, 37, 38]. Even with improvements in modern computing, less than stellar consideration of memory allocation can easily overwhelm anything short of a computing cluster. Further still, making sense of secondary, tertiary, quaternary interactions can quickly become overbearing. To such ends an entire field of graph theory, graph drawing, exists for attempting to provide visual clarity to large graphs which may contain hundreds if not tens of thousands of vertices. While many creative and novel graph drawing layouts exists to help turn "hairball' graphs (graphs so large and dense practically no information

can be extracted from it visually) into something manageable, even learning to intuitively understand such layouts (e.g. BioFabric) can be a trying process in its own right [39]. Even relatively simple graphs, layout depending, can be hard to parse. In light of such difficulties some researchers have shifted focus towards the development of interactive web applications for the use of understanding regulation networks [40, 41, 29, 30, 31]. Some of these tools, like GeneMANIA and Pathway Commons, are based off the integrative cPath-2 gene regulation database (see table 1.2 for a list of its constitute databases). Other tools, such as STRING and STITCH may make use of some of the same underlying database of cPath, but generally follow the same paradigm; namely, a user provides a set of genes and the application, often with a hairball graph, returns the gene regulation network between any two genes in the provided set. As knowledge of regulatory pathways increases, such tools may become more convoluted for parsing how one gene of interest is regulated in relation to a set of others.

Therefore the web application Knock-In / Knock-Out Network Interaction Tools (KNIT), available at http://knit.ims.bio/ was developed. During development sil was utilized midst preprocessing, whilst ksp and mag assist with API requests on the backend and apoll is utilized on the client (see table 1.1 for an overview of libraries and packages). Like GeneMANIA and Pathway Commons, KNIT leverages the cPath database as its foundation for constructing the composite graph for the given user query [29, 30, 31, 32, 33, 34, 35]. Unlike these tools, however, KNIT utilizes Yen's k-best paths algorithm to identify which pathways to return [36]. The cost for traversal along the edges are proportional to publications found within the cPath integrative database supporting it [33, 34]. This allows users to specify their gene of interest, a set of genes they would like to know regulation pathways relating to the gene of interest, and the number of pathways to return. Additionally, unlike the majority of gene regulatory network tools, KNIT utilizes a hierarchical layered graph layout. While visually less dense than other layout algorithms, hierarchical layered drawings

Source	Date collected	Version	Pathways	Interactions
ChEBI Ontology	01-Jun-2017	152		
UniProtKB/Swiss-Prot	07-Jun-2017			
UniChem	19-Jun-2017			
Reactome	23-Jun-2017	61	16771	42349
PID	27-Jul-2015	final	14707	10526
PhosphoSite	15-Jun-2017		29007	16168
HumanCyc	2016	20	6669	5029
HPRD PSI-MI	13-Apr-2010	9	39826	9542
PANTHER Pathways	04-Jul-2016	3.4.1	5736	7850
DIP	05-Feb-2017		9025	4968
BioGRID	25-May-2017	3.4.149	394749	789498
IntAct	03-Jun-2017		247237	611820
IntAct Complex	03-Jun-2017		0	2515
BIND	15-Dec-2010		35451	72508
CORUM	17-Feb-2012		0	4401
MSigDB	Sep-2016	5.2	131239	13455
MiRTarBase	15-Sep-2015	6.1	337227	17395
DrugBank	01-Apr-2017	5.0.6	19555	16427
Recon X: Reconstruction of the Human Genome	2013	2.02	10816	8324
Toxicogenomics Database	06-Jun-2017	06-Jun-2017 release	602966	73712
KEGG	Jul-2011		3566	3349
Small Molecule Pathway Database	05-Jun-2016	2.0	4948	4958
INOH	22-Mar-2011	4.0	5433	17134
NetPath	Dec-2011		6351	3275
WikiPathways	29-Sep-2015		9756	9561

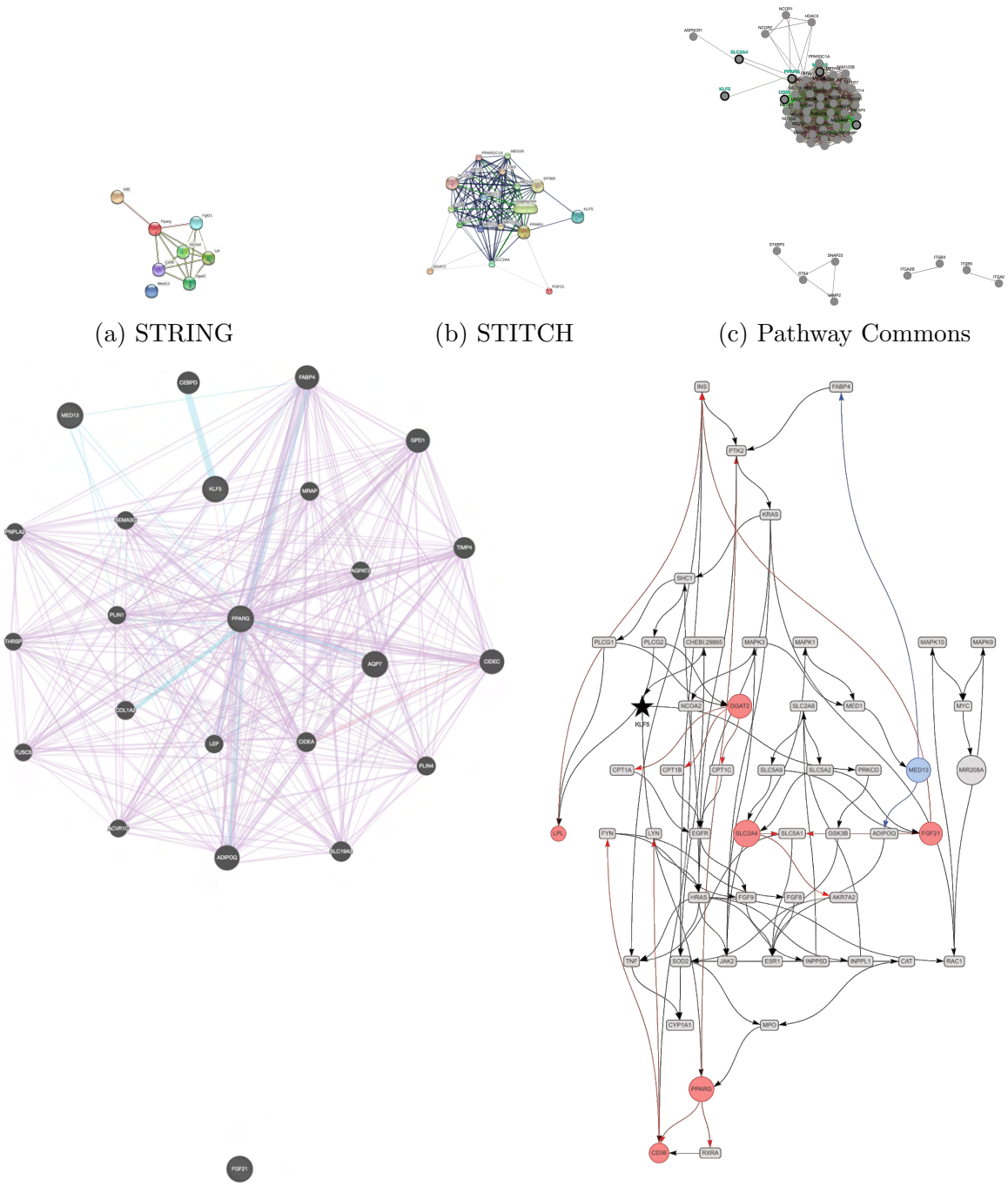
Table 1.2: Overview of sources from Pathway Commons (v9). Adapted from Pathway Commons data-sources.

are useful for seeing information flow (see figure 1.6).

1.3.1.2 SCADEN

The Single-cell Assisted Deconvolutional Network (SCADEN) - available at https://-scaden.ims.bio/ - is demonstrative of the increased access a web application can provide for a novel method (1.7). Depending on operating system, system rights, familiarity with package managers / installation, etc one may find the well documented SCADEN package (https://scaden.readthedocs.io) overwhelming. However, the terse and dry documentation which may deter researchers from utilizing the method is negated via immediate access to SCADEN via the PWA (see figure 1.8), developed with the reusable schema outlined in section 1.2.2.

Without the need for a GPU, understanding of deep neural networks or any installations, SCADEN allows users to generate gene expression profiles to deconvolve changes in gene expression from cell type [42]. As with most neural networks, SCA-



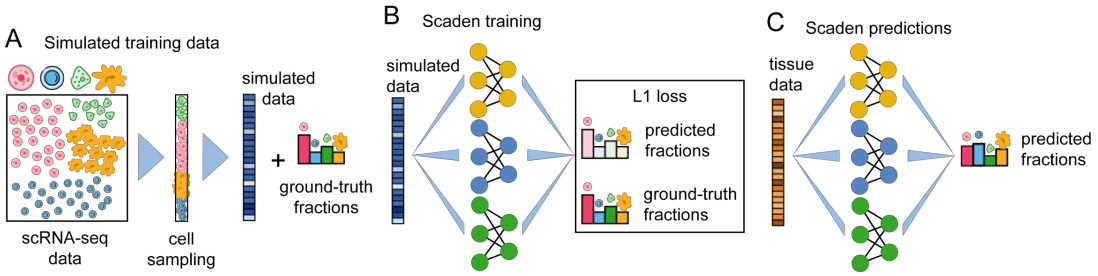
(a) STRING (b) STITCH (c) Pathway Commons

(d) GeneMania (e) KNIT

Figure 1.6: Visual output of KNIT, STRING, STITCH, GeneMANIA, and Pathway Commons provided the gene of interest, KLF5, and a gene list (Pparg1, Pparg2, Lpl, Cd36, and Dgat2).

DEN depends heavily on domain data which can be hard to find - tissue depending. However, SCADEN outperforms other deconvolution methods like MuSiC and CIBERSORTx [43, 44]. One of the largest drawbacks to SCADEN is the hardware

requirements, which are otherwise mitigated with the PWA.



(a) 135 Artificial bulk samples are generated by subsampling random cells from a scRNA - seq datasets 136 and merging their expression profiles.

(b) Model training and parameter optimization on 137 simulated tissue RNA - seq data by comparing cell fraction predictions to ground - truth cell 138 composition

(c) Cell deconvolution of real tissue RNA - seq data using SCADEN.

Figure 1.7: Overview of training data generation and cell type deconvolution with SCADEN.

1.3.1.3 SEA

The small RNA Expression Atlas (SEA) is a large scale integrative platform in combination with Oasis 2.0 [21, 45]. sRNA sequencing datasets from the public domain, via Oasis 2.0, underwent a standardized analysis pipeline to facilitate cross dataset comparisons. Yet merely standardizing and storing these datasets leaves much to be

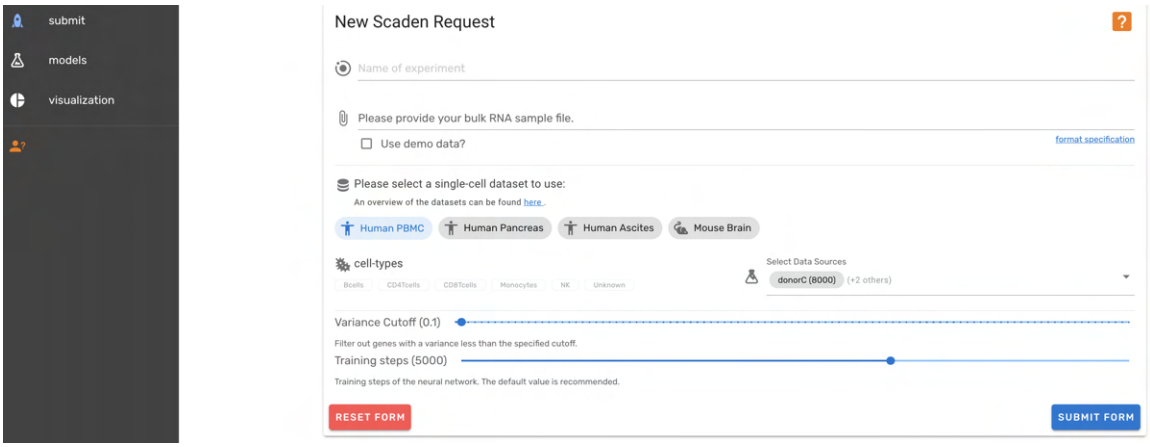


Figure 1.8: Example of the user friendly, no setup or installation required SCADEN forum powered by a component-based web application

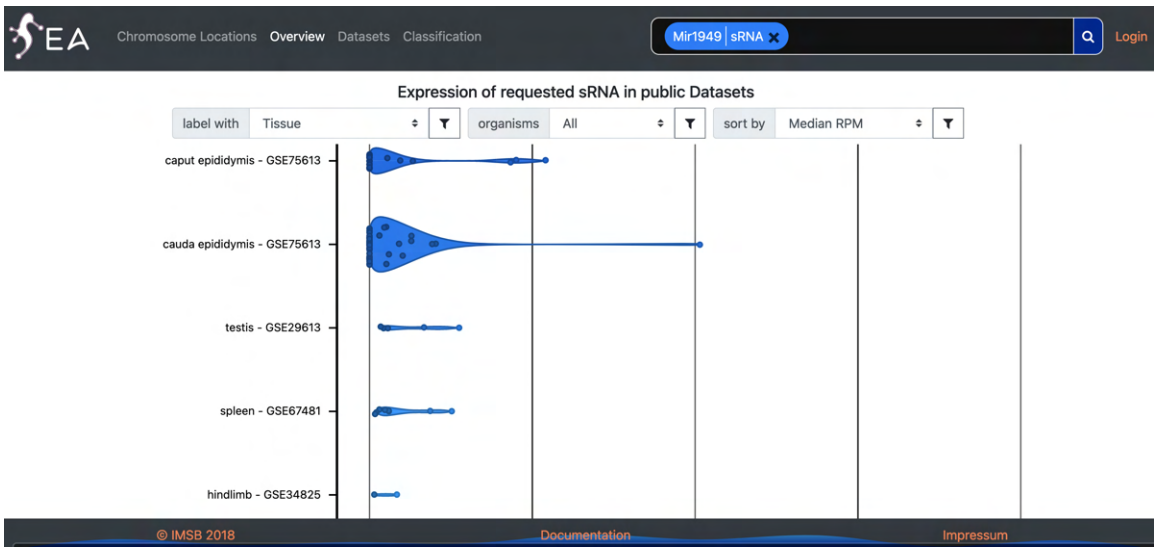


Figure 1.9: Interactive, filterable visual components of the SEA platform. SEA provides users a convenient an engaging way to search and browse datasets.

desired. With over 4000 samples across more than 350 datasets finding which ones to compare and inspecting them for outliers is a nontrivial task. To facilitate ease of use, SEA's search engine expresses complex behavior depending on the entities entered and combinations thereof e.g. if an ontology is specified or not. Further, results returned by SEA is not merely a list of datasets; rather SEA provides interactive visual insights into the datasets to help guide users (see figure 1.9). These insights include less readily supported but vital chart types like UpSet [46]. Additionally, SEA offers users the ability to select datasets from their search results and readily compare them as in the case of figure 1.10. The extensive functionality of SEA is supported by the apoll, d3sm, and tagahead packages (table 1.1).

1.3.1.4 Oasis2.0

Given the prominence of sRNA in disease (due to dysregulation), analysis thereof is highly relevant [47]. To this end Oasis 2.0 (the second major release of the Oasis suite) was developed to assist researchers in the processing of deep sequencing data e.g. sRNA detection, classification, etc. Oasis 2.0 promotes an improved classification module. As the classification module was previously based on Random Forest (RF),

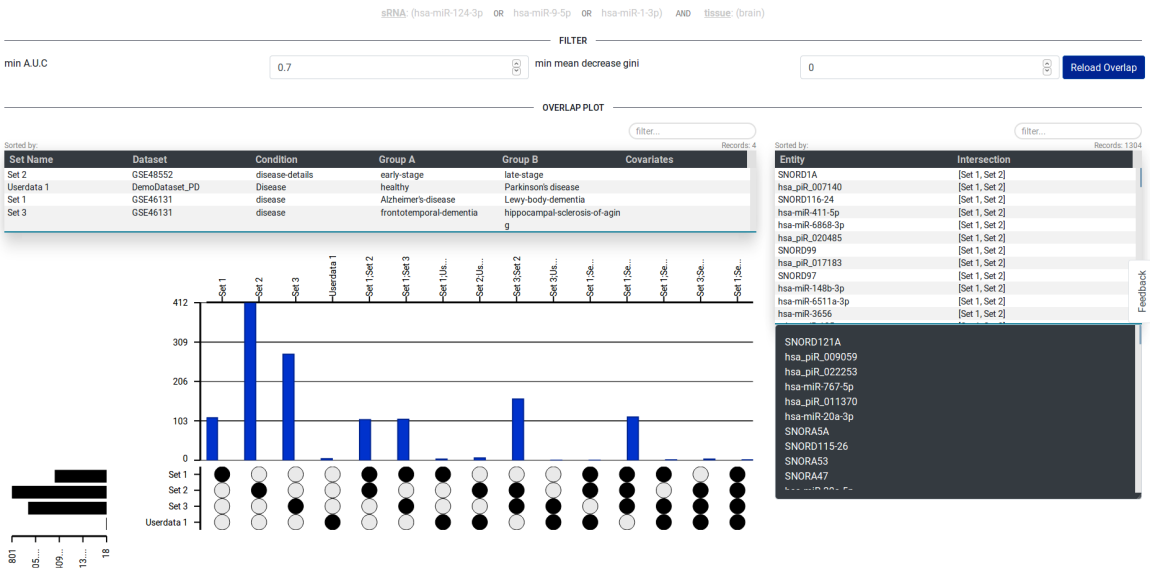


Figure 1.10: SEA Overlap Visual Analyses. Leftmost table: overview of the sRNA and tissue datasets that were searched and selected. Rightmost table: entities (here sRNA) that pass the specified the given filter parameters. Visual component: the UpSet plot for the corresponding venn diagram of the specified sets of elements.

improvements stem from feature pruning as well as better sampling [48]. Additionally, model evaluation metrics were made easier to evaluate via interactive and responsive charts. These charts laid the foundation for d3sm, which is an abstraction of these d3 powered scripts for improved reusability (see table 1.1).

1.3.2 Data Analysis Projects

Interactive progressive web applications are readily applicable to novel methods, algorithms, and tools. Not all publications are either as substantial as a novel methodology or yield insights from which an application can be built around. For example, many biology papers involve analysis of tissue samples of two or more conditions. These projects, which leverage analytical tools (such as those deployed as a PWA), are also fundamental to academia. This thesis includes two such projects. The first is related to COVID-19's scope of invasive behavior and the second assists in analyzing single cell data when few samples exists. A third unpublished project emphasizing how a new tool can become a web application is featured in section 4.1.

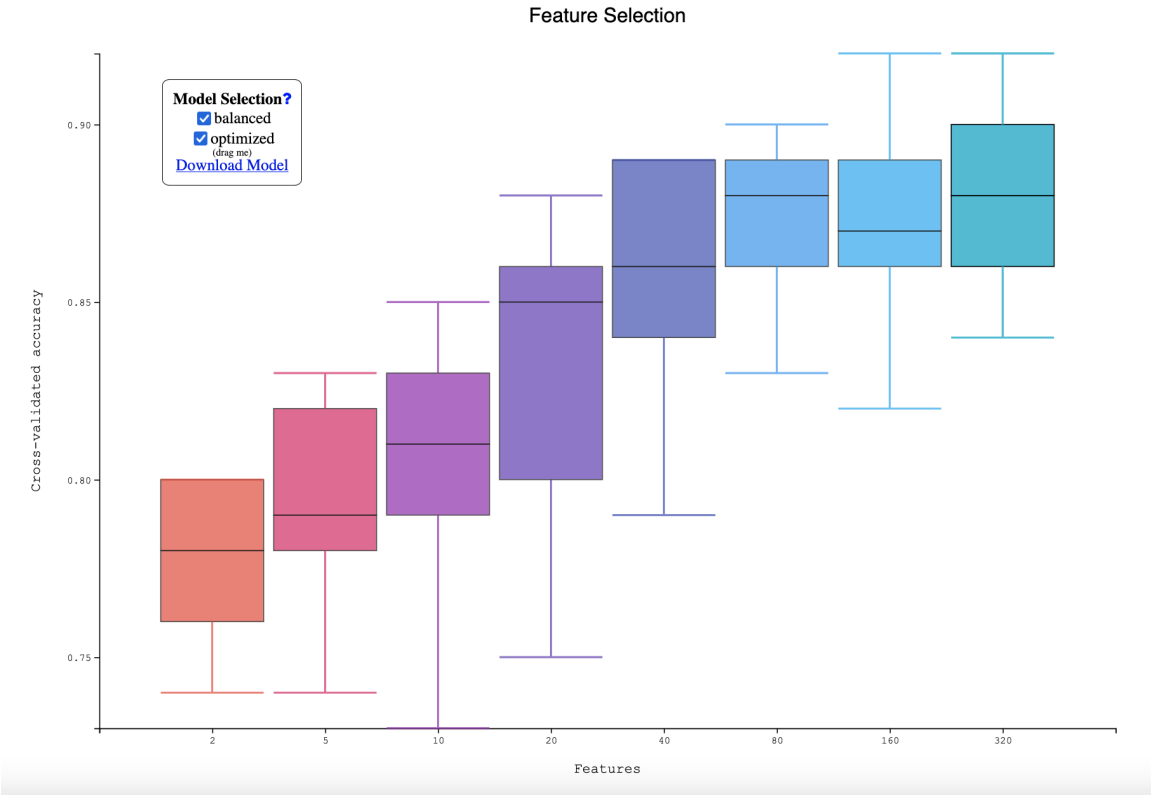


Figure 1.11: Oasis 2.0 classification output feature importance (cross validated prediction error) of random forest models. Random forests are trained by incrementally adding features according to their gini ranking.

1.3.2.1 COVID-19

It is well known that viruses are quite capable of traversing the blood-brain barrier and entering the brain. Inflammation of the central nervous system (CNS) is a serious condition. Additionally viral loads in the CNS during pregnancy has been linked to psychological disorders in children. With the ongoing pandemic of COVID-19 the question as to whether or not it also had the same penetrative capabilities was raised. Assessment of postmortem brain samples indicated that COVID-19 indeed can pass through the blood-brain barrier and be associated with CNS inflammation. Utilizing the scGANs and known marker genes, cell type clusters were annotated. The results of which suggest that neurons, glial and endothelial cells may contribute to COVID-19 infection.

1.3.2.2 scGANs

It is not uncommon for biological experiments to be sparse on samples. The reasons for this varies from ethics to funding. Nonetheless it is self explanatory how an increase of samples, be it a specific underserved sample type or in general, is beneficial. State-of-the-art machine learning techniques like variational autoencoders (VAE) and generative adversarial neural networks (GANs) have had record success at augmenting datasets. Given the ready application of in silico data augmentation, it is worthwhile to produce a proof-of-concept example. To this end GANs were applied to single cell data and demonstrated reliable results (scGANs). These results were deemed viable via four-fold authentication. Further, the GANs can be conditions (cscGANs) to produce specific cell types as needed.

1.4 Discussion

All of the developed methods, tools and applications - Oasis2.0, SEA, SCADEN, KNIT, BED.AI, scGANs - have similar pre-existing methodologies. Given the ex-

panse of domains in-depth comparisons to said methods and applications are left to their corresponding papers. Briefly, however, the novelty spans all aspects of the application development process (new method inclusive) and generally encompasses methodological improvement and breadth of features.

Naturally, the largest contrary point to the development of PWAs for novel software is that the author(s) of such software must not only invest the time into learning the requisite web development skills but also then devote a non-negligible portion of their research hours to a non-research based task. Whilst a large undertaking upfront, with the increasing support for component-based development producing PWAs has a lower barrier to entry than ever before. Additionally, the more components an author develops for their PWA, the more can be reused in subsequent applications. Thus progress web applications can become progressively easier to make. It may seem that aside from the time invested towards learning how to make and developing these applications, PWAs have purely upside for the author. However, with deployment the author also takes on the responsibility of fees and maintenance for having the site hosted. Further, the momentary advantages of having developed a PWA may be quickly dismissed should a publisher choose to support applications.

At the moment most journals, while accepting of web applications, have made little headway in the integration of interactive elements to their online platforms. There are several journals, e.g. Distill.pub, that have made an active effort in promoting online-only interactive articles. These articles, while affording authors greater flexibility, are still limited in comparison to deployment of one's own application. In the future, journals may take a more active approach in promoting research accessibility. However, as many journals still require payment to read a publication, such a future seems far off. In conclusion, encapsulating one's novel software inside of a PWA (alongside the publication and standalone repository when applicable) not only promotes accessibility but may improve reuseability and impact of one's research.

2 Bibliography

[1] Ewen Callaway. Will the pandemic permanently alter scientific publishing? *Nature*, 582:167–168, 6 2020.

[2] Karen White. Publications output: U.s. trends and international comparisons, 2019.

[3] Peder Olesen Larsen and Markus von Ins. The rate of growth in scientific publication and the decline in coverage provided by science citation index. *Scientometrics*, 84:575–603, 2010.

[4] Pietro Della Briotta Parolo, Raj Kumar Pan, Rumi Ghosh, Bernardo A. Huberman, Kimmo Kaski, and Santo Fortunato. Attention decay in science. *Journal of Informetrics*, 9:734–745, 10 2015.

[5] Marco Campani and Ruggero Vaglio. A simple interpretation of the growth of scientific/technological research impact leading to hype-type evolution curves. *Scientometrics*, 103:75–83, 4 2015.

[6] Lutz Bornmann and Rüdiger Mutz. Growth rates of modern science: A bibliometric analysis based on the number of publications and cited references. *Journal of the Association for Information Science and Technology*, 66:2215–2222, 11 2015.

[7] Andy Cockburn, Pierre Dragicevic, Lonni Besançon, and Carl Gutwin. Threats

of a replication crisis in empirical computer science. *Communications of the ACM*, 63:70–79, 8 2020.

[8] Daniele Fanelli and Vincent Larivière. Researchers' individual publication rate has not increased in a century. *PLoS ONE*, 11, 3 2016.

[9] Erik Von Elm, Greta Poglia, Bernhard Walder, and Martin R Tramè. Different patterns of duplicate publication an analysis of articles used in systematic reviews. *JAMA*, 2004.

[10] Frank Truth. Pay big to publish fast: Academic journal rackets. *Journal for Critical Education Policy Studies*, 2012.

[11] Sam Richard and Pete LePage. What are progressive web apps?, 2 2020.

[12] H. Gruber, J. Hätönen, and P. Koutroumpis. Broadband access in the eu: An assessment of future economic benefits. *Telecommunications Policy*, 38:1046–1058, 2014.

[13] Arnold Picot and Christian Wernick. The role of government in broadband access. *Telecommunications Policy*, 31:660–674, 11 2007.

[14] Nina Czernich, Oliver Falck, Tobias Kretschmer, and Ludger Woessmann. Broadband infrastructure and economic growth*. *The Economic Journal*, 121:505–532, 5 2011.

[15] Toni Janevski. Mobile broadband: Next generation mobile networks. *NGN Architectures, Protocols and Services*, pages 141–179, 3 2014.

[16] Sarah Aldridge and Sarah A. Teichmann. Single cell transcriptomics comes of age. *Nature Communications*, 11, 12 2020.

[17] Schubert C. Single-cell analysis: The deepest differences. *Nature*, 480:133–137, 12 2011.

[18] Renchao Chen, Xiaoji Wu, Lan Jiang, and Yi Zhang. Single-cell rna-seq reveals hypothalamic cell diversity. *Cell Reports*, 18:3227–3241, 3 2017.

[19] Jeffrey M. Perkel. Single-cell analysis enters the multiomics age. *Nature*, 595:614–616, 7 2021.

[20] Paul Datlinger, André F. Rendeiro, Thorina Boenke, Martin Senekowitsch, Thomas Krausgruber, Daniele Barreca, and Christoph Bock. Ultra-high-throughput single-cell rna sequencing and perturbation screening with combinatorial fluidic indexing. *Nature Methods*, 18:635–642, 6 2021.

[21] Raza-Ur Rahman, Anna-Maria Liebhoff, Vikas Bansal, Maksims Fiosins, Ashish Rajput, Abdul Sattar, Daniel S Magruder, Sumit Madan, Ting Sun, Abhivyakti Gautam, Sven Heins, Timur Liwinski, Jörn Bethune, Claudia Trenkwalder, Juliane Fluck, Brit Mollenhauer, and Stefan Bonn. Seaweb: the small rna expression atlas web application. *Nucleic Acids Research*, 48:D204–D219, 1 2020.

[22] Laurens Van Der Maaten. Accelerating t-SNE using Tree-Based Algorithms. *Journal of Machine Learning Research*, 15:1–21, 2014.

[23] Laurens Van Der Maaten and Geoffrey Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[24] Xiaojie Qiu, Yan Zhang, Shayan Hosseinzadeh, Dian Yang, Angela Pogson, Li Wang, Matt Shurtleff, Ruoshi Yuan, Song Xu, Yian Ma, Joseph Replogle, Spyros Darmanis, Ivet Bahar, Jianhua Xing, and Jonathan Weissman. Mapping transcriptomic vector fields of single cells. 2019.

[25] Van Hoan Do and Stefan Canzar. A generalization of t-sne and umap to single-cell multimodal omics. *Genome Biology 2021 22:1*, 22:1–9, 5 2021.

[26] CO Ciccolella, R Anno, R Halpert, J Spidlen, JE Snyder-Cappione, and AC Belkina. Automated optimized parameters for t-distributed stochastic neighbor em-

bedding improve visualization and analysis of large datasets. *Nature Communications*, 10:1–12, 12 2019.

[27] Pavlin G. Poličar, Martin Stražar, and Blaž Zupan. Embedding to reference t-sne space addresses batch effects in single-cell classification. *Machine Learning 2021*, pages 1–20, 8 2021.

[28] Dmitry Kobak and Philipp Berens. The art of using t-sne for single-cell transcriptomics. *Nature Communications 2019 10:1*, 10:1–14, 11 2019.

[29] Khalid Zuberi, Max Franz, Harold Rodriguez, Jason Montojo, Christian Tannus Lopes, Gary D. Bader, and Quaid Morris. GeneMANIA Prediction Server 2013 Update. *Nucleic Acids Research*, 41(W1):W115–W122, jul 2013.

[30] Jason Montojo, Khalid Zuberi, Harold Rodriguez, Gary D Bader, and Quaid Morris. GeneMANIA: Fast gene network construction and function prediction for Cytoscape. *F1000Research*, 3:153, 2014.

[31] Sara Mostafavi, Debajyoti Ray, David Warde-Farley, Chris Grouios, and Quaid Morris. GeneMANIA: a real-time multiple association network integration algorithm for predicting gene function. *Genome biology*, 9 Suppl 1(Suppl 1):S4, 2008.

[32] James Vlasblom, Khalid Zuberi, Harold Rodriguez, Roland Arnold, Alla Gagarinova, Viktor Deineko, Ashwani Kumar, Elisa Leung, Kamran Rizzolo, Bahram Samanfar, Luke Chang, Sadhna Phanse, Ashkan Golshani, Jack F. Greenblatt, Walid A. Houry, Andrew Emili, Quaid Morris, Gary Bader, and Mohan Babu. Novel function discovery with GeneMANIA: a new integrated resource for gene function prediction in Escherichia coli. *Bioinformatics*, 31(3):306–310, feb 2015.

[33] Ethan G Cerami, Gary D Bader, Benjamin E Gross, and Chris Sander. cPath: open source software for collecting, storing, and querying biological pathways. *BMC Bioinformatics*, 7(1):497, dec 2006.

[34] Ethan G Cerami, Gary D Bader, Benjamin E Gross, and Chris Sander. cPath: open source software for collecting, storing, and querying biological pathways. *BMC Bioinformatics*, 7(1):497, nov 2006.

[35] E. G. Cerami, B. E. Gross, E. Demir, I. Rodchenkov, O. Babur, N. Anwar, N. Schultz, G. D. Bader, and C. Sander. Pathway Commons, a web resource for biological pathway data. *Nucleic Acids Research*, 39(Database):D685–D690, jan 2011.

[36] J. Y. Yen. Finding the K Shortest Loopless Paths in a Network. *Management Science*, 17(11), 1971.

[37] Kozo Sugiyama, Shojiro Tagawa, and Mitsuhiko Toda. Methods for Visual Understanding of Hierarchical System Structures. *Transactions on Systems, Man, and cybernetics*, 11(2):109–125, 1981.

[38] Christian Bachmaier. A radial adaptation of the Sugiyama framework for visualizing hierarchical information. In *IEEE Transactions on Visualization and Computer Graphics*, 2007.

[39] William JR Longabaugh. Combing the hairball with biofabric: a new approach for visualization of large networks. *BMC Bioinformatics 2012 13:1*, 13:1–16, 10 2012.

[40] Michael Kuhn, Christian von Mering, Monica Campillos, Lars Juhl Jensen, and Peer Bork. STITCH: interaction networks of chemicals and proteins. *Nucleic acids research*, 36(Database issue):D684–8, jan 2008.

[41] Damian Szklarczyk, Andrea Franceschini, Stefan Wyder, Kristoffer Forslund, Davide Heller, Jaime Huerta-Cepas, Milan Simonovic, Alexander Roth, Alberto Santos, Kalliopi P Tsafou, Michael Kuhn, Peer Bork, Lars J Jensen, and Christian von Mering. STRING v10: protein-protein interaction networks, integrated

over the tree of life. *Nucleic acids research*, 43(Database issue):D447–52, jan 2015.

[42] Alexandre Kuhn, Doris Thu, Henry J Waldvogel, Richard L M Faull, and Ruth Luthi-Carter. Population-specific expression analysis (psea) reveals molecular changes in diseased brain. *Nature Methods 2011 8:11*, 8:945–947, 10 2011.

[43] Xuran Wang, Jihwan Park, Katalin Susztak, Nancy R. Zhang, and Mingyao Li. Bulk tissue cell type deconvolution with multi-subject single-cell expression reference. *Nature Communications 2019 10:1*, 10:1–9, 1 2019.

[44] Aaron M. Newman, Chloé B. Steen, Chih Long Liu, Andrew J. Gentles, Aadel A. Chaudhuri, Florian Scherer, Michael S. Khodadoust, Mohammad S. Esfahani, Bogdan A. Luca, David Steiner, Maximilian Diehn, and Ash A. Alizadeh. Determining cell type abundance and expression from bulk tissues with digital cytometry. *Nature Biotechnology 2019 37:7*, 37:773–782, 5 2019.

[45] Vincenzo Capece, Julio C Garcia Vizcaino, Ramon Vidal, Raza-Ur Rahman, Tonatiuh Pena Centeno, Orr Shomroni, Irantzu Suberviola, Andre Fischer, and Stefan Bonn. Oasis: Online Analysis of Small RNA Deep Sequencing Data. *Bioinformatics*, 31(13):2205–2207, jul 2015.

[46] Alexander Lex, Nils Gehlenborg, Hendrik Strobelt, Romain Vuillemot, and Hanspeter Pfister. Upset: Visualization of intersecting sets.

[47] Witwer KW. Circulating microrna biomarker studies: pitfalls and potential solutions. *Clinical chemistry*, 61:56–63, 1 2015.

[48] Yali Amit and Donald Geman. Communicated by shimon ullman shape quantization and recognition with randomized trees. *Neural Computation*, 9:1545–1588, 1997.

[49] Igor Splawski, Katherine W Timothy, Niels Decher, Pradeep Kumar, Frank B Sachse, Alan H Beggs, Michael C Sanguinetti, and Mark T Keating. Severe

Arrhythmia Disorder Caused by Cardiac L-Type Calcium Channel Mutations. *Proc. Natl. Acad. Sci. U.S.A.*, 102(23):8088–8089, jun 2005.

[50] Igor Splawski, Katherine W Timothy, Leah M Sharpe, Niels Decher, Pradeep Kumar, Raffaella Bloise, Carlo Napolitano, Peter J Schwartz, Robert M Joseph, Karen Condouris, Helen Tager-Flusberg, Silvia G Priori, Michael C Sanguinetti, and Mark T Keating. Ca(V)1.2 Calcium Channel Dysfunction Causes a Multisystem Disorder Including Arrhythmia and Autism. *Cell*, 119(1):19–31, oct 2004.

[51] Johannes A Mayr, Franz A Zimmermann, Rita Horváth, Hans-Christian Schneider, Benedikt Schoser, Elke Holinski-Feder, Birgit Czermin, Peter Freisinger, and Wolfgang Sperl. Deficiency of the Mitochondrial Phosphate Carrier Presenting as Myopathy and Cardiomyopathy in a Family with Three Affected Children. *Neuromuscul. Disord.*, 21(11):803–808, nov 2011.

[52] Charles J David, Mo Chen, Marcela Assanah, Peter Canoll, and James L Manley. HnRNP Proteins Controlled by c-Myc Deregulate Pyruvate Kinase mRNA Splicing in Cancer. *Nature*, 463(7279):364–368, jan 2010.

[53] Eric T Wang, Rickard Sandberg, Shujun Luo, Irina Khrebtukova, Lu Zhang, Christine Mayr, Stephen F Kingsmore, Gary P Schroth, and Christopher B Burge. Alternative Isoform Regulation in Human Tissue Transcriptomes. *Nature*, 456(7221):470–476, nov 2008.

[54] Qun Pan, Ofer Shai, Leo J Lee, Brendan J Frey, and Benjamin J Blencowe. Deep Surveying of Alternative Splicing Complexity in the Human Transcriptome by High-Throughput Sequencing. *Nat. Genet.*, 40(12):1413–1415, dec 2008.

[55] Mikita Suyama. Mechanistic Insights into Mutually Exclusive Splicing in Dynamin 1. *Bioinformatics*, 29(17):2084–2087, jan 2013.

[56] Klas Hatje, Raza-Ur Rahman, Ramon O Vidal, Dominic Simm, Björn Hammesfahr, Vikas Bansal, Ashish Rajput, Michel Edwar Mickael, Ting Sun, Stefan Bonn, and Martin Kollmar. The landscape of human mutually exclusive splicing. *Molecular systems biology*, 13(12):959, dec 2017.

[57] Meena Kishore Sakharkar, Vincent T K Chow, and Pandjassarame Kangueane. Distributions of Exons and Introns in the Human Genome. *In Silico Biol. (Gedrukt)*, 4(4):387–393, 2004.

[58] Holger Pillmann, Klas Hatje, Florian Odronitz, Björn Hammesfahr, and Martin Kollmar. Predicting Mutually Exclusive Spliced Exons Based on Exon Length, Splice Site and Reading Frame Conservation, and Exon Sequence Homology. *BMC Bioinformatics*, 12:270, 2011.

[59] P Baldi, S Brunak, Y Chauvin, and A Krogh. Naturally Occurring Nucleosome Positioning Signals in Human Exons and Introns. *J. Mol. Biol.*, 263(4):503–510, nov 1996.

[60] Joao Curado, Camilla Iannone, Hagen Tilgner, Juan Valcárcel, and Roderic Guigó. Promoter-like Epigenetic Signatures in Exons Displaying Cell Type-Specific Splicing. *Genome Biol*, 16, 2015.

[61] Eva Schad, Lajos Kalmar, and Peter Tompa. Exon-Phase Symmetry and Intrinsic Structural Disorder Promote Modular Evolution in the Human Genome. *Nucleic Acids Res*, 41(8):4409–4422, apr 2013.

[62] Shijia Zhu, Guohua Wang, Bo Liu, and Yadong Wang. Modeling Exon Expression Using Histone Modifications. *PLoS One*, 8(6), jun 2013.

[63] Serafim Batzoglou, Lior Pachter, Jill P Mesirov, Bonnie Berger, and Eric S Lander. Human and Mouse Gene Structure: Comparative Analysis and Application to Exon Prediction. *Genome Res.*, 10(7):950–958, jan 2000.

[64] Simon E Fisher, Alfredo Ciccodicola, Karo Tanaka, Anna Curci, Sonia Desicato, Michele D'urso, and Ian W Craig. Sequence-Based Exon Prediction around the Synaptophysin Locus Reveals a Gene-Rich Area Containing Novel Genes in Human Proximal Xp. *Genomics*, 45(2):340–347, oct 1997.

[65] Christof Angermueller, Tanel Pärnamaa, Leopold Parts, and Oliver Stegle. Deep Learning for Computational Biology. *Mol. Syst. Biol.*, 12(7):878, 2016.

[66] Chris Burge and Samuel Karlin. Prediction of complete gene structures in human genomic DNA11Edited by F. E. Cohen. *Journal of Molecular Biology*, 268(1):78–94, 1997.

[67] M Ahmad, A Abdullah, and K Buragga. A better way for exon identification in DNA splicing. In *2010 IEEE EMBS Conference on Biomedical Engineering and Sciences (IECBES)*, pages 422–426, nov 2010.

[68] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. 2015.

[69] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks.

[70] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86:2278–2323, 1998.

[71] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. jun 2014.

[72] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. dec 2015.

[73] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn.

IEEE Transactions on Pattern Analysis and Machine Intelligence, 42:386–397, 3 2017.

[74] Zifeng Wu, Chunhua Shen, and Anton van den Hengel. Wider or deeper: Revisiting the resnet model for visual recognition. *Pattern Recognition*, 90:119–133, 6 2019.

[75] A Garcia-Garcia, S Orts-Escolano, S O Oprea, V Villena-Martinez, and J Garcia-Rodriguez. A review on deep learning techniques applied to semantic segmentation. *arxiv*, 2017.

[76] Moisès Burset and Roderic Guigó. Evaluation of gene structure prediction programs. *Genomics*, 34:353–367, 6 1996.

[77] Burge C and Karlin S. Prediction of complete gene structures in human genomic dna. *Journal of molecular biology*, 268:78–94, 4 1997.

[78] Anders Krogh. Using database matches with hmmgene for automated gene detection in drosophila. *Genome Research*, 10:523, 4 2000.

3 Publications

3.1 KNIT, 2021

Bioinformatics, 37(2), 2021, 276–278
doi: 10.1093/bioinformatics/btaa1107
Advance Access Publication Date: 8 January 2021
Applications Note

Gene expression

Interactive gene networks with KNIT

D. S. Magruder [1,2,*], A. M. Liebhoff[1], J. Bethune[1] and S. Bonn [1,*]

[1]Institute for Medical Systems Biology, bAIome – Center for Biomedical AI, Center for Molecular Neurobiology, University Medical Center Hamburg-Eppendorf, 20246 Hamburg, Germany and [2]Genevention GmbH, 37079 Goettingen, Germany

*To whom correspondence should be addressed.

Associate Editor: Pier Luigi Martelli

Received on May 26, 2020; revised on December 28, 2020; editorial decision on December 29, 2020; accepted on January 3, 2021

Abstract

Summary: KNIT is a web application that provides a hierarchical, directed graph on how a set of genes is connected to a particular gene of interest. Its primary aim is to aid researchers in discerning direct from indirect effects that a gene might have on the expression of other genes and molecular pathways, a very common problem in omics analysis. As such, KNIT provides deep contextual information for experiments where gene or protein expression might be changed, such as gene knock-out and overexpression experiments.

Availability and implementation: KNIT is publicly available at http://knit.ims.bio. It is implemented with Django and Nuxtjs, with all major browsers supported.

Contact: sumner.magruder@zmnh.uni-hamburg.de or sbonn@uke.de

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

A common method to understand the functional role of a gene is by altering its expression status and measuring the subsequent molecular changes of the cell. Alterations in gene expression have been functionally grouped into gene deletion, expression attenuation or gene over-expression using various Molecular Biology methods. Subsequent measurements of molecular changes are usually obtained using omics technologies, such as next generation sequencing in the case of gene expression.

While these studies present molecular changes with unprecedented depth they tend to lack information on which functional changes are directly caused by the gene of interest, and which ones are compensatory or indirect changes to sustain cell homeostasis. Therefore, differentially expressed genes (DEGs) in a gene deletion study are not necessarily a direct consequence of initial perturbation of the system. Identifying which of these downstream effects are relevant to the primary gene of interest (e.g. the gene which was deleted, over-expressed, etc.) can be cumbersome.

To contextualize a set of genes in relation to the primary gene of interest, gene-gene network tools like GeneMANIA, Pathway Commons, STRING and STITCH may be leveraged (Cerami *et al.*, 2006, 2011; Kuhn *et al.*, 2008; Mostafavi *et al.*, 2008; Szklarczyk et al., 2015). However, these tools do not provide a query matching the paradigm, as they search for connections between the set of all requested genes rather than querying for pathways to or from the primary gene from or to the rest of the genes in the set. In conjunction with the use of force-based graph layouts, visualizing the directional relationship from a primary gene of interest to a set of genes becomes convoluted.

Here, we present KNIT, a web application that provides visual and query-able information on how a set of genes is connected to a particular gene of interest. KNIT uses hierarchical, directed layouts to provide visual cues of potentially direct effects, aiding researchers in defining the true molecular function of a gene of interest. In addition, KNIT supports enrichment analysis of a given graph, guiding the formulation of hypotheses on the underlying biology. While KNIT was designed with the gene knock-in (KI) and knock-out (KO) paradigm in mind, KNIT clearly generalizes to any exploratory question between a gene of interest and set of genes.

2 Materials and methods

KNIT facilitates the exploration of the directional relationship between a gene of interest (e.g. KI or KO) and the entries in a gene list (e.g. DEGs) by constructing a composite graph from the human data collected via cPath, which is made accessible by Pathway Commons and through metadata from NCBI (Cerami *et al.*, 2006, 2011). As the goal of KNIT is the identification of targeted relationships between a gene of interest and another set of genes, KNIT constructs a composite graph by utilizing the k-shortest paths to and from the gene of interest and each entry in the gene list, where k is set by the user. As querying for the pathways may be computationally expensive, each pathway request (source, target, k) is conducted asynchronously on the backend, allowing users to see their composite graph develop in real time. In case the user supplies gene expression fold change information or P-values for the gene set genes, KNIT will incorporate this information in the resultant graph (Fig. 1). While individual connections are valid, as they are retrieved from an established database (Pathway Commons), not every sequence of connections in the graph is a pathway, as KNIT displays aggregated

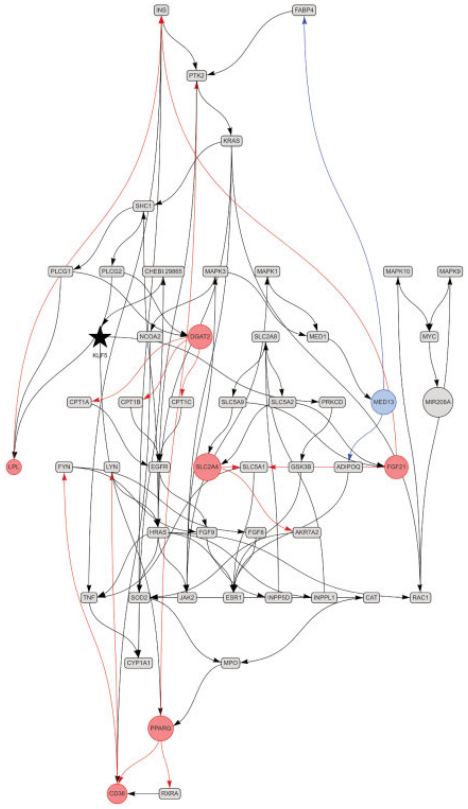


Fig. 1. The composite graph returned by KNIT. The primary gene of interest (KLF5) is highlighted as a black star, while a set of DEGs (circles) are colored and scaled by their metadata (expression change and significance, where red and blue symbols indicate up- and down-regulation, respectively). The graph suggests that the up-regulation of Pparg may be a direct consequence of KLF knockout. Together with the interactively accessible metadata, the graph allows users to formulate hypotheses of how their primary gene interacts with a given set of genes

information from various sources. Pathway information can, however, be retrieved from provided metadata. Therefore, KNIT should be understood as a simple-to-use tool for hypothesis formulation and defining the focus in follow up research.

2.1 Path-finding

Path-finding is non-trivial and depends on various preferences such as the length of the path or the cost to travel the path, as highlighted in Madkour *et al.* (2017) for shortest-path algorithms and Pascoal et al. (2006) for *k*-shortest-path algorithms. In addition, viewing the singular 'best' path between two entities in an interaction network provides an incomplete image of how these two entities interact. Therefore KNIT computes the *k*-best paths utilizing Yen's algorithm (Yen, 1971), which utilizes Dijstkra's algorithm for shortest-path finding. Currently, KNIT weights arcs in the graph proportionally to the number of publications supporting the arc.

2.2 Web application

KNIT's architecture consists of a single page Nuxt.js application and a singular module Django application. A Node.js server provides the frontend, while Nginx serves the backend. Non-blocking asynchronous requests to build the composite graph for each source-target pair and for the composite graph's metadata are sent from the frontend via the axios.js library to the backend, allowing for scalability (Supplementary Fig. S1). The *k*-shortest paths for each source—target pair are calculated by the backend and as this data is returned to the frontend, the graph is rendered utilizing vis.js. An overview of this architecture can be seen in the Supplementary Materials. Once all queried paths are found, the meta information for the resultant sub-graph is requested. Four main types of meta information are provided: (i) data sources: origin of evidence for the composite graph together with a brief overview of the sources, (ii) interaction types: summary of the interaction information between entities of the sub-graph, as well as the relative percentage of publications that support that interaction type, (iii) pathways: known pathways of edges of the returned sub-graph are a part of and (iv) publications: list of the supporting publications. In addition, KNIT provides a feature for interactive computation of enrichment. The user can select meta-information which will update the graph. As an edge may have multiple sources of evidence which support it, the edge will only be removed from the graph if every supporting evidence is deselected.

3 Usage and case study

KNIT has a rich online documentation, explaining its basic functionality and how to interpret analysis results. In addition, KNIT supports batch upload of data, which makes it easy to query a list of e.g. 50 differentially expressed genes with *P*-value and fold change information. To exemplify KNIT's salient features we used cardiomyocyte data to compare analysis results for KNIT, STRING, STITCH, Genemania and Pathway Commons using default settings (Cerami *et al.*, 2006, 2011; Kuhn *et al.*, 2008; Mostafavi *et al.*, 2008; Szklarczyk *et al.*, 2015) (Fig. 1, Supplementary Fig. S2). More specifically, the data by Pol *et al.* (2019) highlights the effect of cardiomyocyte KLF5 signaling (black star in Fig. 1) on white adipose tissue using a murine *Klf5* knocked-out model. The *Klf5* knock-out resulted in increased weight of the mice and increased mRNA levels of genes involved in the adipocyte lipid metabolism: Pparg1, Pparg2, Lpl, Cd36 and Dgat2 (Fig. 1, red and blue circles). As can be seen in Figure 1 and Supplementary Figures S2 and S3, KNIT shows clearly which genes might be directly affected by the *Klf5* KO and which are not, while visualizing positive and negative interactions, the interaction type and all relevant meta-information interactively. Additional validation is provided in Supplementary Materials and in Supplementary Table S1.

4 Conclusion

KNIT provides users an intuitive GUI to readily find interactions to their primary gene of interest along with associated meta-data. Further KNIT interactive exploration aids researchers in framing the relationship between the conditions of their experiment and the results. KNIT is the first web application that allows to query a target gene and a gene set of interest for potential directed signaling, supporting researchers in differentiating direct from indirect interactions.

Financial Support: none declared.

Conflict of Interest: none declared.

References

Cerami,E.G. *et al.* (2006) cPath: open source software for collecting, storing, and querying biological pathways. *BMC Bioinformatics*, **7**, 497.

Cerami,E.G. *et al.* (2011) Pathway Commons, a web resource for biological pathway data. *Nucleic Acids Res.*, **39**, D685–D690.

Kuhn,M. *et al.* (2008) STITCH: interaction networks of chemicals and proteins. *Nucleic Acids Res.*, **36**, D684–8.

Madkour,A. *et al.* (2017) A Survey of Shortest-Path Algorithms. Technical Report. arXiv.

Mostafavi,S. *et al.* (2008) GeneMANIA: a real-time multiple association network integration algorithm for predicting gene function. *Genome Biol.*, **9**, S4.

Pol,C.J. *et al.* (2019) Cardiac myocyte KLF5 regulates body weight via alteration of cardiac FGF21. *Biochim. Biophys. Acta Mol. Basis Dis.*, **1865**, 2125–2137.

Pascoal,M.B. *et al.* (2006) A comprehensive survey on the quickest path problem. *Ann. Oper. Res.* **147**, 5–21.

Szklarczyk,D. *et al.* (2015) STRING v10: protein-protein interaction networks, integrated over the tree of life. *Nucleic Acids Res.*, **43**, D447–52.

Yen,J.Y. (1971) Finding the K shortest loopless paths in a network. *Manag. Sci.*, **17**, 712–716.

3.2 Neuropathy, 2020

Neuropathology of patients with COVID-19 in Germany: a post-mortem case series

Jakob Matschke, Marc Lütgehetmann, Christian Hagel, Jan P Sperhake, Ann Sophie Schröder, Carolin Edler, Herbert Mushumba, Antonia Fitzek, Lena Allweiss, Maura Dandri, Matthias Dottermusch, Axel Heinemann, Susanne Pfefferle, Marius Schwabenland, Daniel Sumner Magruder, Stefan Bonn, Marco Prinz, Christian Gerloff, Klaus Püschel, Susanne Krasemann, Martin Aepfelbacher, Markus Glatzel

Summary

Background **Prominent clinical symptoms of COVID-19 include CNS manifestations. However, it is unclear whether severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2), the causative agent of COVID-19, gains access to the CNS and whether it causes neuropathological changes. We investigated the brain tissue of patients who died from COVID-19 for glial responses, inflammatory changes, and the presence of SARS-CoV-2 in the CNS.**

Methods **In this post-mortem case series, we investigated the neuropathological features in the brains of patients who died between March 13 and April 24, 2020, in Hamburg, Germany. Inclusion criteria comprised a positive test for SARS-CoV-2 by quantitative RT-PCR (qRT-PCR) and availability of adequate samples. We did a neuropathological workup including histological staining and immunohistochemical staining for activated astrocytes, activated microglia, and cytotoxic T lymphocytes in the olfactory bulb, basal ganglia, brainstem, and cerebellum. Additionally, we investigated the presence and localisation of SARS-CoV-2 by qRT-PCR and by immunohistochemistry in selected patients and brain regions.**

Findings **43 patients were included in our study. Patients died in hospitals, nursing homes, or at home, and were aged between 51 years and 94 years (median 76 years [IQR 70–86]). We detected fresh territorial ischaemic lesions in six (14%) patients. 37 (86%) patients had astrogliosis in all assessed regions. Activation of microglia and infiltration by cytotoxic T lymphocytes was most pronounced in the brainstem and cerebellum, and meningeal cytotoxic T lymphocyte infiltration was seen in 34 (79%) patients. SARS-CoV-2 could be detected in the brains of 21 (53%) of 40 examined patients, with SARS-CoV-2 viral proteins found in cranial nerves originating from the lower brainstem and in isolated cells of the brainstem. The presence of SARS-CoV-2 in the CNS was not associated with the severity of neuropathological changes.**

Interpretation **In general, neuropathological changes in patients with COVID-19 seem to be mild, with pronounced neuroinflammatory changes in the brainstem being the most common finding. There was no evidence for CNS damage directly caused by SARS-CoV-2. The generalisability of these findings needs to be validated in future studies as the number of cases and availability of clinical data were low and no age-matched and sex-matched controls were included.**

Funding **German Research Foundation, Federal State of Hamburg, EU (eRARE), German Center for Infection Research (DZIF).**

Introduction

COVID-19, the disease caused by severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2), has evolved into a global pandemic since the first recorded cases in December, 2019. Although SARS-CoV-2 primarily targets the respiratory tract,[1] other organ systems such as the renal and cardiovascular systems are also affected.[2,3] Additionally, neurological symptoms are common in COVID-19 and include anosmia and ageusia, non-specific symptoms such as dizziness and headache, and severe conditions such as ischaemic stroke, haemorrhagic encephalopathy, and posterior reversible encephalopathy syndrome with epileptic seizures.[4–7] Furthermore, clinical data and laboratory investigations suggest that encephalitis,[8–10] meningitis,[9,10] polyneuritis cranialis, and Guillain-Barré and Miller Fisher syndromes[11–13] might also be associated with COVID-19.

Why SARS-CoV-2 infection leads to neurological symptoms, and whether and how the virus gains access to the CNS are not well understood. The two main competing hypotheses are based on neurotropism and direct invasion of SARS-CoV-2 into the CNS, and indirect mechanisms mediated by the cytokine storm induced by systemic SARS-CoV-2 infection.

In-depth neuropathological assessment can elucidate if and how SARS-CoV-2 gains access to or damages the brain.[14] However, only a few reports of the neuropathological findings of patients with COVID-19 have been published. Two case reports showed no gross CNS abnormalities at autopsy,[15] and two case series documented no signs of encephalitis or CNS vasculitis.[16,17] Additionally, loss of white matter and axonal injury were described in one case report,[18] while massive intracranial haemorrhage

Lancet Neurol 2020; 19: 919–29

Published Online October 5, 2020 https://doi.org/10.1016/S1474-4422(20)30308-2

See Comment page 883

Institute of Neuropathology (J Matschke MD, C Hagel MD, M Dottermusch MD, S Krasemann PhD, Prof M Glatzel MD), **Institute of Medical Microbiology, Virology, and Hygiene** (M Lütgehetmann MD, S Pfefferle MD, Prof M Aepfelbacher MD), **Institute of Legal Medicine** (Prof J P Sperhake, A S Schröder MD, C Edler MD, H Mushumba MD, A Fitzek MD, A Heinemann MD, Prof K Püschel MD), **I Department of Medicine** (Prof M Dandri PhD, L Allweiss PhD), **Institute of Medical Systems Biology** (D Sumner Magruder MSc, Prof S Bonn PhD), **and Department of Neurology** (Prof C Gerloff MD), **University Medical Center, Hamburg-Eppendorf, Hamburg, Germany; Institute of Neuropathology** (Prof M Prinz MD, M Schwabenland MD), **Center for Basics in Neuromodulation, Faculty of Medicine** (Prof M Prinz), **and Signaling Research Centers BIOSS and CIBSS** (Prof M Prinz), **University of Freiburg, Freiburg, Germany; Center for Infection Research, Partner Site Hamburg-Borstel-Lübeck-Riems, Germany** (M Lütgehetmann, Prof M Dandri); **and German Center for Neurodegenerative Diseases, Tübingen, Germany** (Prof S Bonn)

Correspondence to: Prof Dr Markus Glatzel, Institute of Neuropathology, University Medical Center Hamburg-Eppendorf, 20246 Hamburg, Germany **m.glatzel@uke.de**

For more on the **global spread of COVID-19 cases** see https://covid19.who.int/

Research in context

Evidence before this study

We searched PubMed for studies focusing on the neuropathology of severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) infection, published in German or English, until Aug 2, 2020. Search terms included "COVID-19" OR "SARS-CoV-2" AND "neuropathology" OR "neurodegeneration" OR "encephalitis" OR "central nervous system" OR "brain". Studies examining the neuropathology of COVID-19 in animal models were also considered. The literature gave a heterogeneous picture regarding the neuropathological presentation of COVID-19. Two studies described no signs of encephalitis or nervous system vasculitis, and mainly reactive changes unrelated to SARS-CoV-2 infection, whereas other studies show pan-encephalitis, cerebral haemorrhage, and areas of necrosis with loss of white matter and axonal injury attributed to SARS-CoV-2 infection. All studies were subject to sampling bias and small patient numbers, and most examined brains originated from patients who died in hospital under intensive care unit treatment, which can itself lead to neuropathological alterations independently of SARS-CoV-2 infection.

Added value of this study

To our knowledge this study represents the world's largest case series of brain autopsies from patients with COVID-19 to date. We included 43 patients who died under intensive care unit treatment, in regular hospital wards, in nursing homes, or in their own homes, ranging in age from 51 years to 94 years. Fresh ischaemic lesions were found in the brains of six patients, and almost all patients showed astrocytic reactions in all assessed brain regions. Neuroimmune activation was observed in all examined brains, with prominent involvement of the brainstem and neuroimmune reaction, in line with involvement of the adaptive and innate immune systems. The presence of SARS-CoV-2 did not seem to be associated with the severity of neuroimmune activation. Neuroimmune activation was also observed in patients who died from COVID-19 at home or in nursing homes.

Implications of all the available evidence

The emerging evidence, including the current study, shows that neuropathological alterations in the brains of patients who die from COVID-19 are relatively mild, although the virus is able to gain access to the brain. The neuropathological alterations are most likely to be immune-mediated, and there does not seem to be fulminant virus-induced encephalitis nor direct evidence for SARS-CoV-2-caused CNS damage. Further studies are needed to define how SARS-CoV-2 gains access to the brain, to define the neuroimmune activation, and to describe the distribution of SARS-CoV-2 in the brain.

and pan-encephalitis were described in a case series,[19] and one case series reported only the detection of SARS-CoV-2 in the brain.[20]

The current study aimed to investigate the neuropathological features of COVID-19, including glial response, inflammatory changes, and the presence and distribution of SARS-CoV-2 in the brain of patients who died from COVID-19.

Methods

Study design and participants

Consecutive patients who had died following a diagnosis of SARS-CoV-2 infection were autopsied at the Institute of Legal Medicine, University Medical Center of Hamburg-Eppendorf (Hamburg, Germany) between March 13 and April 24, 2020, upon order issued by the Hamburg public health authorities in accordance with section 25(4) of the German Infection Protection Act. Organisation of autopsies and adequate collection of samples were logistically challenging as this time period coincided with the peak incidence of COVID-19 in Hamburg. Inclusion criteria for this study were a confirmed diagnosis of SARS-CoV-2 infection, with SARS-CoV-2 RNA detected by quantitative RT-PCR (qRT-PCR) analysis of pharyngeal swabs, and the availability of sufficient high-quality brain tissue samples. Clinical presentation and neuroradiological findings did not form part of the inclusion criteria.

The study was approved by the local ethics committee of the Hamburg Chamber of Physicians (approval number PV7311) and the study is in line with the Declaration of Helsinki.

Procedures

We assessed patients' clinical data, including pre-existing medical conditions, medical course before death, and ante-mortem diagnostic findings. Where logistically feasible, before fixation, specimens were taken from 23 brains for cryopreservation to allow investigation of the presence of SARS-CoV-2 in non-fixed tissue. All brains were fixed in buffered 4% formaldehyde, examined macroscopically, and underwent routine neuropathological workup.

Single-cell gene expression analysis

Human brain single-cell transcriptome data taken from Darmanis and colleagues' study[21] were processed and analysed with use of the methods described by Marouf and colleagues.[22] In brief, cell-type clusters were annotated using known marker genes as reported previously.[21] Mean cell type-specific RNA levels of angiotensin-converting enzyme 2 (*ACE2*), cathepsin L (*CTSL*), transmembrane serine protease 2 (*TMPRSS2*), transmembrane serine protease 4 (*TMPRSS4*), neuropilin 1 (*NRP1*), and two pore segment channel 2 (*TPCN2*) were normalised per gene (cell type-specific expression divided by the sum of gene

expression across cell types), and were subsequently plotted as a heatmap.

Histological and immunohistochemical evaluations

Formalin-fixed paraffin-embedded tissue (FFPE) samples from the olfactory bulb, superior frontal gyrus, basal ganglia including the putamen, upper and lower medulla oblongata, and cerebellar hemisphere were processed and stained with haematoxylin and eosin using standard laboratory procedures. Immunohistochemical staining was also done with a Ventana Benchmark XT Autostainer (Ventana, Tucson, AZ, USA), in accordance with the manufacturer's recommendations, using antibodies against human glial fibrillary acidic protein (GFAP; clone 6F2; Dako, Glostrup, Denmark; dilution 1:200), HLA-DR (mouse anti-HLA-DP, DQ, DR antibody, clone CR3/43; Dako; 1:200), transmembrane protein 119 (TMEM119; catalogue number ab185333; Abcam, Cambridge, UK; 1:250), ionized calcium-binding adaptor molecule 1 (IBA1; clone EPR16588; Abcam, Cambridge; 1:1000), CD68 (clone PG-M1; Dako; 1:200), and CD8 (clone SP239; Spring Bioscience, Pleasanton, USA; 1:100). Double-immunolabelling for IBA1 and CD8 was done sequentially with Permanent Red (Monosan Permanent AP-Red Kit; Monosan, Uden, Netherlands) as chromogen.

Slides were examined by experienced neuropathologists (JM, CH, and MG). At least two neuropathologists, masked to patients' clinical findings, assessed each slide, and disagreements were resolved by consensus. Slides were screened at low magnification and areas with the most pronounced changes (ie, strongest staining) were used for quantification, and were electronically scanned at high magnification (×40) as high-resolution images (1900×1200 pixels) with a NanoZoomer 2.0-HT (Hamamatsu Photonics, Hamamatsu, Japan).

The degree of astrogliosis and microgliosis was classified as none, slight, moderate, or severe, using a three-tiered semi-quantitative approach (appendix p 3), based on GFAP as an astrocyte marker and HLA-DR as a marker of activated microglia. CD68 was used to judge phagocytic activity, and IBA1 as an additional marker for microglia activity.

For semi-quantitative assessment of cytotoxic T lymphocyte infiltration, cells with positive CD8 staining were counted per high-power field (HPF) of 0·5 mm². Infiltration was categorised as none, mild (one to nine cells per HPF), moderate (ten to 49 cells per HPF), or severe (≥50 cells per HPF; appendix p 3).

qRT-PCR analysis of SARS-CoV-2

qRT-PCR was used to quantify SARS-CoV-2 presence in specimens with enough high-quality material available. RNA was isolated from frozen or paraffin-embedded tissue samples. Frozen tissue was ground with a Precellys 24 tissue homogeniser (Bertin, Rockville, USA) using 2 mL tubes pre-filled with ceramic beads (Precellys Lysing Kit; Bertin) and 1 mL RNAse-free and DNAse-free PCR-grade water. 200 µL of the tissue homogenate was transferred to a MagNA Pure 96 instrument (Roche, Mannheim, Germany), and automated nucleic acid extraction was done according to the manufacturer's recommendation with whole process control (RNA Process Control Kit; Roche), with a final elution volume of 100 µL. Slides of paraffin-embedded tissue were deparaffinised and RNA was extracted with the Maxwell 16 LEV RNA FFPE Purification Kit and a Maxwell RNA extraction system (Promega, Fitchburg, USA).

PCR and virus quantification were done as previously described.[20] In brief, an assay targeting the E gene of SARS-CoV-2[23] was used for the amplification and detection of SARS-CoV-2 RNA. A cycle threshold value for the target was determined with use of the second derivative maximum method.[24] For quantification, standard in-vitro transcribed RNA of the E gene of SARS-CoV-2 was used (catalogue number 001K-03884; European Virus Archive, Charité, Berlin, Germany). The linear range of the assay is between 1×10^3 and 1×10^9 copies per mL. To normalise for input, quantitative β-globin PCR was done with a commercial TaqMan primer kit (catalogue number Hs00758889_s1; Thermo Fisher Scientific, Waltham, MA, USA), and the amount of DNA was normalised with use of a KAPA Human Genomic DNA Quantification and QC DNA Standard (catalogue number 07960638001; KAPA Biosystems, Cape Town, South Africa).

Immunohistochemical detection of SARS-CoV-2 spike protein and nucleoprotein

Antibodies for detecting SARS-CoV-2 in FFPE tissues were first validated on SARS-CoV-2-infected (Hamburg isolate) and non-infected Vero cells that were processed to FFPE blocks (appendix p 2). In specimens with sufficient high-quality tissue, we tested for the presence of the virus with immunohistochemistry using antibodies against viral nucleocapsid protein (catalogue numbers 40143-R001 [dilution 1:5000] and 40143-T62 [dilution 1:1000]; Sino Biological, Eschborn, Germany) and spike protein (clone 1A9, catalogue number GTX632604; GeneTex, Irvine, USA; dilution 1:300).[25] Immunohistochemical staining was done with a Ventana Benchmark XT Autostainer. All slides were examined by two experienced morphologists (SK and MG), and any disagreements were resolved by consensus.

See Online for appendix

Role of the funding source

The funder of the study had no role in study design, data collection, data analysis, data interpretation, or writing of the report. The corresponding author had full access to all the data in the study and had final responsibility for the decision to submit for publication.

Results

From the 110 patients with diagnosed or suspected SARS-CoV-2 infection who were autopsied between March 13 and April 24, 2020, 43 (39%) with a positive qRT-PCR test for SARS-CoV-2 and adequate samples

	Sex	Age, years	Place of death	Post-mortem interval, days	Cause of death	Comorbidities	Brain weight, g	Brain oedema	Brain atrophy	Arteriosclerosis	Macroscopic findings
Case 1	Female	87	Nursing home	0	Pneumonia	COPD, dementia, IHD, renal insufficiency	1215	None	Mild	Moderate	None
Case 2	Female	85	Hospital ward	0	Pneumonia	Atrial fibrillation, cardiac insufficiency, IHD, myelofibrosis, renal insufficiency	1240	None	Mild	Moderate	Fresh infarction in territory of PCA
Case 3	Male	88	Hospital ward	5	Pneumonia	Emphysema, IHD, renal insufficiency	1490	Moderate	None	Moderate	Fresh infarction in territory of MCA
Case 4	Male	75	ICU	4	Pulmonary arterial embolism, pneumonia	Atrial fibrillation, emphysema, hypertension, renal insufficiency	1475	Mild	None	Moderate	Fresh infarction in territory of PCA
Case 5	Female	86	Nursing home	0	Pneumonia	COPD, dementia, IHD	1250	None	Mild	Severe	Fresh infarction in territory of PCA
Case 6	Male	90	Nursing home	2	Pneumonia	Atrial fibrillation, dementia, diabetes, history of stroke	1015	None	Moderate	Severe	Old infarctions in territory of PCA
Case 7	Male	90	Hospital ward	3	Emphysema with respiratory decompensation	Cardiac insufficiency, COPD	1440	None	Mild	Moderate	None
Case 8	Male	77	Hospital ward	2	Pneumonia	Aortic aneurysm, atrial flutter, cardiac hypertrophy, emphysema, renal insufficiency	1590	Moderate	None	Moderate	None
Case 9	Male	76	ICU	3	Pulmonary arterial embolism, respiratory tract infection	Cardiac insufficiency, COPD	1460	Mild	None	Moderate	None
Case 10	Male	76	ICU	3	Sepsis, aortic valve endocarditis, pneumonia	AML, cardiomyopathy, thyroid cancer	1270	None	Mild	Mild	None
Case 11	Male	70	Hospital ward	1	Pneumonia (aspiration)	Cardiac insufficiency, COPD, IHD, Parkinson's disease	1430	Mild	None	Severe	None
Case 12	Male	93	Hospital ward	3	Pneumonia	Diabetes, hypertension	1400	Mild	None	Moderate	None
Case 13	Male	66	Emergency room	2	Pneumonia	Diabetes, IHD	1450	Mild	None	Severe	None
Case 14	Female	54	Hospital ward	1	Pneumonia	Trisomy 21, epilepsy	950	None	Severe	Mild	Grey matter heterotopia
Case 15	Male	82	Hospital ward	1	Pneumonia	Diabetes, IHD, Parkinson's disease	1170	None	Mild	Moderate	Old infarctions in territory of PCA
Case 16	Male	86	Nursing home	2	Sepsis, pneumonia	Emphysema, epilepsy, hypoxic brain damage, IHD, renal insufficiency	1210	None	Mild	Moderate	None
Case 17	Female	87	Home	1	Pneumonia	Cardiac insufficiency, COPD	1180	None	Mild	Severe	None
Case 18	Female	70	ICU	3	Pneumonia	Cardiac insufficiency	1150	None	Mild	Moderate	None
Case 19	Female	75	ICU	4	Pneumonia	Cardiac arrythmia, IHD	1210	None	Mild	Severe	None
Case 20	Male	93	Hospital ward	2	Pneumonia	Atrial fibrillation, cardiac insufficiency, diabetes, IHD, obstructive sleep apnoea syndrome	1000	None	Moderate	Moderate	Old cerebellar infarction
Case 21	Female	82	Hospital ward	4	Purulent bronchitis	COPD, history of pulmonary embolism, renal insufficiency	1080	None	Moderate	Moderate	None
Case 22	Male	63	ICU	1	Pulmonary arterial embolism, pneumonia	Cardiac insufficiency	1435	Mild	None	Mild	Fresh infarction in territory of ACA
Case 23	Male	84	Hospital ward	5	Pneumonia, septic encephalopathy	Diabetes, history of stroke, hypertension, IHD, ulcerative colitis	1350	Mild	None	Severe	None

(Table continues on next page)

	Sex	Age, years	Place of death	Post-mortem interval, days	Cause of death	Comorbidities	Brain weight, g	Brain oedema	Brain atrophy	Arteriosclerosis	Macroscopic findings
(Continued from previous page)											
Case 24	Male	71	ICU	2	Pulmonary arterial embolism, pneumonia	Cardiac insufficiency, diabetes, lung granuloma	1665	Moderate	None	Mild	None
Case 25	Male	75	Nursing home	3	Sudden cardiac death	Parkinson's disease	1110	Mild	None	Moderate	None
Case 26	Male	52	Home	1	Pulmonary arterial embolism, pneumonia	Cardiac insufficiency	1520	Moderate	None	Severe	None
Case 27	Male	85	ICU	2	Pneumonia	COPD, aortic valve replacement, hypertension, IHD	1400	Mild	None	Moderate	None
Case 28	Female	75	Home	2	Pulmonary arterial embolism	Hypertension, IHD	1095	None	Moderate	Moderate	None
Case 29	Male	59	Hospital ward	12	Pneumonia	Cardiomyopathy	1575	Moderate	None	Mild	None
Case 30	Male	85	Hospital ward	15	Pneumonia	Atrial fibrillation, COPD, hypothyroidism, lung cancer, renal insufficiency	1540	Moderate	None	Moderate	Cerebellar metastasis of non-small cell lung cancer
Case 31	Female	76	Hospital ward	2	Pneumonia	Breast cancer, hypertension	1180	None	Mild	Moderate	None
Case 32	Male	73	Home	9	Sudden cardiac death	Cardiomyopathy, emphysema, IHD	1430	Mild	None	Severe	None
Case 33	Male	70	ICU	9	Pneumonia	Dementia, IHD, hypertension	1370	None	None	Moderate	None
Case 34	Female	90	Nursing home	3	Pneumonia	Cardiomyopathy, dementia, emphysema, renal insufficiency	1090	None	Severe	Moderate	None
Case 35	Female	94	Hospital ward	2	Sepsis	Atrial fibrillation, cardiac insufficiency, dementia, history of stroke, IHD, renal insufficiency	1220	Mild	Mild	Moderate	Old infarction in territory of PCA
Case 36	Female	87	Hospital ward	3	Sepsis, pneumonia	Colon cancer, emphysema, paranoid schizophrenia	1310	None	None	Mild	None
Case 37	Female	54	ICU	1	Pneumonia	Mild cardiomyopathy	1470	Mild	None	Mild	None
Case 38	Female	79	Hospital ward	5	Pneumonia	COPD, myelodysplastic syndrome, IHD	1290	None	Mild	Mild	None
Case 39	Male	51	Home	8	Pneumonia	Liver cirrhosis	1255	Mild	Mild	Mild	None
Case 40	Male	85	Hospital ward	3	Pneumonia	Atrial fibrillation, cardiac insufficiency, dysphagia, emphysema, hypertension, IHD	1290	Mild	None	Moderate	None
Case 41	Male	56	Hospital ward	3	Pneumonia	Cardiac insufficiency, COPD, diabetes, IHD, renal insufficiency	1230	Mild	None	Mild	Old infarctions in territory of PCA and lenticulostriate arteries
Case 42	Male	76	ICU	3	Aortic valve endocarditis, pneumonia	AML, cardiomyopathy, thyroid cancer	1270	Mild	None	Mild	None
Case 43	Female	59	ICU	1	Pneumonia	Multiple myeloma	1220	Mild	None	Mild	Fresh infarction in territory of MCA

COPD=chronic obstructive pulmonary disease. IHD=ischaemic heart disease. PCA=posterior cerebral artery. MCA=middle cerebral artery. ICU=intensive care unit. AML=acute myeloid leukaemia. ACA=anterior cerebral artery.

Table: **Summary of cases and brain autopsy findings**

available were included in this study. The autopsy findings, excluding any neuropathological analysis, of 37 (86%) of these patients were previously reported in a separate report of the first 80 consecutive individuals who died of SARS-CoV-2 infection in Hamburg, Germany.[26] The remaining six (14%) cases have not been previously reported. 40 (93%) patients had adequate samples for the detection of SARS-CoV-2 by immunohistochemistry, and 27 patients (63%) had samples available for the detection of SARS-CoV-2 by qRT-PCR. Data on SARS-CoV-2 RNA in

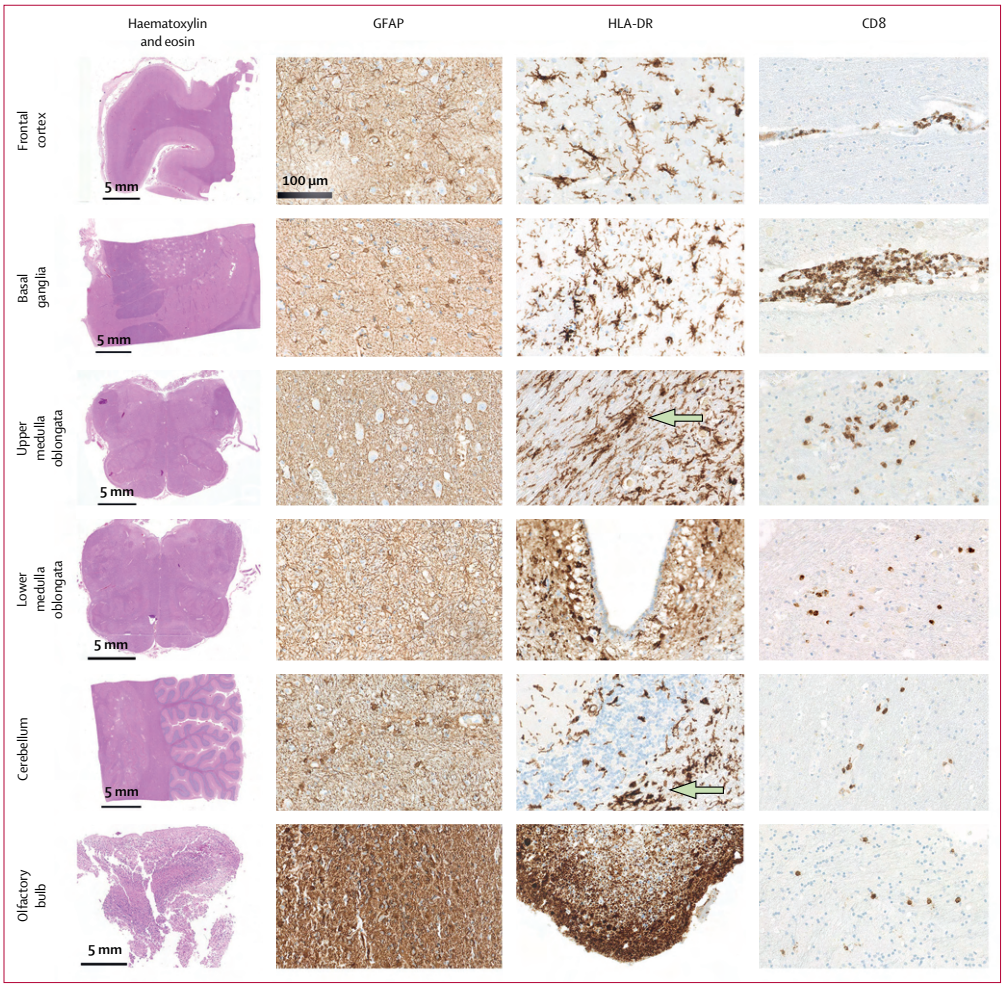


Figure 1: **Common neuropathological findings in the brains of patients who died from COVID-19**
An overview of each brain region with haematoxylin and eosin staining is shown in the first column. Immunohistochemical staining for the astrocytic marker GFAP showed variable degrees of reactive astrogliosis. Immunohistochemical staining for the microglia marker HLA-DR showed reactive activation of the microglia with occasional microglial nodules in the medulla oblongata and cerebellum (green arrows). Staining for the cytotoxic T lymphocyte marker CD8 (brown) revealed perivascular and parenchymal infiltration with CD8-positive cells. GFAP=glial fibrillary acidic protein.

the brain tissue from 22 of these cases have been reported previously.[20]

The median age of the 43 patients was 76 years (IQR 70–86; range 51–94), 16 (37%) patients were women and 27 (63%) were men. 40 (93%) had relevant pre-existing chronic medical conditions (mainly cardiorespiratory problems), and 13 (30%) had pre-existing neurological diseases, such as neurodegenerative disease or epilepsy (table). 11 patients (26%) died outside of a hospital (five at home and six in a nursing facility) and 32 (74%) died in a hospital. 12 (28%) patients who died in hospital were treated in intensive care units (ICUs). Cause of death was mainly attributed to the respiratory system, with viral pneumonia as the underlying condition in most cases (table).

The mean post-mortem interval was 3·3 days (SD 3·1) after two patients with extremely long post-mortem intervals of 12 days and 14 days were excluded as outliers (table). The mean weight of the unfixed brains was 1302 g (SD 171; median 1270 g [1195–1438]; range 950–1665; table) with 23 patients (53%) showing signs of mild to moderate brain oedema, commonly seen as unspecific agonal changes. Arteriosclerosis of the basal vessels was mild in 12 (28%) patients, moderate in 22 (51%), and

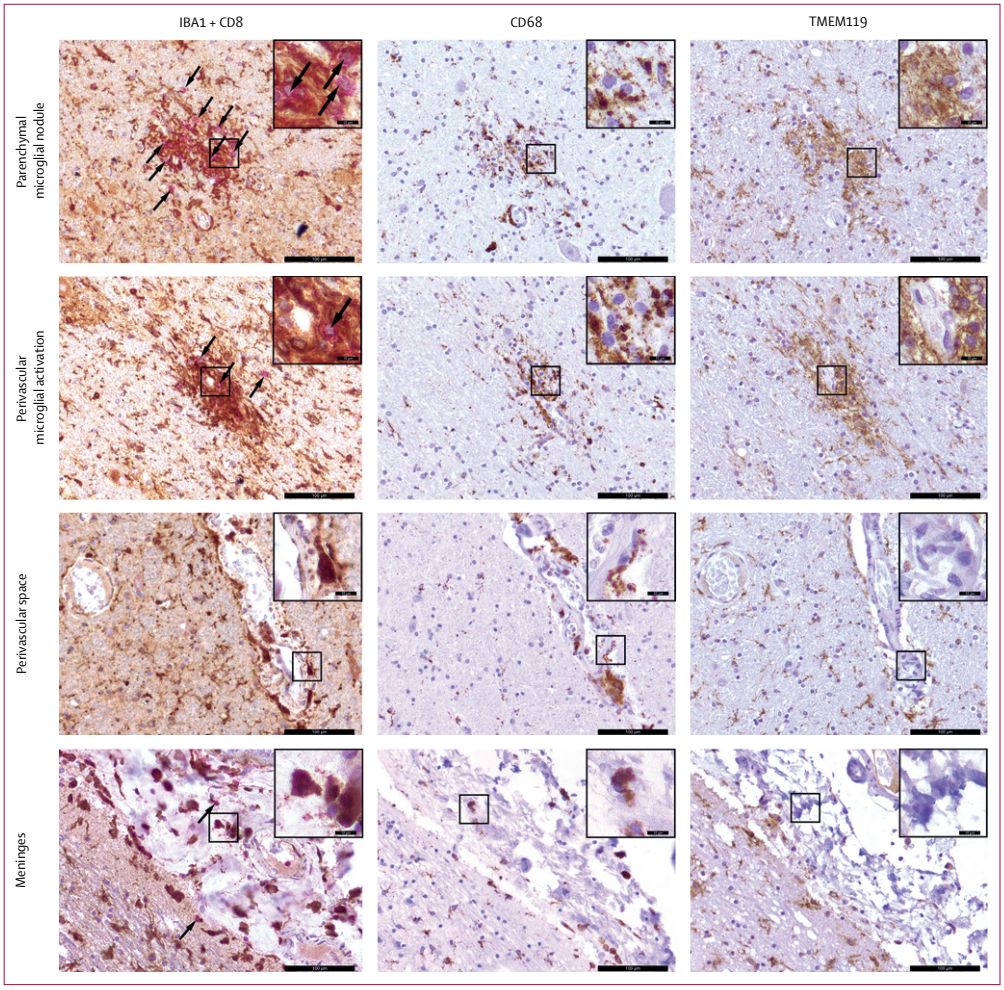


Figure 2: **Concomitant activation of the adaptive and innate immune systems in the brain of one patient (case 2) who died from COVID-19**
Representative images of double-chromogenic immunohistochemical labelling for IBA1 (brown) and CD8 (pink), as well as immunohistochemical staining for CD68 (brown), and TMEM119 (brown) at different CNS interfaces in the upper medulla oblongata. Counterstaining was done with haematoxylin (blue). Scale bars represent 100 μm (10 μm in the inset images). Arrows indicate CD8-positive T cells.

severe in nine (21%; table). 13 brains (30%) showed gross macroscopic abnormalities (fresh territorial ischaemic lesions in six patients, older territorial ischaemic lesions in five patients, grey matter heterotopia in one patient with trisomy 21, and cerebellar metastasis of a non-small cell lung carcinoma in one patient; table).

There was no evidence of cerebral bleeding or small-vessel thromboses. We found rare instances (two cases) of neuronophagy, and no acute necrotising lesions. Six (14%) patients had fresh ischaemic infarctions: three in the territory of the posterior cerebral artery, two in the territory of the anterior cerebral artery, and one in the territory of the middle cerebral artery, which were most likely due to thromboembolic events. A highly variable degree of astrogliosis was seen in all patients, with 37 patients (86%) showing astrogliosis in all assessed regions. Diffuse activation of microglia, with occasional microglial nodules, was pronounced in the brainstem and cerebellum. Additionally, we found distinct positive staining for HLA-DR in subpial and subependymal regions, a pattern not commonly observed in classic encephalitis. Parenchymal and perivascular microglia expressed the lysosomal marker CD68 while retaining the microglia core marker TMEM119 on their surfaces (figures 1, 2; appendix p 5).

To assess which cell types in the CNS might be prone to SARS-CoV-2 infection, we did an in-silico analysis

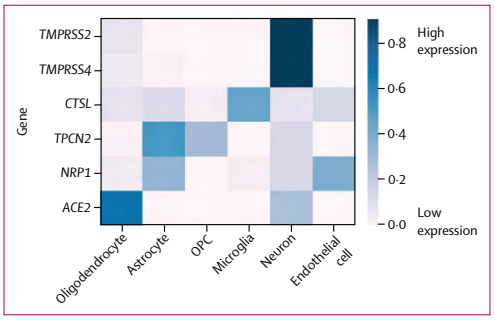


Figure 3: **In-silico analysis of the distribution of genes relevant to severe acute respiratory syndrome coronavirus 2 in the CNS**
Human temporal lobe cell type-specific expression of *TMPRSS2*, *TMPRSS4*, *CTSL*, *TPCN2*, *NRP1*, and *ACE2*. The heatmap shows the per-gene normalised mean expression across cell types (expression sums to 1 across the cell types). OPC=oligodendrocyte precursor cell.

of publicly available datasets. The analysis focused on cell-specific expression within the cerebral cortex of genes that have been shown to contribute to viral entry into the cell[27] and viral persistence,[25] including *ACE2, TMPRSS2, TPCN2, TMPRSS4, NRP1,* and *CTSL*. Our analysis showed that these genes are expressed in neurons, glial cells, and endothelial cells, suggesting their possible capacity to support SARS-CoV-2 infection. Expression of *ACE2* was highest in oligodendrocytes, while *TMPRSS2* and *TMPRSS4* were highest in neurons, *CTSL* was highest in microglia, and *TPCN2* was highest in astrocytes (figure 3).

Cytotoxic T lymphocytes could be seen in small amounts (up to 49 cells per HPF) in the frontal cortex and basal ganglia, but their presence was more pronounced in the brainstem, where they were mostly concentrated in perivascular regions but were also observed in the parenchyma as clusters in close vicinity to IBA-1-positive microglia (figures 1, 4). Infiltration of the meningeal compartments by cytotoxic T lymphocytes was seen in 34 (79%) patients (moderate in six patients and mild in 28 patients; table, figure 4), with an enrichment of IBA-1-postive, CD68-positive, and TMEM119-negative perivascular and meningeal macrophages (figure 2). The olfactory bulb showed a high degree of astrogliosis and microgliosis, but only minor infiltration by cytotoxic T lymphocytes (figures 1, 4).

SARS-CoV-2 RNA was detected by qRT-PCR in cryopreserved frontal lobe tissue from nine (39%) of 23 patients with available samples, and in FFPE medulla oblongata tissue from four (50%) of eight patients with available samples. In total, SARS-CoV-2 was found in the brain tissues of 13 (48%) of 27 patients who had at least one available sample (four patients had both types of sample available; figure 4). A median 4700 copies of SARS-CoV-2 RNA were detected (IQR 1350–29 400; range <1000 to $1{\cdot}62\times10^5$) among these 13 cases.

Samples from 40 (93%) patients underwent immunohistochemical staining for SARS-CoV-2 spike and nucleocapsid proteins. SARS-CoV-2-positive structures (cells and nerve fibres) were found scattered throughout the brain tissue. In eight (61%) of the 13 cases for which SARS-CoV-2 was detected in the brain by qRT-PCR, at least one SARS-CoV-2 protein could be detected (both spike and nucleocapsid in four cases, spike alone in three cases, and nucleocapsid alone in one case). Notably, in eight patients who were untested or tested negative on qRT-PCR analysis of SARS-CoV-2 RNA in the brain tissues, viral proteins were detectable by immunohistochemistry in the medulla oblongata (spike and nucleocapsid in one case, nucleocapsid alone in one case, and spike alone in six cases; figure 4). In the 16 (40%) cases positive for SARS-CoV-2 proteins on immunohistochemistry, spike protein was detected much more frequently (14 [88%] cases) than nucleocapsid protein (seven [44%] cases). By immunohistochemistry, SARS-CoV-2 could be mapped to isolated cells within the medulla oblongata and in the cranial nerves (either the glossopharyngeal or vagal nerves) originating from the brainstem (figure 5, appendix p 4). Overall, SARS-CoV-2 RNA or proteins were detected in the brain tissues of 21 (53%) of the 40 investigated patients, with eight (20%) patients having both SARS-CoV-2 RNA and protein detected (figure 4).

Cases 2, 16, 21, and 41 showed more brainstem inflammation (in terms of infiltration by cytotoxic CD8-positive T cells or activation of microglia) than all others (figure 4). Among these four cases, viral proteins were detected in the brain of one patient (case 2) and viral RNA in the brains of two patients (cases 2 and 21), and none died under ICU treatment.

Discussion

To our knowledge, this is the most comprehensive report of neuropathological findings of patients who died from COVID-19. In this post-mortem case series, we observed substantial yet highly variable degrees of astrogliosis in all assessed regions. Astrocytes are key regulators of homoeostasis, responding to stimuli through upregulation of GFAP and astroglial hypertrophy.[28] Because astrogliosis occurs in a variety of pre-existing medical conditions, and because critical illness also contributes to astrogliosis, a causal connection to SARS-CoV-2 cannot be drawn at present.

Activation of microglia and infiltration of cytotoxic T lymphocytes were mostly confined to the brainstem and cerebellum, with little involvement of the frontal lobe, in line with clinical findings pointing to an involvement of the brainstem.[5] The staining pattern of activated microglia with occasional microglial nodules is reminiscent of mild viral and autoimmune encephalitides.[29] We observed cytotoxic T lymphocytes in close vicinity to IBA-1-positive microglia, suggesting that these glial cells activate lymphocytes and potentially induce T-cell stimulation.[30] Microglia strongly expressed the lysosomal marker CD68, indicating their increased phagocytic activity. Notably, highly activated phagocyting microglia retained the microglial core marker

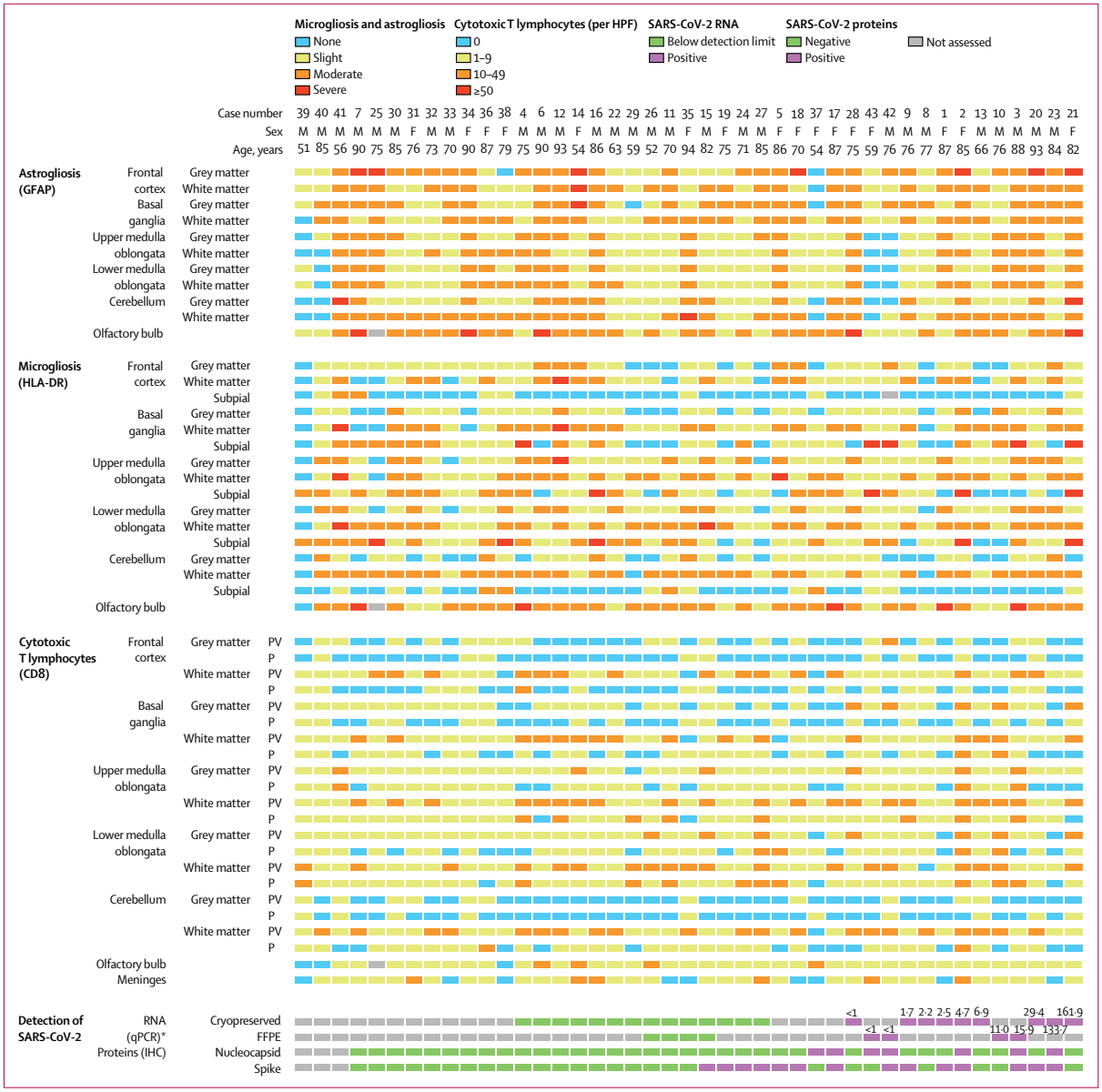


Figure 4: **Neuropathological findings and SARS-CoV-2 viral loads in studied patients (n=43)**
Cases are arranged from left to right on the basis of the presence and quantity of SARS-CoV-2 in the brain. F=female. FFPE=formalin-fixed paraffin-embedded. HPF=high-power field. IHC=immunohistochemistry. M=male. P=parenchymal. PV=perivascular. qPCR=quantitative PCR. SARS-CoV-2=severe acute respiratory syndrome coronavirus 2. *Values shown for positive cases represent number of copies of SARS-CoV-2 RNA ($\times 10^3$/mL); detection was done in the frontal lobe in cryopreserved specimens and in the upper medulla oblongata in FFPE specimens.

TMEM119 on their surfaces. Anosmia has been linked to COVID-19,[5] and might be related to the pronounced astrogliosis and microgliosis in the olfactory bulb observed in this study.

Clinically, nuchal rigidity has been identified in patients with COVID-19 as a possible sign of SARS-CoV-2-associated meningitis.[10] We found mostly mild meningeal infiltrates consisting of cytotoxic T lymphocytes, most likely

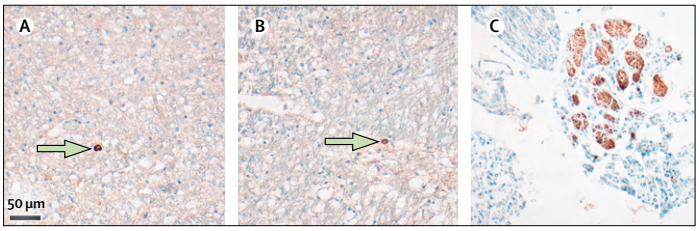


Figure 5: **Distribution of SARS-CoV-2 within the CNS**
Representative images of viral protein-positive cells (green arrows) in the medulla oblongata detected by anti-nucleocapsid protein antibody (A) or anti-spike protein antibody (B). (C) SARS-CoV-2 nucleoprotein (brown staining) could also be detected in subsets of cranial nerves originating from the lower brainstem. SARS-CoV-2=severe acute respiratory syndrome coronavirus 2.

indicative of non-specific meningeal reaction rather than viral meningitis.

CNS involvement with destructive lesions has been documented in cases of SARS-CoV infection.[31,32] Additionally, influenza A virus infects the CNS in some patients with a severe disease course. In these patients, acute subarachnoid haemorrhage and acute necrotising encephalopathy with multifocal brain lesions have been observed,[29] and such findings have also been reported in one patient with COVID-19.[4] Pan-encephalitis and intracranial haemorrhage[19] in addition to myelin loss has been described in patients with COVID-19.[18] In our cohort, we did not find evidence of myelin loss, cerebral bleeding, or acute necrotising lesions, congruent with previous findings.[16] Whether the absence of these pathologies was due to the small number of patients in our study or because they are not caused by SARS-CoV-2 remains an open question. Intracranial haemorrhagic lesions can also occur as a result of necessary treatments in ICUs, such as extracorporeal membrane oxygenation.[33] Some patients with COVID-19 present with neurological symptoms indicative of cerebral ischaemia, and, in patients younger than 50 years of age, large-vessel stroke has been proposed to constitute a presenting feature.[7] We observed signs of fresh territorial ischaemic lesions in six (16%) patients, of whom four were older than the average age of our sample and two were younger (63 years and 59 years). Thus, patients with stroke were not disproportionately younger among our cases. Morphologically, the ischaemic lesions followed a vascular pattern and presumably were of thromboembolic origin. These data are in line with previously published data showing thromboembolic events in a proportion of patients with COVID-19.[2]

SARS-CoV-2 was detected by qRT-PCR or immunostaining in the brains of 21 (53%) of all tested patients. Furthermore, immunohistochemical analysis revealed viral proteins in the cranial nerves (either glossopharyngeal or vagal) originating from the lower medulla oblongata and in single cells within the medulla oblongata. Although qRT-PCR might lack sensitivity, and immunostaining for viral proteins is prone to artifacts, our findings are consistent with those of previous studies of SARS-CoV, in which viral proteins could be seen in single cells in the brains of some patients who died following infection with the virus.[31] Thus, effects of SARS-CoV-2 on the brainstem could be correlates of, or contribute to, unusually rapidly deteriorating respiratory function, as has been observed in some patients with COVID-19 given non-invasive ventilation.[34]

The presence of SARS-CoV-2 RNA and proteins in the brains of patients with COVID-19 in this study is in line with the hypothesis that SARS-CoV-2 can infiltrate the CNS.[14] However, the presence of SARS-CoV-2 was not associated with the severity of neuropathological changes. Thus, CNS damage and neurological symptoms might be due to additional factors such as cytokine storm, neuroimmune stimulation, and systemic SARS-CoV-2 infection, rather than by direct CNS damage caused by the virus. We saw a surprisingly uniform presentation of neuropathological findings (ie, activation of microglia, infiltration with CD8-positive T cells) in our patients, irrespective of the clinical severity of COVID-19 in each case. Notably, the neuropathological presentation in patients who died in a domestic setting or in a nursing home did not differ from that in patients who died in hospital wards or ICUs.

The main limitation of our study is its descriptive nature and the absence of age-matched and sex-matched controls. Our study cohort was assembled during the peak of the SARS-CoV-2 pandemic and, for logistic reasons, simultaneous collection of case controls was not feasible, and it was not possible to use historical controls because of differences in sampling protocols. Thus, the proposed mechanisms of viral entry, viral replication, and putative pathophysiological principles underlying tissue damage must be interpreted in this context. Furthermore, we assessed only a small number of post-mortem specimens, and the selected regions might not be fully representative of the whole brain. Sample preservation could have influenced the analyses. Additionally, due to the heterogeneity of the places of death of studied patients, no systematically validated clinical data (eg, systematically documented neurological information) were available, and establishing conclusive clinicopathological correlations was not possible.

In summary, our results show that SARS-CoV-2 RNA and proteins can be detected in the CNS. The brain shows mild neuropathological changes with pronounced neuroinflammation in the brainstem being the most common finding. However, the presence of SARS-CoV-2 in the CNS was not associated with the severity of neuropathological changes.[16,17] Careful neuropathological interpretation will be essential to disentangle which changes are attributable to SARS-CoV-2. All such changes must be mapped against neuropathological changes caused by pre-existing medical conditions often present in patients with COVID-19, as well as neuropathological changes caused by invasive treatments that are used in severe cases of COVID-19.[35]

Contributors

JM, CH, and MG designed the study. MG and JM wrote the manuscript. JM, ML, JPS, ASS, CE, LA, MDa, AH, AF, SP, SB, HM, KP, SK, MA, MP, MS, and MG performed experiments and collected or analysed the data. ML, LA, MDa, SK, and MA analysed the presence and distribution of the virus. JM, CH, SK, MP, MS, and MG performed morphological analyses. DSM and SB performed in single-cell gene expression analysis. All authors discussed the results. JM, SK, MDo, MP, MS, and MG created the figures. MG and CG reviewed and discussed clinicopathological interpretations. All authors read, edited, and approved the manuscript.

Declaration of interests

MDa reports grants from the German Center for Infection Research during the conduct of the study and grants from the German Research Foundation outside the submitted work. All other authors declare no competing interests.

Data sharing

Data collected for the study and data from sample analyses will be made available upon reasonable request to the corresponding author.

Acknowledgments

We thank Ulrike Rumpf, Claudia Oye Attah, and Kristin Hartmann (Institute of Neuropathology, University Medical Center Hamburg-Eppendorf, Hamburg, Germany) for technical help. MG was supported by the German Research Foundation (SFB877). ML, MDa, LA, and MG were supported by Hamburg state research funding (Landesforschungsförderung; "mechanisms of cell-communication during infection"). DSM and SB were supported by eRARE Maxomod and the German Research Foundation (SFB1286).

References

1 Zou L, Ruan F, Huang M, et al. SARS-CoV-2 viral load in upper respiratory specimens of infected patients. *N Engl J Med* 2020; **382:** 1177–79.

2 Wichmann D, Sperhake JP, Lutgehetmann M, et al. Autopsy findings and venous thromboembolism in patients with COVID-19: a prospective cohort study. *Ann Intern Med* 2020; **173:** 268–77.

3 Varga Z, Flammer AJ, Steiger P, et al. Endothelial cell infection and endotheliitis in COVID-19. *Lancet* 2020; **395:** 1417–18.

4 Poyiadji N, Shahin G, Noujaim D, Stone M, Patel S, Griffith B. COVID-19-associated acute hemorrhagic necrotizing encephalopathy: CT and MRI features. *Radiology* 2020; **296:** 201187.

5 Mao L, Jin H, Wang M, et al. Neurologic manifestations of hospitalized patients with coronavirus disease 2019 in Wuhan, China. *JAMA Neurol* 2020; **77:** 683–90.

6 Helms J, Kremer S, Merdji H, et al. Neurologic features in severe SARS-CoV-2 infection. *N Engl J Med* 2020; **382:** 2268–70.

7 Oxley TJ, Mocco J, Majidi S, et al. Large-vessel stroke as a presenting feature of COVID-19 in the young. *N Engl J Med* 2020; **382:** e60.

8 Huang YH, Jiang D, Huang JT. SARS-CoV-2 detected in cerebrospinal fluid by PCR in a case of COVID-19 encephalitis. *Brain Behav Immun* 2020; **87:** 149.

9 Bernard-Valnet R, Pizzarotti B, Anichini A, et al. Two patients with acute meningo-encephalitis concomitant to SARS-CoV-2 infection. *Eur J Neurol* 2020; **27:** e43–44.

10 Moriguchi T, Harii N, Goto J, et al. A first case of meningitis/encephalitis associated with SARS-coronavirus-2. *Int J Infect Dis* 2020; **94:** 55–58.

11 Gutierrez-Ortiz C, Mendez A, Rodrigo-Rey S, et al. Miller Fisher syndrome and polyneuritis cranialis in COVID-19. *Neurology* 2020; **95:** e601–05.

12 Zhao H, Shen D, Zhou H, Liu J, Chen S. Guillain-Barré syndrome associated with SARS-CoV-2 infection: causality or coincidence? *Lancet Neurol* 2020; **19:** 383–84.

13 Toscano G, Palmerini F, Ravaglia S, et al. Guillain-Barré syndrome associated with SARS-CoV-2. *N Engl J Med* 2020; **382:** 2574–76.

14 De Felice FG, Tovar-Moll F, Moll J, Munoz DP, Ferreira ST. Severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) and the central nervous system. *Trends Neurosci* 2020; **43:** 355–57.

15 Barton LM, Duval EJ, Stroberg E, Ghosh S, Mukhopadhyay S. COVID-19 autopsies, Oklahoma, USA. *Am J Clin Pathol* 2020; **153:** 725–33.

16 Solomon IH, Normandin E, Bhattacharyya S, et al. Neuropathological features of COVID-19. *N Engl J Med* 2020; published online June 12. https://doi.org/10.1056/nejmc2019373.

17 Schaller T, Hirschbuhl K, Burkhardt K, et al. Postmortem examination of patients with COVID-19. *JAMA* 2020; **323:** 2518.

18 Reichard RR, Kashani KB, Boire NA, Constantopoulos E, Guo Y, Lucchinetti CF. Neuropathology of COVID-19: a spectrum of vascular and acute disseminated encephalomyelitis (ADEM)-like pathology. *Acta Neuropathol* 2020; **140:** 1–6.

19 von Weyhern CH, Kaufmann I, Neff F, Kremer M. Early evidence of pronounced brain involvement in fatal COVID-19 outcomes. *Lancet* 2020; **395:** e109.

20 Puelles VG, Lutgehetmann M, Lindenmeyer MT, et al. Multiorgan and renal tropism of SARS-CoV-2. *N Engl J Med* 2020; **383:** 590–92.

21 Darmanis S, Sloan SA, Zhang Y, et al. A survey of human brain transcriptome diversity at the single cell level. *Proc Natl Acad Sci USA* 2015; **112:** 7285–90.

22 Marouf M, Machart P, Bansal V, et al. Realistic in silico generation and augmentation of single-cell RNA-seq data using generative adversarial networks. *Nat Commun* 2020; **11:** 166.

23 Pfefferle S, Reucher S, Norz D, Lutgehetmann M. Evaluation of a quantitative RT-PCR assay for the detection of the emerging coronavirus SARS-CoV-2 using a high throughput system. *Euro Surveill* 2020; **25:** 2000152.

24 Tichopad A, Dilger M, Schwarz G, Pfaffl MW. Standardized determination of real-time PCR efficiency from a single reaction set-up. *Nucleic Acids Res* 2003; **31:** e122.

25 Rockx B, Kuiken T, Herfst S, et al. Comparative pathogenesis of COVID-19, MERS, and SARS in a nonhuman primate model. *Science* 2020; **368:** 1012–15.

26 Edler C, Schroder AS, Aepfelbacher M, et al. Dying with SARS-CoV-2 infection—an autopsy study of the first consecutive 80 cases in Hamburg, Germany. *Int J Legal Med* 2020; **134:** 1275–84.

27 Hoffmann M, Kleine-Weber H, Schroeder S, et al. SARS-CoV-2 cell entry depends on ACE2 and TMPRSS2 and is blocked by a clinically proven protease inhibitor. *Cell* 2020; **181:** 271–80.e8.

28 Verkhratsky A, Zorec R, Parpura V. Stratification of astrocytes in healthy and diseased brain. *Brain Pathol* 2017; **27:** 629–44.

29 Ludlow M, Kortekaas J, Herden C, et al. Neurotropic virus infections as the cause of immediate and delayed neuropathology. *Acta Neuropathol* 2016; **131:** 159–84.

30 Tröscher AR, Wimmer I, Quemada-Garrido L, et al. Microglial nodules provide the environment for pathogenic T cells in human encephalitis. *Acta Neuropathol* 2019; **137:** 619–35.

31 Gu J, Gong E, Zhang B, et al. Multiple organ infection and the pathogenesis of SARS. *J Exp Med* 2005; **202:** 415–24.

32 Desforges M, Le Coupanec A, Stodola JK, Meessen-Pinard M, Talbot PJ. Human coronaviruses: viral and cellular factors involved in neuroinvasiveness and neuropathogenesis. *Virus Res* 2014; **194:** 145–58.

33 Le Guennec L, Cholet C, Huang F, et al. Ischemic and hemorrhagic brain injury during venoarterial-extracorporeal membrane oxygenation. *Ann Intensive Care* 2018; **8:** 129.

34 Li YC, Bai WZ, Hashikawa T. The neuroinvasive potential of SARS-CoV2 may play a role in the respiratory failure of COVID-19 patients. *J Med Virol* 2020; **92:** 552–55.

35 Glatzel M. Neuropathology of COVID-19: where are the neuropathologists? *Brain Pathol* 2020; **30:** 729.

3.3 SCADEN, 2020

SYSTEMS BIOLOGY

Deep learning–based cell composition analysis from tissue expression profiles

Kevin Menden[1]*, Mohamed Marouf[2], Sergio Oller[2], Anupriya Dalmia[1], Daniel Sumner Magruder[2,3], Karin Kloiber[2], Peter Heutink[1], Stefan Bonn[1,2]*

We present Scaden, a deep neural network for cell deconvolution that uses gene expression information to infer the cellular composition of tissues. Scaden is trained on single-cell RNA sequencing (RNA-seq) data to engineer discriminative features that confer robustness to bias and noise, making complex data preprocessing and feature selection unnecessary. We demonstrate that Scaden outperforms existing deconvolution algorithms in both precision and robustness. A single trained network reliably deconvolves bulk RNA-seq and microarray, human and mouse tissue expression data and leverages the combined information of multiple datasets. Because of this stability and flexibility, we surmise that deep learning will become an algorithmic mainstay for cell deconvolution of various data types. Scaden's software package and web application are easy to use on new as well as diverse existing expression datasets available in public resources, deepening the molecular and cellular understanding of developmental and disease processes.

INTRODUCTION

The analysis of tissue-specific gene expression using next-generation sequencing [RNA sequencing (RNA-seq)] is a centerpiece of the molecular characterization of biological and medical processes (*1*). A well-known limitation of tissue-based RNA-seq is that it typically measures average gene expression across many molecularly diverse cell types that can have distinct cellular states (*2*). A change in gene expression between two conditions can therefore be attributed to a change in the cellular composition of the tissue or a change in gene expression in a specific cell population, or a mixture of the two. To deconvolve the cell type composition from a change in gene expression is especially important in systems with cellular proliferation (e.g., cancer) or cellular death (e.g., neuronal loss in neurodegenerative diseases) due to systematic cell population differences between experimental groups (*3*).

To account for this problem, several computational cell deconvolution methods have been proposed during the last years (*4*, *5*). These algorithms use gene expression profiles (GEPs) of cell type–specifically expressed genes to estimate cellular fractions using linear regression to detect, interpret, and possibly correct for systematic differences in cellular abundance between samples (*4*). While the best-performing linear regression algorithms for deconvolution seem to be variations of support vector regression (*6*–*10*), the selection of an optimal GEP is a field of active research (*10*, *11*). It has been recently shown that the design of the GEP is the most important factor in most deconvolution methods, as results from different algorithms strongly correlate given the same GEP (*11*).

In theory, an optimal GEP should contain a set of genes that are predominantly expressed within each cell population of a complex sample (*12*). They should be stably expressed across experimental conditions, for example, across health and disease, and resilient to experimental noise and bias. However, bias is typically inherent to biomedical data and is imparted, for instance, by intersubject variability, variations across species, different data acquisition methods, different experimenters, or different data types. The negative impact of bias on deconvolution performance can be partly improved by using large, heterogeneous GEP matrices (*11*). It is therefore expected that recent advancement in cell deconvolution relied almost exclusively on sophisticated algorithms to normalize the data and engineer optimal GEPs (*10*).

While GEP-based approaches lay the foundational basis of modern cell deconvolution algorithms, we hypothesize that deep neural networks (DNNs) could create optimal features for cell deconvolution, without relying on the complex generation of GEPs. DNNs such as multilayer perceptrons are universal function approximators that achieve state-of-the-art performance on classification and regression tasks. Whereas this feature is of little importance for strictly linear input data, it makes DNNs superior to linear regression algorithms as soon as data deviate from ideal linearity. This means, for instance, that as soon as data are noisy or biased and classical linear regression algorithms may falter, the hidden layer nodes of the DNN learn to represent higher-order latent representations of cell types that do not depend on input noise and bias. We theorize, therefore, that by using gene expression information as network input, hidden layer nodes of the DNN would represent higher-order latent representations of cell types that are robust to input noise and technical bias.

An obvious limitation of DNNs is the requirement for large training data to avoid overfitting of the machine learning model. While ground-truth information on tissue RNA-seq cell composition is scarce, one can use single-cell RNA-seq (scRNA-seq) data to obtain large numbers of in silico tissue datasets of predefined cell composition (*7*–*9*, *13*–*15*). We do this by subsampling and subsequently merging cells from scRNA-seq datasets, this approach being limited only by the availability of tissue-specific scRNA-seq data. It is to be noted that scRNA-seq data suffer from biases, such as dropout, to which RNA-seq data are not subject (*16*). While this complicates the use of scRNA-seq data for GEP design (*8*), we surmise that latent network nodes could represent features that are robust to these biases.

On the basis of these assumptions, we developed a single cell–assisted deconvolutional DNN (Scaden) that uses simulated bulk RNA-seq samples for training and predicts cell type proportions for input expression samples of cell mixtures. Scaden is available as downloadable software package and web application (https://scaden.ims.bio).

[1]German Center for Neurodegenerative Diseases, Tuebingen, Germany. [2]Institute of Medical Systems Biology, University Medical Center Hamburg-Eppendorf, Hamburg, Germany. [3]Genevention GmbH, Goettingen, Germany.
*Corresponding author. Email: sbonn@uke.de (S.B.); kevin.menden@dzne.de (K.M.)

Scaden is trained on publicly available scRNA-seq and RNA-seq data, does not rely on specific GEP matrices, and automatically infers informative features. Last, we show that Scaden deconvolves expression data into cell types with higher precision and robustness than existing methods that rely on GEP matrices.

RESULTS

Scaden overview, model selection, and training

In this part, we focus on the design and optimization of Scaden by training, validation, and testing on in silico data. Note that the generation of in silico data is a strictly linear mathematical operation. Our aim in this context, to corroborate Scaden's basic functionality, is to show that Scaden's performance compares with (but not necessarily exceeds) that of state-of-the-art algorithms.

The basic architecture of Scaden is a DNN that takes gene counts of RNA-seq data as input and outputs predicted cell fractions (Fig. 1). To optimize the performance of the DNN, it is trained on data that contain both the gene expression and the real cell type fraction information (Fig. 1B). The network then adjusts its weights to minimize the error between the predicted cell fractions and the real cell fractions (Fig. 1C). We restricted feature selection to the removal of "uninformative" genes that have either zero expression or an expression variance below 0.1, leaving ~10,000 genes for training. In our hands, this feature selection step decreases training time and memory usage.

For the model selection and training, we made use of the large numbers of artificial bulk RNA-seq datasets with defined composition that can be generated in silico from published scRNA-seq and RNA-seq datasets (simulated tissues; Fig. 1A and tables S1 and S2). The only constraint is that the scRNA-seq and RNA-seq data must come from the same tissue as the bulk data subject to deconvolution.

To find the optimal DNN architecture for cell deconvolution, we generated bulk peripheral blood mononuclear cell (PBMC) RNA-seq data from four publicly available scRNA-seq datasets (tables S1 and S3). We performed leave-one-dataset-out cross-validation, training Scaden on mixtures of synthetic datasets from three scRNA-seq datasets and evaluating the performance on simulated tissue from a fourth scRNA-seq dataset.

We used the root mean square error (RMSE), Pearson's correlation coefficient (r), the slope and intercept of the regression fitted for ground-truth and predicted cell fractions, and Lin's concordance correlation coefficient (CCC) (*17*) to assess algorithmic performance. The CCC is a measure sensitive not only to scatter but also to deviations from linearity (slope and intercept). Within the main text, we report on CCC and RMSE values only; other metrics can be found in the Supplementary Materials.

The final Scaden model is an ensemble of the three best-performing models (table S4), and the final cell type composition estimates are the averaged predictions of all three ensemble models (Fig. 1 and fig. S1). Using an ensemble of models increased the deconvolution performance as compared to single best models (table S6). Details of the model and hyperparameters are given in table S5. We also evaluated the effect of the size of the training dataset on Scaden deconvolution performance, repeating leave-one-dataset-out cross-validation on PBMC data with training dataset sizes from 150 up to 15,000 samples (fig. S2). The increase in CCC value starts to level off from about 1500 simulated samples for this dataset but continues to increase slowly with sample size. We specifically addressed the question to what degree the DNN, trained on simulated samples, tends to overfit, failing to generalize to real bulk RNA-seq data. To understand after how many steps a model trained on in silico data overfits on real RNA-seq data, we trained Scaden on simulated data from an ascites scRNA-seq dataset (table S1; 6000 samples) and evaluated the loss function on a corresponding annotated RNA-seq dataset (*18*) (table S2; three samples) as a function of the number of steps (fig. S3). All models converged after approximately 5000 steps and slightly overfit when trained for longer. On the basis of this result, we opted for an early-stop approach after 5000 steps for evaluation on real bulk RNA-seq data.

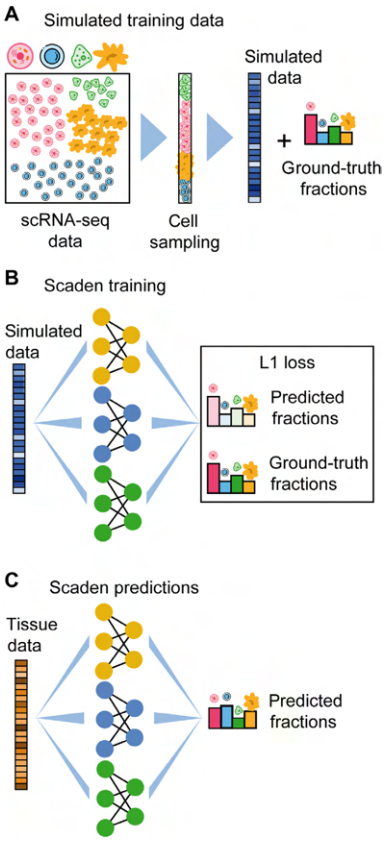


Fig. 1. Overview of training data generation and cell type deconvolution with Scaden. (**A**) Artificial bulk samples are generated by subsampling random cells from an scRNA-seq dataset and merging their expression profiles. (**B**) Model training and parameter optimization on simulated tissue RNA-seq data by comparing cell fraction predictions to ground-truth cell composition. (**C**) Cell deconvolution of real tissue RNA-seq data using Scaden.

We then compared Scaden to four state-of-the-art GEP-based cell deconvolution algorithms, CIBERSORT (CS) (*6*), CIBERSORTx (CSx) (*7*), Multi-subject Single Cell deconvolution (MuSiC) (*8*), and Cell Population Mapping (CPM) (*9*). While CS relies on hand-curated GEP matrices, CSx, MuSiC, and CPM can generate GEPs using scRNA-seq data as input.

To get an initial estimate of Scaden's deconvolution fidelity, we trained the model on 24,000 simulated PBMC RNA-seq samples from three datasets and tested its performance in comparison to CS, CSx,

MuSiC, and CPM on a fourth dataset of 500 samples each (e.g., training on data6k, data8k, and donorA and evaluation on donorC). We used corresponding scRNA-seq datasets for the construction of GEPs as input for CSx and MuSiC, and CPM. For CS, we used the PBMC-optimized LM22 GEP matrix (*6*), which was developed by the CS authors for the deconvolution of human PBMC data.

For two of four test datasets (donorA and donorC), Scaden obtained the highest CCC and lowest RMSE, followed by CSx, MuSiC, CS, and CPM (fig. S4 and table S7). CSx and MuSiC obtained the highest CCC values for the data8k and data6k datasets, respectively. Scaden obtained the highest average CCC and lowest RMSE (0.88 and 0.08, respectively), followed by MuSiC (0.85 and 0.10), CSx (0.83 and 0.11), CS (0.63 and 0.15), and CPM (0 and 0.20; fig. S4). As expected, all algorithms that use scRNA-seq data as reference performed well, with the notable exception of CPM. We want to mention that CPM focuses on the reconstruction of continuous spectra of cellular states, while it incorporates cell deconvolution as an additional feature. We therefore report CPM's deconvolution performance in the Supplementary Materials from here on. On average, Scaden also obtained the highest correlation and the best intercept and slope values on simulated PBMC data (table S7). A closer inspection on a per–cell type basis (Fig. 2A) revealed that Scaden yields consistently higher CCC values and lower RMSEs when compared to the other algorithms.

A specific feature of the MuSiC algorithm is that it preferentially weighs genes according to low intersubject and intracell cluster variability for its GEP, which increases deconvolution robustness when high-expression heterogeneity is observed between human participants, for example (*8*). To understand whether Scaden can use multisubject information to increase its deconvolution performance, we trained Scaden, CSx, and MuSiC on scRNA-seq pancreas data from several participants (*19*) and assessed the performance on a separate simulated pancreas RNA-seq dataset (*20*). To allow for direct comparison, we chose the same pancreas training and test datasets that were used in the original MuSiC publication (table S1). To enable Scaden to leverage the heterogeneity of multisubject data, training data were generated separately for every participant in the dataset (see Methods). CSx cannot profit from multisubject data but performed well on the artificial PBMC datasets and was therefore included in the comparison. The best average performance (across cell types) is achieved by Scaden (CCC = 0.98), closely followed by MuSiC (CCC = 0.93), while CSx does not perform as well (CCC = 0.75; Fig. 2B and table S8). On a per–cell type basis, Scaden's predictions are clearly superior to the other two algorithms for all cell types. This provides strong evidence that Scaden, by separating training data generation for each participant, can learn intersubject heterogeneity and outperform specialized multisubject algorithms such as MuSiC on the cell type deconvolution task.

In addition, we wanted to test how the best-performing deconvolution algorithms Scaden, MuSiC, and CSx behave when unknown cell content is part of the mixture. To test this, all cells falling into the "Unknown" category were removed from the training or reference PBMC datasets but added to the simulated mixture samples at fixed percentages (5, 10, 20, and 30%; see Methods). Scaden obtains the highest CCC for all tested percentages of unknown cell content (fig. S5 and table S9). The general deconvolution performance declines linearly with increasing percentage of unknown content for all tested algorithms, indicating that Scaden, MuSiC, and CSx have a similar robustness against unknown mixture content.

We next compared the runtime and memory footprint of Scaden and MuSiC on an Intel Xeon six-core central processing unit (CPU)

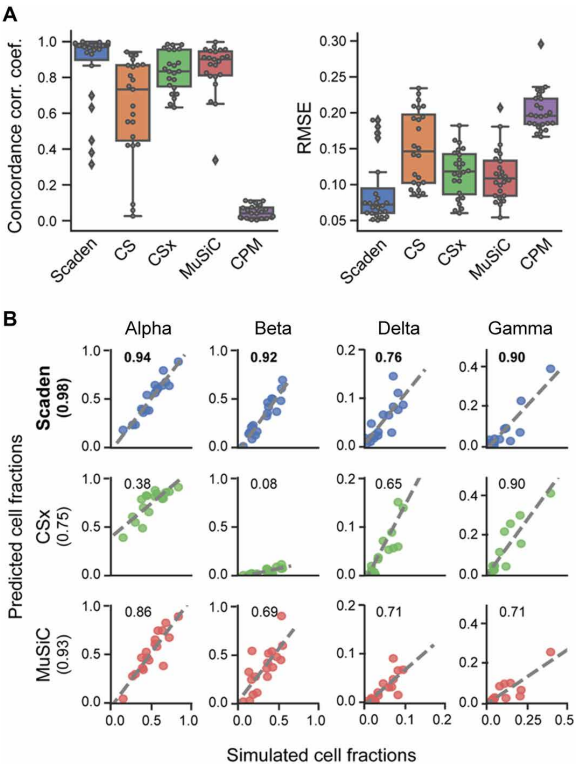


Fig. 2. Performance comparison of deconvolution algorithms on simulated tissue data. (**A**) Boxplots of the cell type prediction CCC and RMSE for four simulated PBMC datasets. Tables S14 and S16 contain information on the five (six for CS) cell types used. (**B**) Scatterplots for four pancreas cell types of ground-truth (*x* axis) and predicted values (*y* axis) for Scaden, CSx, and MuSiC on artificial pancreas data (*20*). Numbers inside the plotting area and in parenthesis signify CCC values.

to the runtime of the CSx web application. Scaden is the only algorithm that requires the generation of in silico training data, which takes 13 min for 2000 samples with a peak memory usage of 8 GB. Similar values were obtained for the human brain data. Next, we used the PBMC data to benchmark the runtime and memory consumption of the deconvolution task. For Scaden, model training took ~11 min and cell fraction prediction ~8 s for 500 samples, using less than 1-GB memory. We used the web application of CSx with batch correction to deconvolve the 500 PBMC samples in 35 min. MuSiC took only 2 min and 15 s to deconvolve all 500 samples, with the memory usage peaking at 4.5 GB. As Scaden can take advantage of a graphics processing unit (GPU), we additionally compared training duration on an AMD Ryzen 5 2600 CPU and GeForce RTX 2600 GPU on the same machine. Training on the CPU took 9 min and 39 s, while it took only 3 min and 2 s on the GPU, corresponding to a roughly three times shorter runtime for Scaden if a GPU is available.

Robust deconvolution of bulk expression data

The true use case of cell deconvolution algorithms is the cell fraction estimation of tissue RNA-seq data. In particular for noisy and biased bulk RNA-seq data, we hypothesize that Scaden's latent feature

representations might help it to more robustly predict cell fractions as compared to GEP-based algorithms.

We therefore assessed the performance of Scaden, CS, CSx, and MuSiC to deconvolve two publicly available human PBMC bulk RNA-seq datasets, for which curated GEP matrices and RNA-seq data with associated ground-truth cell type compositions from flow cytometry are available (see the "Data availability" section). We will refer to these datasets that consists of 12 samples each as PBMC1 (*21*) and PBMC2 (*10*) (table S2). Both datasets have similar cell type compositions across samples, with CD4 and CD8 T cells making up the biggest fractions. Deconvolution for all methods was performed as described in the previous section, with the difference that data from all four PBMC scRNA-seq datasets were now deployed for Scaden training. Results are given in Fig. 3 (A to C) and tables S10 and S11.

On the PBMC1 dataset and using all cell types, Scaden obtained the highest CCC and lowest RMSE (0.56 and 0.13), while CSx (0.55 and 0.16) and CS (0.43 and 0.15) performed well yet notably worse than Scaden (Fig. 3A and tables S10 and S11). CPM (0 and 0.18) and MuSiC (−0.19 and 0.32) both failed to deconvolve the cell fractions of the PBMC1 data. Scaden also obtained the best CCC and RMSE (0.68 and 0.08) on the PBMC2 dataset, while CS (0.58 and 0.10) and CSx (0.42 and 0.13) obtained good deconvolution results. Similar to the PBMC1 data deconvolution results, CPM (−0.16 and 0.11) and MuSiC (−0.13 and 0.30) did not perform well on the PBMC2 deconvolution task. In addition to CCC and RMSE metrics, Scaden achieves the best correlation, intercept, and slope on both PBMC datasets (tables S10 and S11).

In particular, Scaden outperforms classical algorithms on a per–cell type basis (Fig. 3, B and C). These results show weaker correlations and a strong dependence on the cell type. A closer examination of the metrics in table S11 and fig. S6 shows that the largest variations are found in the slope and intercept.

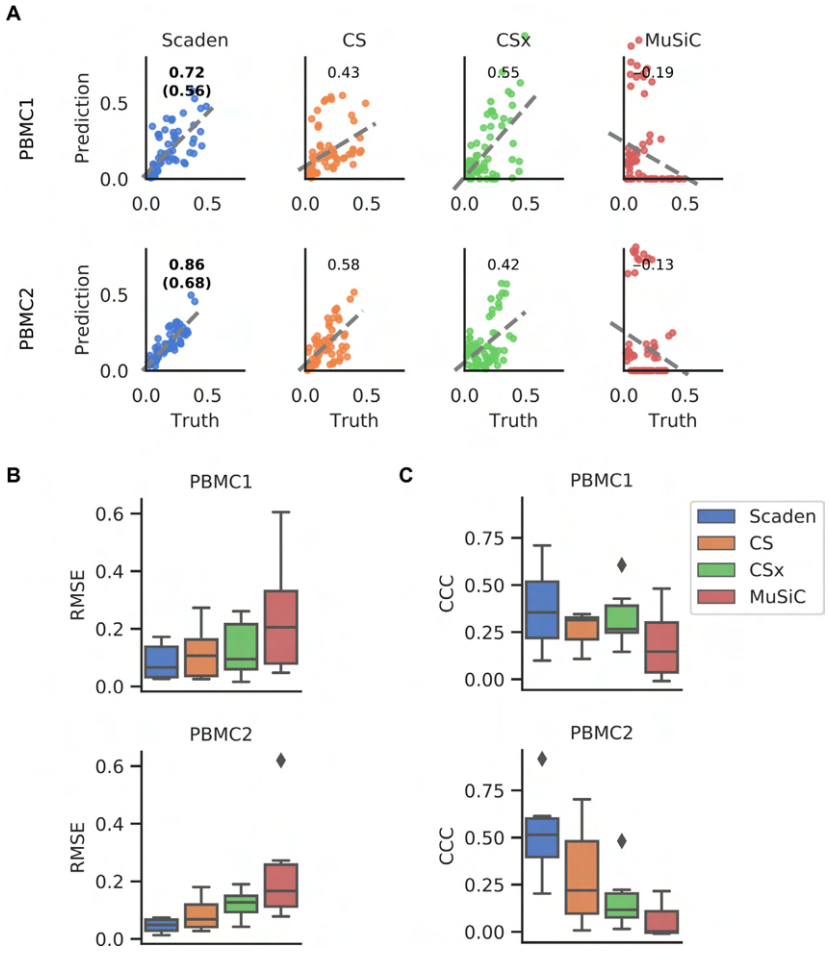


Fig. 3. Comparison of deconvolution algorithms on PBMC tissue RNA-seq data. (**A**) Per–cell type scatterplots of ground-truth (*x* axis) and predicted values (*y* axis) for Scaden, CS, CSx, and MuSiC on real PBMC1 and PBMC2 cell fractions. Numbers inside the plotting area signify CCC values. For Scaden, the CCC using only scRNA-seq training data is shown in parenthesis, and the CCC using mixed scRNA-seq and RNA-seq training data is shown without parentheses. (**B**) Boxplots of RMSE values for real PBMC1 and PBMC2 data. (**C**) CCC values for real PBMC1 and PBMC2 data.

We further evaluated how good the Scaden ensemble performs compared to the best single DNN model (M512, 512 nodes input layer). While the M512 model shows good deconvolution performance on the PBMC1 (CCC, 0.57) and PBMC2 (CCC, 0.68) datasets, the ensemble model achieves the best average cross-validation performance (table S6). We therefore opted to use the ensemble method to reduce interdataset performance variation observed with M512 and other single models.

An additional algorithmic feature of Scaden is that it seamlessly integrates increasing amounts of training data, which can be of different types, such as a combination of simulated tissue and real tissue data with cell fraction information. In theory, even limited real tissue training data could make Scaden robust to data type bias and consequently improve Scaden's deconvolution performance on real tissue data. We therefore trained Scaden on a mix of simulated PBMC and real PBMC2 (12 samples) data and evaluated its performance on real PBMC1 data (Fig. 3, A and B, fig. S6, and tables S10 and S11). While the training contained very little (~2%) real data, Scaden's CCC increased from 0.56 to 0.72, and the RMSE decreased from 0.13 to 0.10. We observed similar performance increases when Scaden was trained on simulated PBMC and real PBMC1 data and evaluated on real PBMC2 data (Fig. 3, A and B, fig. S6, and tables S10 and S11). Next, we wanted to investigate how a Scaden model trained on only few real samples compares to the models trained on simulated or simulated and real data. While a Scaden model trained on only bulk PBMC1 samples ($n = 12$) deconvolves PBMC2 data with a CCC of 0.62, it does not reach the CCC of models trained on simulated data (CCC of 0.68) or on simulated and bulk data (CCC of 0.86). We would also not advise training models on so few training samples, as these models are usually overfit.

This further validates that Scaden reliably deconvolves tissue RNA-seq data into the constituent cell fractions and that very accurate deconvolution results can be obtained if reference and target datasets are from the same experiment.

We next wanted to test how the algorithm performs on postmortem human brain tissue of a subsample from the Religious Orders Study and Memory and Aging Project (ROSMAP) study (*22*), for which ground-truth cell composition information was recently measured by immunohistochemistry (41 samples with all cell types given) (*23*). The data provided by this study consist of bulk RNA-seq data from the dorsolateral prefrontal cortex and pose a special challenge due to the complexity of its cell type composition, which is further complicated by the fact that the data originate from brains of healthy individuals as well as patients with Alzheimer's disease (AD) at various stages of neuronal loss. As reference datasets, we used the scRNA-seq dataset provided by Darmanis *et al.* (*24*) from the anterior temporal lobe of living patients and the Lake dataset that isolates nuclei of neurons from two (visual and frontal) cortical regions from a postmortem brain and subjects them to RNA-seq (*25*). From these, we generated 2000 training samples (Darmanis) and 4000 samples (two regions from the Lake dataset).

Figure 4A shows the deconvolution results for all three algorithms with the Darmanis (scRNA-seq) reference dataset. Scaden achieves the highest CCC value (0.92) followed by MuSiC (0.87) and CSx (0.81; table S12). Compared to Scaden, MuSiC and CSx overestimate neural percentages, leading to higher RMSE values of 0.09 and 0.12, respectively (Scaden, 0.06). Notably, all methods showed a lower CCC on the per–cell type level (Fig. 3B), demonstrating that some per–cell type correlations are poor, either in slope, intercept, variance, or a combination of them. This emphasizes the need for a cell type–specific inspection of results and highlights that, depending on the dataset, cell type–specific deconvolution results can be far from perfect.

In addition to comparing the predictive power of Scaden, CSx, and MuSiC on human brain tissue with different reference datasets,

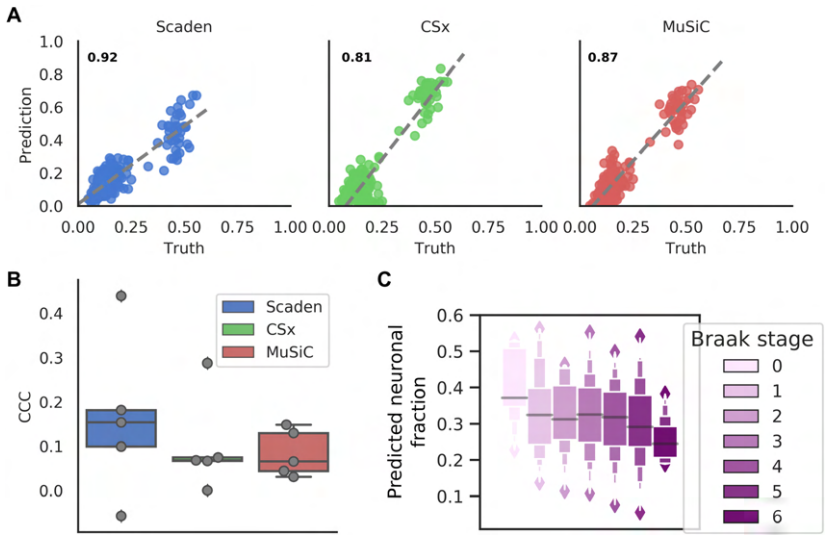


Fig. 4. Deconvolution performance comparison on brain tissue RNA-seq data. (**A**) Prediction of human brain cell fractions of the ROSMAP dataset using the Darmanis dataset as a reference: scatterplots of ground-truth (*x* axis) and predicted values (*y* axis) for Scaden, CSx, and MuSiC of data. CCC values are shown as inserts. (**B**) Per–cell type CCC values for ROSMAP using the Darmanis data as a reference. (**C**) Neuronal content determined by Scaden trained on mouse brain data and evaluated on the Braak stage of the ROSMAP study.

we also tested how the choice of reference datasets affected Scaden's deconvolution results. Notably, all methods substantially drop in performance when the Lake single-nucleus RNA-seq dataset is used as a reference as we had presumed (fig. S7A). We want to emphasize that Scaden, in contrast to CSx and MuSiC, has the possibility to simultaneously use both datasets as reference, whereas for CSx and MuSiC, the user has to choose one of the two, unaware of which will give the correct results.

We found that the performance of Scaden was almost unaffected when the Lake dataset was added to the Darmanis training samples (CCC = 0.90, RMSE = 0.06; fig. S7A and table S12). These results show that cell deconvolution with Scaden is robust to training data bias (Darmanis single-cell versus Lake single-nucleus data). An added benefit of Scaden is that it allows for the inclusion and mixing of different scRNA-seq experiments in the training dataset, further increasing its robustness (fig. S7A). Last, when calculating the CCC values on a per-sample basis, Scaden achieves the best scores for most samples (fig. S7B).

In a next step, we wanted to assess whether Scaden's deconvolution performance was robust across species by trying to predict the cell fractions of the ROSMAP study (*22*) with a Scaden model trained on in silico data from five mouse brain scRNA-seq datasets (table S1). Intriguingly, Scaden was able to achieve a CCC value of 0.83 and an RMSE of 0.079, showing that Scaden can reliably deconvolve RNA-seq data across related species.

The ROSMAP study also contains information on the Braak stages (*26*) corresponding to 390 human postmortem prefrontal cortex samples, which correlate with the severity and progression stage of AD and the degree of neuronal loss. We used the Scaden model trained on artificial data generated from five mouse brain scRNA-seq datasets to predict neuronal cell fractions of this larger human dataset. Overall, Scaden's cell fraction predictions capture the increased neuronal loss with increasing Braak stage (Fig. 4C). The largest drop in neural percentage is observed at stage 5, when the neurodegeneration typically reaches the prefrontal cortex of the brain.

Given the robustness with which Scaden predicts tissue RNA-seq cell fractions using scRNA-seq training data, even across species, we next wanted to investigate whether an scRNA-seq–trained Scaden model can also deconvolve other data types. To this end, we measured the deconvolution performance on a bulk PBMC microarray dataset (20 samples) (*6*) of a Scaden model trained on scRNA-seq and RNA-seq PBMC data (see above). We compared Scaden to CS using the microarray-derived LM22 matrix. CS achieved a slightly higher CCC and slightly lower total RMSE (0.72 and 0.11) than Scaden (0.71 and 0.13), while Scaden obtained the highest average CCC (0.50) compared to CS (0.39; fig. S8 and table S13). Notably, in this scenario, Scaden was trained entirely on simulated scRNA-seq and RNA-seq data, while CS's LM22 GEP was optimized on PBMC microarray data.

Overall, we provide strong evidence that Scaden robustly deconvolves tissue data across tissues, species, and even data types.

DISCUSSION

Scaden is a novel deep learning–based cell deconvolution algorithm that, in many instances, compares favorably in both prediction robustness and accuracy to existing deconvolution algorithms that rely on GEP design and linear regression. We believe that Scaden's performance relies to a large degree on the inherent feature engineering of the DNN. The network does not only select features (genes) for regression but also creates new features that are optimal for the regression task in the nodes of the hidden layers. These hidden features are nonlinear combinations of the input features (gene expression), which makes it notoriously difficult to explain how a DNN works (*27*). It is important to highlight that this feature creation is fundamentally different from all other existing cell deconvolution algorithms, which rely on heuristics that select a defined subset of genes as features for linear regression.

Another advantage of this inherent feature engineering is that Scaden can be trained to be robust to input noise and bias (e.g., batch effects). Noise and bias are all prevalent in experimental data, because of different sample quality, sample processing, experimenters, and instrumentation, for example. If the network is trained on different datasets of the same tissue, however, then it learns to create hidden features that are robust to noise and bias, such as batch effects. This robustness is pivotal in real-world cell deconvolution use cases, where the bulk RNA data for deconvolution and the training data (and therefore the network and GEP) contain different noise and biases. In this study, we tested Scaden with training data from scRNA-seq datasets generated with a variety of different protocols and could not identify a specific protocol that is not suitable. While especially recent cell deconvolution algorithms include batch correction heuristics before GEP construction, Scaden optimizes its hidden features automatically when trained on data from various batches. Potential protocol-specific biases can therefore be alleviated when employing training data from multiple protocols.

The robustness to noise and bias, which might be due to hidden feature generation, is especially evident in Scaden's ability to deconvolve across data types. A network trained on in silico bulk RNA-seq data can seamlessly deconvolve microarray data of the same tissue. This is quite noteworthy, as microarray data are known to have a reduced dynamic range and several hybridization-based biases compared to RNA-seq data. In other words, Scaden can deconvolve bulk data of types that it has never been trained on, even in the face of strong data type bias. This raises the possibility that Scaden trained on scRNA-seq data might reliably deconvolve other bulk omics data as well, such as proteomic and metabolomic data. This assumption is strengthened by the fact that Scaden, trained on scRNA-seq data, attains state-of-the-art performance on the deconvolution of bulk RNA-seq data, two data types with very distinct biases (*16*).

As highlighted in the introduction, a drawback for many DNNs is the large amount of training data required to obtain robust performance. Here, we used scRNA-seq data to create in silico bulk RNA-seq data of predefined type (target tissue) with known composition, across datasets. This immediately highlights Scaden's biggest limitation, the dependency on scRNA-seq data of the target tissue. In this study, we have shown that Scaden, trained solely on simulated data from scRNA-seq datasets, can outperform GEP-based deconvolution algorithms. We did observe, however, that the addition of labeled RNA-seq samples to the training data did substantially improve deconvolution performance in the case of PBMC data. We therefore believe that efforts to increase the similarity between simulated training data and the target bulk RNA-seq data could increase Scaden's performance further. Mixtures of in silico bulk RNA-seq data and publically available RNA-seq data, of purified cell types, for example, could further increase the deconvolution performance of Scaden. Furthermore, domain adaptation methods can be used to improve performance of models that are trained on data (here, scRNA-seq data) that are similar to the target data (here, RNA-seq

data) (*28*). In future versions, Scaden's simple multilayer perceptron architecture could leverage domain adaptation to further stabilize and improve its cell deconvolution performance.

Scaden uses an ensemble approach by averaging the predictions of three different models to increase performance and improve generalization. Increasing the number of models per ensemble would allow for the estimation of the prediction uncertainty. While not implemented in this study, this could be an interesting extension to Scaden's ensemble architecture.

Recent cell deconvolution algorithms have used cell fraction estimates to infer cell type–specific gene expression from bulk RNA-seq data. It is straightforward to use Scaden's cell fraction estimates to infer per-group (*3*) and per-sample (*7*) cell type–specific gene expression using simple regression or non-negative matrix factorization, respectively. We would like to add a note of caution, however, as the error of cell fraction estimates, which can be quite large, is propagated into the gene expression calculations and will affect any downstream statistical analysis.

While Scaden achieves good performance on the samples and tissues used in this study, it is important to keep in mind that cell type similarity, sample heterogeneity, and complexity, as well as experimental noise and bias, can severely limit deconvolution accuracy. Furthermore, Scaden is currently not attempting to model cell size differences in its algorithm, which might be useful to consider for the interpretation of prediction results.

In summary, the deconvolution performance, robustness to noise and bias, and the flexibility to learn from large numbers of in silico datasets, across data types (scRNA-seq and RNA-seq mixtures) and potentially even tissues, make us believe that DNN-based architectures will become an algorithmic mainstay of cell type deconvolution.

METHODS

Datasets and preprocessing

scRNA-seq datasets

The following human PBMC scRNA-seq datasets were downloaded from the 10X Genomics data download page: 6k PBMCs from a Healthy Donor, 8k PBMCs from a Healthy Donor, Frozen PBMCs (Donor A), and Frozen PBMCs (Donor C) (*29*). Throughout this paper, these datasets are referred to with the handles data6k, data8k, donorA, and donorC, respectively. It was not intended to incorporate as many datasets as possible. Instead, these four datasets were chosen with the goal to dispose of a set of samples with consistent cell types and gene expression. This limited our choice to datasets that displayed clearly identifiable cell types for the majority of cells. The Ascites scRNA-seq dataset was downloaded from https://figshare.com as provided by Schelker *et al.* (*18*). Pancreas and mouse brain datasets were downloaded from the scRNA-seq dataset collection of the Hemberg laboratory (https://hemberg-lab.github.io/scRNA.seq.datasets/). The human brain datasets from Darmanis *et al.* (*24*) and Lake *et al.* (*25*) where downloaded from Gene Expression Omnibus (GEO) with accession numbers GSE67835 and GSE97930, respectively. A table listing all datasets including references to the original publications can be found in table S1.

scRNA-seq preprocessing and analysis

All datasets were processed using the Python package Scanpy (v. 1.2.2) (*30*) following the Scanpy's reimplementation of the popular Seurat's clustering workflow. First, the corresponding cell-gene matrices were filtered for cells with less than 500 detected genes and genes expressed in less than five cells. The resulting count matrix for each dataset was filtered for outliers with high or low numbers of counts. Gene expression was normalized to library size using the Scanpy function "normalize_per_cell." The normalized matrix of all filtered cells and genes was saved for the subsequent data generation step.

The following processing and analysis steps had the sole purpose of assigning cell type labels to every cell. All cells were clustered using the louvain clustering implementation of the Scanpy package. The louvain clustering resolution was chosen for each dataset, using the lowest possible resolution value (low-resolution values lead to less clusters) for which the calculated clusters appropriately separated the cell types. The top 1000 highly variable genes were used for clustering, which were calculated using Scanpy's "filter_genes_dispersion" function with parameters min_mean = 0.0125, max_mean = 3, and min_disp = 0.5. Principal components analysis was used for dimensionality reduction.

To identify cell types, marker genes were investigated for all cell types in question. For PBMC datasets, useful marker genes were adopted from public resources such as the Seurat tutorial for 2700 PBMCs (*31*). Briefly, interleukin-7 receptor (IL7R) was taken as marker for CD4 T cells, LYZ for monocytes, MS4A1 for B cells, GNLY for natural killer cells, FCER1A for dendritic cells, and CD8A and CCL5 as markers for CD8 T cells. For all other scRNA-seq datasets, marker genes and expected cell types were inferred from the original publication of the dataset. For instance, to annotate cell types of the mouse brain dataset from Zeisel *et al.* (*32*), we used the same marker genes as Zeisel and colleagues. We did not use the same cell type labels from the original publications because a main objective was to assure that cell type labeling is consistent between all datasets of a certain tissue.

Cell type annotation was performed manually across all the clusters for each dataset, such that all cells belonging to the same cluster were labeled with the same cell type. The cell type identity of each cluster was chosen by crossing the cluster's highly differentially expressed genes with the curated cell type's marker genes. Clusters that could not be clearly identified with a cell type were grouped into the "Unknown" category.

Tissue datasets for benchmarking

To assess the deconvolution performance on real tissue expression data, we used datasets for which the corresponding cell fractions were measured and published. The first dataset is the PBMC1 dataset, which was obtained from Zimmermann *et al.* (*21*). The second dataset, PBMC2, was downloaded from GEO with accession code GSE107011 (*10*). This dataset contains both RNA-seq profiles of immune cells (S4 cohort) and from bulk individuals (S13 cohort). As we were interested in the bulk profiles, we only used 12 samples from the S13 cohort from these data. Flow cytometry fractions were collected from the Monaco *et al.* publication (*10*).

In addition to the above mentioned two PBMC datasets, we used Ascites RNA-seq data. This dataset was provided by the authors, and cell type fractions for this dataset were taken from the supplementary materials of the publication (*18*).

For the evaluation on pancreas data, artificial bulk RNA-seq samples created from the scRNA-seq dataset of Xin *et al.* (*20*) were used. This dataset was downloaded from the resources of the MuSiC publication (*8*). The artificial bulk RNA-seq samples used for evaluation were then created using the "bulk_construct" function of the MuSiC tool.

To assess how Scaden and the GEP algorithms deal with the presence of unknown cell types, we generated PBMC bulk RNA samples

from the four scRNA-seq datasets (6000 each). The undefined amount of unknown cells that was generated by this approach was removed to be replaced by defined amounts of 5, 10, 20, and 30% of unknown cells, respectively. Cell fractions of all four samples were predicted with Scaden trained on the other three.

Performance on these samples was then assessed to test robustness against unseen cell types in the bulk mixture. Scaden was trained on samples from all datasets but the test dataset, while CSx and MuSiC used data8k as a reference.

The microarray dataset GSE65133 was downloaded from GEO, and cell type fractions were taken from the original CS publication (*6*).

Last, we wanted to get insights into neurodegenerative cell fraction changes in the brain. While it is known that neurodegenerative diseases like AD are accompanied by a gradual loss of brain neurons, stage-specific cell type shifts are still hard to come by. Here, we use the ROSMAP study cortical RNA-seq dataset along with the corresponding clinical metadata, to infer cell type composition over six clinically relevant stages of neurodegeneration (*22*). Furthermore, to assess deconvolution accuracy on postmortem human brain tissue, we used 41 samples from the ROSMAP, for which cell composition information from immunohistochemistry (*23*) was recently released and for which fractions for all cell types were reported. The ROSMAP RNA-seq data were downloaded from www.synapse.org/. The cell composition values were provided by the authors of the study (*23*).

RNA-seq preprocessing and analysis

For the RNA-seq datasets analyzed in this study, we did not apply any additional processing steps but used the obtained count or expression tables directly as downloaded for all datasets except the ROSMAP dataset. For the latter, we generated count tables from raw FastQ files using Salmon (*33*) and the GRCh38 reference genome. FastQ files from the ROSMAP study were downloaded from Synapse (www.synapse.org).

Simulation of bulk RNA-seq samples from scRNA-seq data

Scaden's DNN requires large amounts of training RNA-seq samples with known cell fractions. This explains why the generation of artificial bulk RNA-seq data is one of the key elements of the Scaden workflow.

To generate the training data, preprocessed scRNA-seq datasets were used (see the "Datasets and preprocessing" section), comprising the gene expression matrix and the cell type labels. Artificial RNA-seq samples were simulated by subsampling cells from individual scRNA-seq datasets; cells from different datasets were not merged into samples to preserve within-subject relationships. Datasets generated from multiple participants were split according to participant, and each subsampling was constrained to cells from one participant to capture the cross-subject heterogeneity and keep subject-specific gene dependencies.

The exact subsampling procedure is described in the following. First, for every simulated sample, random fractions were created for all different cell types within each scRNA-seq dataset using the random module of the Python package NumPy. Briefly, a random number was chosen from a uniform distribution between 0 and 1 using the NumPy function "random.rand()" for each cell type, and then this number was divided by the sum of all random numbers created to ensure the constraint of all fractions adding up to 1

$$f_c = \frac{r_c}{\sum_{C_{\text{all}}} r_c}$$

where r_c is the random number created for cell type c and C_{all} is the set of all cell types. Here, f_c is the calculated random fraction for cell type c. Then, each fraction was multiplied with the total number of cells selected for each sample, yielding the number of cells to choose for a specific cell type

$$N_c = f_c * N_{\text{total}}$$

where N_c is the number of cells to select for the cell type c, and N_{total} is the total number of cells contributing to one simulated RNA-seq sample (500, in this study). Next, N_c cells were randomly sampled from the scRNA-seq gene expression matrix for each cell type c. Afterward, the randomly selected single-cell expression profiles for every cell type are then aggregated by summing their expression values, to yield the artificial bulk expression profile for this sample.

Using the above-described approach, cell compositions that are strongly biased toward a certain cell type or are missing specific cell types are rare among the generated training samples. To account for this and to simulate cell compositions with a heavy bias to and the absence of certain cell types, a variation of the subsampling procedure was used to generate samples with sparse compositions, which we refer to as sparse samples. Before generating the random fractions for all cell types, a random number of cell types was selected to be absent from the sample, with the requirement of at least one cell type constituting the sample. After these leave-out cell types were chosen, random fractions were created and samples generated as described above. The average cell type proportions of the training dataset generated as described above are equal for all cell types. This allows for unbiased deconvolution as the true cell composition of a given tissue is not known beforehand. Using different sampling distributions (e.g., Gaussian and Uniform) or excluding sparse samples did not change Scaden's deconvolution performance notably on the simulated PBMC datasets. This shows that Scaden is relatively robust to training data generated by different sampling procedures.

Using this procedure, we generated 32,000 samples for the human PBMC training dataset, 14,000 samples for the human pancreas training dataset, 6000 samples for human brain, and 30,000 samples for the mouse brain training dataset (table S3).

Artificial bulk RNA-seq datasets were stored in "h5ad" format using the Anndata package (*30*), which allows to store the samples together with their corresponding cell type ratios while also keeping information about the scRNA-seq dataset of origin for each sample. This allowed to access samples from specific datasets, which is useful for cross-validation.

Scaden overview

The following section contains an overview of the input data preprocessing, the Scaden model, model selection, and how Scaden predictions are generated.

Input data preprocessing

The data preprocessing step is aimed to make the input data more suitable for machine learning algorithms. To achieve this, an optimal preprocessing procedure should transform any input data from the simulated samples or from the bulk RNA-seq to the same feature scale. Before any scaling procedure can be applied, it must be ensured that both the training data and the bulk RNA-seq data subject to prediction share the same features. Therefore, before scaling, both datasets are limited to contain features (genes) that are available in both datasets. In addition, uninformative genes that have

either zero expression or an expression variance below 0.1 were removed, leaving ~10,000 genes for model training and inference. The two-step processing procedure used for Scaden is described in the following:

First, to account for heteroscedasticity, a feature inherent to RNA-seq data, the data were transformed into logarithmic space by adding a pseudocount of 1 and then taking the Logarithm (base 2).

Second, every sample was scaled to the range [0,1] using the MinMaxScaler() class from the Sklearn preprocessing module. Per-sample scaling, unlike per-feature scaling that is more common in machine learning, assures that intergene relative expression patterns in every sample are preserved. This is important, as our hypothesis was that a neural network could learn the deconvolution from these intergene expression patterns

$$x_{\text{scaled},i} = (x_i - \min(\boldsymbol{X}_i)) / (\max(X_i) - \min(X_i))$$

where $x_{\text{scaled},i}$ is the $\log_2$ expression value of gene x in sample i, X_i is the vector of $\log_2$ expression values for all genes of sample i, $\min(\boldsymbol{X}_i)$ is the minimum gene expression of vector X_i, and $\max(X_i)$ is the maximum gene expression of vector X_i.

Note that all training datasets are stored as expression values and are only processed as described above. In the deployment use case, the simulated training data should contain the same features as in the bulk RNA-seq sample that shall be deconvolved.

Model selection

The goal of model selection was to find an architecture and hyperparameters that robustly deconvolve simulated tissue RNA-seq data and, more importantly, real bulk RNA-seq data. Because of the very limited availability of bulk RNA-seq datasets with known cell fractions, model selection was mainly optimized on the simulated PBMC datasets. To capture interexperimental variation, we used leave-one-dataset-out cross-validation for model optimization: A model was trained on simulated data from all but one dataset, and performance was tested on simulated samples from the left-out dataset. This allows to simulate batch effects between datasets and helps to test the generalizability of the model. In the process of model selection and (hyper-) parameter optimization, performed on PBMC and Ascites datasets, we found three models with different architectures and dropout rates but comparable performance. To address overfitting in individual models, we decided to use a combination of models, expecting this to serve as another means of regularization. We did not test multiple combinations but rather used an informed choice with varying layer sizes and dropout regularization, with the goal to increase model diversity. We observed that the average of an ensemble of models generalized better to the test sets than individual models. Model training and prediction is done separately for each model, with the prediction averaging step combining all model predictions (fig. S1 and tables S4 and S6). We provide a list of all tested parameters in the Supplementary Materials (table S5).

Final Scaden model

The Scaden model learns cell type deconvolution through supervised training on datasets of simulated bulk RNA-seq samples simulated with scRNA-seq data. To account for model biases and to improve performance, Scaden consists of an ensemble of three DNNs with varying architectures and degrees of dropout regularization. All models of the ensemble use four layers of varying sizes between 32 and 1024 nodes, with dropout regularization implemented in two of the three ensemble models. The exact layer sizes and dropout rates are listed in table S4. The rectified linear unit is used as activation function in every internal layer. We used a Softmax function to predict cell fractions, as we did not see any improvements in using a linear output function with consecutive non-negativity correction and sum-to-one scaling. Python (v. 3.6.6) and the TensorFlow library (v. 1.10.0) were used for implementation of Scaden. A complete list of all software used for the implementation of Scaden is provided in table S15.

Training and prediction

After the preprocessing of the data, a Scaden ensemble can be trained on simulated tissue RNA-seq data or mixtures of simulated and real tissue RNA-seq data. Parameters are optimized using Adam with a learning rate of 0.0001 and a batch size of 128. We used an L1 loss as optimization objective

$$L1(y_i, \hat{y}_i) = |y_i - \hat{y}_i|$$

where y_i is the vector of ground-truth fractions of sample i and $\hat{y}_i$ is the vector of predicted fractions of sample i. Each of the three ensemble models is trained independently for 5000 steps. This "early stopping" serves to avoid domain overfitting on the simulated tissue data, which would decrease the model performance on the real tissue RNA-seq data. We observed that training for more steps lead to an average performance decrease on real tissue RNA-seq data. To perform deconvolution with Scaden, a bulk RNA-seq sample is fed into a trained Scaden ensemble, and three independent predictions for the cell type fractions of this sample are generated by the trained DNNs. These three predictions are then averaged per cell type to yield the final cell type composition for the input bulk RNA-seq sample

$$\hat{y}_c = \frac{\hat{y}_c^1 + \hat{y}_c^2 + \hat{y}_c^3}{3}$$

where $\hat{y}_c$ is the final predicted fraction for cell type c and $\hat{y}_c^i$ is the predicted fraction for cell type c of model i.

Scaden requirements

Currently, a disadvantage of the Scaden algorithm is the necessity to train a new model for deconvolution if no perfect overlap in the feature space exists. This constraint limits the usefulness of pretrained models. Once trained, however, the prediction runtime scales linearly with sample numbers and is usually in the order of seconds, making Scaden a useful tool if deconvolution is to be performed on very large datasets. While the requirements are dataset dependent, the Scaden demo was profiled to require a peak of 3.2 GB of random-access memory (RAM) during the DNN training process, so a computer with 8 GB of RAM should be able to run it smoothly. In our tests with an Intel(R) Xeon(R) CPU E5-1630 workstation, the demo could run in 22 min, spending most of the CPU time in the DNN training process. The most prominent and obvious issue of Scaden is the difference between simulated scRNA-seq data used for training and the bulk RNA-seq data subject to inference. While Scaden is able to transfer the learned deconvolution between the two data types and achieves state-of-the-art performance, we hypothesize that efforts to improve this translatability could improve Scaden's prediction accuracy even further. Algorithmic improvements are therefore likely to address this issue and are planned for future releases.

Algorithm comparison

We used several performance measures to compare Scaden to four existing cell deconvolution algorithms, CS with LM22 GEP, CSx,

MuSiC, and CPM. To compare the performance of the five deconvolution algorithms, we measured the RMSE, Lin's CCC, Pearson product moment correlation coefficient r, and R^2 values, comparing real and predicted cell fractions estimates. In addition, to identify systematic prediction errors and biases, slope and intercept for the regression lines were calculated. These metrics are defined as follows

$$\mathrm{RMSE}(y,\hat{y}) = \sqrt{\mathrm{avg}\,(y-\hat{y})^2}$$

$$r(y,\hat{y}) = \frac{\mathrm{cov}(y,\hat{y})}{\sigma_y \sigma_{\hat{y}}}$$

$$R^2(y,\hat{y}) = r(y,\hat{y})^2$$

$$\mathrm{slope}(y,\hat{y}) = \frac{\Delta y}{\Delta \hat{y}}$$

$$\mathrm{CCC}(y,\hat{y}) = \frac{2r\sigma_y \sigma_{\hat{y}}}{\sigma_y{}^2 + \sigma_{\hat{y}}{}^2 + (\mu_x - \mu_{\hat{y}})}$$

where y are the ground-truth fractions, $\hat{y}$ are the prediction fractions, σ_x is the SD of x, $\mathrm{cov}(y,\hat{y})$ is the covariance of y and $\hat{y}$, and $\mu_y, \mu_{\hat{y}}$ are the mean of the predicted and ground-truth fractions, respectively.

All metrics were calculated for all data points of a dataset and separately for all data points of a specific cell type. For the latter approach, we then averaged the resulting values to recover single values. While the metrics calculated on all data points might be sufficient, we deem that the cell type–specific deconvolution might, in many instances, be of even greater interest. It is noteworthy in this context that cell type–specific deconvolution performance can be quite weak, depending on the dataset. This is true for all tested deconvolution algorithms, while Scaden achieves best performance.

CIBERSORT

CS is a cell convolution algorithm based on specialized GEPs and support vector regression. Cell composition estimations were obtained using the CS web application (https://cibersort.stanford.edu/). For all deconvolutions with CS, we used the LM22 GEP, which was generated by the CS authors from 22 leukocyte subsets profiled on the HGU133A microarray platform.

Because the LM22 GEP matrix contains cell types at a finer granularity than what was used for this study, predicted fractions of subcell types were added together. For cell grouping, we used the mapping of subcell types to broader types given by figure 6 from Monaco *et al.* (*10*). We provide a table with the exact mappings used here in the Supplementary Materials (table S13). The deconvolution was performed using 500 permutations with quantile normalization disabled for all datasets but GSE65133 (Microarray), as is recommended for RNA-seq data. We used default settings for all other CS parameters.

CIBERSORTx

CSx is a recent variant of CS that can generate GEP matrices from scRNA-seq data and use these for deconvolution. For additional deconvolution robustness, it applies batch normalization to the data. All signature matrices were created by uploading the labeled scRNA-seq expression matrices and using the default options. Quantile normalization was disabled. For deconvolution on simulated data, no batch normalization was used. For all bulk RNA-seq datasets, the S-Mode batch normalization was chosen. All PBMC datasets were deconvolved using a GEP matrix generated from the data6k dataset (for simulated samples from data6k, a donorA GEP matrix was chosen).

MuSiC

MuSiC is a deconvolution algorithm that uses multisubject scRNA-seq datasets as GEP matrices in an attempt to include heterogeneity in the matrices to improve generalization. While MuSiC tries to address similar issues of previous deconvolution algorithms by using scRNA-seq data, the approach is very different. For deconvolution, MuSiC applies a sophisticated GEP-based deconvolution algorithm that uses weighted non-negative least-squares regression with an iterative estimation procedure that imposes more weight on informative genes and less weight on noninformative genes.

The MuSiC R package contains functionality to generate the necessary GEP matrix given an scRNA-seq dataset and cell type labels. To generate MuSiC deconvolution predictions on PBMC datasets, we used the data8k scRNA-seq dataset as reference data for MuSiC and follow the tutorial provided by the authors to perform the deconvolution. For deconvolution of artificial samples generated from the data8k dataset, we provided MuSiC with the data6k dataset as a reference instead.

MuSiC was developed with a focus on multisubject scRNA-seq datasets, in which the algorithm tries to take advantage from the added heterogeneity that these datasets contain, by calculating a measure of cross-subject consistency for marker genes. To assess how Scaden performs on multisubject datasets compared to MuSiC, we evaluated both methods on artificial bulk RNA-seq samples from human pancreas. We used the bulk_construct function from MuSiC to combine the cells from all 18 participants contained in the scRNA-seq dataset from Xin *et al.* (*20*) to generate artificial bulk samples for evaluation. Next, as a multisubject reference dataset, we used the pancreas scRNA-seq dataset from Segerstolpe *et al.* (*19*), which contains single-cell expression data from 10 different participants, 4 of which with type 2 diabetes. For Scaden, the Segerstolpe scRNA-seq dataset was split by participants, and training datasets were generated for each participant, yielding in total 10,000 samples. For MuSiC, a processed version of this dataset was downloaded from the resources provided by the MuSiC authors (*8*) and used as an input reference dataset for the MuSiC deconvolution. Deconvolution was then performed according to the MuSiC tutorial, and performance was compared according to the above-defined metrics.

Cell Population Mapping

CPM is a deconvolution algorithm that uses single-cell expression profiles to identify a so-called "cell population map" from bulk RNA-seq data (*9*). In CPM, the cell population map is defined as composition of cells over a cell-state space, where a cell state is defined as a current phenotype of a single cell. Contrary to other deconvolution methods, CPM tries to estimate the abundance of all cell states and types for a given bulk mixture, instead of only deconvolving the cell types. As input, CPM requires an scRNA-seq dataset and a low-dimensional embedding of all cells in this dataset, which represents the cell-state map. As CPM estimates abundances of both cell states and types, it can be used for cell type deconvolution by summing up all estimated fractions for all cell states of a given cell type, a method that is implemented in the scBio R package, which contains the CPM method. To perform deconvolution with CPM, we used the data6k PBMC scRNA-seq dataset as an input reference for all PBMC samples. For samples simulated from the data6k dataset, we used the data8k dataset as a reference. According to the CPM paper, a dimension reduction method can be used to obtain the cell-state space. We therefore used Uniform Manifold Approximation and Projection (UMAP), a dimension reduction method widely used for scRNA-seq

data, to generate the cell-state space mapping for the input scRNA-seq data. Deconvolution was then performed using the CPM function of the scBio package with an scRNA-seq dataset and accompanying UMAP embedding as input.

Code and software availability

The source code for Scaden is available at https://github.com/KevinMenden/scaden. Documentation is published at https://scaden.readthedocs.io. Code to generate the figures along with the training datasets used in this study is published at figshare: https://figshare.com/projects/Scaden/62834. The Scaden web application can be accessed at https://scaden.ims.bio.

SUPPLEMENTARY MATERIALS

Supplementary material for this article is available at http://advances.sciencemag.org/cgi/content/full/6/30/eaba2619/DC1

View/request a protocol for this paper from *Bio-protocol*.

REFERENCES AND NOTES

1. R. Hrdlickova, M. Toloue, B. Tian, RNA-Seq methods for transcriptome analysis. *Wiley Interdiscip. Rev. RNA* **8**, e1364 (2017).
2. M. Egeblad, E. S. Nakasone, Z. Werb, Tumors as organs: Complex tissues that interface with the entire organism. *Dev. Cell* **18**, 884–901 (2010).
3. A. Kuhn, D. Thu, H. J. Waldvogel, R. L. M. Faull, R. Luthi-Carter, Population-specific expression analysis (PSEA) reveals molecular changes in diseased brain. *Nat. Methods* **8**, 945–947 (2011).
4. F. Avila Cobos, J. Vandesompele, P. Mestdagh, K. De Preter, Computational deconvolution of transcriptomics data from mixed cell populations. *Bioinformatics* **34**, 1969–1979 (2018).
5. S. Mohammadi, N. Zuckerman, A. Goldsmith, A. Grama, A critical survey of deconvolution methods for separating cell types in complex tissues. *Proc. IEEE* **105**, 340–366 (2017).
6. A. M. Newman, C. L. Liu, M. R. Green, A. J. Gentles, W. Feng, Y. Xu, C. D. Hoang, M. Diehn, A. A. Alizadeh, Robust enumeration of cell subsets from tissue expression profiles. *Nat. Methods* **12**, 453–457 (2015).
7. A. M. Newman, C. B. Steen, C. L. Liu, A. J. Gentles, A. A. Chaudhuri, F. Scherer, M. S. Khodadoust, M. S. Esfahani, B. A. Luca, D. Steiner, M. Diehn, A. A. Alizadeh, Determining cell type abundance and expression from bulk tissues with digital cytometry. *Nat. Biotechnol.* **37**, 773–782 (2019).
8. X. Wang, J. Park, K. Susztak, N. R. Zhang, M. Li, Bulk tissue cell type deconvolution with multi-subject single-cell expression reference. *Nat. Commun.* **10**, 380 (2019).
9. A. Frishberg, N. Peshes-Yaloz, O. Cohn, D. Rosentul, Y. Steuerman, L. Valadarsky, G. Yankovitz, M. Mandelboim, F. A. Iraqi, I. Amit, L. Mayo, E. Bacharach, I. Gat-Viks, Cell composition analysis of bulk genomics using single-cell data. *Nat. Methods* **16**, 327–332 (2019).
10. G. Monaco, B. Lee, W. Xu, S. Mustafah, Y. Y. Hwang, C. Carré, N. Burdin, L. Visan, M. Ceccarelli, M. Poidinger, A. Zippelius, J. Pedro de Magalhães, A. Larbi, RNA-Seq signatures normalized by mRNA abundance allow absolute deconvolution of human immune cell types. *Cell Rep.* **26**, 1627–1640.e7 (2019).
11. F. Vallania, A. Tam, S. Lofgren, S. Schaffert, T. D. Azad, E. Bongen, W. Haynes, M. Alsup, M. Alonso, M. Davis, E. Engleman, P. Khatri, Leveraging heterogeneity across multiple datasets increases cell-mixture deconvolution accuracy and reduces biological and technical biases. *Nat. Commun.* **9**, 4735 (2018).
12. D. Venet, F. Pecasse, C. Maenhaut, H. Bersini, Separation of samples into their constituents using gene expression data. *Bioinformatics* **17**, S279–S287 (2001).
13. E. Shapiro, T. Biezuner, S. Linnarsson, Single-cell sequencing-based technologies will revolutionize whole-organism science. *Nat. Rev. Genet.* **14**, 618–630 (2013).
14. Tabula Muris Consortium, Single-cell transcriptomics of 20 mouse organs creates a *Tabula Muris*. *Nature* **562**, 367–372 (2018).
15. K. W. Kelley, H. Nakao-Inoue, A. V. Molofsky, M. C. Oldham, Variation among intact tissue samples reveals the core transcriptional features of human CNS cell classes. *Nat. Neurosci.* **21**, 1171–1184 (2018).
16. S. C. Hicks, F. W. Townes, M. Teng, R. A. Irizarry, Missing data and technical variability in single-cell RNA-sequencing experiments. *Biostatistics* **19**, 562–578 (2018).
17. L. I. Lin, A concordance correlation coefficient to evaluate reproducibility. *Biometrics* **45**, 255–268 (1989).
18. M. Schelker, S. Feau, J. Du, N. Ranu, E. Klipp, G. MacBeath, B. Schoeberl, A. Raue, Estimation of immune cell content in tumour tissue using single-cell RNA-seq data. *Nat. Commun.* **8**, 2032 (2017).
19. Å. Segerstolpe, A. Palasantza, P. Eliasson, E. M. Andersson, A. C. Andréasson, X. Sun, S. Picelli, A. Sabirsh, M. Clausen, M. K. Bjursell, D. M. Smith, M. Kasper, C. Ämmälä, R. Sandberg, Single-cell transcriptome profiling of human pancreatic islets in health and type 2 diabetes. *Cell Metab.* **24**, 593–607 (2016).
20. Y. Xin, J. Kim, H. Okamoto, M. Ni, Y. Wei, C. Adler, A. J. Murphy, G. D. Yancopoulos, C. Lin, J. Gromada, RNA sequencing of single human islet cells reveals type 2 diabetes genes. *Cell Metab.* **24**, 608–615 (2016).
21. M. T. Zimmermann, A. L. Oberg, D. E. Grill, I. G. Ovsyannikova, I. H. Haralambieva, R. B. Kennedy, G. A. Poland, System-wide associations between DNA-methylation, gene expression, and humoral immune response to influenza vaccination. *PLOS ONE* **11**, e0152034 (2016).
22. D. A. Bennett, A. S. Buchman, P. A. Boyle, L. L. Barnes, R. S. Wilson, J. A. Schneider, Religious orders study and rush memory and aging project. *J. Alzheimers Dis.* **64**, S161–S189 (2018).
23. E. Patrick, M. Taga, A. Ergun, B. Ng, W. Casazza, M. Cimpean, C. Yung, J. A. Schneider, D. A. Bennett, C. Gaiteri, P. L. De Jager, E. M. Bradshaw, S. Mostafavi, Deconvolving the contributions of cell-type heterogeneity on cortical gene expression. *bioRxiv* **2019**, 566307 (2019).
24. S. Darmanis, S. A. Sloan, Y. Zhang, M. Enge, C. Caneda, L. M. Shuer, M. G. H. Gephart, B. A. Barres, S. R. Quake, A survey of human brain transcriptome diversity at the single cell level. *Proc. Natl. Acad. Sci. U.S.A.* **112**, 7285–7290 (2015).
25. B. B. Lake, R. Ai, G. E. Kaeser, N. S. Salathia, Y. C. Yung, R. Liu, A. Wildberg, D. Gao, H. L. Fung, S. Chen, R. Vijayaraghavan, J. Wong, A. Chen, X. Sheng, F. Kaper, R. Shen, M. Ronaghi, J. B. Fan, W. Wang, J. Chun, K. Zhang, Neuronal subtypes and diversity revealed by single-nucleus RNA sequencing of the human brain. *Science* **352**, 1586–1590 (2016).
26. H. Braak, E. Braak, Neuropathological stageing of Alzheimer-related changes. *Acta Neuropathol.* **82**, 239–259 (1991).
27. J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, H. Lipson, Understanding Neural Networks Through Deep Visualization (2015); http://arxiv.org/abs/1506.06579.
28. B. Athiwaratkun, M. Finzi, P. Izmailov, A. G. Wilson, Improving consistency-based semi-supervised learning with weight averaging. *Jmlr* **17**, 1–35 (2018).
29. M. Zhang, K. T. Ma, J. H. Lim, Q. Zhao, J. Feng, Deep future gaze: Gaze anticipation on egocentric videos using adversarial networks, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 21 to 26 July 2017.
30. F. A. Wolf, P. Angerer, F. J. Theis, SCANPY: Large-scale single-cell gene expression data analysis. *Genome Biol.* **19**, 15 (2018).
31. R. Satija, J. A. Farrell, D. Gennert, A. F. Schier, A. Regev, Spatial reconstruction of single-cell gene expression data. *Nat. Biotechnol.* **33**, 495–502 (2015).
32. A. Zeisel, H. Hochgerner, P. Lönnerberg, A. Johnsson, F. Memic, J. van der Zwan, M. Häring, E. Braun, L. E. Borm, G. La Manno, S. Codeluppi, A. Furlan, K. Lee, N. Skene, K. D. Harris, J. Hjerling-Leffler, E. Arenas, P. Ernfors, U. Marklund, S. Linnarsson, Molecular architecture of the mouse nervous system. *Cell* **174**, 999–1014.e22 (2018).
33. M. I. Love, C. Soneson, R. Patro, Swimming downstream: Statistical analysis of differential transcript usage following Salmon quantification. *F1000Res.* **7**, 952 (2018).
34. M. Baron, A. Veres, S. L. Wolock, A. L. Faust, R. Gaujoux, A. Vetere, J. H. Ryu, B. K. Wagner, S. S. Shen-Orr, A. M. Klein, D. A. Melton, I. Yanai, A single-cell transcriptomic map of the human and mouse pancreas reveals inter- and intra-cell population structure. *Cell Syst.* **3**, 346–360.e4 (2016).
35. B. Tasic, V. Menon, T. N. Nguyen, T. K. Kim, T. Jarsky, Z. Yao, B. Levi, L. T. Gray, S. A. Sorensen, T. Dolbeare, D. Bertagnolli, J. Goldy, N. Shapovalova, S. Parry, C. Lee, K. Smith, A. Bernard, L. Madisen, S. M. Sunkin, M. Hawrylycz, C. Koch, H. Zeng, Adult mouse cortical cell taxonomy revealed by single cell transcriptomics. *Nat. Neurosci.* **19**, 335–346 (2016).
36. R. A. Romanov, A. Zeisel, J. Bakker, F. Girach, A. Hellysaz, R. Tomer, A. Alpár, J. Mulder, F. Clotman, E. Keimpema, B. Hsueh, A. K. Crow, H. Martens, C. Schwindling, D. Calvigioni, J. S. Bains, Z. Máté, G. Szabó, Y. Yanagawa, M.-D. Zhang, A. Rendeiro, M. Farlik, M. Uhlén, P. Wulff, C. Bock, C. Broberger, K. Deisseroth, T. Hökfelt, S. Linnarsson, T. L. Horvath, T. Harkany, Molecular interrogation of hypothalamic organization reveals distinct dopamine neuronal subtypes. *Nat. Neurosci.* **20**, 176–188 (2017).
37. J. N. Campbell, E. Z. Macosko, H. Fenselau, T. H. Pers, A. Lyubetskaya, D. Tenen, M. Goldman, A. M. J. Verstegen, J. M. Resch, S. A. McCarroll, E. D. Rosen, B. B. Lowell, L. T. Tsai, A molecular census of arcuate hypothalamus and median eminence cell types. *Nat. Neurosci.* **20**, 484–496 (2017).
38. R. Chen, X. Wu, L. Jiang, Y. Zhang, Single-cell RNA-seq reveals hypothalamic cell diversity. *Cell Rep.* **18**, 3227–3241 (2017).

Acknowledgments: We would like to thank the people of the Genome Biology of Neurodegenerative Diseases group and Institute of Medical Systems Biology for helpful discussions and suggestions. **Funding:** This study was supported, in part, by RiMod-FTD, an EU Joint Programme–Neurodegenerative Disease Research (JPND), and the NOMIS Foundation to K.M., A.D., and P.H. and BMBF Integrative Data Semantics for

Neurodegenerative research (IDSN), ERA-Net E-Rare MAXOMOD, CRC 1286/Z2, CRU 296 P8, and CRU 306 P-C to M.M., S.O., D.S.M., K.K., and S.B. **Author contributions:** K.M. and S.B. initiated the project. K.M., P.H., and S.B. designed the study, deep learning models, and analysis. K.M., M.M., and S.O. built the deep learning models. K.M., M.M., K.K., and A.D. analyzed the data. D.S.M. and S.O. built the Scaden web application. K.M., K.K., and S.B. wrote the manuscript. M.M., A.D., and P.H. contributed to the manuscript writing. **Competing interests:** The authors declare that they have no competing interests. **Data and materials availability:** All data needed to evaluate the conclusions in the paper are present in the paper and/or the Supplementary Materials. Additional data related to this paper may be requested from the authors. Only publicly available datasets were used during this study. The scRNA-seq PBMC datasets donorA, donorC, data6k, and data8k were all downloaded from 10X Genomics (https://support.10xgenomics.com/single-cell-gene-expression/datasets), where they are listed as "Frozen PBMCs (Donor A)," "Frozen PBMCs (Donor C)," "6k PBMCs from a Healthy Donor," and "8k PBMCs from a Healthy Donor," respectively. The pancreas scRNA-seq dataset from Segerstolpe *et al.* (*19*) was downloaded from ArrayExpress with accession code E-MTAB-5061. The scRNA-seq datasets from Baron *et al.* (*34*) (pancreas), Tasic *et al.* (*35*), Zeisel *et al.* (*32*), Romanov *et al.* (*36*), Campbell *et al.* (*37*), and Chen *et al.* (*38*) (all mouse brain) were all downloaded from https://hemberg-lab.github.io/scRNA.seq.datasets/. The ascites scRNA-seq dataset was downloaded from https://figshare.com/s/711d3fb2bd3288c8483a. The bulk RNA-seq dataset PBMC1 is accessible from ImmPort with accession code SDY67. The PBMC2 dataset was downloaded from GEO with accession code GSE107011. The ROSMAP human brain RNA-seq dataset was downloaded from Synapse (ID: syn3219045). The bulk RNA-seq data from ascites was provided by Schelker *et al.* (*18*). The pancreas scRNA-seq dataset from Xin *et al.* (*20*) was accessed from the MuSiC tutorial site (https://xuranw.github.io/MuSiC/articles/pages/data.html).

Submitted 18 November 2019
Accepted 5 June 2020
Published 22 July 2020
10.1126/sciadv.aba2619

Citation: K. Menden, M. Marouf, S. Oller, A. Dalmia, D. S. Magruder, K. Kloiber, P. Heutink, S. Bonn, Deep learning–based cell composition analysis from tissue expression profiles. *Sci. Adv.* **6**, eaba2619 (2020).

Deep learning–based cell composition analysis from tissue expression profiles

Kevin Menden, Mohamed Marouf, Sergio Oller, Anupriya Dalmia, Daniel Sumner Magruder, Karin Kloiber, Peter Heutink and Stefan Bonn

Sci Adv **6** (30), eaba2619.
DOI: 10.1126/sciadv.aba2619

3.4 SEA, 2019

D204–D219 *Nucleic Acids Research, 2020, Vol. 48, Database issue* *Published online 10 October 2019*
doi: 10.1093/nar/gkz869

SEAweb: the small RNA Expression Atlas web application

Raza-Ur Rahman [1,†], Anna-Maria Liebhoff[1,†], Vikas Bansal[1,2,‡], Maksims Fiosins[1,2,3,‡], Ashish Rajput[1,‡], Abdul Sattar[1], Daniel S. Magruder[1,3], Sumit Madan[4,5], Ting Sun[1,6], Abhivyakti Gautam[1], Sven Heins[1], Timur Liwinski[7], Jörn Bethune[1], Claudia Trenkwalder[8,9], Juliane Fluck[4,10,11], Brit Mollenhauer[8,12] and Stefan Bonn [1,2,*]

[1]Institute of Medical Systems Biology, Center for Molecular Neurobiology, University Medical Center Hamburg-Eppendorf, 20251 Hamburg, Germany, [2]German Center for Neurodegenerative Diseases, 72076 Tübingen, Germany, [3]Genevention GmbH, 37079 Göttingen, Germany, [4]Fraunhofer Institute for Algorithms and Scientific Computing, Schloss Birlinghoven, 53757 Sankt Augustin, Germany, [5]Rheinische Friedrich-Wilhelms-Universität Bonn, 53113 Bonn, Germany, [6]Department of Neurogenetics, Max Planck Institute of Experimental Medicine, 37075 Göttingen, Germany, [7]Department of Medicine, University Medical Center Hamburg-Eppendorf, 20251 Hamburg, Germany, [8]Paracelsus-Elena-Klinik, 34128 Kassel, Germany, [9]Department of Neurosurgery, University Medical Center Göttingen, 37075 Göttingen, Germany, [10]Institute of Geodesy and Geoinformation, University of Bonn, 53115 Bonn, Germany, [11]German National Library of Medicine (ZB MED) - Information Centre for Life Sciences, 53115 Bonn, Germany and [12]Institute of Neurology, University Medical Center Göttingen, 37075 Göttingen, Germany

Received July 19, 2019; Revised September 14, 2019; Editorial Decision September 23, 2019; Accepted October 01, 2019

ABSTRACT

We present the Small RNA Expression Atlas (SEAweb), a web application that allows for the interactive querying, visualization and analysis of known and novel small RNAs across 10 organisms. It contains sRNA and pathogen expression information for over 4200 published samples with standardized search terms and ontologies. In addition, SEAweb allows for the interactive visualization and re-analysis of 879 differential expression and 514 classification comparisons. SEAweb's user model enables sRNA researchers to compare and re-analyze user-specific and published datasets, highlighting common and distinct sRNA expression patterns. We provide evidence for SEAweb's fidelity by (i) generating a set of 591 tissue specific miRNAs across 29 tissues, (ii) finding known and novel bacterial and viral infections across diseases and (iii) determining a Parkinson's disease-specific blood biomarker signature using novel data. We believe that SEAweb's simple semantic search interface, the flexible interactive reports and the user model with rich analysis capabilities will enable researchers to better understand the potential function and diagnostic value of sRNAs or pathogens across tissues, diseases and organisms.

INTRODUCTION

Small RNAs (sRNAs) are a class of short, non-coding RNAs with important biological functions in nearly all aspects of organismal development in health and disease. Especially in diagnostic and therapeutic research, sRNAs such as miRNAs and piRNAs received recent attention (1). The increasing number of deep sequencing sRNA studies (sRNA-seq) is reflecting the importance of sRNAs in biological processes as well as disease diagnosis and therapy. In addition, recent evidence highlights the pivotal roles of viral and bacterial-derived sRNAs in the pathogenesis of infectious diseases, across the animal and plant kingdoms (2–4). Viral sRNAs play vital roles in the viral replication, persistence, the immune escape and host cell transformation (2,3). Many DNA and RNA viruses encode various classes of small RNAs, which associate with host RNAs and proteins and affect their stability and function. The introduction of sRNA deep sequencing (sRNA-seq) allowed for the quantitative analysis of sRNAs of a specific organism, but its generic nature also enables the simultaneous detection of microbial and viral reads. sRNA-seq data therefore naturally lends itself for the analysis of host-pathogen interactions, which has been recently exemplified for RNA-seq

*To whom correspondence should be addressed. Tel: +49 40 7410 55082; Email: sbonn@uke.de
†The authors wish it to be known that, in their opinion, the first two authors should be regarded as Joint First Authors.
‡The authors wish it to be known that, in their opinion, the third, fourth and fifth authors should be regarded as Joint Second Authors.

data (5). Using the vast amount of publicly available sRNA-seq data in repositories such as Sequence Read Archive (SRA) (6) and Gene Expression Omnibus (7) enables the unbiased charting of viral and bacterial infections across tissues, diseases, species, age and sex. This would not only highlight novel causal or comorbid relationships between diseases and infections, it could also shed mechanistic insights onto how the infectious agent interacts with and modifies the host cell. To harvest the true potential of existing data, it is important to allow for querying, visualization and analysis of sRNA-seq data across organisms, tissues, cell types and disease states. This would allow researchers, for example, to search for disease-specific sRNA or pathogenic biomarker signatures across all disease entities investigated. Data integration and interoperability require (i) a streamlined analysis workflow to reduce analysis bias between experiments (ii) also necessitates standardized annotation using ontologies to search and retrieve relevant samples and (iii) flexible and interactive visualization of the data.

To date, several web-based sRNA-seq expression profile databases are available that differ in their level of information, portfolio, performance and user-friendliness. Recent additions to sRNA web based databases include miRmine (8), provides expression of a single or multiple miRNAs for a specific tissue, cell-line or disease. Results are displayed in multiple interactive, graphical and downloadable formats. miratlas (9) allows for searching miRNA expression profiles as well as sRNA-seq experiments and provides information on the miRNA modification analysis. YM500v3 (10) provides interactive web reports on sRNA expression profiles, novel miRNA expression profiles, miRNA modification analysis, sRNA differential expression and miRNA gene targets. SPAR (11) is a user-friendly web server for the analysis, annotation and visualization of sRNA-seq data. It provides expression profiles of 10 different types of sRNAs across different tissues and cell types of human (hg19, hg38) and mouse (mm10). Currently, SPAR is the only tool that allows users to compare their input experimental data against the reference datasets from ENCODE (12) and DASHR (13). Moreover, it supports different genome versions of an organism. DASHR2 (14) supports sRNA expression profiles across different genome versions of the same species across tissues and cell types and supports 10 types of sRNAs. Results are provided in an interactive manner, such as sncRNA locus sorting and filtering by biological features. All annotation and expression information are downloadable and accessible as UCSC genome browser tracks.

Although many good web platforms for the sRNA-seq data exist, some important aspects for storing and searching have yet to be integrated. For example, no current web application allows for the ontology based search of sRNA-seq experiments. Current tools lack an important association of miRNAs with disease. miRNA disease associations are provided by HMDD (15), but it does not provide miRNA expression information. Except for YM500v3, current tools do not provide miRNAs and gene targets. Of note YM500v3 is only limited to cancer miRNome studies. Also, there is currently no web application that allows for the identification of biomarkers of disease via machine-learning. The above mentioned web platforms do not provide expression of novel miRNAs in known disease state or tissues, including the structure and probability of the novel miRNA prediction. To our knowledge no other data repository provides pathogenic signatures from sRNA-seq data including their differential expression in healthy and diseased condition. Except for SPAR, current sRNA-seq web services do not allow for the user data upload. At last, in current tools users can only search for the results that are stored in the database, there is no option for the users to reanalyze data with the samples of their choice. This feature would greatly facilitate researchers to perform differential expression between male and female of an experiment or to compare old aged patients (samples) with young ones in the same group. In the end, these functionalities should be paired with a flexible and interactive visualization of the sRNA-seq data supporting more species and cross study comparisons.

In order to address the above mentioned limitations, we hereby present the **s**mall-RNA **E**xpression **A**tlas (SEAweb), a web application that allows for querying, visualization and analysis of over 4200 published sRNA-seq expression samples. SEAweb automatically downloads and re-analyzes published data using Oasis 2 (16), semantically annotates relevant meta-information using standardized terms (the annotations are later checked and corrected manually), synchronizes sRNA information with other databases, allows for the querying of terms across ontological graphs and presents quality curated sRNA expression information as interactive web reports. In addition, SEAweb stores sRNA differential expression, sRNA based classification, pathogenic sRNA signatures from bacteria and viruses and pathogen differential expression. Gene targets and disease associations for miRNAs are also incorporated into SEAweb.

One of the most useful features of SEAweb is to enable users to upload their analysis results of differential expression and classification from Oasis 2. This allow users to compare their data to over 4200 experimental samples across different conditions. Using SEAweb's interactive visualizations, users can upload their data into their own workspace, select the published datasets to compare to, and define if differential expression or classification results should be compared. SEAweb also provides users with an option to perform on the fly analysis such as overlapping differentially expressed (DE) sRNAs or pathogens across different studies or the most important features (sRNAs) identified with classification. At last, SEAweb enables end users to re-submit samples from interactive plots for differential expression or classification, this helps users to choose samples of their choice from an experiment. It currently supports 10 organisms (Table 1) and is continuously updated with novel published sRNA-seq datasets and relevant sRNA information from various online resources. A detailed comparison of SEAweb to other existing sRNA expression databases (Table 2) highlights that SEAweb is superior in terms of supported organism, ontological annotations, diseases, tissues, sRNA based classification, pathogen k-mer DE, known miRNA disease associations, user specific experimental data upload, cross study comparisons and re-analysis with selected samples. SEAweb contains

Table 1. Supported SEAweb organisms and their corresponding genome versions

Organism	genome-version	genome-date
Bos taurus	UMD3.1	2009-11
Caenorhabditis elegans	WBcel235	2012-12
Danio rerio	GRCz10	2014-09
Drosophila melanogaster	BDGP6	2014-07
Mus musculus	GRCm38	2012-01
Gallus gallus	Galgal4	2011-11
Rattus norvegicus	Rnor_6.0	2014-07
Homo sapiens	GRCh38	2013-12
Sus scrofa	Sscrofa10.2	2011-08
Anopheles gambiae	AgamP4	2006-02

Table 2. Comparison of sRNA expression databases

Feature	SEAweb	miRmine[1]	DASHR2[2]	miratlas[3]	YM500v3[4]	SPAR[5]
Organisms	10	1	1	2	1	2
sRNA types	5	1	10	1	5	10
Samples	>4200	304	802	461	>8000*	365[$]
Novel miRNAs	+				+	
Ontology search[#]	+					
sRNA DE	+				+	
sRNA classification	+					
Pathogen k-mer expression	+					
Pathogen k-mer DE	+					
miRNA targets	+				+	
miRNA disease associations	+					
User data upload	+					+
Cross study comparisons	+					+
Re-analysis with selected samples	+					
Dataset search	+			+		+
Genome versions			+			+
Modification analysis				+	+	
Tissue specificity			+			+

This table includes recent sRNA expression databases and a list of features we deem relevant.
*Supports mainly cancer-related datasets.
[#]Use of ontological graphs for the annotation and querying of samples.
[$]Number of datasets based on [2](14) (information about number of samples cannot be obtained).
[1](8), [2](14), [3](9), [4](10), [5](11). For number of samples per organism, see Supplementary Material Table S5.

over 4200 samples in its database, which is considerably less than YM500v3, which hosts over 8000 cancer samples. It is to be noted, however, that the YM500v3 database only supports cancer datasets and no other disease types (Table 2). Additionally, SEAweb also stores in-house data (for a month) from the end users to enable comparison with the data in SEA.

MATERIALS AND METHODS

User data

In case users want to upload their in-house data for comparing it to all the available data in SEAweb, they need to create an account. User-DB, stores their account information as well as sRNA-seq data uploaded by the users. Moreover, user uploaded data is shown only from their respective account and is not available to other users. Users have the option to include their data in the SEAweb for a limited time (30 days). We do not provide users to include their data in the SEAweb permanently or publicly for several reasons: (i) these data are unpublished and we can run into data protection issues. (ii) The ontological annotations of these data by the end users might not be consistent with ours and hence not comparable. (iii) Users might not want to provide information about their experiments such as tissue or disease etc. (iv) End users might not be able trust the system, if anyone could add any quality of data. Data that are added by us follows a manual curation for quality checks. With these measures, we encourage users to upload their data (temporarily), without any data protection issues.

sRNA tissue specificity

To compute tissue specificity indices (TSI) for human sRNAs we calculated median of reads per million (RPM) expression per dataset and tissue. sRNAs with a median RPM expression of at least three were considered in all the tissue specificity analysis. Moreover, sRNAs which had no expression in any tissue and tissues with no sRNAs expression were excluded from the TSI analysis. Healthy and diseased samples were mixed for tissues within the same dataset (Figure 2; Supplementary Figure S3 and Table S1). To remove potential biases introduced by diseased samples we also calculated TSI for non-diseased samples only (Supplementary Figures S2-3, Tables S1 and 4). These analyses were performed for two sets of sRNAs, miRNAs (Figure 2; Supplementary Figure S2 and Table S1) and all non-miRNA sRNAs including piRNA, snoRNA, snRNA and rRNA in SEAweb (Supplementary Figures S3-4 and Table S4). Shannon entropy from BioQC R package was used to calculate TSI for each miRNA across tissues. In the end, 1522 miRNAs across 64 datasets were considered for miRNA tissue specificity in healthy and diseased mixed samples, 1365 miRNAs across 43 datasets were considered for miRNA tissue specificity in non-diseased samples, 4300 sRNAs (piRNA, snoRNA, snRNA and rRNA) across

64 datasets were considered for sRNA tissue specificity in healthy and diseased mixed samples, and 1672 sRNAs (piRNA, snoRNA, snRNA and rRNA) across 43 datasets were considered for sRNA tissue specificity in non-diseased samples (Supplementary Tables S1 and 4).

Novel miRNA gene targets

miRDB (17) was used to obtain targets of the novel miRNAs. We restricted the analysis to highly probable gene targets having a score of 70 or more.

Text mining pipeline

To extract miRNA–gene targets, a dedicated text mining pipeline that reads unstructured text data and outputs structured data that includes the detected and normalized genes and miRNAs as well as the relations between them. Named entity recognition software ProMiner (18) and MiRNADetector (19) are used to detect and normalize genes and miRNAs, respectively. Both detectors are incorporated in the BELIEF text mining pipeline (20) that contains machine learning models to detect specific relations from the complete Medline abstracts.

Gene enrichment analysis

Gene enrichment analysis was performed using webgestalt R package version 0.3.0.

In-house Parkinson's disease data

Isolation of total RNA from peripheral blood sample. Peripheral blood samples were collected into PAXgene Blood RNA tube (PreAnalytiX) from consenting patients and healthy controls, the tubes were gently inverted for multiple times, incubated for 20–24 h under room temperature and stored under −80°C until processing. Total RNA was isolated using the PAXgene Blood RNA kit (PreAnalytiX) according to the manufacturer's protocol. The purity and concentration of isolated RNA were measured with NanoDrop™ 2000 spectrophotometer (Thermo Fisher Scientific). The RNA integrity was determined by Agilent RNA 6000 Nanochip (Agilent Technologies) using the 2100 Bioanalyzer (Agilent Technologies).

Small RNA library preparation. Small RNA libraries were prepared using 1 μg high-quality RNA following the protocol of Illumina TrueSeq small RNA library kit (Illumina). In brief, 3′adapter was denatured for 2 min under 70°C, and ligated to the RNA with T4 RNA Ligase 2 deletion mutant for 1 h at 28°C. Then the reaction was stopped with stop solution for 15 min under 28°C. Subsequently, 5′ adapter was denatured for 2 min at 70°C, then added to the RNA with adenosine triphosphate and T4 DNA ligase for 1 h under 28°C. After adaptors ligation, the RNA was reverse transcribed to complement DNA (cDNA) by using SuperScript II Reverse Transcriptase (Thermo Fisher Scientific) and dNTPs for 1 h at 50°C. Then, the cDNA was indexed and amplified with polymerase chain reaction (PCR) mix and primers supplied in the kit for 12 cycles (denaturing at 98°C for 30 s, annealing at 60°C for 30 s, extension at 72°C for 15 s, with a final extension at 72°C for 10 min). Amplified and indexed cDNAs were then pooled together, mixed with DNA loading dye and loaded on a 5% Tris-borate-EDTA (TBE) acrylamide gels (Bio-Rad). After 57 min electrophoresis under 145 V, the gel was stained with Midori Green for 5 min and viewed under the UV transilluminator, fragments between Illumina's custom ladder 145 and 160 bp were cut out for library preparation. The gel was centrifuged at 20 000 × *g* for 2 min through a Gel Breaker tube (Bio-Cat). Then cDNA was eluted from the homogenized gel by adding 300 μl UltraPure water and shaking under 800 × rpm for 2 h. Then the gel was transferred on a 5 um filter tube (Bio-Cat) and centrifuged for 10 s under 600 × *g* and the gel debris was separated. Afterward, 2 μl Glycoblue, 30 μl of 3M sodium acetate and 975 μl 100% ethanol (pre-chilled under −20°C) were added and well mixed to the sample, following an immediate centrifuge at 20 000 × *g* for 20 min under 4°C. After remove and discard the supernatant, the pellet was washed with 500 μl 70% pre-chilled ethanol. The supernatant was discarded after sample being centrifuged at 20 000 × *g* for 2 min under room temperature, and the pellet was dried in a 37°C heat block for 10 min with open lid. At last, the pellet was resuspended in 10 μl 10 mM Tris–HCL (pH 8.5) and the sample quality was checked using Agilent High Sensitivity DNA chip (Agilent Technologies) using the 2100 Bioanalyzer (Agilent Technologies). All high quality libraries were then sequenced on Illumina HiSeq 2000 Sequencer.

Classification feature pruning

We used Oasis 2 to identify Parkinson's disease (PD) biomarker using 47 PD and 53 frequency-matched healthy controls. For classification analysis, we used all small RNAs ($n = 49\,965$) in Oasis 2. The random forest (RF) classifier in Oasis 2 selected these 18 sRNAs by filtering for informative features while removing the non-informative ones. In brief, The RF selects part of the features for the construction of each tree (mtry parameter, which is by default equal to $\sqrt{n}$ where n is total number of features). If there is a big number of non-informative features ('noise'), many trees can be build based on noise only and therefore affect the classification quality. The way to avoid trees built of noise is feature pruning. The idea is to arrange the variables based on their importance in the full model and then remove less important variables one-by-one, calculating model performance at each step. At the end, the subset of variables with the best performance are considered as important features. We used cross-validation-based backward selection, implemented in the R caret package with 10-fold cross-validation, repeated 10 times at each step for the performance calculation.

SYSTEM DESIGN

SEAweb stores sRNA expression information, sRNA differential expression, sRNA-based classification, pathogenic sRNA signatures from bacteria and viruses, pathogen differential expression, miRNA gene targets and disease association as well as deep and standardized metadata on the samples, analysis workflows and databases used. Metadata information is normalized using ontologies to allow

for standardized search and retrieval across ontological hierarchies (section 'Semantic data layer' and Supplementary Material). The following sections will detail the system design of SEAweb (Figure 1).

Acquisition and analysis of sRNA datasets

SEAweb acquires raw published sRNA-seq datasets and their primary annotation from Gene Expression Omnibus (GEO) and NCBI's Sequence Reads Archive (SRA) repository (Supplementary Material). Novel datasets are downloaded and stored in SEAweb's raw data repository while corresponding annotations are stored in SEAweb's annotation database and are manually curated. In order to retrieve relevant samples for downloading, we optimized our search queries to look for the datasets that have, (i) Experiment type as non-coding RNA profiling by high throughput sequencing, (ii) Sequencing platform as Illumina, (iii) Tissue, cell type, disease or cell line information and (iv) is one of the 10 organisms that SEAweb supports at the moment (Table 1). Raw data are downloaded and subsequently processed automatically by SEAweb's sRNA analysis workflow using Oasis 2.0 (http://oasis.ims.bio/) (Supplementary Material). Subsequently, sRNA counts of high-quality samples are stored in the sRNA expression database. For all the experiments with samples from different conditions such as disease, tissue, cell line or cell type; sRNA differential expression and classification was performed within the experiment using Oasis 2. All possible comparisons for an experiment were taken into account such as healthy versus disease stage 1, healthy versus disease stage 2, disease stage 1 versus disease stage 2 as explained in Supplementary Section 3.4. Additionally, differential expression analysis of detected pathogens was performed using DESeq2 package (21). In order to reduce bias that could be introduced into the data by using different analysis routines, every sample in SEAweb has been analyzed by identical analysis workflows using identical databases and genome versions. Moreover, SEAweb stores analysis workflow parameters used to analyze the samples such as adapter sequence, genome, number of mismatches, minimum and maximum read lengths along with the versioning information about the software and databases used for the analysis. In case of changes in databases or analysis routines, we completely re-analyze all SEAweb' data for consistency.

Additionally, sample annotations are processed automatically with SEAweb's annotation workflow. Processed files and annotations are subsequently semi-automatically curated (Supplementary Sections 2.3 and 3).

Data storage

Once the raw sequencing data is analyzed, the next step is to store the analysis results to the database for downstream analysis and querying. Most metadata is quite different between experiments. Some experiments may have information such as disease, tissue, cell line, gender, age of patient while others may completely lack this. Due to this sparse nature of the biological experimental data, we opted to use NoSQL database management systems such as MongoDB and Neo4J for hierarchical (connected) normalized data. A multi-database management system architecture was used to store different types of data:

In brief, **Expression-DB** is created to save sRNA expression profiles, sRNA differential expression, sRNA based classification as well as pathogen detection and pathogen differential expression. This database stores the identification and description of the experiment (dataset), information about dataset processing (pipeline information and parameters), information about samples. **Association-DB** is used to store genomic coordinates for sRNAs, miRNA gene targets and miRNA diseases association**.** It contains information about sRNA's and gene's chromosomal locations, miRNA target genes and miRNA disease associations. Chromosomal coordinates were obtained from miRBase version 21 (22), ensemble version 84 (23) and piRNA bank (24), miRNA gene targets were obtained from mirTarBase version 7.0 (25) as well as from BELIEF text mining pipeline (20) ('Materials and Methods' section), miRNA disease associations were obtained from HMDD database version 2.0 (15). In order to support the aggregation and comparison of these different types of data we normalized the identifiers across databases. To enable search by ontological terms, **Annotation-DB** is created using the Neo4J database management system. Neo4J is a graph database, representing elements as graph nodes or vertices. **Annotation-DB** (supplementary Figure S1) stores the following three node types: (i) Experiments (datasets), this type of node stores information about the experiment such as description of the experiment, reference to database, experimental design and any global level information, which is common among all the samples. (ii) Sample node type is used to store information about individual sample, such as description of a sample, reference to database, sample-specific processing parameters. (iii) Annotation term node type stores annotation term information of samples such as organism, disease, tissue, cell type, cell line, age, gender, condition (treated\untreated) and extracted molecule for sequencing etc. We normalize organism with the NCBI taxonomy ontology (26), tissue with the BRENDA tissue ontology (27), disease with the human disease ontology (28), cell type with the cell ontology (29) and cell line with the cell line ontology (30) or experimental factor ontology (31) (Table 3). If the annotation term is normalized, it stores ontology reference (term identifier and preferred level). The nodes are connected if they have a relation (dataset/sample, sample/term and term/term) (supplementary Figure S1). To allow for fast ontological search, all parents of a term in the ontology are also stored in the database and connected with their corresponding annotation terms (section 'Semantic data layer' and Supplementary Material). **User-DB** stores in-house sRNA-seq data (differential expression and classification from Oasis) uploaded by the users. This database allows users to compare their own data to the huge and diverse sRNA-seq published data. User uploaded data are deleted after 30 days.

In addition, SEAweb contains information about the GEO series accession (GSE) and sample accession (GSM) identifiers along with the sample identifier from the SRA database (SRR) in the Annotation-DB together with the annotations and in the Expression-DB together with expression profiles, differential expression and classification anal-

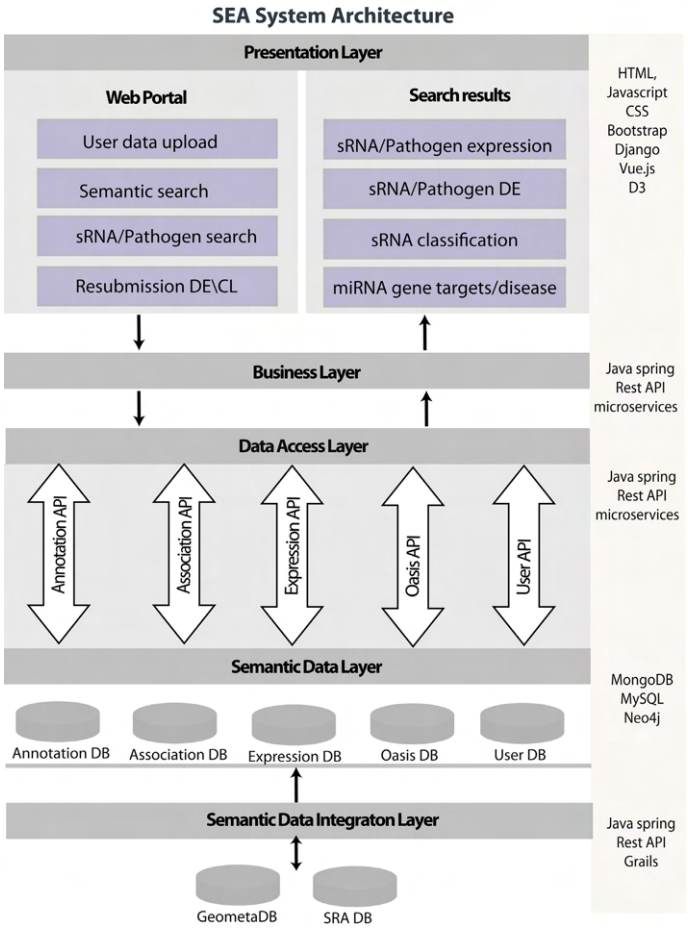


Figure 1. SEAweb system architecture. SEAweb system was developed using the modular system design approach (model-view-controller). The system has a presentation layer for user interface and visualization of search results. Presentation layer is followed by a business layer which transform complex user queries and distribute particular requests to the data access layer REST API services. There is a semantic data layer, to store and access primary and derived data together with annotations and links to secondary data. Annotation-DB stores metadata for experiments, samples, corresponding ontological terms as well as relations between dataset/sample, sample/term and term/term. Association-DB contains information about sRNAs and genes chromosomal locations, miRNA target genes and miRNA disease associations. Expression-DB stores sRNA expression profiles, sRNA differential expression; sRNA based classification as well as pathogen detection and pathogen differential expression. It also store details about dataset processing pipeline and parameters. Oasis-DB was used to store novel predicted miRNA information. User DB contains in-house data uploaded by the end users from Oasis 2 pipeline. Semantic data integration layer integrates primary and secondary data into the mentioned databases. Microservices were implemented in order to achieve strong encapsulation and well-defined interfaces via REST APIs.

ysis. We optimize search and retrieval times by indexing for the most common queries and most relevant terms.

Semantic data layer

Given the diversity of the biological data, users of the SEAweb system are given a possibility to interpret data independently using common terminologies**.** In order to enable users to browse data autonomously using common well-structured terminology, a standardized semantic layer for data retrieval is developed (Figure 1). It includes semantic annotations of data and semantic search, linking data with semantic lookup platform (OLS), as well as storing primary and derived data together with provenance information and references to secondary data.

One of the most important aspects of semantic layer are ontology-based data annotations. They enable interoperability of the data, as well as using of standard terminologies for data retrieval. It is important to standardize annotations using ontologies and semantic mappings (32). Ontologies define not only standard classes, but also the relations between terms, which enables semantic search by term hierarchies, for example, by parent terms. In SEAweb, we connect (normalize) annotations with ontologies in

Table 3. SEAweb keys and used ontologies (as of June 2019)

Key	Ontology(s)	# Annotations	# Terms
Organism	NCBI Taxonomy	4235	126
Tissue	BRENDA tissue/enzyme source	3021	190
Disease	Human Disease Ontology	1951	287
Cell type	Cell Ontology	732	304
Cell line	Cell Line Ontology	663	132
	Experimental Factor Ontology	134	76

a semi-automatic way, i.e. first automatically extract possible annotation terms from GEO descriptions and normalize them, and later curate annotations manually (Supplementary Section 3). The Ontologies and the number of normalized terms in SEAweb are listed in Table 3. To enable the search across ontological hierarchies we integrated data with the relevant ontologies into the graph database Neo4J (supplementary Figure S1).

Ontology Lookup Service (OLS) is a service which allows to extract relevant terms from ontologies together with term information. SEAweb uses OLS for annotation normalization and accesses ontologies via the OLS REST interface, which supports complex and compound queries and query auto-completion (33). Details about annotation criteria, processing and group annotation are described in Supplementary Section 3.2.

Another aspect of the semantic layer is storing of the primary and the derived data together with provenance information. For SEAweb, primary data are FASTQ files, retrieved from the NCBI SRR database. This data are not stored after Oasis analysis, only provenance data about source and analysis details is saved. So for SEAweb, primary data are sRNA counts. Based on those counts, DE and classification results are obtained and are also saved to allow data interpretation. From derived data, the provenance information allows to retrieve raw counts and check how those results are obtained.

Querying and visualization

Application programming interfaces (APIs) are developed to access data in SEAweb databases (Supplementary Section 3.5). The APIs help to use the multi-database system components independently as well as in combination. In brief, we extend the SEAweb backend application with RESTful web services, such as Annotation-API, Association-API, Expression-API, User-Expression-API, Predicted miRNA-API to access Annotation-DB, Association-DB, Expression-DB, User-Expression-DB and Oasis-DB, respectively. Additionally the SEAweb business logic API is created in order to combine all those APIs and make necessary data transformations between frontend and other APIs. As a result, the user can make queries to answer biological questions like; what is the expression of hsa-miR-488-5p across all human tissues? Is hsa-miR-488-5p expressed higher in adenocarcinomas as compared to other cancer types? Is a particular sRNA/pathogen DE in Alzheimer's disease? What are common DE sRNAs/pathogens or potential sRNA based biomarkers in a particular disease or tissue? What is the expression of a novel miRNA for known disease states? All API calls are described in Supplementary Section 3.5.

Table 4. SEAweb browser compatibility

Browser	Version
Chrome	61.0.3163.100, 62.0.3202.62
Mozilla Firefox	55.0.3, 56.0 (64-bit), 57.0 (64-bit)
Chromium	62.0.3202.75
Safari	11.0.1
Internet explorer	11

Browsers that are used to test SEAweb functionalities.

In addition, users can browse and query all datasets using the browse link. A three-panel browse function (Supplementary Figure S8) facilitates searching for specific small RNAs (miRNA, piRNA, snoRNA, snRNA and rRNA), annotation terms (organism, tissue, cell type, cell line and disease), and pathogens (bacteria or viruses). By selecting single or several terms from the three panel browse function the user can make arbitrarily specific searches in SEAweb. For example, the user can click on a small RNA and cancer to see its expression profiles in the cancer datasets.

In brief, the SEAweb system is developed using the modular system design approach (Figure 1). We build micro services to achieve strong encapsulation and well-defined interfaces via REST APIs. An object oriented programming approach is used to build the SEAweb application using the spring framework and Java 8. The SEAweb user interface (UI) is developed in Django framework version 2.0, HTML version 5, D3 and CSS 3. SEAweb visualizes the results depending on the user query, such as a violin plot for the expression of sRNAs or pathogens. Upset plots are shown for the overlap of sRNAs or pathogens (based on DE or classification) across experiments. SEAweb enables the download of search results in the form of CSV files. The functionality is tested on all major browsers (Table 4).

SEAweb usage

SEAweb is a publicly available data repository and a web server and users can use it without an account or login. In case users want to upload and compare their own data to the data in SEAweb they need to create an account. Users have an option to sign in with their google account or they can register in the SEAweb system directly with a valid email address, choosing a username and password for their account. We have created User-DB to store their account information as well as sRNA-seq data uploaded by the users. Moreover, user-uploaded data are only accessible from the user's account. Users have the option to include their data in SEAweb for 30 days. For the data protection, security, and storage space reasons, we currently do not allow users to add data permanently to SEAweb ('Materials and Methods' section).

APPLICATION OF SEA

In this section, we describe a few examples that illustrate how SEAweb can be employed to answer biological questions and to uncover unappreciated properties of sRNA data integration with interactive result visualization. First, we took advantage of the diverse and massive sRNA-seq data in SEAweb to present the most comprehensive set of tissue specific miRNAs till date. Second, we utilized the pathogenic reads in sRNA-seq to find their association to diseases. At last, we show a use case of SEAweb by comparing an in-house PD sRNA-ome to other neurodegenerative diseases sRNA expression profiles available in SEAweb.

sRNA tissue specificity

Several studies have shown tissue specificity for miRNAs. Recently, Ludwig *et al.*, (34) analyzed several human tissue biopsies of different organs from two individuals to define the distribution of miRNAs using tissue specificity index (TSI) and found several groups of miRNAs with tissue-specific expression. Similarly, Lee *et al.*, (35) provides the expression of 201 miRNAs across nine human tissues to find tissue specificity of miRNAs. miRNAs whose expression is 20-fold or higher in a certain tissue compared with the mean of all the other tissues were characterized as tissue specific. According to Lee *et al.*, skeletal muscle, brain, heart and pancreas are the tissues expressing the most specific miRNAs. Moreover, Guo *et al.*, (36) manually extracted 116 tissue-specific miRNAs across 12 human tissues. We used Shannon entropy to calculate TSI for each human miRNA across all the human tissues available in SEAweb ('Materials and Methods' section). In order to calculate tissue specificity, we mixed healthy and diseased human samples (where available) within an experiment (Figure 2 and Supplementary Table S1). We used very stringent criteria: miRNAs with Shannon entropy score more than 0.8 were considered as tissue specific and $\leq$0.2 were considered as ubiquitous miRNAs (Figure 2 and Supplementary Table S1). We were able to provide by far the most comprehensive set of 591 distinct tissue-specific miRNAs across 29 tissues; blood plasma, skin, blood serum, liver, bone marrow, serum, testis, blood, semen, prefrontal cortex, peripheral blood, colon, brain, cornea, breast, renal cortex, bladder, embryo, placenta, lung, tongue, tonsil, skeletal muscle, kidney, lymph node, heart, muscle, thyroid gland and neocortex (Figure 2 and Supplementary Table S1). In order to compare the TSI for miRNAs in SEAweb with the existing findings, we merged the list of miRNAs from the above studies and retained all the 12 tissues. Out of 12 tissues, we did not have sequencing data for four of them: thymus, pancreas, spleen and bone.

We were able to detect two out of the three **heart** specific miRNAs (miR-1 and miR-302d) from Lee *et al.*, study, and 6 out of 10 heart specific miRNAs (hsa-miR-1-5p, hsa-miR-208a-3p, hsa-miR-208b-5p, hsa-miR-208b-3p, hsa-miR-302d-3p, hsa-miR-133b, hsa-miR-302a-3p, hsa-miR-302a-5p, hsa-miR-302b-3p) from the manually curated list of Guo *et al.* miR-208 is obtained from an old annotation, because the latest release of miRBase has more specific annotation like miR-208a-3p, miR-208b-3/5p. **Interestingly** we were able to find the whole family of miR-208 as heart specific. We were not able to detect miR-126, miR-302c, miR-367, hsa-miR-133a-5p in heart. Of note, none of these three is heart specific in the Lee *et al.*, study.

Muscle and **brain** were the only two tissues covered by all the three above mentioned studies. In muscle, we were able to detect two out of the three muscle specific miRNAs (miR-133b, miR-1-3p) from Ludwig *et al.*, three out of four (miR-95 was not found to be muscle specific) from Lee *et al.*, and 4 out of 10 for Guo *et al.*, compilation. We were not able to detect miR-206, miR-133a, miR-134, miR-193a, miR-95 and miR-128a. Note that from the same study miR-134 is mentioned as muscle as well as testis specific and miR-128a as muscle as well as brain specific. Moreover miR-95 is the only miRNA that is muscle specific in all of the three studies.

Another tissue covered by all of the three studies is the brain. In total 30 miRNAs were known to be brain specific, only 1 out of 30 (miR-7) is common among all the three studies and only three in two studies (miR-124, miR-9, miR-218) one of which is in the curated list. In our study, we found 26 miRNAs to be brain specific but none from the known ones.

Tissue with the most number (n = 43) of known specific miRNA was **placenta** provided by Guo *et al.* Interestingly, miRNAs associated with placenta were mostly evolutionary related. We were able to detect these evolutionary related miRNAs to be placenta specific as well. In short, we detected **517**a/b/c, **518**a/b/c/d/e/f, **519**a/b/c/d/e, **520**a/d/e/f/g (not detecting **520**b/c/h). Moreover we were also able to detect miR-371, miR-372, miR-512, miR-522, miR-523, miR-524, miR-525, miR-526b and miR-527. Out of 43, we detected 35 and did not detected miR-377, miR-526a, miR-184, miR-154, miR-381, miR-503, miR-450 and miR-136. We detected only 2 (miR-513c-5p, miR-202-3p) out of 15 for **testes**. There were two tissues, **lung** and **liver**; mentioned only in one study Guo *et al.*, we could not detect the only miRNA miR-126 for lung. Interestingly this miRNA is also mentioned as heart specific in the same study. We also did not find the four liver specific miRNAs miR-122, miR-483, miR-92a, miR-192; two (miR-483, miR-92a) of which are shown as bone specific in the same study. In **kidney** we could not detect any miRNA out of eight kidney specific in Guo *et al.* Of note Lee *et al.*, also found only one miRNA miR-204 to be kidney specific and does not have any evidence for the rest of the seven miRNAs. In brief, there is no significant consensus on the tissue specific miRNAs in the previous studies. However, our work still aligns reasonably well to their findings.

To understand if disease samples might affect the tissue specificity calculations we also performed a tissue specificity analysis using only non-disease samples (Supplementary Figure S2, 4, Table S1 and 4). Using only non-disease samples we found three additional tissue specific miRNAs, hsa-miR-503-3p in placenta, hsa-miR-1-3p and hsa-miR-133a-5p in muscle and heart. Overall, our miRNA tissue predictions, mixed as well as non-disease only, were consistent with published information on tissue-specific miRNA expression (34–36). As Ludwig *et al.*, used only two individual's tissues, Lee *et al.*, also performed own experiments in a control (same laboratory, same protocols) environment and used different statistical methods compared to ours, we

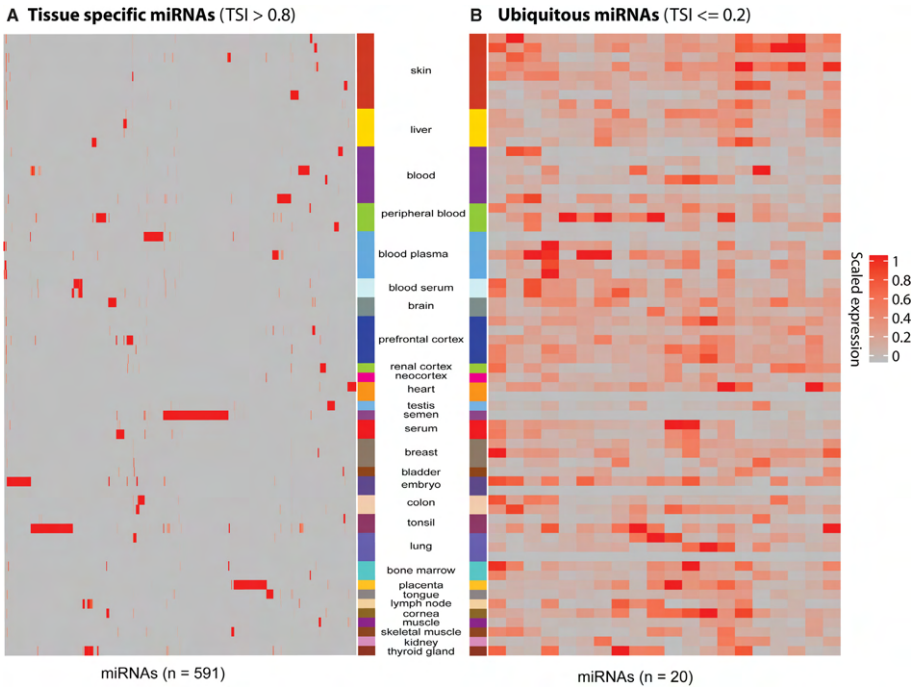


Figure 2. Tissue specific miRNAs. The heatmaps show the scaled expression (0-1) of (**A**) tissue specific or (**B**) ubiquitous miRNAs across all the tissues. (A) Tissue specific miRNAs. miRNA expression across all the tissues with TSI > 0.8 ($n = 591$). (B) Ubiquitous miRNAs. miRNA expression across all the tissues with TSI $\leq$ 0.2 ($n = 20$). miRNA names are omitted for simplicity. A complete list of tissue specific and ubiquitous miRNAs with their Shannon entropy score can be found in Supplementary Table S1. These calculations are based on healthy and disease samples within an experiment ('Materials and Methods' section).

were still able to get a reasonable overlap with tissue-specific miRNAs considering diverse (different laboratories, different protocols) and massive data. Therefore, we think that this work provides the most comprehensive set of tissue-specific miRNAs till date ($n = 591$ miRNAs) (Supplementary Table S1). In order to explore the tissue specificity of other types of sRNAs in SEAweb such as piRNA, snoRNA, snRNA and rRNA, we repeated the above analysis with exactly the same set of samples once for the healthy and diseased mixed and once for the non-diseased samples (Supplementary Figures S3, 4 and Table S4). We found 3445 out of 4300 (filtered for minimum reads, see 'Materials and Methods' section) sRNAs to be tissue specific and only 73 to be ubiquitous in the healthy and disease mixed samples (Supplementary Figure 3 and Table S4). In the non-diseased samples, we found 1005 sRNAs to be specific and 45 to be ubiquitously expressed across tissues (Supplementary Figure S3 and Table S4).

Known and novel bacterial or viral infections

We have validated our approach of pathogen detection in Oasis 2 (16) using sRNA datasets with defined viral or bacterial infections. Overall, the prediction of bacterial (*Mycobacterium abscessus*) and viral (HIV, HHV4, HHV5, Gallid_herpesvirus_2) infections resulted in high F-scores, recall and precision, especially when the top five predicted pathogen species are reported. However, the current work additionally involves differential expression analysis of pathogens and therefore we validated our approach of pathogen differential regulation using seven datasets with known infection status. The samples in these datasets are known to be infected with seven bacterial pathogens and three viral pathogens. Of note, we focused on within-dataset comparison in order to avoid technical confounders (Supplementary Table S2). For each sample, k-mer counts were calculated for all infectious species present in Kraken database (4336 viral and 2784 bacterial/archaeal genomes) and differential abundance analysis was carried out for those species that have at least three counts (baseMean) in a particular comparison. As expected, in all comparisons the known pathogen represented the best hit (i.e. smallest adjusted *P*-value) except Vaccinia virus (Figure 3A). However, Vaccinia virus has the highest $\log_2$ fold change as expected within the dataset (GSE54235) comparison. It is worthy to note that Chlamydia trachomatis detection is based on sRNA-seq performed on conjunctival tissue from children with follicular trachoma and children with healthy conjunctivae, indicating a good performance of our pathogen detection pipeline from tissues.

Next, we aimed to find novel associations of pathogens with disease. We took all the comparisons, which has

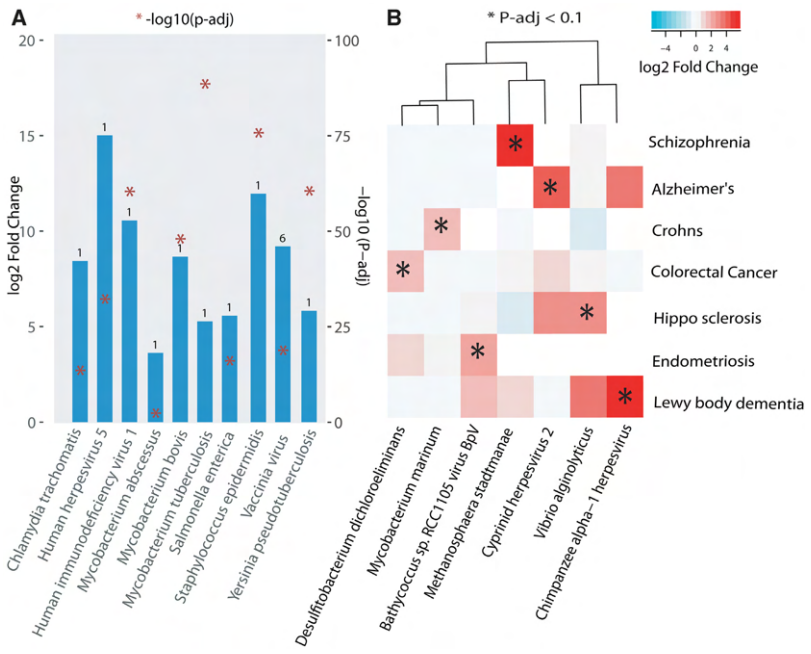


Figure 3. Known and Novel bacterial or viral infections. **(A)** Known associations. Pathogen detection using seven datasets known to be infected with seven bacterial and three viral pathogens. Bar represents pathogen $\log_2$-fold difference between the uninfected and infected state (Supplementary Table S2). Number on top of the bar denotes rank of the pathogen compared to all the other DE pathogens within the comparison (i.e. smallest adjusted *P*-value). **(B)** Novel associations. Heatmap shows $\log_2$-fold difference of pathogens significantly upregulated in disease as compared to healthy (fold change > 1 and padj < 0.1) (Supplementary Table S2). Comparisons that have less than six pathogens significantly DE are selected for specificity. Details about dataset, comparison groups, $\log_2$fold and padj for both (A and B) are provided in (Supplementary Table S2).

'healthy' and at least a disease state annotation (Supplementary Table S2). In order to achieve more specificity we took only comparisons that have less than six pathogens significantly upregulated in disease as compared to healthy (FC > 1 and padj < 0.1). There were a total of eight comparisons but we removed 'GSE69837' as this was a known case (Chlamydia trachomatis already shown in Figure 3A). It was interesting to find viruses and bacteria significantly upregulated in sRNA-seq data in certain disease compared to healthy patients (Figure 3B). Some of the most interesting cases are highlighted in this section below.

Mycobacterium marinum in patients with ileal Crohn's disease. In the original study, expression of microRNAs in mucosae of patients with a normal pouch after colectomy for intractable ulcerative colitis was compared to several control cohorts, among them was a cohort of patients with Crohn's disease (CD) of the terminal ileum (37). CD patients were previously not exposed to immunosuppression. Compared to patients with non-inflamed ileal pouch, patients with ileal CD showed an increased mucosal expression of *Mycobacterium marinum*. The bacterial genus *Mycobacterium* causes diverse diseases in humans, of which Tuberculosis is the most serious with around one-quarter of the world population latently infected and ∼1.6 million deaths in 2017 on a global scale. *M. marinum* is a non-tuberculous (also termed 'atypical') *Mycobacterium* species, which is ubiquitously abundant in aquatic environments (38). Infection of humans is well known, but it is considered a rare event. It typically occurs after exposure to contaminated water or infected marine animals, and it is more common in immunosuppressed individuals. The most commonly affected organ is the skin, in more severe cases involvement of muscles, bones or joints is reported (38). Opportunistic infection with *M. marinum* in CD is recognized in those patients receiving anti-tumor necrosis factor therapy (e.g. infliximab) (39). However, to the best of our knowledge, enteric super-infection with *M. marinum* has not been reported in the literature so far. Interestingly, due to the resemblance of the granulomatous intestinal inflammation in CD with enteric infection caused by other Mycobacteria, it has been hypothesized that Mycobacterial infection is involved in the pathogenesis of CD, with much focus on *Mycobacterium avium paratuberculosis* (40). However, the aetiological significance of this pathogen in CD remains uncertain. Hence, the gut mucosal prevalence of *M. marinum* and its potential pathophysiologic significance in patients with CD should be further explored.

Methanosphaera stadtmanae in patients with schizophrenia. We detected an overabundance of *Methanosphaera stadtmanae* in neurons derived from induced pluripotent stem cells (iPSC) of patients with schizophrenia, compared to healthy controls. *M. stadtmanae* is an Archaeal microor-

ganism which is frequently detected in the healthy human gut microbiota (41). It is involved in intestinal methanogenesis and associated fermentative dynamics. *M. stadtmanae* is recognized by the innate immune system, therefore it can induce inflammatory cytokine responses and could have diverse immunomodulatory functions (42). Interestingly, *M. stadtmanae* was found with an increased prevalence in faecal samples of patients with inflammatory bowel diseases (IBD) Crohn's disease (CD) and ulcerative colitis with antigen-specific IgG-responses (43). Immune system processes have been proposed to be involved in the pathogenesis of schizophrenia (44). Regarding the immunegenetic basis of schizophrenia, genome-wide pleiotropy has been reported between schizophrenia and CD as well as an increased prevalence of schizophrenia in patients with IBD (45). Therefore, the potential immunogenic importance of *M. stadtmanae* in schizophrenia should be investigated.

Chimpanzee herpesvirus in Lewy body dementia. We detected an increased abundance of a viral pathogen identified as *chimpanzee herpesvirus* (ChHV) in the cerebral cortex of patients with lewy body dementia (LBD) compared to non-demented controls (46). ChHV is an alphaherpesvirus closely related to human herpes simplex virus type 2 (HSV-2) (47). LBD is a neurodegenerative disorder, which underlies 4.2% of all dementia cases, second only to Alzheimer's dementia (AD) (48). The aetiology of LBD is obscure, but growing evidence points toward neuro inflammation as a key pathophysiologic factor, analogous to the pathogenesis of AD (49). In AD it is assumed that multiple pathogens infecting the brain are key triggers of neural dysfunctional protein accumulation and neuro inflammation in genetically vulnerable individuals (50). Among the pathogens detected in brains of AD patients, multiple lines of evidence point at herpes simplex virus type 1 (HSV-1) and HSV-2 as two of the main drivers of AD neurodegeneration (50,51). Given the close phylogenetic relationship between ChHV and HSV-2, ChHV might play a role in inflammatory neurodegenerative processes in LBD similar to the other herpesviruses in AD. Therefore, the association detected in the present study should be further elaborated.

Analyzing in-house data and comparing with SEAweb data

One of the key features available in SEAweb is uploading the in-house data and comparing it with the already integrated data. Mostly, researchers use different analysis pipelines to carry out differential expression or classification, which makes it very hard to compare the results with the publicly available data. Therefore, we require a database with interactive visualizations that has all the publicly available data analyzed using the same pipeline with same parameters. For SEAweb, we have analyzed and integrated all the data using Oasis 2 pipeline. We expect that comparing the in-house data with the data in SEAweb will yield disease-specific signatures, in this case a sRNA or group of sRNAs. Note that uploading to SEAweb requires the output of Oasis 2 (Supplementary Material).

In order to test this feature, we uploaded in-house sRNA-seq data from well characterized 47 PD and 53 frequency-matched healthy controls, which is a baseline data from the longitudinal *de novo* Parkinson disease (DeNoPa) cohort (Supplementary Table S3) and available as 'demo user data' in SEAweb. SEAweb gives us a unique opportunity to identify PD-specific biomarkers associated with early-stage PD that can eventually help us in early diagnosis, therefore, better treatment of the disease. Below we describe the differential expression and classification results from PD data and an approach in order to identify PD-specific biomarkers that do not overlap with other neurodegenerative diseases.

We found four significantly DE miRNAs with adjusted *P*-value < 0.1. Out of these, two are upregulated in PD (hsa-miR-502-3p and hsa-miR-532-5p) and two are downregulated in PD (hsa-miR-30d-5p and hsa-miR-22-5p) (Supplementary Table S3). Next, we overlapped these four DE miRNAs with all the neurodegenerative disease-related datasets integrated in SEAweb. We focused on nine comparisons (from five datasets) in which one of the conditions is a healthy state and the other is a diseased condition (Alzheimer's disease (AD), LBD, tangle-predominant dementia, Huntington's disease (HD), Frontotemporal dementia or Hippocampal sclerosis of aging). Out of the two upregulated miRNAs in PD, one (hsa-miR-502-3p) is upregulated in Alzheimer's disease and one (hsa-miR-532-5p) is upregulated in both Alzheimer's and Huntington's disease (Figure 4A). In contrast, none of the downregulated miRNAs in PD were found to be significantly down in any of these nine comparisons. Interestingly, it has been shown that the expression of miR-22 is downregulated in a 6-hydroxydopamine-induced cell model of PD using RT-PCR (52). Moreover, Margis *et al.*, found that hsa-miR-22 has reduced expression in the blood of *de novo* PD patients (53). Furthermore, family members of hsa-miR-30d-5p are known to be deregulated in PD (54) and putatively target the PD-related gene, **LRRK2 (PARK8)** (55). These results confirms, the potential role of hsa-miR-30d-5p and hsa-miR-22-5p in PD. To explore the mechanism by which these two miRNA are involved in PD, we performed gene ontology (GO) analysis of the validated and predicted targets using webgestalt (56). The top ten terms ranked according to FDR adjusted *P*-value are shown in the (Figure 4B). The top significant hit (FDR < 0.1) is axon development. Recent publications (57–59) have suggested the role of massive and unmyelinated axonal arbor in PD. In substantia nigra pars compacta (SNc), the axonal arbor of dopamine neurons is very large as compared to other neuronal types. This leads to the hypothesis that these dopamine neurons have selective and exceptional vulnerability in PD, and have a higher energy demand that may play a crucial role in cell death (57).

To obtain a unique PD biomarker we explored the classification results integrated in SEAweb. PD and healthy were classified with an AUC of 0.89 (Figure 4D). Interestingly, the classifier used only 18 sRNAs ('Materials and Methods' section) to separate the two states (Supplementary Table S3 and Figure S5). Moreover, only two sRNAs hsa-miR-30d-5p (downregulated in PD) and hsa-miR-502-3p (upregulated in PD) are DE between healthy and PD out of the 18 sRNAs identified by the classifier (Supplementary Figure S6). We overlapped these 18 sRNAs with the classification results from other neurodegenerative diseases integrated in SEAweb (Figure 4C). There are only three sRNA that are

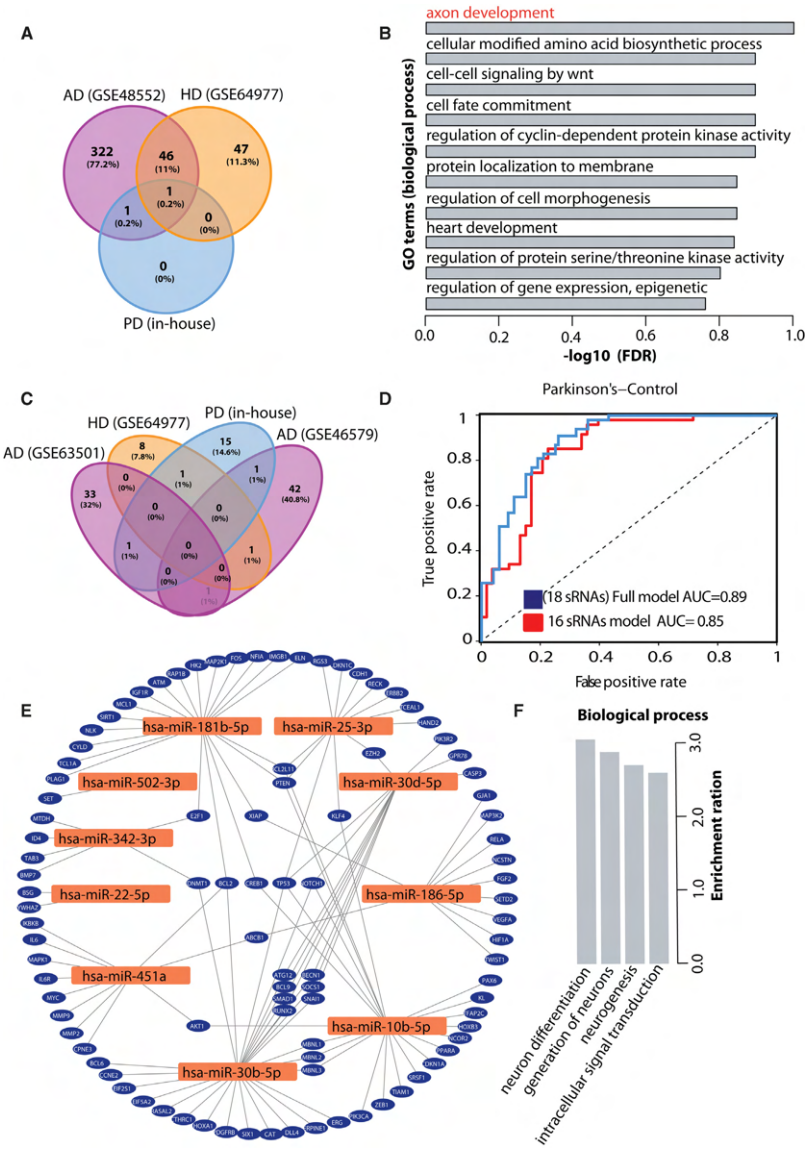


Figure 4. In-house *de novo* Parkinson disease (DeNoPa). (**A**) sRNA DE Overlap. Overlap of upregulated sRNAs between in-house denopa (blue), AD (purple) and HD (orange). Overall nine neurodegenerative disease comparisons were considered and overlap was found with these two datasets. (**B**) GO terms. Top 10 GO terms associated with the target genes of the two downregulated sRNAs. (**C**) sRNA classification Overlap. Overlap of classification features (sRNAs) between in-house denopa (blue), AD (two datasets) (purple) and HD (orange). (**D**) DeNoPa classification. Receiver-operating characteristic (ROC) curve showing true- and false-positive rates for DeNoPa disease prediction based on sRNA expression profile using 18 sRNAs in full model (blue) and 16 unique (not found in other neurodegenerative diseases) sRNAs (red). (**E**) PD associated genes. Network of PD associated genes and 13 known miRNAs from the classification. (**F**) GO terms for novel miRNAs. GO terms associated with the target genes of the three novel miRNAs from the classification.

also found in AD or HD but they have opposite change of expression. This suggests the specificity of these sRNAs to PD as compared to other neurodegenerative diseases. Furthermore, to filter out sRNAs known to be associated with other neurodegenerative diseases, we used the association database of sRNA-disease association available in SEAweb. The results showed that hsa-miR-342-3p has been associated with other neurodegenerative diseases (60,61). Next, we also filtered out sRNAs if the base mean read count is less than five and also, hsa-miR-502-3p that was found to be upregulated in AD (Figure 4A). Then we run a random forest classifier using the normalized counts for the remaining 15 sRNAs and hsa-miR-22-5p that is downregulated in our data. (Figure 4D) shows that using 16 sRNAs to classify PD and controls, yielded 85% area under the curve (AUC) with 83% recall and 77% of precision. Furthermore, to find the relevance of the 13 known miRNAs (out of 16 sRNAs) in PD, we obtained their target genes from SEAweb (only 10 miRNAs out of 13 have targets supported by strong evidence) and overlapped with the targets genes of PD associated miRNAs in SEAweb. Interestingly, these 10 known miRNAs targets 96 genes, which are known to be associated with PD (Figure 4E and Supplementary Table S3). The list includes **TP53** (62) that contributes to the apoptotic deterioration taking place in PD, **PTEN** (63) that has been linked to PD via DNA damage and DNA repair machinery, **SMAD1** (64) is an important regulator required for neurite growth, **EZH2** (65) is a lysine methyltransferase component of polycomb repressive complex 2 that has been associated with PD and **BCL2** (66) is required for proper development of the dopaminergic system and has been implicated in the pathogenesis of PD. To gain further insights into the three novel predicted miRNAs (out of 16 sRNAs) used to classify PD and controls, we performed gene enrichment analysis on their target genes using webgestalt (67). The novel miRNAs were p-hsa-miR-113, p-hsa-miR-247 and p-hsa-miR-235-**1/2/3** (Supplementary Material). We used miRDB (17) to get target genes for the mature sequences of these predicted miRNAs ('Materials and Methods' section). Interestingly the GO terms for these miRNAs were **neuron differentiation, generation of neurons**, **neurogenesis** and **regulation of intracellular signal transduction** (Figure 4F). All these processes are highly related to PD, and hence we think these novel miRNAs should further be explored and validated in the laboratory. Predicted structure of these miRNAs can be found in Supplementary Material.

All together, these results make a strong case in favor of using SEAweb in order to retrieve disease-specific biomarkers.

CONCLUSION

SEAweb is designed for the biological or medical end-user that is interested to define where and when a sRNA of interest is expressed. Prototypical questions that can be addressed with SEAweb are: What is the expression of hsa-miR-488-5p across all human tissues? Is hsa-miR-488-5p expressed higher in adenocarcinomas as compared to other cancer types? Is the tissue-specific expression of hsa-miR-488-5p conserved in mice? Its unique selling points are the deep and standardized annotation of meta-information, the re-analysis of published data with Oasis 2 to reduce analysis bias, a user-friendly search interface that supports complex queries and the fast and interactive visualization of analysis results across 10 organisms (Table 1) and various sRNA-species. SEAweb also contains information on the expression of currently 769 high-quality predicted miRNAs, across organisms and tissues.

In addition, SEAweb also stores sRNA differential expression, sRNA based classification, pathogenic sRNA signatures from bacteria and viruses and pathogen differential expression. Furthermore, SEAweb can be used to search gene targets or diseases associated with a miRNA. Moreover, SEAweb allows end users to upload their analysis results of differential expression and classification from Oasis 2. This will allow users to compare their data to over 4200 experimental samples across different conditions. SEAweb also provides users with an option to perform on the fly analysis such as overlapping DE sRNAs or pathogens across different studies or the most important features (sRNAs) identified with classification. SEAweb enables end users to re-submit samples from interactive plots for differential expression or classification, this will help users to choose samples of their choice from an experiment (Supplementary Figure S7).

Moreover, SEAweb is continuously growing and aims to eventually encompass all sRNA-seq datasets across all organisms deposited in GEO and other repositories. In order to keep SEAweb up to-date with the current small RNA sequencing data or the data that will be published to GEO in the future, we have written programs that automatically search GEO and SRA databases every two weeks (consistent with the GEO update cycle). These programs download raw fastq files, submit these to Oasis 2, and assign responsibility to another program for the semi-automated annotation for tissue, cell line, cell type and other meta-data available. In case the system cannot fully annotate all fields, automatic annotation is followed by manual curation using a front-end curation system. Currently manual annotation QA is the rate-limiting step, which is why we actively develop deep learning-based annotation prediction routines for future versions of SEAweb (68). Genome versions will be updated with every major release of SEAweb. SEAweb will be backward compatible in the future by allowing users to choose previous genome versions and annotations.

A detailed comparison of SEAweb to other existing sRNA expression databases highlights that SEAweb is superior in terms of supported organism, annotations, diseases, tissues, sRNA based classification, pathogen k-mer DE, known miRNA disease associations, user specific experimental data upload, cross study comparisons and reanalysis with selected samples (Table 2).

As far as we are aware, SEAweb is the only sRNA-seq database that supports ontology-based queries, supporting single or combined searches for five predefined keys (organism, tissue, disease, cell type and cell line) across all datasets. However, the SEAweb database system contains additional (meta)-information including age, gender, developmental stage, genotype as well as technical experimental details such as the sequencing instrument and proto-

col details (e.g. library kit, RNA extraction procedure). We plan to normalize most of this additional information in future versions of SEAweb. This will allow users, for example, to query and analyze sRNA expression effects that are introduced by library kit or sequencing platform differences (both of these features can introduce large biases in the detection and expression of sRNAs). Other future developments will include information on sRNA editing, modifications and mutation events.

In summary, SEAweb supports interactive result visualization on all levels, from querying and displaying of sRNA expression information to the mapping and quality information for each of the over 4200 samples. SEAweb is a fast, flexible, and fully interactive web application for the investigation of sRNA and pathogen expression across cell lines, tissues, diseases, organisms and sRNA-species. As such, SEAweb should be a valuable addition to the landscape of sRNA expression databases.

Additionally, we presented the most comprehensive set of tissue specific miRNAs till date. We were able to provide by far the most complete set of 591 distinct tissue specific miRNAs across 30 tissues. To our knowledge this is by far the most comprehensive analysis (set) of tissue-specific miRNAs.

In the current work, we also found pathogen signatures from sRNA-seq data. We found signatures of pathogens in severe diseases like dementia. In brief, we found differential regulation of *M. marinum* in patients with ileal crohn's disease, methanosphaera stadtmanae in patients with schizophrenia and chimpanzee herpesvirus in LBD.

From our in-house PD data, we were able to find potential biomarkers based on differential expression and classification for the early detection of PD. The top term for the GO analysis of the two downregulated miRNAs is axon development, suggesting their role in PD. Moreover, gene targets of the sRNAs for the top important features (potential biomarkers) for PD using classification were overlapping with the targets of the known PD miRNAs. Additionally, GO analysis for the targets of the three novel miRNAs are neuron differentiation, generation of neurons, neurogenesis and regulation of intracellular signal transduction (Figure 4F). We think these novel miRNAs should be further explored and validated in the laboratory.

At last, researchers have used massive sRNA data from SEAweb for other tasks, for example, it enables to use deep learning for data augmentation problem such as predicting sex and tissue based on sRNA expression profiles (68). As such, SEAweb should be a valuable addition to the landscape of sRNA-seq web applications.

DATA AVAILABILITY

SEAweb is implemented in Java, J2EE, spring, Django, html5, css3, JavaScript, Bootstrap, Vue.js, D3, mongodb and neo4j. It is freely available at http://sea.ims.bio/.

SUPPLEMENTARY DATA

Supplementary Data are available at NAR Online.

ACKNOWLEDGEMENTS

We would like to thank Mariah Snyder, Yu Zhao, the ZMNH IT and all of the SEAweb users for helpful suggestions.

Authors' contributions: S.B. initiated the study and designed the web application as well as analyses together with R.R. and V.B. R.R., A.S., A.L. designed and implemented the expression database. R.R. and M.F. designed and implemented association database. M.F., S.M. and J.F. designed and developed the semantic integration service. R.R., A.S., A.L., M.F. implemented the APIs. R.R. and A.G. implemented the pipeline to automatically download and submit sRNA-seq data to Oasis 2. R.R. and A.G. implemented the predicted miRNA API. S.H. designed the development and deployment system infrastructures. T.S. annotated the experiments and samples. D.S.M. and A.L. developed the interactive user interface. A.L., J.B., V.B. and R.R. made the user manual and tutorials. V.B., R.R., A.L. and T.L. analyzed the sRNA-seq data mentioned in the manuscript. A.R., C.T. and B.M. provided and sequenced the Denopa sRNA samples. S.B., R.R., V.B. and T.L. wrote the manuscript. All authors read and approved the final manuscript.

FUNDING

Network of Centres of Excellence in Neurodegeneration (CoEN) Initiative; BMBF Integrative Data Semantics in Neurodegeneration [031L0029B, IDSN]; KFO 296 Fetomaternal immune cross talk [255154572]; KFO 306 Primär Sklerosierende Cholangitis [278045702]. Funding for open access charge: BMBF Integrative Data Semantics in Neurodegeneration [031L0029B, IDSN].

Conflict of interest statement. None declared.

REFERENCES

1. Witwer,K.W. (2015) Circulating microRNA biomarker studies: pitfalls and potential solutions. *Clin. Chem.*, **61**, 56–63.
2. Tycowski,K.T., Guo,Y.E., Lee,N., Moss,W.N., Vallery,T.K., Xie,M. and Steitz,J.A. (2015) Viral noncoding RNAs: more surprises. *Genes Dev.*, **29**, 567–584.
3. Bruscella,P., Bottini,S., Baudesson,C., Pawlotsky,J.-M., Feray,C. and Trabucchi,M. (2017) Viruses and miRNAs: more friends than foes. *Front. Microbiol.*, **8**, 824.
4. Ahmed,W., Zheng,K. and Liu,Z.-F. (2016) Small non-coding RNAs: new insights in modulation of host immune response by intracellular bacterial pathogens. *Front. Immunol.*, **7**, 431.
5. Simon,L.M., Karg,S., Westermann,A.J., Engel,M., Elbehery,A.H.A., Hense,B., Heinig,M., Deng,L. and Theis,F.J. (2018) MetaMap: an atlas of metatranscriptomic reads in human disease-related RNA-seq data. *Gigascience*, **7**, doi:10.1093/gigascience/giy070.
6. Leinonen,R., Sugawara,H., Shumway,M and International Nucleotide Sequence Database Collaboration (2011) The sequence read archive. *Nucleic Acids Res.*, **39**, D19–D21.
7. Barrett,T., Wilhite,S.E., Ledoux,P., Evangelista,C., Kim,I.F., Tomashevsky,M., Marshall,K.A., Phillippy,K.H., Sherman,P.M., Holko,M. *et al.* (2013) NCBI GEO: archive for functional genomics data sets–update. *Nucleic Acids Res.*, **41**, D991–D995.
8. Panwar,B., Omenn,G.S. and Guan,Y. (2017) miRmine: a database of human miRNA expression profiles. *Bioinformatics*, **33**, 1554–1560.
9. Vitsios,D.M., Davis,M.P., van Dongen,S. and Enright,A.J. (2017) Large-scale analysis of microRNA expression, epi-transcriptomic features and biogenesis. *Nucleic Acids Res.*, **45**, 1079–1090.
10. Chung,I.-F., Chang,S.-J., Chen,C.-Y., Liu,S.-H., Li,C.-Y., Chan,C.-H., Shih,C.-C. and Cheng,W.-C. (2017) YM500v3: a

database for small RNA sequencing in human cancer research. *Nucleic Acids Res.*, **45**, D925–D931.
11. Kuksa,P.P., Amlie-Wolf,A., Katanić,Ž., Valladares,O., Wang,L.-S. and Leung,Y.Y. (2018) SPAR: small RNA-seq portal for analysis of sequencing experiments. *Nucleic Acids Res.*, **46**, W36–W42.
12. Sloan,C.A., Chan,E.T., Davidson,J.M., Malladi,V.S., Strattan,J.S., Hitz,B.C., Gabdank,I., Narayanan,A.K., Ho,M., Lee,B.T. *et al.* (2016) ENCODE data at the ENCODE portal. *Nucleic Acids Res.*, **44**, D726–D732.
13. Leung,Y.Y., Kuksa,P.P., Amlie-Wolf,A., Valladares,O., Ungar,L.H., Kannan,S., Gregory,B.D. and Wang,L.-S. (2016) DASHR: database of small human noncoding RNAs. *Nucleic Acids Res.*, **44**, D216–D222.
14. Kuksa,P.P., Amlie-Wolf,A., Katanić,Ž., Valladares,O., Wang,L.-S. and Leung,Y.Y. (2019) DASHR 2.0: integrated database of human small non-coding RNA genes and mature products. *Bioinformatics*, **35**, 1033–1039.
15. Li,Y., Qiu,C., Tu,J., Geng,B., Yang,J., Jiang,T. and Cui,Q. (2014) HMDD v2.0: a database for experimentally supported human microRNA and disease associations. *Nucleic Acids Res.*, **42**, D1070–D1074.
16. Rahman,R.-U., Gautam,A., Bethune,J., Sattar,A., Fiosins,M., Magruder,D.S., Capece,V., Shomroni,O. and Bonn,S. (2018) Oasis 2: improved online analysis of small RNA-seq data. *BMC Bioinformatics*, **19**, 54.
17. Wong,N. and Wang,X. (2015) miRDB: an online resource for microRNA target prediction and functional annotations. *Nucleic Acids Res.*, **43**, D146–D152.
18. Hanisch,D., Fundel,K., Mevissen,H.-T., Zimmer,R. and Fluck,J. (2005) ProMiner: rule-based protein and gene entity recognition. *BMC Bioinformatics*, **6**, S14.
19. Bagewadi,S., Bobić,T., Hofmann-Apitius,M., Fluck,J. and Klinger,R. (2015) Detecting miRNA mentions and relations in biomedical literature [version 3; peer review: 2 approved, 1 approved with reservations]. *F1000Research*, **3**, 205.
20. Madan,S., Hodapp,S., Senger,P., Ansari,S., Szostak,J., Hoeng,J., Peitsch,M. and Fluck,J. (2016) The BEL information extraction workflow (BELIEF): evaluation in the BioCreative V BEL and IAT track. *Database.*, **2016**, baw136.
21. Love,M.I., Huber,W. and Anders,S. (2014) Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biol.*, **15**, 550.
22. Kozomara,A. and Griffiths-Jones,S. (2014) miRBase: annotating high confidence microRNAs using deep sequencing data. *Nucleic Acids Res.*, **42**, D68–D73.
23. Zerbino,D.R., Achuthan,P., Akanni,W., Amode,M.R., Barrell,D., Bhai,J., Billis,K., Cummins,C., Gall,A., Girón,C.G. *et al.* (2018) Ensembl 2018. *Nucleic Acids Res.*, **46**, D754–D761.
24. Sai Lakshmi,S. and Agrawal,S. (2008) piRNABank: a web resource on classified and clustered Piwi-interacting RNAs. *Nucleic Acids Res.*, **36**, D173–D177.
25. Chou,C.-H., Shrestha,S., Yang,C.-D., Chang,N.-W., Lin,Y.-L., Liao,K.-W., Huang,W.-C., Sun,T.-H., Tu,S.-J., Lee,W.-H. *et al.* (2018) miRTarBase update 2018: a resource for experimentally validated microRNA-target interactions. *Nucleic Acids Res.*, **46**, D296–D302.
26. Federhen,S. (2012) The NCBI Taxonomy database. *Nucleic Acids Res.*, **40**, D136–D143.
27. Gremse,M., Chang,A., Schomburg,I., Grote,A., Scheer,M., Ebeling,C. and Schomburg,D. (2011) The BRENDA Tissue Ontology (BTO): the first all-integrating ontology of all organisms for enzyme sources. *Nucleic Acids Res.*, **39**, D507–D513.
28. Schriml,L.M., Mitraka,E., Munro,J., Tauber,B., Schor,M., Nickle,L., Felix,V., Jeng,L., Bearer,C., Lichenstein,R. *et al.* (2019) Human disease ontology 2018 update: classification, content and workflow expansion. *Nucleic Acids Res.*, **47**, D955–D962.
29. Diehl,A.D., Meehan,T.F., Bradford,Y.M., Brush,M.H., Dahdul,W.M., Dougall,D.S., He,Y., Osumi-Sutherland,D., Ruttenberg,A., Sarntivijai,S. *et al.* (2016) The Cell Ontology 2016: enhanced content, modularization, and ontology interoperability. *J. Biomed. Semantics*, **7**, 44.
30. Sarntivijai,S., Lin,Y., Xiang,Z., Meehan,T.F., Diehl,A.D., Vempati,U.D., Schürer,S.C., Pang,C., Malone,J., Parkinson,H. *et al.* (2014) CLO: The cell line ontology. *J. Biomed. Semantics*, **5**, 37.
31. Malone,J., Holloway,E., Adamusiak,T., Kapushesky,M., Zheng,J., Kolesnikov,N., Zhukova,A., Brazma,A. and Parkinson,H. (2010) Modeling sample variables with an Experimental Factor Ontology. *Bioinformatics*, **26**, 1112–1118.
32. Schuurman,N. and Leszczynski,A. (2008) Ontologies for bioinformatics. *Bioinform. Biol. Insights*, **2**, 187–200.
33. Côté,R., Reisinger,F., Martens,L., Barsnes,H., Vizcaino,J.A. and Hermjakob,H. (2010) The ontology lookup service: bigger and better. *Nucleic Acids Res.*, **38**, W155–W160.
34. Ludwig,N., Leidinger,P., Becker,K., Backes,C., Fehlmann,T., Pallasch,C., Rheinheimer,S., Meder,B., Stähler,C., Meese,E. *et al.* (2016) Distribution of miRNA expression across human tissues. *Nucleic Acids Res.*, **44**, 3865–3877.
35. Lee,E.J., Baek,M., Gusev,Y., Brackett,D.J., Nuovo,G.J. and Schmittgen,T.D. (2008) Systematic evaluation of microRNA processing patterns in tissues, cell lines, and tumors. *RNA*, **14**, 35–42.
36. Guo,Z., Maki,M., Ding,R., Yang,Y., Zhang,B. and Xiong,L. (2014) Genome-wide survey of tissue-specific microRNA and transcription factor regulatory networks in 12 tissues. *Sci. Rep.*, **4**, 5150.
37. Ben-Shachar,S., Yanai,H., Sherman Horev,H., Elad,H., Baram,L., Issakov,O., Tulchinsky,H., Pasmanik-Chor,M., Shomron,N. and Dotan,I. (2016) MicroRNAs Expression in the ileal pouch of patients with ulcerative colitis is robustly Up-Regulated and correlates with disease phenotypes. *PLoS One*, **11**, e0159956.
38. Johnson,M.G. and Stout,J.E. (2015) Twenty-eight cases of Mycobacterium marinum infection: retrospective case series and literature review. *Infection*, **43**, 655–662.
39. Ferreira,J., Grochowsky,J., Krakower,D., Zuromskis,P., Baden,R. and Cheifetz,A.S. (2012) Mycobacterium marinum: an increasingly common opportunistic infection in patients on infliximab. *Am. J. Gastroenterol.*, **107**, 1268–1269.
40. McMullen,L., T Leach,S., A Lemberg,D. and S Day,A. (2015) Current roles of specific bacteria in the pathogenesis of inflammatory bowel disease. *AIMS Microbiol.*, **1**, 82–91.
41. Dridi,B., Henry,M., El Khéchine,A., Raoult,D. and Drancourt,M. (2009) High prevalence of Methanobrevibacter smithii and Methanosphaera stadtmanae detected in the human gut using an improved DNA detection protocol. *PLoS One*, **4**, e7063.
42. Bang,C., Weidenbach,K., Gutsmann,T., Heine,H. and Schmitz,R.A. (2014) The intestinal archaea Methanosphaera stadtmanae and Methanobrevibacter smithii activate human dendritic cells. *PLoS One*, **9**, e99411.
43. Blais Lecours,P., Marsolais,D., Cormier,Y., Berberi,M., Haché,C., Bourdages,R. and Duchaine,C. (2014) Increased prevalence of Methanosphaera stadtmanae in inflammatory bowel diseases. *PLoS One*, **9**, e87734.
44. Pouget,J.G. (2018) The Emerging Immunogenetic Architecture of Schizophrenia. *Schizophr. Bull.*, **44**, 993–1004.
45. Bernstein,C.N., Hitchon,C.A., Walld,R., Bolton,J.M., Sareen,J., Walker,J.R., Graff,L.A., Patten,S.B., Singer,A., Lix,L.M. *et al.* (2019) Increased burden of psychiatric disorders in inflammatory bowel disease. *Inflamm. Bowel Dis.*, **25**, 360–368.
46. Hébert,S.S., Wang,W.-X., Zhu,Q. and Nelson,P.T. (2013) A study of small RNAs from cerebral neocortex of pathology-verified Alzheimer's disease, dementia with lewy bodies, hippocampal sclerosis, frontotemporal lobar dementia, and non-demented human controls. *J. Alzheimers. Dis.*, **35**, 335–348.
47. Severini,A., Tyler,S.D., Peters,G.A., Black,D. and Eberle,R. (2013) Genome sequence of a chimpanzee herpesvirus and its relation to other primate alphaherpesviruses. *Arch. Virol.*, **158**, 1825–1828.
48. Vann Jones,S.A. and O'Brien,J.T. (2014) The prevalence and incidence of dementia with Lewy bodies: a systematic review of population and clinical studies. *Psychol. Med.*, **44**, 673–683.
49. Surendranathan,A., Rowe,J.B. and O'Brien,J.T. (2015) Neuroinflammation in Lewy body dementia. *Parkinsonism Relat. Disord.*, **21**, 1398–1406.
50. Harris,S.A. and Harris,E.A. (2015) Herpes simplex virus type 1 and other pathogens are key causative factors in sporadic alzheimer's disease. *J. Alzheimers. Dis.*, **48**, 319–353.
51. Sochocka,M., Zwolińska,K. and Leszek,J. (2017) The infectious etiology of alzheimer's disease. *Curr. Neuropharmacol.*, **15**, 996–1009.
52. Yang,C.P., Zhang,Z.H., Zhang,L.H. and Rui,H.C. (2016) Neuroprotective role of MicroRNA-22 in a

6-Hydroxydopamine-Induced cell model of parkinson's disease via regulation of its target gene TRPM7. *J. Mol. Neurosci.*, **60**, 445–452.
53. Margis,R., Margis,R. and Rieder,C.R.M. (2011) Identification of blood microRNAs associated to Parkinson's disease. *J. Biotechnol.*, **152**, 96–101.
54. Leggio,L., Vivarelli,S., L'Episcopo,F., Tirolo,C., Caniglia,S., Testa,N., Marchetti,B. and Iraci,N. (2017) microRNAs in Parkinson's disease: From pathogenesis to novel diagnostic and therapeutic approaches. *Int. J. Mol. Sci.*, **18**, E2698.
55. Heman-Ackah,S.M., Hallegger,M., Rao,M.S. and Wood,M.J.A. (2013) RISC in PD: the impact of microRNAs in Parkinson's disease cellular and molecular pathogenesis. *Front. Mol. Neurosci.*, **6**, 40.
56. Wang,J., Vasaikar,S., Shi,Z., Greer,M. and Zhang,B. (2017) WebGestalt 2017: a more comprehensive, powerful, flexible and interactive gene set enrichment analysis toolkit. *Nucleic Acids Res.*, **45**, W130–W137.
57. Pissadaki,E.K. and Bolam,J.P. (2013) The energy cost of action potential propagation in dopamine neurons: clues to susceptibility in Parkinson's disease. *Front. Comput. Neurosci.*, **7**, 13.
58. Bolam,J.P. and Pissadaki,E.K. (2012) Living on the edge with too many mouths to feed: why dopamine neurons die. *Mov. Disord.*, **27**, 1478–1483.
59. Surmeier,D.J., Obeso,J.A. and Halliday,G.M. (2017) Selective neuronal vulnerability in Parkinson disease. *Nat. Rev. Neurosci.*, **18**, 101–113.
60. Saba,R., Goodman,C.D., Huzarewich,R.L.C.H., Robertson,C. and Booth,S.A. (2008) A miRNA signature of prion induced neurodegeneration. *PLoS One*, **3**, e3652.
61. Montag,J., Hitt,R., Opitz,L., Schulz-Schaeffer,W.J., Hunsmann,G. and Motzkus,D. (2009) Upregulation of miRNA hsa-miR-342-3p in experimental and idiopathic prion disease. *Mol. Neurodegener.*, **4**, 36.
62. Alves da Costa,C. and Checler,F. (2011) Apoptosis in Parkinson's disease: is p53 the missing link between genetic and sporadic Parkinsonism? *Cell. Signal.*, **23**, 963–968.
63. Ogino,M., Ichimura,M., Nakano,N., Minami,A., Kitagishi,Y. and Matsuda,S. (2016) Roles of PTEN with DNA repair in parkinson's disease. *Int. J. Mol. Sci.*, **17**, E954.
64. Hegarty,S.V., Sullivan,A.M. and O'Keeffe,G.W. (2018) Inhibition of *miR-181a* promotes midbrain neuronal growth through a Smad1/5-dependent mechanism: implications for Parkinson's disease. *Neuronal. Signal.*, **2**, NS20170181.
65. Södersten,E., Feyder,M., Lerdrup,M., Gomes,A.-L., Kryh,H., Spigolon,G., Caboche,J., Fisone,G. and Hansen,K. (2014) Dopamine signaling leads to loss of Polycomb repression and aberrant gene activation in experimental parkinsonism. *PLoS Genet.*, **10**, e1004574.
66. van der Heide,L.P. and Smidt,M.P. (2013) The BCL2 code to dopaminergic development and Parkinson's disease. *Trends Mol. Med.*, **19**, 211–216.
67. Zhang,B., Kirov,S. and Snoddy,J. (2005) WebGestalt: an integrated system for exploring gene sets in various biological contexts. *Nucleic Acids Res.*, **33**, W741–W748.
68. Fiosina,J., Fiosins,M. and Bonn,S. (2019) *Bioinformatics Research and Applications:Deep Learning and Random Forest-Based Augmentation of sRNA Expression Profiles*. Chen,Z-Z, Ueta,S, Li,J and Wang,L (eds). Springer International Publishing, 159–170.

3.5 scGANs, 2018

nature COMMUNICATIONS

ARTICLE

https://doi.org/10.1038/s41467-019-14018-z OPEN

Realistic in silico generation and augmentation of single-cell RNA-seq data using generative adversarial networks

Mohamed Marouf[1,5], Pierre Machart[1,5], Vikas Bansal[1], Christoph Kilian[1,2], Daniel S. Magruder[1,3], Christian F. Krebs[2] & Stefan Bonn[1,4*]

A fundamental problem in biomedical research is the low number of observations available, mostly due to a lack of available biosamples, prohibitive costs, or ethical reasons. Augmenting few real observations with generated in silico samples could lead to more robust analysis results and a higher reproducibility rate. Here, we propose the use of conditional single-cell generative adversarial neural networks (cscGAN) for the realistic generation of single-cell RNA-seq data. cscGAN learns non-linear gene–gene dependencies from complex, multiple cell type samples and uses this information to generate realistic cells of defined types. Augmenting sparse cell populations with cscGAN generated cells improves downstream analyses such as the detection of marker genes, the robustness and reliability of classifiers, the assessment of novel analysis algorithms, and might reduce the number of animal experiments and costs in consequence. cscGAN outperforms existing methods for single-cell RNA-seq data generation in quality and hold great promise for the realistic generation and augmentation of other biomedical data types.

[1] Institute of Medical Systems Biology, University Medical Center Hamburg-Eppendorf, Hamburg, Germany. [2] Center for Internal Medicine, III. Medical Clinic and Polyclinic, University Medical Center Hamburg-Eppendorf, Hamburg, Germany. [3] Genevention GmbH, Goettingen, Germany. [4] German Center for Neurodegenerative Diseases, Tuebingen, Germany. [5] These authors contributed equally: Mohamed Marouf, Pierre Machart. *email: sbonn@uke.de

Biological systems are usually highly complex, as intracellular and intercellular communication, for example, are orchestrated via the non-linear interplay of tens to hundreds of thousands of different molecules[1]. Recent technical advances have enabled scientists to scrutinize these complex interactions, measuring the expression of thousands of genes at the same time, for instance[2]. Unfortunately, this complexity often becomes a major hurdle as the number of observations can be relatively small, due to economical or ethical considerations or simply because the number of available patient samples is low. Next to technically induced measurement biases, this problem of too few observations, in the face of many parameters, might be one of the most prominent bottlenecks in biomedical research[1]. Thus, a small sample size might not reflect the population well, an imbalance that can decrease the reproducibility of experimental results[3].

While the number of biological samples might be limited, realistic in silico generation of observations could accommodate for this unfavorable situation. In practice, in silico generation has seen success in computer vision when used for data augmentation, whereby in silico-generated samples are used alongside the original ones to artificially increase the number of observations[4]. In this manuscript, we focus on augmenting real with newly generated samples, in their original high-dimensional gene space, and whose distribution mimics the original data distribution. While classically, data modeling relies on a thorough understanding of the priors on invariants underlying the production of such data, current methods of choice for photorealistic image generation rely on deep learning-based generative adversarial networks (GANs)[5–8] and variational autoencoders (VAEs)[9,10].

GANs involve a generator that outputs realistic in silico-generated samples. This is achieved with a neural network that learns to transform a simple, low-dimensional distribution into a high-dimensional distribution that is virtually indistinguishable from the real training distribution (Supplementary Fig. 1).

While data augmentation has been a recent success story in various fields of computer science, the development and usage of GANs and VAEs for omics data augmentation has yet to be investigated. As a proof of concept that realistic in silico generation could potentially be applied to biomedical omics data, we focus on the generation of single-cell RNA (scRNA) sequencing data using GANs. scRNA sequencing has made it possible to evaluate genome-wide gene expression of thousands to millions of cells in a single experiment[11]. This detailed information across genes and cells opens the door to a much deeper understanding of cell type heterogeneity in a tissue, cell differentiation, and cell type-specific disease etiology.

In this manuscript, we establish how a single-cell GAN (scGAN) can be leveraged to generate realistic scRNA-seq data. We further demonstrate that our scGAN can use conditioning (cscGAN) to produce specific cell types or subpopulations, on-demand. Finally, we show how our models can successfully augment sparse cell populations to improve the quality and robustness of downstream classification. To the best of our knowledge, this constitutes the first attempt to apply these groundbreaking methods for the augmentation of sequencing data.

Results

Realistic generation of scRNA-seq data using an scGAN. Given the great success of GANs in producing photorealistic images, we hypothesize that similar approaches could be used to generate realistic scRNA-seq data (i.e. matrices where each row corresponds to a cell and each column to the expression level of a gene). In this work "realistic" is referring to the generation of data that mimics the distribution of the real data, in their original space, without merely replicating them. To distinguish experimental scRNA-seq data from data produced by GANs we will use the terms "real" and "generated" cells, respectively.

To build and evaluate different GAN models for scRNA-seq data generation we used a peripheral blood mononuclear cell (PBMC) scRNA-seq dataset with 68,579 cells[12] (Supplementary Table 1, Methods). The PBMC dataset contains many distinct immune cell types, which yield clear clusters that can be assigned their cell type identity with marker genes (genes specifically expressed in a cluster). The aforementioned features of the PBMC dataset make it ideal for the evaluation of our scGAN performance.

Since it is notoriously difficult to evaluate the quality of generative models[13,14] we used four evaluation criteria inspired by single-cell data analysis: t-SNE, marker gene correlation, maximum mean discrepancy (MMD), and classification performance (see Methods for evaluation details). These metrics are used as quantitative and qualitative measures to assess the synthesized cells. Based on these criteria, the best performing single-cell GAN (scGAN) model was a GAN minimizing the Wasserstein distance[15], relying on two fully connected neural networks with batch normalization (Supplementary Fig. 1). We found that the quality of the generated cells greatly improved when the training cells were scaled to exhibit a constant total count of 20,000 reads per cell. In addition to this preprocessing step, we added a custom library-size normalization (LSN) function to our scGAN's generator so that it explicitly outputs generated cells with a total read count equal to that of the training data (20,000 reads per cell) (Supplementary Fig. 1). Our LSN function greatly improved training speed and stability and gave rise to the best performing models based on the aforementioned metrics. Further details of the model selection and (hyper)-parameter optimization can be found in the Methods section.

For a qualitative assessment of the results, we used t-SNE[16,17] to obtain a two-dimensional representation of generated and real cells from the test set (Fig. 1a–c, Supplementary Fig. 2). The scGAN generates cells that represent every cluster of the data it was trained on and the expression patterns of marker genes are accurately learned by scGAN (Supplementary Fig. 3).

Furthermore, the scGAN is able to model intergene dependencies and correlations, which are a hallmark of biological gene-regulatory networks[18]. To prove this point we computed the correlation and distribution of the counts of cluster-specific marker genes (Fig. 1d) and 100 highly variable genes between generated and real cells (Supplementary Fig. 4). We then used SCENIC[19] to understand if scGAN learns regulons, the functional units of gene-regulatory networks consisting of a transcription factor (TF) and its downstream regulated genes. scGAN trained on all cell clusters of the Zeisel dataset[20] (see Methods) faithfully represent regulons of real test cells, as exemplified for the Dlx1 regulon in Supplementary Fig. 4G–J, suggesting that the scGAN learns dependencies between genes beyond pairwise correlations.

To show that the scGAN generates realistic cells, we trained a Random Forest (RF) classifier[21] to distinguish between real and generated data. The hypothesis is that a classifier should have a (close to) chance-level performance when the generated and real data are highly similar. Indeed the RF classifier only reaches 0.65 area under the curve (AUC) when discriminating between the real cells and the scGAN-generated data (blue curve in Fig. 1e) and 0.52 AUC when tasked to distinguish real from real data (positive control).

Finally, we compared the results of our scGAN model to two state-of-the-art scRNA-seq simulations tools, Splatter[22] and SUGAR[23] (see Methods for details). While Splatter models some marginal distribution of the read counts well (Supplementary Fig. 5), it struggles to learn the joint distribution of these counts,

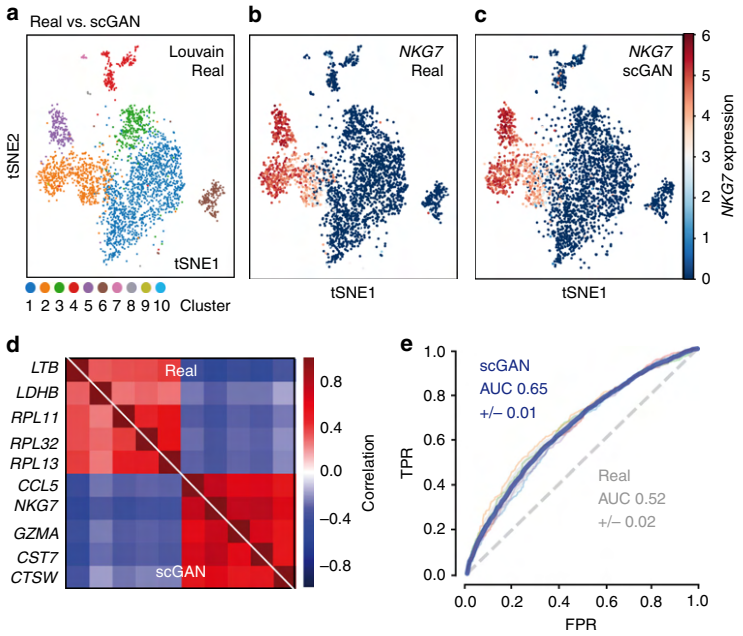


Fig. 1 Evaluation of the scGAN-generated PBMC cells. a–c t-SNE visualization of the Louvain-clustered real cells (**a**) and the *NKG7* gene expression in real (**b**) and scGAN-generated (**c**) cells. **d** Pearson correlation of marker genes for the scGAN-generated (bottom left) and the real (upper right) data. **e** Cross-validation ROC curve (true positive rate against false positive rate) of an RF classifying real and generated cells (scGAN in blue, chance-level in gray).

as observed in t-SNE visualizations with one homogeneous cluster instead of the different subpopulations of cells of the real data, a lack of cluster-specific gene dependencies, and a high MMD score (129.52) (Supplementary Table 2, Supplementary Fig. 4). SUGAR, on the other hand, generates cells that overlap with every cluster of the data it was trained on in t-SNE visualizations and accurately reflects cluster-specific gene dependencies (Supplementary Fig. 6). SUGAR's MMD (59.45) and AUC (0.98), however, are significantly higher than the MMD (0.87) and AUC (0.65) of the scGAN and the MMD (0.03) and AUC (0.52) of the real data (Supplementary Table 2, Supplementary Fig. 6). It is worth noting that SUGAR can be used, like here, to generate cells that reflect the original distribution of the data. It was, however, originally designed and optimized to specifically sample cells belonging to regions of the original dataset that have a low density, which is a different task than what is covered by this manuscript. While SUGAR's performance might improve with the adaptive noise covariance estimation, the runtime and memory consumption for this estimation proved to be prohibitive (see Supplementary Fig. 6F–I and Methods).

The results from the t-SNE visualization, marker gene correlation, MMD, and classification corroborate that the scGAN generates realistic data from complex distributions, outperforming existing methods for in silico scRNA-seq data generation. The realistic modeling of scRNA-seq data entails that our scGAN does not denoise nor impute gene expression information, while they potentially could[24]. Nevertheless, an scGAN that has been trained on imputed data using MAGIC[25] generates realistic imputed scRNA-seq data (Supplementary Fig. 7). Of note, the fidelity with which the scGAN models scRNA-seq data seems to be stable across several tested dimensionality reduction algorithms (Supplementary Fig. 8).

Realistic modeling across tissues, organisms, and data size. We next wanted to assess how faithful the scGAN learns very large, more complex data of different tissues and organisms. We therefore trained the scGAN on the currently largest published scRNA-seq dataset consisting of 1.3 million mouse brain cells and measured both the time and performance of the model with respect to the number of cells used (Supplementary Table 1, Supplementary Fig. 9). Qualitative assessment using t-SNE visualization shows that the scGAN generates cells that represent every cluster of the data it was trained on. The expression patterns of marker genes are accurately learned (Supplementary Fig. 9).

The actual time required to train an scGAN depends on the data size and complexity and on the computer architecture used, necessitating at least one high-performance GPU card. However, it should be noted that scGAN uses batch training so that its memory consumption does not depend on the number of cells and its runtime scales linearly, at worst, with it.

Our results demonstrate that the scGAN performs consistently well on scRNA-seq datasets from different organisms, tissues, and with varying complexity and size, learning realistic representations of millions of cells.

Conditional generation of specific cell types. scRNA-seq in silico data generation reaches its full potential when specific cells of interest could be generated on demand, which is not directly possible with the scGAN model. This conditional generation of cell types could be used to increase the number of a sparse, specific population of cells that might represent only a small fraction of the total cells sequenced.

While specific cell types of interest can be obtained by scGAN cell generation followed by clustering and cell selection, we developed and evaluated various conditional scGAN (cscGAN)

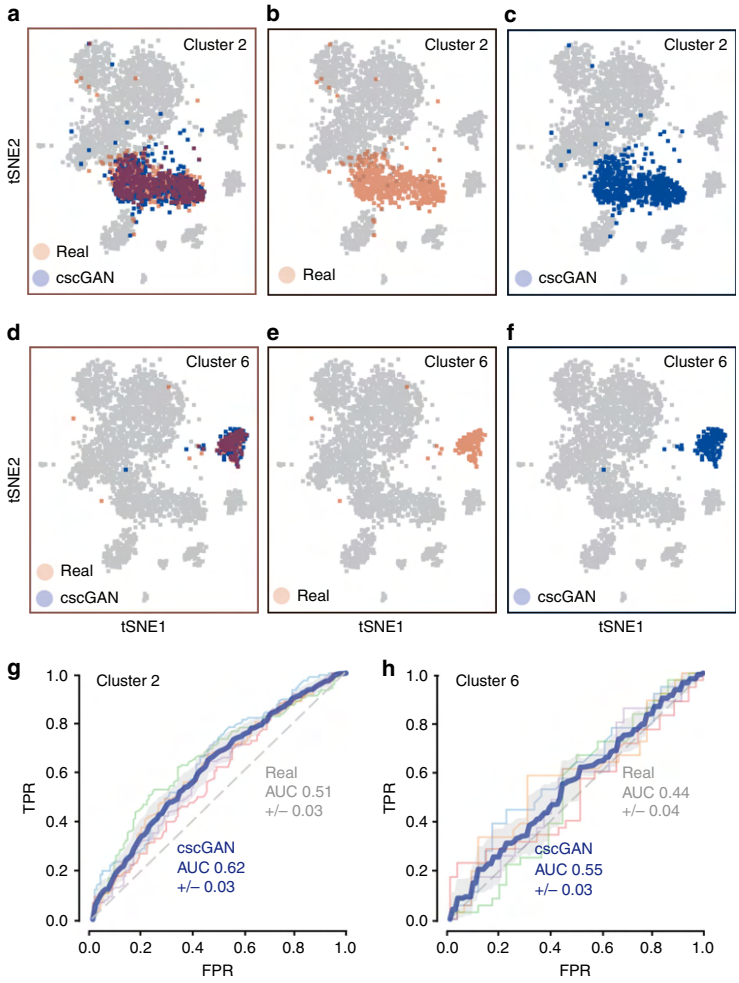


Fig. 2 Evaluation of the conditional generation of PBMC cells. a–c t-SNE visualization of cluster 2 real cells (red, panels **a**, **b**), cluster 2 generated cells (blue, panels **a**, **c**), and other real cells (gray, all panels). **d–f** Same as **a–c** for cluster 6 cells. **g** Cross-validation ROC curve of an RF classifying cluster 2 real from cscGAN-generated cells (cscGAN in blue, chance-level in gray). **h** Same as **g** for cluster 6 cells.

architectures that can directly generate cell types of interest. Common to all these models is that the cscGAN learns to generate cells of specific types while being trained on the complete multiple cell type dataset. The cell type information is then associated to the genes' expression values of each cell during the training. These tags can then be used to generate scRNA-seq data of a specific type, in our case of a specific cluster of PBMCs. The best performing cscGAN model utilized a projection discriminator[26], along with Conditional Batch Normalization[27] and an LSN function in the generator. Again, model architecture selection and optimization details can be found in the Methods.

Model performance was assessed on the PBMC dataset using t-SNE, marker gene correlation, and classification. The cscGAN learns the complete distribution of clusters of the PBMC data (Fig. 2) and can conditionally represent each of the ten clusters on demand. The t-SNE results for the conditional generation of cluster 2 and cluster 6 cells are shown in Fig. 2a–c and Fig. 2d–f, respectively. Figure 2a–c highlights the real (red, panels a and b) and generated (blue, panels a and c) cells for cluster 2, while the real cells of all other clusters are shown in gray. The cscGAN generates cells that are overlapping with the real cluster of interest in the t-SNE visualizations. In addition, the cscGAN also accurately captures inter- and intra-cluster gene–gene dependencies as visualized in the marker gene correlation plots in Supplementary Fig. 10. The assumption that the cscGAN generates conditional cells that are very similar to the real cells of the cluster of interest is substantiated in the final classification task. An RF classifier reaches an AUC between 0.62 (cluster 2, Fig. 2g) and 0.55 (cluster 6, Fig. 2h) when trying to distinguish cluster-specific cscGAN-generated cells from real cells, a value that is reasonably close to the perfect situation of random classification (AUC of 0.5) (Supplementary Table 4). The MMD distances between cscGAN generated and real cells is 0.286 (cluster 2, Fig. 2g) and 0.238 (cluster 6, Fig. 2h) while distances between real and real cells (positive control) were 0.037 and 0.129, respectively (Supplementary Table 3).

It is interesting to observe that the cscGAN and scGAN generate cells of very similar quality, as an RF classifier reaches an AUC of 0.61 (MMD of 0.674) to distinguish between

cscGAN-generated and real cells and an AUC of 0.65 (MMD of 0.547) for scGAN-generated and real cells (differences not significant).

The results of this section demonstrate that the cscGAN can generate high-quality scRNA-seq data for specific clusters or cell types of interest, while rivaling the overall representational power of the scGAN. Importantly, the fidelity with which the cscGAN models scRNA-seq data seems to be independent of the tested Louvain and K-means clustering algorithms (Supplementary Fig. 11, Supplementary Table 5).

Improved classification of sparse cells using augmented data. We now investigate how we can use the conditional generation of cells to improve the quality and robustness of downstream classification of rare cell populations. The underlying hypotheses are two-fold. (i) A few cells of a specific cluster might not represent the cell population of that cluster well, potentially degrading the quality and robustness of downstream classification. (ii) This degradation might be mitigated by augmenting the rare population with cells generated by the cscGAN. The base assumption is that the cscGAN might be able to learn good representations for small clusters by using gene expression and correlation information from the whole dataset.

To test the two parts of our hypothesis, we first artificially reduce the number of cells of the PBMC cluster 2 (downsampling) and observe how it affects the ability of an RF model to accurately distinguish cells from cluster 2 from cells of other clusters. In addition, we train the cscGANs on the same downsampled datasets, generate cells from cluster 2 to augment the downsampled population, retrain an RF with this augmented dataset, and measure the gain in their ability to correctly classify the different populations.

More specifically, cluster 2 comprises 15,008 cells and constitutes the second largest population in the PBMC dataset. Such a large number of cells makes it possible to obtain statistically sound classification results. By deliberately holding out large portions of this population, we can basically quantify how the results would be affected if that population was arbitrarily small. We produce eight alternate versions of the PBMC dataset, obtained by downsampling the cluster 2 population (keeping 50%, 25%, 10%, 5%, 3%, 2%, 1%, and 0.5% of the initial population) (Supplementary Fig. 12, Supplementary Table 6). We then proceed to train RF classifiers (for each of those eight downsampled datasets) (Supplementary Fig. 13A), on 70% of the total amount of cells and kept aside 30% to test the performance of the classifier (Supplementary Fig. 13B). The red line in Fig. 3a and Supplementary Fig. 14 very clearly illustrates how the performance of the RF classifier, measured through the F1 score, gradually decreases from 0.95 to 0.45 while the downsampling rate goes from 50% to 0.5%. To see if we could mitigate this deterioration, we tested two ways of augmenting our alternate datasets. First, we used a naïve method, which we call upsampling, where we simply enlarged the cluster 2 population by duplicating the cells that were left after the downsampling procedure (Supplementary Fig. 13A). The orange line in Fig. 3a shows that this naive strategy actually mitigates the effect of the downsampling, albeit only to a minor extent (F1 score of 0.6 obtained for a downsampling rate of 0.5%). It is important to note that adding noise (e.g. standard Gaussian) to the upsampled cluster 2 cells usually deteriorated the classification performance (data not shown).

In order to understand whether in silico-generated cluster 2 cells could improve the RF performance, we next trained the cscGANs on the eight downsampled datasets (Supplementary Fig. 13C). We then proceeded to augment the cluster 2 population with the cells generated by the cscGAN (Supplementary Fig. 13A). Figure 3c shows that using as little as 2% (301 cells) of the real cluster 2 data for training the cscGAN suffices to generate cells that overlap with real test cells. When less cells are used the t-SNE overlap of cluster 2 training cells and generated cells slightly decreases (Fig. 3d, Supplementary Fig. 15). These results strongly suggest that the cluster-specific expression and gene dependencies are learned by the cscGAN, even when very few cells are available. In line with this assumption, the blue curves in Fig. 3a and Supplementary Fig. 14 show that augmenting the cluster 2 population with cluster 2 cells generated by the cscGAN almost completely mitigates the effect of the downsampling (F1 score of 0.93 obtained for a downsampling rate of 0.5%). We obtained similar results with RFs that have been optimized for the number of trees and features per tree (Supplementary Fig. 14D), showing that augmentation robustly increases classification performance across RF hyper-parameter space. Interestingly, the RF improves with increasing numbers of generated cells used for the classifiers' training (Supplementary Fig. 16).

Two conclusions can be obtained from these results. First the obvious, few cluster-specific cells do not represent the population well. Second, the usage of cscGAN-generated scRNA-seq data can mitigate this effect and increases the performance of downstream applications like classification when limited samples of a specific cluster are available.

Improved trajectory analysis using augmented data. The previous results highlight the ability of the cscGAN to specifically generate cells corresponding to different types or clusters. Such discrete states, however, are not sufficient to capture intermediate and transitional cellular states of an organism. Erythrocytes, for example, are derived in the red bone marrow from pluripotent stem cells that give rise to all types of blood cells. This differentiation process contains transitional cellular states that can be visualized (Supplementary Fig. 17A–C) using a pseudo-time analysis of bone marrow scRNA-seq data[28] (see Supplementary Table 1, Methods). The outcome of pseudo-time analyses, however, depends heavily on how well the variety of continuous states of erythrocytes is represented in the data. To highlight this property, we manually downsampled a subpopulation of erythrocytes in the bone marrow dataset. We can observe in Supplementary Fig. 17D–F that such downsampling directly affects the structure of the graph inferred by the pseudo-time analysis.

To show that the scGAN can reliably model populations that exist in continuous cellular states, we trained it on the downsampled bone marrow dataset. We then replaced the cells that were re-moved from the original data with handpicked scGAN-generated cells that belonged to the same subpopulation of erythrocytes. Adding the cells generated by scGAN allows to restore the original structure of the graph (Supplementary Fig. 17G–I). These results suggest that scGANs are able to model discrete and continuous cellular states and cell trajectories.

cscGAN learns and translates gene-regulatory syntax. The fidelity with which the (c)scGAN creates cells of very sparse populations is striking and it is tempting to speculate if the model actually learns and translates gene-regulatory information from abundant cell clusters to sparse ones.

We trained scGANs on decreasing amounts of cluster 2 cells (keeping 50%, 25%, 10%, 5%, 3%, 2%, 1%, 0.5%, 0.2%, and 0.1% of the initial population) and compared scGAN-generated cluster 2 cells to real test cluster 2 cells. In addition, we trained cscGANs on the same number of cluster 2 cells and all other clusters (see also previous section). We then compared scGAN (trained only

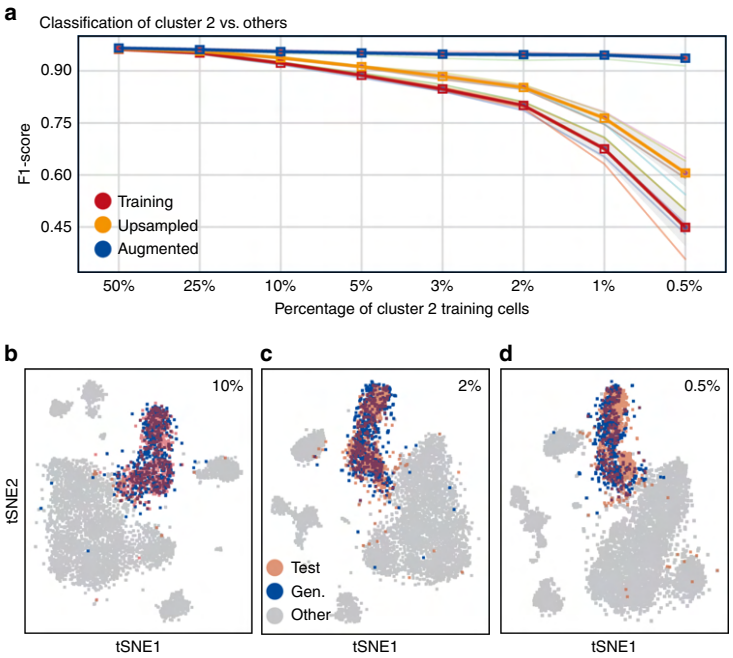


Fig. 3 Effects of data downsampling and augmentation on classification and clustering. a F1 score of an RF classifier trained to discriminate cluster 2 test from other test cells when trained on training (red), upsampled (orange), or augmented (blue) cells for eight levels of downsampling (50% to 0.5%). **b**–**d** t-SNE representation of cluster 2 real test (red) and cscGAN-generated (blue) cells for three levels of downsampling (10% panel **b**, 2% panel **c**, and 0.5% panel **d**). Other test cells are shown in gray.

on cluster 2) and cscGAN (trained on all clusters) generated cluster 2 cells to real test cluster 2 cells using RF classification and t-SNE visualization. The underlying hypothesis is that if the cscGAN can learn and translate general rules of gene regulation from abundant to sparse cell populations, it should provide more realistic cells for sparse clusters than a scGAN that was only trained on the latter.

We first assessed model fidelity by the ability of an RF classifier to distinguish scGAN and cscGAN-generated cluster 2 cells from real test cluster 2 cells. While the scGAN trained on a large number of cluster 2 cells generates more realistic cluster 2 cells than a cscGAN trained on all cell clusters, the cscGAN generates realistic cells of much wider variety than the scGAN when only few cluster 2 training cells are available (Supplementary Fig. 18). While the cscGAN seems to leverage gene-regulatory information from the more abundant clusters to compensate for the missing cluster 2 observations, the scGAN seems to re-create the few cluster 2 cells it has learned from, failing to generalize to unseen cluster 2 test cells (Supplementary Fig. 18C, D).

These results suggest that (c)scGAN can learn fundamental gene-regulatory rules that are valid across the observed cells (clusters and types). The (c)scGAN seems to learn those rules from the cells of large clusters and might apply them when generating cells of very small cell clusters.

Discussion

This work shows how cscGAN can be used to generate realistic scRNA-seq representations of complex scRNA-seq data with multiple distinct cell types and millions of cells. cscGAN outperforms current methods in the realistic generation of scRNA-seq data and scales sublinearly in the number of cells. Most importantly, we provide compelling evidence that generating in silico scRNA-seq data improves downstream applications, especially when sparse and underrepresented cell populations are augmented by the cscGAN-generated cells. We specifically show how the classification of cell types can be improved when the available data are augmented with in silico-generated cells, leading to classifiers that rival the predictive power of those trained on real data of similar size.

It may be surprising or even suspicious that our cscGAN is able to learn to generate cells coming from very small subpopulations (e.g. 16 cells) so well. We speculate that although cells from a specific type may have very specific functions, or exhibit highly singular patterns in the expression of several marker genes, they also share a lot of similarities with the cells from other types, especially with those that share common precursors. In other words, the cscGAN is not only learning the expression patterns of a specific subpopulation from the (potentially very few) cells of that population, but also from the (potentially very numerous) cells from other populations. This hypothesis actually aligns with the architecture of the cscGAN. In the generator, the only parameters that are cluster specific are those learned in the Conditional Batch Normalization layers (BLN). On the other hand, all the parameters of each of the Fully Connected (FC) layers are shared across all the different cell types.

While focusing on the task of cell type classification in this manuscript, many other applications will most probably gain from data augmentation, including—but not limited to—clustering itself, cell type detection, and data denoising. Indeed, a recent manuscript used Wasserstein GANs (WGAN) to denoise scRNA-seq data[24]. For this purpose, the (low-dimensional) representation obtained at the output of the single hidden layer of a critic network was used. These lower-dimensional representations keep cell type-determining factors while they discard noisy

information such as batch effects. In general, GAN models allow for the simulation of cell development or differentiation through simple arithmetic operations applied in the latent space representation of the GAN, operations for which our conditional cscGAN is especially suited.

Throughout this manuscript, we solely focused on using cell types as a side information to condition the generation on. It is worth mentioning that any other kind of side information (partitioning of the sample) could equally be used. For instance, a cscGAN could be conditioned and trained on a combination of case and control samples. While many other choices could lead to interesting applications, we leave this avenue of research for future work.

It is tempting to speculate how well the scRNA-seq data generation using cscGAN can be applied to other biomedical domains and data types. It is easy to envision, for example, how cscGAN variants could generate realistic (small) RNA-seq or proteomic data. Moreover, cscGAN variants might successfully generate whole genomes with predefined features such as disease state, ethnicity, and sex, building virtual patient cohorts for rare diseases, for example. In biomedical imaging, in silico image generation could improve object detection, disease classification, and prognosis, leading to increased robustness and better generalization of the experimental results, extending clinical application.

We hypothesize that data augmentation might be especially useful when dealing with human data, which is notoriously heterogeneous due to genetic and environmental variation. Data generation and augmentation might be most valuable when working with rare diseases or when samples with a specified ethnicity or sex, for example, are simply lacking.

Lastly we would like to emphasize that the generation of realistic in silico data has far reaching implications beyond enhancing downstream applications. In silico data generation can decrease human and animal experimentation with a concomitant reduction in experimental costs, addressing important ethical and financial questions.

Methods

Datasets and preprocessing. *PBMC*: We trained and evaluated all models using a published human dataset of 68,579 PBMCs (healthy donor A)[12]. The dataset was chosen as it contains several clearly defined cell populations and is of reasonable size. In other words, it is a relatively large and complex scRNA-seq dataset with very good annotation, ideal for the learning and evaluation of generative models.

The cells were sequenced on Illumina NextSeq 500 High Output with ~20,000 reads per cell. The cell barcodes were filtered as in ref. [12] and the filtered gene matrix is publicly available on the 10x Genomics website.

In all our experiments, we removed genes that are expressed in less than three cells in the gene matrix, yielding 17,789 genes. We also discarded cells that have less than 10 genes expressed. This, however, did not change the total number of cells. Finally, the cells were normalized for the library size by first dividing UMI counts by the total UMI counts in each cell and then multiplied by 20,000. See Supplementary Table 1 for an outlook of this dataset.

Brain Large: In addition to the PBMC dataset we trained and evaluated our best performing scGAN model on the currently largest available scRNA-seq dataset of ~1.3 million mouse brain cells (10x Genomics). The dataset was chosen to prove that the model performance scales to millions of scRNA-seq cells, even when the organism, tissue, and the sample complexity varies. The sequenced cells are from the cortex, hippocampus, and the subventricular zone of two E18 mice.

The barcodes filtered matrix of gene by cell expression values is available on the 10x Genomics website. After removing genes that are expressed in less than three cells, we obtained a gene matrix of 22,788 genes. We also discarded cells that have less than 10 genes expressed, which did not affect the overall number of cells. The cells were normalized for the library size as described in the PBMC section.

Brain Small: We also examined the performance of the generative models proposed in this manuscript on a subset of the Brain Large dataset provided by 10x Genomics, which consists of 20,000 cells. The preprocessing of the Brain Small dataset was identical to that of the Brain Large dataset, yielding a matrix of 17,970 genes by 20,000 cells (Supplementary Table 1).

Bone Marrow: In order to understand the ability of the scGAN to learn the distribution from imputed cells we used a mouse bone marrow cell dataset (GSE72857)[29]. The cells were collected using a plate-based MARS-seq protocol in order to identify myeloid progenitor subpopulations. The sparsity and the heterogeneity of cells in this dataset makes it suitable for imputation. Data preprocessing was performed as described above, yielding a matrix of 12,443 genes by 2,730 cells (Supplementary Table 1). Processed bone marrow cells were either used directly for scGAN modeling or after imputation using MAGIC (see section Expression imputation with MAGIC).

Zeisel: Finally, we trained scGANs on somatosensory cortex (S1) and hippocampal CA1 cells (GSE60361)[20], which consists of 3,005 high-quality single cells (including neurons, glia, and endothelial cells). After preprocessing we obtained a matrix of 18,738 genes by 3,005 cells (Supplementary Table 1).

Clustering: Throughout this manuscript we use the Cell Ranger workflow for the scRNA-seq secondary analysis[12]. First, the cells were normalized by UMI counts. Then, we took the natural logarithm of the UMI counts. Afterwards, each gene was normalized such that the mean expression value for each gene is 0, and the standard deviation is 1. The top 1000 highly variable genes were selected based on their ranked normalized dispersion. PCA was applied on the selected 1000 genes. In order to identify cell clusters, we used Louvain clustering[30] on the first 50 principal components of the PCA. This replaced the k-means clustering used in Cell Ranger R analysis workflow, as the Scanpy[31] tutorial on clustering the PBMC dataset advises. The number of clusters were controlled by the resolution parameter of scanpy.api.tl.louvain. The higher resolution made it possible to find more and smaller clusters.

For the PBMC and the Brain Large dataset we used a resolution of 0.15 which produced 10 and 13 clusters, respectively. The Brain Small dataset was clustered using a resolution of 0.1 which gives 8 clusters.

To understand if the selection of different clustering algorithms might affect the fidelity with which the cscGAN models scRNA-seq data, we compared the results obtained with Louvain clustering compare to that of K-means clustering on the PBMC dataset. We used the scikit-learn package[32] to apply the clustering on the first 50 principal components extracted as mentioned above. The K-means scikit-learn function default parameters were used for the clustering except for the number of centroids to generate in the data, which was set to 10 (the number of clusters previously obtained with the Louvain algorithm). This produces 10 clusters (Supplementary Fig. 11). The results obtained are very similar to those with the Louvain clustering in terms of the ability of an RF classifier to discriminate between real cells and cscGAN-generated cells (Supplementary Table 5).

Definition of marker genes: In several experiments we investigated the expression levels and correlation of genes. For this purpose, a group of 10 marker genes was defined by taking the five most highly upregulated genes for the largest two clusters in the dataset (clusters 1 and 2 for the PBMC dataset). Significant upregulation was estimated using the logarithm of the Louvain-clustered cells with the scanpy.api.tl.rank_genes_groups function with its default parameters (Scanpy 1.2.2)[31].

Model description. *scGAN*: In this section, we outline the model used for the scGAN by defining the loss function it optimizes, the optimization process, and key elements of the model architecture. GANs typically involve two Artificial Neural Networks: a generator, which, given some input random noise, trains to output realistic samples, and a critic that trains to spot the differences between real cells and the ones that the generator produces (Supplementary Fig. 1). An adversarial training procedure allows for those entities to compete against each other in a mutually beneficial way. Formally, GANs minimize a divergence between the distributions of the real samples and of the generated ones. Different divergences are used giving rise to different GAN variants. While original GANs[5] minimize the so-called Jensen–Shannon divergence, they suffer from known pitfalls making their optimization notoriously difficult to achieve[33]. For instance, they are known to be prone to mode collapse, where the generated samples are realistic albeit only representing a fraction of the variety of the samples it was trained on (i.e. only a few but not all modes of the distribution of the real samples is learned). On the other hand, WGANs[15,34] use a Wasserstein distance, with compelling theoretical and empirical arguments. In our hands, WGANs showed no evidence of mode collapse and showed stable and robust training with respect to hyper-parameter optimization. On a side note, early attempts to train an original GAN on scRNA-seq data never yielded convergence, while an out-of-the-box implantation of a WGAN did. This does not imply that it is impossible to successfully train an original GAN on such data.

Let us denote by P_r and P_s the distributions of the real and of the generated cells respectively. The Wasserstein distance between them, also known as the Earth Mover distance, is defined as follows:

$$W(P_r, P_s) = \inf_{\gamma \in \Pi(P_r, P_s)} \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\gamma} ||\mathbf{x} - \mathbf{y}||, \quad (1)$$

where x and y are random variables and $\prod(P_r, P_s)$ is the set of all joint distributions $\gamma(\mathbf{x}, \mathbf{y})$ whose marginals are P_r and P_s, respectively. Those distributions represent all the ways (called transport plans) you can move masses from x to y in order to transform P_r into P_s. The Wasserstein distance is then the cost of the optimal transport plan.

However, in this formulation, finding a generator that will generate cells coming from a distribution P_s such that it minimizes the Wasserstein distance with the distribution of the real cells is intractable.

Fortunately, we can use a more amenable, equivalent formulation for the Wasserstein distance, given by the Kantorovich–Rubinstein duality:

$$W(P_r, P_s) = \sup_{||f||_L \leq 1} \mathbb{E}_{\mathbf{x}\sim P_r} f(\mathbf{x}) - \mathbb{E}_{x\sim P_s} f(\mathbf{x}), \quad (2)$$

where $||f||_L \leq 1$ is the set of 1-Lipschitz functions with values in $\mathbb{R}$. The solution to this problem is approximated by training a Neural Network that we previously referred to as the critic network, and whose function will be denoted by f_c.

The input of the generator are realizations of a multivariate noise whose distribution is denoted by P_n. As it is common in the literature, we use a centered Gaussian distribution with unit diagonal covariance (i.e. a multivariate white noise). The dimension of the used Gaussian distribution defines the size of the latent space of the GAN. The dimension of that latent space should reflect the intrinsic dimension of the scRNA-seq expression data we are learning from, and is expected to be significantly smaller than their apparent dimension (i.e. the total number of genes).

If we denote by f_g the function learned by our generator network, the optimization problem solved by the scGAN is identical to that of a Wasserstein GAN:

$$\min_{f_g} \max_{||f_c||_L \leq 1} \mathbb{E}_{\mathbf{x}\sim P_r} f_c(\mathbf{x}) - \mathbb{E}_{\mathbf{x}\sim f_g(P_n)} f_c(\mathbf{x}). \quad (3)$$

The enforcement of the Lipschitz constraint is implemented using the gradient penalty term proposed by Gulrajani et al.[34].

Hence, training an scGAN model involves solving a so-called minmax problem. As no analytical solution to this problem can be found, we recourse to numerical optimization schemes. We essentially follow the same recipe as most of the GAN literature[5,33], with an alternated scheme between maximizing the critic loss (for five iterations) and minimizing the generator loss (for one iteration). For both the minimization and the maximization, we use a recent algorithm called AMSGrad[35], which addresses some shortcomings of the widely used Adam algorithm[36], leading to a more stable training and convergence to more suitable saddle points. The AMSGrad exponential decay parameter beta1 was set to 0.5 and beta2 to 0.9.

Regarding the architecture of our critic and generator networks, which is summarized in Supplementary Fig. 1, most of the existing literature on images prescribes the use of convolutional neural networks (CNN). In natural images, spatially close pixels exhibit stronger and more intricate inter-dependencies. Also, the spatial translation of an object in an image usually does not change its meaning. CNNs have been designed to leverage those two properties. However, neither of these properties hold for scRNA-seq data, for which the ordering of the genes is mostly arbitrary and fixed for all cells. In other words, there is no reason to believe that CNNs are adequate, which is why scGAN uses FC layers. We obtained the best results using an MLP with FC layers of 256, 512, and 1024 neurons for the generator and an MLP with FC layers of 1024, 512, 256 for the critic (Supplementary Fig. 1B, C). At the outermost layer of the critic network, following the recommendation from Arjovsky and Bottou[33], we do not use any activation function. For every other layer of both the critic and the generator networks, we use a Rectified Linear Unit (ReLU) as an activation function.

Naturally, the optimal parameters in each layer of the artificial neural network highly depends on the parameters in the previous and subsequent layers. Those parameters, however, change during the training for each layer, shifting the distribution of subsequent layer's inputs slowing down the training process. In order to reduce this effect and to speed up the training process, it is common to use Normalization layers such as Batch Normalization[37] for each training mini-batch. We found that the best results were obtained when using Batch Normalization at each layer of the generator. Finally, as mentioned in the Datasets and preprocessing section, each real sample used for training has been normalized for library size. We now introduce a custom LSN layer that enforces the scGAN to explicitly generate cells with a fixed library size (Supplementary Fig. 1B).

LSN layer: A prominent property of scRNA-seq is the variable range of the genes expression levels across all cells. Most importantly, scRNA-seq data are highly heterogeneous even for cells within the same cell subpopulation. In the field of Machine Learning, training on such data is made easier with the usage of input normalization. Normalizing input yields similarly ranged feature values that stabilize the gradients. scRNA-seq normalization methods that are used include LSN, where the total number of reads per cell is exactly 20,000 (see also Datasets and preprocessing).

We found that training the scGAN on library-size normalized scRNA-seq data helps the training and enhances the quality of the generated cells in terms of our evaluation criteria (model selection method). Providing library-size normalized cells for training of the scGAN implies that the generated cells should have the same property. Ideally, the model will learn this property inherently. In practice, to speed up the training procedure and make training smoother, we added the aforementioned LSN layer at the output of the generator (Supplementary Fig. 1B). Our LSN Layer rescales its inputs ($\bar{\mathbf{x}}$) to have a fixed, total read count (φ) per cell:

$$\mathbf{y}_{\text{relu}} = \text{ReLU}(\bar{\mathbf{x}}\mathbf{W} + \mathbf{b}), \quad (4)$$

$$\mathbf{y}_{\text{output}} = \frac{\varphi}{\sum_i (\mathbf{y}_{\text{relu}})_i} \mathbf{y}_{\text{relu}}, \quad (5)$$

where W and $\mathbf{b}$ are its weights and biases, and $(\mathbf{y}_{\text{relu}})_i$ denotes the ith component of the $\mathbf{y}_{\text{relu}}$ vector.

cscGAN: Our cscGAN leverages conditional information about each cell type, or subpopulation, to enable the further generation of type-specific cells. The integration of such side information in a generative process is known as conditioning. Over the last few years, several extensions to GANs have been proposed to allow for such conditioning[26,38,39]. It is worth mentioning that each of those extensions are available regardless of the type of GAN at hand.

We explore two conditioning techniques, auxiliary classifiers (ACGAN)[39] and projection-based conditioning (PCGAN)[26]. The former adds a classification loss term in the objective. The latter implements an inner product of class labels at the critic's output. While we also report results obtained with the ACGAN (see Supplementary Table 4), the best results were obtained while conditioning through projection.

In practice, the PCGAN deviates from the scGAN previously described by (i) multiple critic output layers, one per cell type and (ii) the use of Conditional BNL[27], whereby the learned singular scaling and shifting factors of the BNL are replaced with one per cell type.

As described in Section 2 and 3 of ref. [26], the success of the projection strategy relies on the hypothesis that the conditional distributions (with respect to the label) of the data at hand are simpler, which helps stabilizing the training of the GAN. When it comes to scRNA-seq data, it is likely that this hypothesis holds as the distribution of the gene expression levels should be simpler within specific cell types or subpopulations.

Model selection and evaluation. Evaluating the performance of generative models is no trivial task[13,14]. We designed several metrics to assess the quality of our generated cells at different levels of granularity. We will now describe in detail how those metrics were obtained. They can be grouped into two categories: the metrics we used for model selection (in order to tune the hyper-parameters of our GANs) and the metrics we introduced in the Results section.

Metrics used for model selection. As described in the previous section, defining our (c)scGAN model entails carefully tuning several hyper-parameters. We hereby recall the most influential ones: (i) the number and size of layers in the Neural Networks, (ii) the use of an LSN layer, and (iii) the use of a Batch Normalization in our generator network.

For each of our models, before starting the training, we randomly pick 3000 cells from our training data and use them as a reference to measure how it performs. We therefore refer to those 3,000 cells as "real test cells".

To optimize those hyper-parameters, we trained various models and evaluated their performance through a few measures, computed during the training procedure: (a) the distance between the mean expression levels of generated cells and real test cells, (b) the mean sparsity of the generated cells, and (c) the intersection between the most highly variable genes between the generated cells and the real test cells.

First, we compute the mean expression value for each gene in the real test cells. During the training procedure, we also compute the mean expression value for each gene in a set of 3,000 generated cells. The discrepancy is then obtained after computing the Euclidean distance between the mean expression values of the real and the generated cells.

scRNA-seq data typically contains a lot of genes with 0 read counts per cell, which we also use to estimate the similarity of generated and real cells. Naturally, similar sparsity values for real and test cells indicate good model performance whereas big differences indicate bad performance.

Finally, using the Scanpy[31] package, we estimate the 1000 most highly variable genes from the real data. During the training, we also estimate what are the 1,000 most highly variable genes from a sample of 3,000 generated cells. We use the size of the intersection between those two sets of 1,000 highly variable genes as a measurement of the quality of the generation.

Gene expression and correlation. To highlight the performance of our models, we used violin plots of the expression of several marker genes along with heatmaps displaying the correlation between those same marker genes as expressed among all clusters, or among specific clusters.

To produce those plots, we used the expression levels of cells (either test real, or generated by scGAN, cscGAN) in a logarithmic scale. For the heatmaps we compute the Pearson product–moment correlation coefficients.

t-SNE plots. To visualize generated and real cells within same t-SNE plot they are embedded simultaneously. In brief, we are trying to assess how realistic the generated cells are. Thus our reference point is the real data itself. The delineation of what constitutes noise and what constitutes biologically relevant signal should be driven by the real data only. Hence we project the generated cells on the first 50 principal components that were computed from the real cells in the Cell Ranger pipeline[12] (see also Datasets and preprocessing). From this low-dimensional representation, we compute the t-SNE embedding.

To show that the results we obtained were not an artifact of using a Principal Components Analysis, we also reported (Supplementary Fig. 8) the results (t-SNE plots and classification results) obtained while using the first 50 components of

ZIFA[40] (Zero-Inflated Factor Analysis), computed on both the real and generated cells, as an alternate dimensionality reduction method.

Classification of real versus generated cells. Building on the 50-dimensional representation of the cells (t-SNE plots section), we trained classifiers to distinguish between real test cells and generated cells. Using this lower-dimensional representation is motivated by the fact that it captures most of the biologically relevant information while discarding most of the noise, which is known to be high in scRNA-seq data. Moreover, it is statistically more sound to use a dimensionality reduction technique prior to classifying data when the number of observations is in the same order of magnitude as the number of variables, as is the case with the datasets we worked with. As mentioned in the Results section, we trained RF classifiers with 1000 trees and a Gini impurity quality metric of the decision split using the scikit-learn package[32]. The maximum depth of the classifier is set so that the nodes are expanded until all leaves are pure or until all leaves contain less than two samples. The maximum number of features used is the square root of the number of genes.

In order to produce Fig. 1e, which highlights the ability to separate real from generated cells, irrespective of which cluster they are coming from, we used the whole real test set along with generated cells. On the other hand, Fig. 2g, h is cluster specific (cluster 2 and cluster 5 respectively). We trained the RFs using only the cells from those specific clusters. To prevent bias due to class imbalance, each model was trained using an equal number of real test cells and generated cells.

We used a five-fold cross-validation procedure to estimate how each classifier generalizes. To assess this generalization performance, we plotted the Receiver Operating Characteristic (ROC) curves obtained for each fold, along with the average of all the ROC curves. We also display the AUC in each of those cases. Supplementary Tables 4 and 5 report more extensive results.

MMD distance. Computing robust distances over empirical distributions is a difficult issue in high dimension. However, a recent framework called kernel two-sample test[41] was proposed as a statistical test to assess whether two samples are coming from the same distribution. It relies on the computation of a distance called MMD. In a nutshell, it compares the first-order moments (means) of the two samples, in a reproducing kernel Hilbert space. As a consequence, the choice of the kernel is of paramount importance.

Following the recommendations from Shaham et al.[42], which uses a deep neural network to minimize the MMD distance between different scRNA-seq data replicates for batch effect removal, we used a kernel that is the sum of three Gaussian kernels:

$$k(\mathbf{x},\mathbf{y}) = \sum_i \exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{\sigma_i^2}\right),$$

where σ_i's are chosen to be $\frac{m}{2}$, m, $2m$ and m is the median of the average distance between a point to its nearest 25 neighbors.

For the sake of consistency with the other measures we proposed (t-SNE plots, RF classification), we proceeded to compute the MMD distances between samples using their 50 PCs representation found with the Cell Ranger pipeline (as described in the "Dataset and preprocessing" part).

We used the MMD implementation from SHOGUN[43], an efficient kernel-based machine learning package.

Downsampling. To assess the impact of cluster size on the ability of the cscGAN to model the cluster we artificially reduced the number of cells of the relatively large PBMC cluster 2. We call this approach "downsampling" throughout the manuscript.

Eight different percentages {50%, 25%, 10%, 5%, 3%, 2%, 1%, 0.5%} of cluster 2 cells were sampled using a random seed and a uniform sampling distribution over all the cluster 2 cells (Supplementary Table 5). We sampled nested subsets (for each seed, the smaller percentage samples are a complete subset of the larger ones). In order to accurately estimate the generalization error across the different experiments and to avoid potential downsampling artifacts, we conducted all our experiments using five different random seeds. For the classification of cell subpopulations (see next paragraph) we report the average F1 score as well as the five individual F1 score values for the different seeds.

Classification of cell subpopulations. To investigate the use of the proposed cscGAN model for data augmentation, we examined the performance of cell subpopulation classification before and after augmenting the real cells with generated cells. For this purpose, and as described in the previous paragraph and the Results section, we produced alternate datasets with sub-sampled cluster 2 populations (Supplementary Fig. 12).

For simplicity, we focus in this section on the experiment where cluster 2 cells were downsampled to 10% using five different random seeds (Supplementary Fig. 13). We advise to use Supplementary Fig. 13 as an accompanying visual guide to this text description.

Using the previously introduced 50-dimensional PC representation of the cells, three RF models were trained to distinguish cluster 2 cells from all other cell populations (RF downsampled, RF upsampled, and RF augmented) (Supplementary Fig. 13A). In the training data for all the three classifiers, 70% of the cells from all the clusters except cluster 2 (i.e. 37,500 cells) were used (light blue boxes in Supplementary Fig. 13A).

RF downsampled: For the first RF classifier, we used 10% of cluster 2 cells (1502 cells) and 70% other cells (37,500 cells) to train the RF model. We refer to this dataset as the "RF downsampled" set in Supplementary Fig. 13A. This dataset was also used to train the cscGAN model, which is used later to generate in silico cluster 2 cells (Supplementary Fig. 13C). It is important to note that RF classifiers for "RF downsampled" datasets always use weights that account for the cluster-size imbalance. The reason for this is that RFs are sensitive to unbalanced classes, leading to classifiers that always predict the much larger class, thereby optimizing the classification error[44].

RF upsampled: For the second RF classifier, we uniformly sampled with replacement 5,000 cells from the 1,502 cluster 2 cells (10%). We added those 5,000 (copied) cells to the original 1,502 cells. This dataset is referred to as "RF upsampled" in Supplementary Fig. 13A. The rationale for this upsampling is that RF multinomial classifiers are sensitive to the class frequencies in the training data. The upsampling was conducted only for cluster 2 cells as a baseline to which the augmentation is compared. As the augmented and upsampled datasets remain unbalanced, we adjusted the class weights during the training to be inversely proportional to the class frequencies, as outlined in the previous paragraph (RF downsampled).

As a side note, we also conducted experiments where we added standard Gaussian noise to the upsampled cells, which always reduced the performance of the RF classifier and are therefore not shown.

RF augmented: Finally, the third classifier training data "RF augmented" consists of 10% cluster 2 cells as well as 5,000 cluster 2 cells generated using the 10% cscGAN model as shown in Supplementary Fig. 9C. The 10% cscGAN model was trained on 10% cluster 2 cells as well as all other cells (53,571 cells, Supplementary Fig. 13C).

The RF classifiers were trained using the same parameters as described in the Classification of real versus generated cells methods section, using 1,000 trees and Gini impurity. The only difference is that here the class weights during the training are adjusted inversely proportional to the class frequencies, as already mentioned above. The scikit-learn package[32] was used to conduct all experiments to classify cell subpopulations.

Test cells: The test cells used to evaluate the classifiers consisted of 30% of the data from all the clusters. Since we are testing the cscGAN's ability to augment different percentages of real cluster 2 cells, we made sure that the 30% of cluster 2 cells used in the test set were selected from the cells which were not seen by any trained cscGAN model (Supplementary Fig. 13B).

To prove that the downsampling limits the ability to classify and that augmenting the dataset mitigates this effect, all three RF classifiers were trained to classify cluster 2 cells versus all other subpopulations. The F1 score of each classifier is calculated and presented in different colors (Fig. 3a).

Furthermore, in order to understand how augmentation helps to separate close clusters, we trained the same three RF classifiers after removing all clusters except cluster 2 and 1 from the corresponding training data. We repeated this procedure for cluster 2 and 5, and cluster 2 and 3. We chose those clusters in particular because their highly differentially expressed genes are also highly expressed in 2 meaning that separating them from cluster 2 is more difficult (Fig. 1a–c). In a similar way, F1 scores for classification of cluster 2 versus 1 (Supplementary Fig. 14A), cluster 2 versus 3 (Supplementary Fig. 14B), and cluster 2 versus 5 (Supplementary Fig. 14C) are calculated and reported.

As mentioned above, we repeated this procedure for different downsampling levels of cluster 2 cells and for five different sampling seeds for each level (Supplementary Table 5).

When training the RF classifier with the augmented dataset, the number of cells used in the augmentation was set to 5,000 cells. This, however, does not necessarily mean that 5,000 cells is the optimal number of cells to be added. The increase in the F1 score due to augmenting the data with generated cells depends on two factors: (i) the number of real cells in the original subpopulation and (ii) the number of cells used for augmentation. To highlight the impact of the number of generated cells used for data augmentation, we trained the previously mentioned RF classifiers using different numbers of generated cells (from 100 to 12,000) while keeping the number of other cells constant (Supplementary Fig. 16).

Splatter comparison. In addition to what has been previously introduced in the Results section, we also compared the performance of the scGAN to Splatter[22], using the metrics described in their manuscript. Briefly, Splatter simulation is based on a gamma-Poisson hierarchical model, where the mean expression of each gene is simulated from a gamma distribution and cell counts from a Poisson distribution. We noticed that Splatter uses the Shapiro–Wilk test to evaluate the library-size distribution, which limits the number of input cells to 5,000. Therefore, we slightly modified the code that allows Splatter to take more than 5,000 cells as input.

While scGAN learns from and generates library-size normalized cells, Splatter is not suited for that task. For the sake of fairness, we used the Splatter package on the non-normalized PBMC training dataset. We then generated (non-normalized)

cells, which we normalized, so that they could be compared to the cells generated by scGAN. Following ref. [22], we used the following evaluation metrics: distribution of the mean expression, of the variance, of the library sizes and ratio of zero read counts in the gene matrix. The results were computed using the Splatter package and are reported in Supplementary Fig. 5.

We observe that the results obtained by Splatter are marginally better than or identical to these of scGAN (Supplementary Fig. 5). The results from those measures suggest that both Splatter and scGAN constitute almost perfect simulations. However, Splatter simulates virtual genes. While those genes share some characteristics with the real genes Splatter infers its parameters from, there is no one-to-one correspondence between any virtual gene simulated in Splatter-generated cells and the real genes. We therefore did not compare Splatter-simulated cells with real cells, as we did to evaluate the quality of (c)scGAN-generated cells. This also prohibits the use of Splatter for data-augmentation purposes.

This being said, we also would like to pinpoint that while the (c)scGAN is able to capture the gene–gene dependencies expressed in the real data (Fig. 1d, Supplementary Fig. 10), this does not hold for Splatter, for which the virtual genes are mostly independent from each other. To prove this point, we extract the 100 most highly variable genes from the real cells, the cells generated by Splatter, and the cells generated by the scGAN. We then proceed to compute the Pearson correlation coefficients between each pair within those 100 genes (Supplementary Fig. 4). It reveals that while those most highly variable genes in the real cells or those generated by the scGAN exhibit some strong correlations, highly variable genes are mostly independent from each other in the cells generated by Splatter. These results are surprising given that the graphical model used in Splatter is expressive enough to accommodate for complex dependencies between genes. It is likely that it is the inference algorithm that is failing at capturing the gene–gene dependencies in the PBMC dataset, while a manual selection of the parameters of Splatter can allow to simulate cells with some gene–gene dependencies.

SUGAR comparison. Another generative model of high-dimensional data that could be used to generate scRNA-seq data is SUGAR (Synthesis Using Geometrically Aligned Random-walks).

Both scGAN and SUGAR share the assumption that the training data lie on a low-dimensional manifold which is the case of single-cell data[23]. The scGAN uses a random variable $\boldsymbol{Z}$ with a fixed distribution $P(\boldsymbol{z})$ and passes it through a neural network based parametric function (the generator) $(\theta)\mathbf{z} \rightarrow \mathbf{x}$. The output of this parametric function P_θ is then learned using an Earth mover distance to be closer to the real distribution P_r. SUGAR, on the other hand, uses a Gaussian kernel to construct the diffused geometry around each data point and then, using a sparsity-based measure, new points are sampled to even out the sparsity along the manifold.

In order to compare the quality of scGAN-generated cells with SUGAR-generated cells, we run SUGAR on a group of training cells from the PBMC dataset using the publicly available MATLAB implementation (https://github.com/KrishnaswamyLab/SUGAR). The training cells were the same cells used to train an scGAN model and were preprocessed as described in Datasets section PBMC.

SUGAR could generate points to explicitly balance the density over the learnt manifold by assuming that there are sparse regions and then generating points to equalize the estimated sparse areas on the learnt manifold. However such an equalization produces cells that, by design, do not follow the original distribution of the real cells. Therefore, we turned off the density equalization option when we generated cells using SUGAR to ensure that the generated cells compare favorably to the real ones in terms of distribution. For the same reason, we also turned off the imputation step. Finally, using the adaptive noise covariance estimation option of SUGAR resulted in scalability issues (Supplementary Fig. 6F–I, training and generating cells on a reduced 3,000 cells × 2,000 genes dataset required 1.3 Terabytes of RAM and computed for over 36 h), precluding the use of this option on the PBMC dataset. Following SUGAR co-author suggestions, we fixed the noise covariance matrix to be the identity matrix in order to allow SUGAR to generate cells in the original genes space using the available MATLAB version. The generated cells using SUGAR contained some negative values which we replaced with zeros to comply with our analysis workflow (logarithmic transformation using Cell Ranger). For the visualization of the SUGAR-generated cells, we used t-SNE to obtain a two-dimensional visualization of the generated and the real cells (Supplementary Fig. 6A–C). We also computed the MMD statistic obtained from the comparison of real (test) data with the generated data using both SUGAR (59.45) and scGAN (0.872) as described in the MMD methods (Supplementary Table 3). It is worth noting that while the Gaussian noise, added to the real cells, is the crux of how SUGAR generates novel cells. It, however, also may be the reason why the samples produced by SUGAR do not follow the original distribution of the data as closely as those produced by scGAN.

To investigate whether the gene–gene dependencies were kept in the SUGAR-generated data we computed the Pearson correlation coefficients of the cluster-specific marker genes (Supplementary Fig. 6D).

Lastly, we trained an RF classifier to distinguish between the real and the SUGAR-generated cells. We conjectured that RF classifier should have close to chance-level performance in the task of distinguishing the generated data from the real data. The RF classifier reaches 0.98 AUC when discriminating between the real and generated cells (blue curve in Supplementary Fig. 6E).

Expression imputation with MAGIC. An important aspect of using an scGAN for generating realistic cells is its fidelity in learning the distribution of the input data regardless of the preprocessing which is applied. Imputation of scRNA-seq data is used to denoise the data, to reduce the amount of drop-outs, and consequently to more accurately recover the gene–gene interactions. For this reason, we investigated the ability of an scGAN to generate realistic imputed cells when real imputed cells are used in the training.

We used MAGIC[25] to impute the scGAN training data, a method developed to impute missing values and to restore the structure of the scRNA-seq data.

The Mouse Bone Marrow dataset was used in this analysis after applying the basic filtering and the LSN we applied in all our experiments (refer to Datasets Supplementary Table 1, preprocessing section). The preprocessed cells are then imputed using the open source MAGIC implementation. In accordance with the MAGIC tutorial all genes were used with four diffusion steps. Afterwards, we trained an scGAN models for 100k steps on both imputed and non-imputed data. Both models were used to generate cells which we used to plot the gene–gene relationships of three genes in the form of scatter plots. To evaluate imputation fidelity we used the three genes that were used in the MAGIC online tutorial (Ifitm1, a stem cell marker, Klf1, an erythroid marker, and Mpo, a myeloid marker).

Regulon detection using SCENIC. We used SCENIC[19] to evaluate whether scGANs model active regulons in the Zeisel RNA-seq dataset. This dataset was used by the authors of SCENIC to show cross-species Dlx1 regulon activity. We selected the top 50 target genes with highest weight for each TF and subsequently found significantly over-represented TF-binding motifs in the set of genes. Modules with enriched TF-binding motifs were kept and defined as active regulons. We then trained an scGAN model on the Zeisel dataset and used it to generate 10,000 library-size normalized cells. The Dlx1 regulon was then found in the real dataset (realDlx1) as well as in the generated one (genDlx1). In addition, we used AUCell to calculate the regulon binarized activity of the realDlx1 regulon in the cells of the generated dataset and the genDlx1 regulon in real cells. Reciprocal activity of realDlx1 and genDlx1 regulons are visualized using t-SNE on real and generated data (Supplementary Fig. 4).

Pseudo-time analysis with PAGA. In this analysis we investigate the ability and the fidelity of the scGAN model to generate scRNA-seq data corresponding to continuous cell states. For this purpose, we used PAGA pseudo-time topology-preserving embedding with partition-based graph abstraction. While clustering enables understanding the biological signals within cell populations, trajectory analysis using pseudo-time and graph embeddings allows for the interpretation of continuous phenotypes and processes such as development and disease progression[28]. We chose an scRNA-seq hematopoiesis dataset (bone marrow dataset) that contains many intermediate and transition states to investigate the performance of such trajectory analysis[29].

In order to examine the ability of scGAN models to learn the manifold the data lie on, we performed a pseudo-time analysis as described in the official git repository of the hematopoiesis scRNA-seq data [https://github.com/theislab/paga]. The cells were first preprocessed using the Zheng preprocessing pipeline. Afterwards the force-directed single-cell graph was built using 20 PCA components[45,46]. The graph is then de-noised and rebuilt using PAGA-initialization as described in the official tutorial of PAGA. Supplementary Figure 17A–C shows the graph of the bone marrow scRNA-seq data. Scanpy pseudo-time analysis was used to infer the progression of cells through geodesic distance along the graph[47].

In the next step, we downsampled a specific transient cell state represented by cells grouped in node 4 of the PAGA graph (Supplementary Fig. 17C–F). The original fourth Louvain group population of 150 cells was downsampled to 13 cells. The downsampled scRNA-seq data (the training data) were then used to train an scGAN model without providing any prior information about the cells' states or clusters. After training, the model was used to generate cells that we compared to the original dataset. After investigating the force-directed single-cell graph of the generated cells combined with original cells, we noticed that the generated cells were covering all cellular states of the original scRNA-seq data (Supplementary Fig. 17C–F).

These results motivated us to investigate the fidelity of the scGAN to learn the downsampled cellular state of transient state 4. Therefore, we searched within the generated cells for cells that are close to the sparse area created by the downsampling process. A group of 137 generated cells were found and added to the downsampled scRNA-seq data. We refer to this combined group of generated and downsampled cells as augmented cells. Our assumption is that the generated cells recover the lost biological signal represented by the downsampled transient state. To prove this assumption, we plotted the force-directed single-cell graph of the augmented data and compared it with the one built from the downsampled data. The cells' graph embeddings were recomputed using PAGA-initialization so that the cells are structured in a meaningful topology-preserving layout that reflects the real cell–cell interconnections and the paths of single cells. Data augmentation of the downsampled cells re-established the developmental trajectories that were observed in the real data and lost in the downsampled data, as shown in

Supplementary Fig. 17. Of note, the scGAN was trained with reduced neuron numbers to accommodate for the small size of the dataset.

Software, packages, and hardware used. For the sake of reproducibility, here is a list of the version of all the packages we used: Tensorflow v1.8, Scanpy v1.2.2, Anndata v0.6.5, Pandas v0.22.0, Numpy v1.14.3, Scipy v1.1.0, Scikit-learn v0.19.1, R v3.5.0 (2018-04-23), loomR v0.2.0, SHOGUN v6.1.3, SingleCellexperiment v1.2.0, Splatter v1.4.0, SUGAR v0.0, MAGIC v1.3.0, SCENIC v0.1.7, GENIE3 v1.0.0, Rcistarget v0.99.0, AUCell v0.99.5, RcisTarget.mm9.motifDatabases.20k v0.1.1, ZIFA v0.1. Regarding hardware, all (c)scGAN models were trained on a single-GPU of an NVIDIA DGX-1 server (Tesla V100 GPUs).

Reporting summary. Further information on research design is available in the Nature Research Reporting Summary linked to this article.

Data availability

The datasets used and analyzed during the current study are available on the 10x Genomics dataset repository at https://support.10xgenomics.com/single-cell-gene-expression/datasets/1.1.0/fresh_68k_pbmc_donor_a for PBMC, at https://support.10xgenomics.com/single-cell-gene-expression/datasets/1.3.0/1M_neurons for Brain small and Brain large, and in the Gene Expression Omnibus repository, at https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE72857 for Bone Marrow, and https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE60361 for Zeisel.

Code availability

Our (c)scGAN Tensorflow[48] implementation can be found on https://github.com/imsb-uke/scGAN, including documentation for the training of the (c)scGAN models. As mentioned before, we used Scanpy[31] to conduct most of the data analysis. We also compared our results to those of Splatter[22], and adapted the code they provided on Github (https://github.com/Oshlack/splatter).

Received: 25 August 2018; Accepted: 13 December 2019;
Published online: 09 January 2020

References

1. Munafò, M. R. et al. A manifesto for reproducible science. *Nat. Hum. Behav.* **1**, 0021 (2017).
2. Karczewski, K. J. & Snyder, M. P. Integrative omics for health and disease. *Nat. Rev. Genet.* **19**, 299–310 (2018).
3. Button, K. S. et al. Power failure: why small sample size undermines the reliability of neuroscience. *Nat. Rev. Neurosci.* **4**, 365–376 (2013).
4. Mariani, G., Scheidegger, F., Istrate, R., Bekas, C. & Malossi, C. BAGAN: data augmentation with balancing GAN. Preprint at *arXiv* https://arxiv.org/abs/1803.09655 (2018).
5. Goodfellow, I. et al. Generative adversarial nets. in *Advances in Neural Information Processing Systems* **27** (Montreal, 2014).
6. Karras, T., Aila, T., Laine, S. & Lehtinen, J. Progressive growing of GANs for improved quality, stability, and variation. Preprint at *arXiv* https://arxiv.org/abs/1710.10196 (2017).
7. Isola, P., Zhu, J.-Y., Zhou, T. & Efros, A. A. Image-to-image translation with conditional adversarial networks. in *2017 IEEE Conference on Computer Vision and Pattern Recognition* (2017).
8. Creswell, A. et al. Generative Adversarial Networks: an overview. *IEEE Signal Process. Mag.* **35**, 53–65 (2017).
9. Kingma, D. P. et al. Improved variational inference with inverse autoregressive flow. in *Advances in Neural Information Processing Systems* (Barcelona, Spain, 2016).
10. Kingma, D. P. & Welling, M. Auto-encoding variational Bayes. Preprint at *arXiv* https://arxiv.org/abs/1312.6114 (2013).
11. Tanay, A. & Regev, A. Scaling single-cell genomics from phenomenology to mechanism. *Nature* **541**, 331–338 (2017).
12. Zheng, G. X. Y. et al. Massively parallel digital transcriptional profiling of single cells. *Nat. Commun.* **8**, 14049 (2017).
13. Theis, L., van den Oord, A. & Bethge, M. A note on the evaluation of generative models. in *International Conference on Learning Representations* (San Juan, Puerto Rico, 2016).
14. Lucic, M., Kurach, K., Michalski, M., Gelly, S. & Bousquet, O. Are GANs created equal? A Large-Scale Study. in *Advances in Neural Information Processing Systems* (Montreal, Canada, 2018).
15. Arjovsky, M., Chintala, S. & Bottou, L. Wasserstein generative adversarial networks. in *International Conference on Machine Learning* (Sydney, Australia, 2017).
16. van der Maaten, L. & Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **9**, 2579–2605 (2008).
17. Van Der Maaten, L., Courville, A., Fergus, R. & Manning, C. Accelerating t-SNE using Tree-based algorithms. *J. Mach. Learn. Res.* **15**, 3221−3245 (2014).
18. Davidson, E. H. Emerging properties of animal gene regulatory networks. *Nature* **468**, 911–920 (2010).
19. Aibar, S. et al. SCENIC: Single-cell regulatory network inference and clustering. *Nat. Methods* **468**, 911–920 (2017).
20. Zeisel, A. et al. Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq. *Science* **347**, 1138–1142 (2015).
21. Breiman, L. Random forests. *Mach. Learn.* **45**, 5–32 (2001).
22. Zappia, L., Phipson, B. & Oshlack, A. Splatter: simulation of single-cell RNA sequencing data. *Genome Biol.* **18**, 174 (2017).
23. Lindenbaum, O., Stanley, J. S., Wolf, G. & Krishnaswamy, S. Geometry-based data generation. in *Advances in Neural Information Processing Systems* (Montreal, Canada, 2018).
24. Ghahramani, A., Watt, F. M. & Luscombe, N. M. Generative adversarial networks uncover epidermal regulators and predict single cell perturbations. Preprint at *bioRxiv* https://www.biorxiv.org/content/10.1101/262501v2 (2018).
25. van Dijk, D. et al. Recovering gene interactions from single-cell data using data diffusion. *Cell* **174**, 716–729.e27 (2018).
26. Miyato, T. & Koyama, M. cGANs with projection discriminator. Preprint at *arXiv* https://arxiv.org/abs/1802.05637 (2018).
27. Dumoulin, V., Shlens, J. & Kudlur, M. A learned representation for artistic style. in *International Conference on Learning Representations* (Toulon, France, 2017).
28. Wolf, F. A. et al. PAGA: graph abstraction reconciles clustering with trajectory inference through a topology preserving map of single cells. *Genome Biol.* **20**, 1–9 (2019).
29. Paul, F. et al. Transcriptional heterogeneity and lineage commitment in myeloid progenitors. *Cell* **163**, 1663–1677 (2015).
30. Traag, V. A. Faster unfolding of communities: speeding up the Louvain algorithm. *Phys. Rev. E* **92**, 032801 (2015).
31. Wolf, F. A., Angerer, P. & Theis, F. J. SCANPY: large-scale single-cell gene expression data analysis. *Genome Biol.* **19**, 15 (2018).
32. Pedregosa, F. et al. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011).
33. Arjovsky, M. & Bottou, L. Towards principled methods for training generative adversarial networks. in *International Conference on Learning Representations* (Toulon, France, 2017).
34. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V. & Courville, A. Improved training of Wasserstein GANs. in *Advances in Neural Information Processing Systems* (Long Beach, Florida, USA, 2017).
35. Reddi, S. J., Kale, S. & Kumar, S. On the convergence of adam and beyond. in *International Conference on Learning Representations* (Vancouver, Canada, 2018).
36. Kingma, D. P. & Ba, J. A. A method for stochastic optimization. in *International Conference on Learning Representations* (San Diego, USA, 2015).
37. Ioffe, S. & Szegedy, C. Batch normalization: accelerating deep network training by reducing internal covariate shift. in *International Conference on Machine Learning* (Lille, France, 2015).
38. Mirza, M. & Osindero, S. Conditional generative adversarial nets. Preprint at *arXiv* https://arxiv.org/abs/1411.1784 (2014).
39. Odena, A., Olah, C. & Shlens, J. Conditional image synthesis with auxiliary classifier GANs. in *Proceedings of the 34th International Conference on Machine Learning* (Sydney, Australia, 2017).
40. Pierson, E. & Yau, C. ZIFA: dimensionality reduction for zero-inflated single-cell gene expression analysis. *Genome Biol.* **19**, 241 (2015).
41. Gretton, A. A kernel two-sample test. *J. Mach. Learn. Res.* **13**, 723–773 (2012).
42. Shaham, U. et al. Removal of batch effects using distribution-matching residual networks. *Bioinformatics* **33**, 2539–2546 (2017).
43. Sonnenburg, S. et al. The SHOGUN machine learning toolbox. *J. Mach. Learn. Res.* **11**,1799–1802 (2010).
44. Zadrozny, B., Langford, J. & Abe, N. Cost-sensitive learning by cost-proportionate example weighting. in *3rd IEEE International Conference on Data Mining* (IEEE, 2003).
45. Islam, S. et al. Characterization of the single-cell transcriptional landscape by highly multiplex RNA-seq. *Genome Res.* **21**, 1160–1167 (2011).
46. Jacomy, M., Venturini, T., Heymann, S. & Bastian, M. ForceAtlas2, a continuous graph layout algorithm for handy network visualization designed for the Gephi software. *PLoS ONE* **9**, e98679 (2014).
47. Haghverdi, L., Büttner, M., Wolf, F. A., Buettner, F. & Theis, F. J. Diffusion pseudotime robustly reconstructs lineage branching. *Nat. Methods* **13**, 845–848 (2016).
48. Abadi, M. et al. Tensorflow: a system for large-scale machine learning. in *12th USENIX Symposium on Operating Systems Design and Implementation* (Savannah, GA, USA, 2016).

Acknowledgements
We would like to thank Ulf Panzer and the Institute of Medical Systems Biology for helpful discussions and suggestions. In particular, we would like to thank Sven Heins and the ZMNH IT for setting up the Deep Learning IT infrastructure and the daily support. This work was supported by the grants SFB 1286/Z2 to P.M. and M.M.; SFB 1192 to C.K., C.F.K., and S.B.; DFG BO 4224/4-1 to D.S.M.; and BMBF IDSN to V.B.

Author contributions
S.B. initiated the project. S.B., P.M., M.M. and D.S.M. designed the study, deep learning models, and analysis. P.M. M.M., and D.S.M. built the deep learning models. P.M., M.M., V.B., and C.K. analyzed the data. S.B., P.M., M.M., D.S.M., V.B., and C.F.K. contributed to the manuscript.

Competing interests
The authors declare no competing interests.

Additional information
Supplementary information is available for this paper at https://doi.org/10.1038/s41467-019-14018-z.

Correspondence and requests for materials should be addressed to S.B.

Peer review information *Nature Communications* thanks Smita Krishnaswamy, Johannes Söding and the other, anonymous, reviewer(s) for their contribution to the peer review of this work. Peer reviewer reports are available.

3.6 Oasis, 2018

Rahman et al. BMC Bioinformatics (2018) 19:54
https://doi.org/10.1186/s12859-018-2047-z

BMC Bioinformatics

SOFTWARE Open Access

Oasis 2: improved online analysis of small RNA-seq data

Raza-Ur Rahman[1,2], Abhivyakti Gautam[1], Jörn Bethune[1,2], Abdul Sattar[1,2], Maksims Fiosins[1,2], Daniel Sumner Magruder[1,2], Vincenzo Capece[1], Orr Shomroni[1] and Stefan Bonn[1,2,3*]

Abstract

Background: Small RNA molecules play important roles in many biological processes and their dysregulation or dysfunction can cause disease. The current method of choice for genome-wide sRNA expression profiling is deep sequencing.

Results: Here we present Oasis 2, which is a new main release of the Oasis web application for the detection, differential expression, and classification of small RNAs in deep sequencing data. Compared to its predecessor Oasis, Oasis 2 features a novel and speed-optimized sRNA detection module that supports the identification of small RNAs in any organism with higher accuracy. Next to the improved detection of small RNAs in a target organism, the software now also recognizes potential cross-species miRNAs and viral and bacterial sRNAs in infected samples. In addition, novel miRNAs can now be queried and visualized interactively, providing essential information for over 700 high-quality miRNA predictions across 14 organisms. Robust biomarker signatures can now be obtained using the novel enhanced classification module.

Conclusions: Oasis 2 enables biologists and medical researchers to rapidly analyze and query small RNA deep sequencing data with improved precision, recall, and speed, in an interactive and user-friendly environment.

Availability and Implementation: Oasis 2 is implemented in Java, J2EE, mysql, Python, R, PHP and JavaScript. It is freely available at https://oasis.dzne.de

Background

Small RNAs (sRNAs) are a class of short, non-coding RNAs with important biological functions in nearly all aspects of organismal development in health and disease. Especially in diagnostic and therapeutic research sRNAs, such as miRNAs and piRNAs, received recent attention [18]. The current method of choice for the quantification of the genome-wide sRNA expression landscape is deep sequencing (sRNA-seq).

To date several local as well as server-based sRNA-seq analysis workflows are available that differ in their analysis portfolio, performance, and user-friendliness. Analysis workflows that need to be installed by the end-user comprise, for example, sRNA workbench [1] for the quantification and identification of differentially expressed sRNAs and CAP-miRSeq [16] for the quantification of known and novel miRNAs including variant calling and subsequent differential expression analysis. While workflows that are installed on a local machine offer greater data security and may provide greater flexibility, they require installation, availability of servers, software and hardware maintenance as well as regular updates.

Recent additions to sRNA analysis web applications include omiRas [11], supporting quantification, differential expression and interactive network visualization; mirTools 2.0 [20] that allows for differential expression and gene ontology analysis of detected sRNAs; MAGI, an all-in-one workflow with detailed interactive web reports [8]; Chimira that allows for the detection of miRNA edits and modifications [17]; sRNAtoolbox [15] performs expression profiling of sRNA-seq data, differential expression as well as target gene prediction and visualization of analysis results; and Oasis [2], which supports the detection and annotation of known and

* Correspondence: sbonn@uke.de
[1]Laboratory of Computational Systems Biology, German Center for Neurodegenerative Diseases, Göttingen, Germany
[2]Institute of Medical Systems Biology, Center for Molecular Neurobiology, University Clinic Hamburg-Eppendorf, Hamburg, Germany
Full list of author information is available at the end of the article

novel sRNAs, multivariate differential expression analysis, biomarker detection, and job automation via an advanced programming interface (API). Here we present Oasis 2, an improved major release of the Oasis web application with many new and enhanced features for Biologists and Bioinformaticians (Table 1).

At the heart of Oasis 2 lies the new sRNA detection workflow that is faster and identifies more sRNAs with higher precision. In addition, Oasis 2 now supports sRNA-seq analyses for any organism, detects potential cross-species miRNAs, and reports viral and bacterial infections in samples with high precision and recall. Oasis 2 predicts and stores novel miRNAs in Oasis-DB and allows users to search and extract information for over 700 predicted high-quality miRNAs across 14 organisms. Oasis 2 classification module is improved with the use of balanced sampling and feature pruning methods that enables robust biomarker detection. Like its predecessor Oasis, Oasis 2's differential expression module supports multiple group comparisons (e.g. control vs. treatment 1 vs. treatment 2) and differential expression using co-variates such as age, gender, and medication. The differential expression and classification modules report various quality metrics including known and predicted targets of miRNAs in a downloadable, interactive web report. This web report allows for the subsequent functional enrichment analysis of miRNAs using GeneMania (interactome and GO analysis) [21], g:Profiler (GO, pathway-Kegg, Reactome) [13], STRING (protein-protein interaction network) [4], STITCH (chemical-protein interaction network) [9], and DAVID (enrichment analysis based on many biological databases) [6]. Oasis 2 is also at the heart of the sRNA Expression Atlas (SEA, https://sea.dzne.de), a web application for the interactive querying, visualization, and analysis for over 2000 published sRNA samples. Lastly Oasis 2 features many new analysis and visualization options such as support for adapter trimmed data, options to trim additional barcodes, and interactive plots for sRNA detection and classification output. It has no restrictions on the size or number of samples and has no limits on the analyses per user.

Implementation

The following paragraphs will describe the technical details of Oasis 2's novel sRNA detection, database, and classification modules. Additional information can be found in the supplementary material.

sRNA detection

One of the key differences between Oasis 2 and its predecessor is the fully revised detection of known and novel sRNAs. The new detection workflow increases the alignment speed, is more accurate, and supports the analysis of any model and non-model organism (Fig. 1, Additional file 1). While Oasis detected sRNAs using a single genome alignment step, Oasis 2 is based upon a four-tiered alignment strategy. Users can upload (un)-compressed data that originates from one of the 14 different organisms provided in Oasis 2 and the data will be aligned to the (i) target organism's (TO) transcripts, (ii) TO's genome, (iii) pathogen genomes, and (iv) non-target organism's (NTO) miRNA transcripts in succession (Fig. 1). In the TO Transcript alignment (step 1), reads are aligned to TO transcripts in Oasis-DB, a database that contains transcript information of miRNAs and other sRNA species (snRNA, snoRNA, rRNA and

Table 1 sRNA-seq web application comparison

Feature	Oasis 2	Oasis	omiRas	mirTools 2.0	MAGI	Chimira	sRNAtoolbox
FASTQ compression	✓	✓			✓	✓	
miRNA prediction	✓	✓	✓	✓	✓		✓
miRNA modifications and edits						✓	✓
Novel miRNA database	✓						
Infection and cross-species analysis	✓						✓
Non-model organism	✓					✓	
Differential expression	✓	✓	✓	✓	✓	✓	✓
Multivariate differential expression	✓	✓					✓
Classification	✓	✓					
Novel miRNA target prediction	✓	✓		✓	✓		✓
Pathway/GO analysis	✓	✓	✓	✓	✓		✓
Batch job submission (API)	✓	✓					
Genome browser							✓

Of note, this comparison does not include all available sRNA analysis web applications. It only considers the most recent web applications that we deemed most competitive and we do not compare to standalone software solutions that have to be locally installed

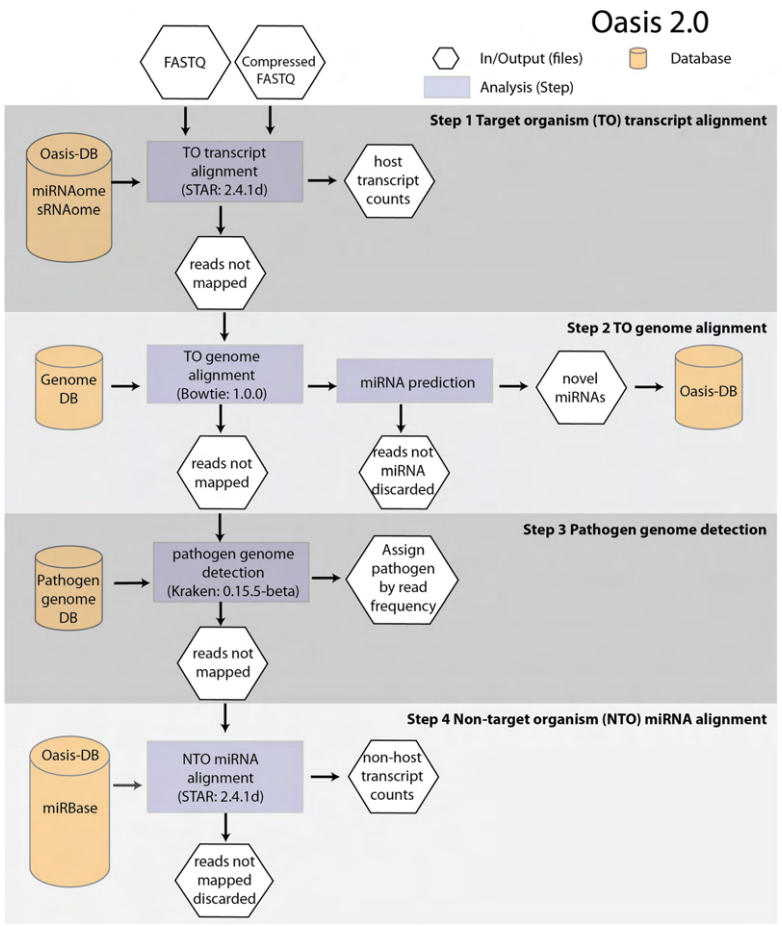


Fig. 1 Detection of sRNAs in Oasis 2: The web application allows for the upload of raw or compressed FASTQ files to Oasis 2's sRNA detection module. After pre-processing (adapter/barcode trimming and length filtering), reads are first aligned to target organism (TO) transcripts that are stored in Oasis-DB (Step 1), including known miRNAs, piRNAs, snoRNAs, snRNAs, rRNAs, and high-stringency predicted miRNAs and their families. Unmapped reads of Step1 are subsequently aligned to the TO's genome (Step 2) to predict and subsequently store novel miRNAs in Oasis-DB. Unmapped reads from step 2 are mapped to bacterial, archaeal, and viral genomes using Kraken (Step 3) to detect potential pathogenic infections or contaminations. Finally, reads that could not be aligned in steps 1–3 are aligned to all non-target organism (NTO) miRNAs in miRBase (Step 4) to detect potentially orthologous or cross-species miRNAs. In case the user's data does not correspond to one of the 14 supplied organisms, Oasis 2 aligns the reads only to NTO miRNAs (Step 4), supporting the detection of miRNA expression in any organism

piRNAs) from miRBase, piRNAbank, Ensembl, predicted novel miRNAs, and sRNA families. In this step reads of length 15–19 nucleotides are aligned with no mismatches whereas reads of length 20–32 nucleotides are mapped allowing for 1 mismatch (Step 2 in Fig. 1). In the TO Genome alignment (step 2), reads that do not align to TO transcripts are subsequently aligned to the reference genome allowing for 1 mismatch and no more than five potential genomic target regions to predict novel, high-quality miRNAs (Additional file 1 section 1.2 'Alignment and counting'). Predicted novel miRNAs are then added to Oasis-DB as described in section 2.2 'Detection and storage of novel miRNAs'. In the Pathogen Genome detection (step 3), reads that could not be aligned to the TO transcriptome or TO genome are used to identify pathogenic sRNA signatures from bacteria and viruses, supplying information on potentially infected samples (Fig.2 & Additional file 1). To this end, we indexed Oasis Pathogen-Genome-DB that consists of 4336 viral and 2784 bacterial/archaeal genomes with

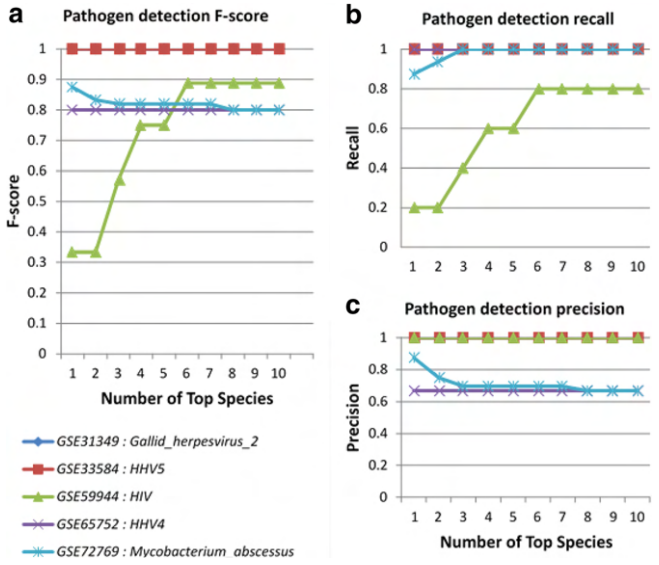


Fig. 2 Pathogen detection performance: To assess the performance of 'pathogen detection module', sRNA datasets with defined viral or bacterial infections were analyzed and the F-score (**a**), recall (**b**), and precision (**c**) of the pathogen predictions were measured for the top 10 reported organisms. Overall, the prediction of bacterial (*M. abscessus*) and viral (*HIV, HHV4, HHV5, Gallid_herpesvirus_2*) infections resulted in high F-scores, recall, and precision, especially when the top 5 predicted pathogen species are reported. In consequence, Oasis 2 currently reports the top five predicted pathogen species based on their read counts

Kraken [19] using a k-mer length of 18. In the Non-TO miRNA alignment (step 4), reads that could not be aligned to TO transcripts, the TO genome or pathogen genomes are aligned without any mismatches to all NTO transcripts of miRBase to detect potential orthologous or cross-species miRNAs. In cases where the data does not belong to one of the 14 supported genomes available in Oasis 2, reads can be aligned to all known and novel predicted miRNAs and miRNA families stored in Oasis-DB (Additional file 1).

In addition to the new alignment strategy, the sRNA detection module also supports data with already trimmed adapters. It also has an option for barcode removal, which is required for the analysis of libraries generated with e.g. the NEXTflex kit. In the case of barcode removal, Oasis 2 first discards the 3′ adapter sequence (in case the adapter is not already trimmed), and then removes an additional N (user defined, default is 0) bases from the adapter-clipped reads.

Detection and storage of novel miRNAs

Another major improvement of Oasis 2 is the ability to query and visualize detailed information for over 700 high-quality predicted miRNAs across 14 organisms (Fig. 1, Additional file 1: Figure S1). Oasis-DB comprises information on all MiRDeep2 [5] predicted miRNAs that pass stringent selection criteria during the sRNA detection step of Oasis 2 (2.1 & Additional file 1), including the miRNA ID, organism, chromosomal location, precursor and mature sequences, structure, read counts, prediction scores, and detailed information on the software and its versions used to predict the miRNA. To assure that Oasis-DB contains only high-quality miRNA entries, novel predicted miRNAs have to pass the three criteria. The log-odds score assigned to the hairpin by miRDeep2 (miRDeep2-score) should be greater than 10, the predicted miRNA hairpin should not have sequence similarity to reference tRNAs or rRNAs, and the estimated randfold *p*-value of the excised potential miRNA hairpin should be equal to or lower than 0.05.

Novel predicted miRNAs are added to Oasis-DB using the standard nomenclature (Additional file 1 section 1.4 'Oasis-DB miRNA insertion and naming').

In addition to novel miRNAs, Oasis-DB also stores information on all other sRNAs and sRNA families (Additonal file 1). To provide access to Oasis-DB we created a novel web frontend, the Oasis 2 'Search' module, which allows users to query miRNAs by mature/precursor ID or sequence, and the organism they come from. Information on high-confidence novel miRNAs is also shared with SEA, a web application that provides expression information of known and novel miRNAs for over 2000 samples (https://sea.dzne.de).

Classification and differential expression

To allow for enhanced sRNA-based biomarker detection several profound changes to the Oasis 2 classification module were made, resulting in more robust biomarker detection with increased accuracy (Additional file 1: Figure S2 , Additional file 1 section 'Oasis 2 classification module'). To increase the performance of the Random Forest-based (RF) classification module we first implemented balanced sampling (Additional file 1), making sure RF predictions would not be biased in the case of uneven class distribution. Since RFs can perform poorly on data that contains few informative and many non-informative features, the classification module was augmented with a feature pruning routine (Additional file 1), reporting prediction performance for the full and best RF models. In addition to providing information on model accuracy using the out-of-bag (OOB) error, Oasis 2 now also provides model performance information based on cross-validation. All classification results can be explored in interactive web reports, allowing for a detailed quality and performance analysis of the predicted biomarkers.

Moreover, we have improved the quality of output plots in the DE module and updated the DESeq2 version for the analysis of differential sRNA expression. Further details about DE module can be found in Additional file 1 section 1.5 'Oasis 2 differential expression module' and Additional file 1: Table S3.

Technologies and compatibility

Oasis 2 is implemented in Java, J2EE, mysql, Python, R, PHP and JavaScript. For the usage JavaScript should be enabled in the browser. Oasis 2 functionality was tested on all major browsers (Table 2). It has no restrictions on the size or number of samples and has no limits on the analyses per user. Potential user-specific problems can arise when i) an institution or university has upload limits, ii) proxy settings that would interrupt or prohibit long uploads, or iii) JavaScript is disabled or blocked. Oasis 2 is freely available at (https://oasis.dzne.de).

Results

We compared the set of analysis options and the analysis speed of Oasis 2 to six state-of-the-art sRNA analysis web applications, including Oasis, omiRas, mirTools 2.0, MAGI, Chimira and sRNAtoolbox, and found that it compares favorably in the number of analysis options (Table 1) and the analysis speed (Table 3). When tested on four publically available datasets, Oasis 2 detected 19 out of 27 (70%) differentially expressed (DE) genes that were previously validated (true positives) and did not detect 4/4 (100%) miRNAs that showed a significant DE in deep sequencing but could not be validated with qPCR (false positives), highlighting both the sensitivity and specificity of Oasis 2. Finally, we compared the performance of the novel classification module to the one implemented in Oasis, showing that prediction accuracy as well as robustness are increased.

Table 2 Oasis 2 browser compatibility

Browser	Version
Chrome	61.0.3163.100, 62.0.3202.62
Mozilla Firefox	55.0.3, 56.0 (64-bit), 57.0 (64-bit)
Chromium	62.0.3202.75
Safari	11.0.1
Internet explorer	11

Browsers that are used to test Oasis 2 functionalities

Detection and differential expression of sRNAs

To estimate if the novel sRNA detection workflow of Oasis 2 identifies and quantifies sRNAs correctly we analyzed four published datasets containing validated sRNA changes using Oasis 2 with default settings. Of note, none of the above-mentioned publications looked into the DE of other small RNA classes (snRNA, snoRNA and rRNA and piRNAs), so the analyses were restricted to miRNAs.

Alzheimer disease data

We started by analyzing an Alzheimer disease (AD) sRNA dataset that consists of 48 Alzheimer and 22 control samples [10] using Oasis 2 and default settings. The original publication uses a Wilcoxon-Mann-Whitney test detecting 125 known DE miRNAs. Oasis 2 detected 103 DE miRNAs using an adjusted p-value < 0.1, of which 62(60%) overlapped with the original analysis. The overlap of 60% seems reasonable, given the different statistical approaches and miRBase versions used for the detection and DE analysis of the miRNAs. In the original publication 8/10 known miRNAs were validated to be differentially expressed in the same direction, whereas two miRNAs (hsa-miR-1285-5p and hsa-miR-26a-5p) were not validated in the same direction (instead of upregulation they showed downregulation in qPCR). Interestingly these two miRNAs were not detected to be differentially expressed by Oasis 2. On the other hand Oasis 2 was able to detect 3/3 upregulated miRNAs (hsa-let-7d-3p, hsa-miR-5010-3p and hsa-miR-151a-3p), 3/5 downregulated miRNAs (hsa-miR-532-5p, hsa-miR-26b-5p and hsa-let-7f-5p), and it did not detect two downregulated miRNAs (hsa-miR-103a-3p, hsa-miR-107). In summary, Oasis 2 was able to detect 6/8 (75%) validated differentially expressed known miRNAs and not detecting 2/2 false positives from the original study. Unfortunately, two novel miRNAs validated in the original study are not added to miRBase yet, therefore we were not able to compare to them.

Table 3 Runtime comparison of different sRNA-seq web applications

Demo Dataset	Oasis 2 (total) [1]	Oasis (total)[1]	MAGI (total)	Chimira (total)	omiRas	mirTools[7] 2.0	sRNAtoolbox
AD (287 GB)[4]	8 h31m50s	12h29m12s	NA[2]	NA[4]	NA[5]	NA	NA
Psoriasis (48 GB)	1h35m17s	5h49m4s	48h[3]	3h3m12s	NA[6]	NA	NA
Renal Cancer (9 GB)	31m43s	1h8m41s	8h[3]	47m11s	9h31m	NA	NA

[1]Run time estimate includes the data compression and decompression, the sRNA Detection, DE Analysis, and Classification. [2] We could not get MAGI to upload all AD files. Most probably it has a problem with the quality or format of one of the files. [3] These values were obtained from the MAGI website. [4] Chimira does not support the analysis of more than 25 files at a time, which prohibited us from getting runtime estimates for the AD dataset. [5] omiRas did not finish uploading files, which prohibited us from getting runtime estimates for the AD dataset. [6] omiRas http uploading error. [7] We cannot compare the runtime of mirTools 2.0 as maximum file size to upload is limited to 30 Mb. The sRNAtoolbox web application has been non-functional since 30/05/2017, which prohibited any runtime comparison (http://bioinfo2.ugr.es:8080/srnatoolbox/quick-start/)

Psoriasis data

Oasis 2's performance was next assessed using a set of 10 Psoriasis and 10 control samples [7]. The original publication uses a hypergeometric test to assess differential expression (Pearson's chi-square test) that is followed by a Bonferroni multiple-testing correction.

In accordance with the analyses performed in the original publication, we only considered non-redundant pre-miRNAs. Oasis 2 found 195 DE miRNAs (166 non-redundant known pre-miRNAs) (adjusted p-value < 0.1) whereas the original publication contains only 98 DE miRNAs (70 non-redundant known pre-miRNAs). Of the 70 DE pre-miRNAs in the original study, 51 (72.85%) could also be found in the list of Oasis 2 DE miRNAs (Table 4). In addition, 5/8 (62.5%) experimentally validated DE miRNAs (miR-21, miR-31,,, miR-944, miR-135band miR-675) were detected by Oasis 2, not identifying validated miRNAs miR-124, miR-431 and miR-219-2-3p that show high expression variation in the original publication. Furthermore, Oasis 2 identified 2/3 (67%) predicted novel DE miRNAs (hsa-miR-203b and hsa-miR-3613) while missing hsa-miR-4490 (miRBase v21). In addition, Oasis 2 did not detect the false positive miR-431* (1/1, 100%) that was predicted to be DE in the original Psoriasis study [7] but could not be validated by qPCR. In summary, Oasis 2 was able to detect 7/11 (64%) validated differentially expressed known and novel miRNAs and did not detect the only available false positive miRNA from the original study.

Of note, Oasis 2' PCA analysis highlights a potentially mis-annotated Psoriasis sample and another outlier sample (Fig. 3A). Removal of these two samples (Fig. 3B) increased the number of significantly (adjusted p-value < 0.1) DE miRNAs from 195 to 256 cases. We would like to emphasize that this data was already analyzed in two publications and to our knowledge this is the first time that these 'problematic' samples were detected, providing strong evidence for the utility of Oasis 2' QC plots.

Renal cancer data

In this work 11 renal cancer and 11 remission samples [12] were analyzed. This is longitudinal data from 11 patients and as such paired but we were unable to extract the pairing information from the GEO database annotations. Therefore the data was analyzed with Oasis 2 in un-paired mode and compared to the published, paired analysis with edgeR [14]. Despite of these technical issues the two analyses showed high overlap. Oasis 2 found 150 DE miRNAs (adjusted p-value < 0.1) whereas the original publication lists only 70 DE miRNAs. Of these 70 DE miRNAs 53 (76%) could also be found in the significant Oasis 2 miRNAs (Table 4). Of note, with

Table 4 Overlap of differentially expressed sRNAs using three datasets

	Statistic[1]	Overlap[2]	Validated overlap[3]	FP overlap[4]
AD	Wilcoxon-Mann-Whitney	60%	75%(6/8)[5]	0% (0/2)
Psoriasis	Pearson's chi-squared	73%	64% (7/11)	0% (0/1)
Renal Cancer	edgeR [14]	76%	80% (4/5)	NA
Schizophrenia	DESeq2 (Dejian et al., 2015)	41%	67%(2/3)	0% (0/1)

[1]Oasis 2 uses a negative binomial distribution as basis for its statistical evaluation of the differential expression. A very similar approach is taken by the edgeR package that has been used in the Renal Cancer study. The Psoriasis data was analyzed using a Pearson's chi-squared test and the AD dataset was analyzed using the non-parametric Wilcoxon-Mann-Whitney test. Schizophrenia dataset used the same approach like Oasis 2. [2]Overlap of differentially expressed miRNAs comparing Oasis 2's results to published data. The percentage is calculated in reference to the shorter DE list. [3]Overlap of differentially expressed miRNAs that have been validated independently in addition to the sRNA-seq experiment. [4]False positive (FP) differentially expressed miRNAs detected by Oasis 2. [5]Only known validated DE miRNAs are considered

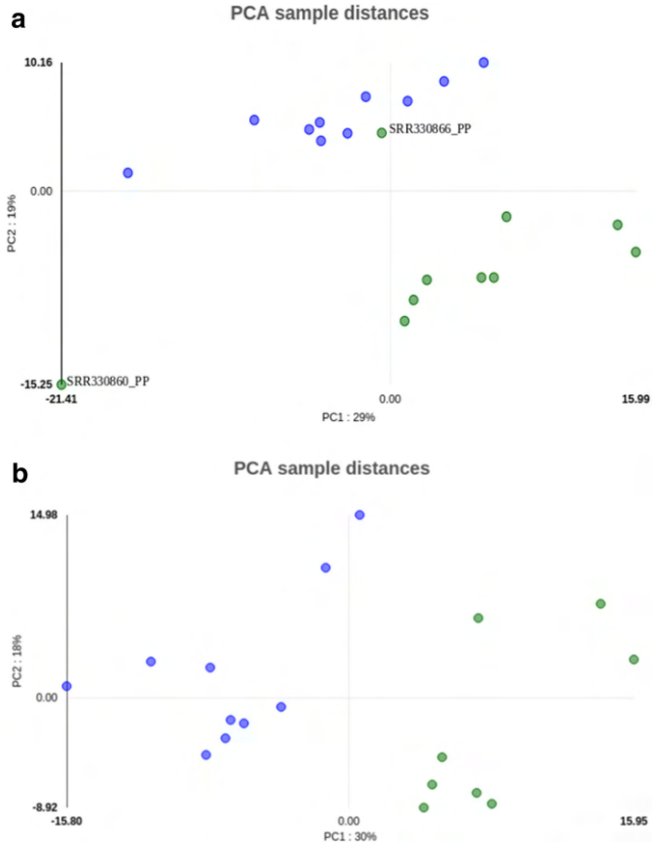


Fig. 3 Oasis 2' (QC) outlier detection: To assess the QC of Oasis 2 and its biological relevance, sRNA Psoriasis data (demo dataset) was analyzed. PCA sample distances of psoriasis (green) and control (blue) is shown. (**a**) PCA of psoriasis and control samples showing a potentially mis-annotated (SRR330866_PP) and an outlier sample (SRR330860_PP). (**b**) PCA of psoriasis and control samples without misclassified/outlier samples. Removal of these two samples increased the number of significantly (adjusted *p*-value < 0.1) DE miRNAs from 195 to 256 cases and increased the AUC from 0.9 to 1 in the classification module, providing strong evidence for the utility of Oasis 2' QC plots

the exception of miR-122 all the validated miRNAs from the original work were detected using Oasis 2 (miR-21-5p, miR-210-3p, miR-199, miR-532-3p).

Schizophrenia and schizoaffective disorder data

In this experiment induced pluripotent stem cells were used to study neuropsychiatric disorders associated with 22q11.2 microdeletions [3]. Controls and patients with 22q11.2 microdeletions diagnosed with a psychotic disorder were compared (9 controls and 7 patients). Oasis 2 found 34 DE miRNAs (adjusted p-value < 0.1) whereas the original publication identified 45 DE miRNAs. Of these 45 DE miRNAs 14 (41%) were also detected as differentially expressed by Oasis 2 (Table 4). In the original publication four miRNAs were validated by qPCR, two significantly up-regulated (miR-23a-5p and miR-146b-3p), one significantly down-regulated (miR-185-5p), and a miRNA that showed no difference in expression (miR-767-5p). Oasis 2 was able to confirm 2/3 (67%) validated differentially expressed miRNAs (miR-23a-5p and miR-185-5p) and did not confirm 1/1 (100%) false positive miRNAs miR-767-5p.

Overall, Oasis 2 detected 19/27 (70%) independently validated DE miRNAs in the published datasets despite of the different statistical approaches and miRBase versions used (Table 4). Detailed analysis results are accessible in Oasis 2's 'Demo Data' webpage. Our results provide strong evidence that Oasis 2 provides biologically meaningful results to the end user.

Pathogen detection and sample classification

To assess the performance of the pathogen detection we analyzed 5 datasets with known viral or bacterial infections (Additional file 1: Table S6). We calculated the precision, recall, and F-score for the detection of the particular pathogen strain in the dataset while considering only the top ranking, first two, three, and up to the first ten reported species (Fig. 2). Species were ordered based on the number of read counts. In general, the viral or bacterial species and strains were detected with high precision and recall, reaching F-scores of ~ 0.8 when the top five viral and bacterial species were considered. In consequence, Oasis 2 currently reports the top five bacterial, archaeal, and viral species found, allowing for the detection of potential infective agents or the discovery of experimental sample contaminations.

To benchmark the improved classification routine, we compared the performance of the old Oasis classification module (unbalanced sampling with all variables) to the new Oasis 2 classification module using balanced sampling and feature optimization using three demo datasets (see Detection and Differential Expression of sRNAs and Additional file 1: Figure S2). From a theoretical perspective, balanced sampling should increase prediction accuracy only in the case of class imbalances. In consequence, the novel classification module enhances the AUC for the imbalanced AD (22 controls, 48 patients) demo dataset by 2% (old AUC 0.95, new AUC 0.97), while it marginally changes classification performance for the balanced Psoriasis (10 control and 10 Psoriasis samples) (old AUC 0.90, new AUC 0.91) and Renal carcinoma (11 control and 11 cancer samples) (new and old AUC 1.00) data. Feature pruning should be crucial when a dataset contains a lot of uninformative features and very few informative features. To this end we have taken an unpublished dataset (6 controls, 6 treatments) that contains at least one feature that perfectly separates the two classes but otherwise contains mostly uninformative features. Whereas the old classification module reaches an AUC of 0 on this dataset, the new module reaches an AUC of 0.833.

Moreover, we also compared the accuracy of the new Oasis 2 classification module on the AD dataset to the published accuracy in the original manuscript [10]. Unfortunately, we were unable to obtain the primary output of the SVM and could not follow the post-processing steps of the machine learning results as performed in the original publication (e.g. removal of miRNAs that also occur in other diseases). In brief, the original publication provides a biomarker signature of 12 miRNAs (10 annotated and two novel) that reaches an average accuracy of 80%. The Oasis 2 classification reaches an accuracy of ~ 87% (AUC of 0.97) using 320 features (no preprocessing for other diseases) and has an out-of-bag error of ~ 10%. Two miRNAs in the original paper list (has-miR-151a-3p, hsa-let-7f-5p) were also found in the top 10 features (miRNAs) obtained with Oasis 2 classification.

The classification analysis of the three demo datasets (see 3.1) yielded stable and robust biomarker predictions that further corroborated the quality of the enhanced classification module.

Runtime estimates

We next estimated the runtime of Oasis 2 using the above-mentioned AD, Psoriasis, and Renal cancer datasets and compared the results to runtime estimates for omiRas, mirTools 2.0, MAGI, Chimira and sRNAtoolbox, five recently developed web applications for the analysis of sRNA-seq data (Table 3, Additional file 1: Table S7). Performances of the sRNA Detection, DE Analysis, and Classification modules were measured on the Oasis 2 server. For benchmarking the Oasis 2 runtime we compared it to the runtime estimates of the above-mentioned web applications by submitting the AD, Psoriasis, and Renal Cancer datasets to the respective services (Table 3). Of note, runtime estimates for MAGI were taken from the MAGI webpage, which we assume constitutes a 'best case scenario' in favor of MAGI (low server analysis load). In addition, we could not compare to mirTools 2.0 as the maximum upload file size is limited to 30 Mb. Furthermore, the sRNAtoolbox web application was also not accessible during the period of testing and writing this manuscript.

Overall, Oasis 2 is significantly faster than MAGI, Chimira, and omiRas. For the smallest dataset (Renal Cancer) Oasis 2 was ~ 1.5 times faster than Chimira, ~ 15 times faster than MAGI, and ~ 18 times faster than omiRas. While the runtime differences between Oasis 2 and Chimira were rather small when only few samples were analyzed, Oasis 2 was ~ 2 times faster than Chimira, ~ 30 times faster than MAGI for the 48 Gb Psoriasis dataset. Unfortunately, we were unable to estimate the runtime of omiRas for the Renal Cancer dataset since it did not finish file upload. Oasis 2 analyzed the largest dataset (AD, 287 Gb) in 8 h31m50s while none of the other tools mentioned above supported the analysis of the AD samples. In summary, Oasis 2 is the fastest of the state-of-the-art web applications we could compare to and has no restrictions on the sample number or size.

Conclusions

Oasis 2 is fast, reliable, and offers several unique features that make it a valuable addition to the ever-growing number of sRNA-seq analysis applications. Especially the analysis support for all organisms, the detection and storage of novel miRNAs, the differential expression and classification modules, and the interactive results visualization supporting GO and pathway enrichment analyses enable

biologists and medical researchers to quickly analyze, visualize, and scrutinize their data. Oasis 2 also offers rich per experiment and per sample quality control, which might be one of the most important steps in the initial data analysis. The utility of a good quality control is exemplified in the analysis of the Psoriasis dataset, which seems to contain a mis-labelled (SRR330866_PP) and an outlier (SRR330860_PP) sample (Fig. 3). The removal of the outlier and mis-labelled samples in the Psoriasis dataset increased the number of significantly DE miRNAs from 195 to 256 cases and increased the classification accuracy for the same dataset from AUC of 0.9 to 1. We would like to emphasize that this data was already analyzed in two publications and to our knowledge this is the first time that these 'problematic' samples were detected, providing strong evidence for the utility of Oasis 2' QC plots. Additionally the modular structure of Oasis 2 (sRNA detection, DE and classification) makes this task even easier, as the user can run only DE (without outliers) rather than going through the sRNA detection step again. In addition Oasis 2 provides PDF and video tutorials that explain its usage and details on how to interpret its results. Future developments will include the detection of small RNA editing, modification, and mutation events as well as more detailed reports on bacterial and viral infections and contaminations.

Additional file

Additional file 1: Oasis2-Suppl-Material.docx: This file contains supplementary material and figures as well. (DOCX 125 kb)

Acknowledgements
We would like to thank Ashish Rajput, Ting Sun, Vikas Bansal, Michel Edwar Mickael, the DZNE IT, and all of the Oasis users for helpful suggestions.

Funding
This work was supported by the DFG (BO4224/4–1), the Network of Centres of Excellence in Neurodegeneration (CoEN) initiative, the Volkswagen Stiftung (Az88705), iMed – the Helmholtz Initiative on Personalized Medicine, and the BMBF grant Integrative Data Semantics in Neurodegeneration (031L0029B, IDSN).

Availability of data and materials
Oasis 2 freely available at https://oasis.dzne.de. Oasis 2' demo data is available at https://oasis.dzne.de/small_rna_demo.php. Additional datasets mentioned and analyzed in this article can
GSE46579
https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE46579
GSE31037
https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE31037
GSE37616
https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE37616
GSE59944
https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE59944
GSE65752
https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE65752
GSE31349
https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE31349
GSE33584
https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE33584
GSE72769
https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE72769

Authors' contributions
SB initiated the study and designed the web application as well as analyses together with RR. RR and AG designed the Oasis-DB to store novel predicted miRNA. MF enhanced the classification module. JB and VC worked on the backend implementations of different modules. AS analyzed sRNA-seq data on different web servers to benchmark Oasis 2. DSM and OS worked the interactive user interface and tutorials. All authors read and approved the final manuscript.

Ethics approval and consent to participate
N/A

Consent for publication
N/A

Competing interests
The authors declare that they have no competing interests.

Publisher's Note

Author details
[1]Laboratory of Computational Systems Biology, German Center for Neurodegenerative Diseases, Göttingen, Germany. [2]Institute of Medical Systems Biology, Center for Molecular Neurobiology, University Clinic Hamburg-Eppendorf, Hamburg, Germany. [3]German Center for Neurodegenerative Diseases, Tübingen, Germany.

Received: 25 August 2017 Accepted: 29 January 2018
Published online: 14 February 2018

References
1. Beckers, et al. Comprehensive processing of high-throughput small RNA sequencing data including quality checking, normalization, and differential expression analysis using the UEA sRNA Workbench. RNA. 2017;823–35.
2. Capece V, et al. Oasis: online analysis of small RNA deep sequencing data. Bioinformatics. 2015;31:2205–7.
3. Dejian, et al. MicroRNA Profiling of Neurons Generated Using Induced Pluripotent Stem Cells Derived from Patients with Schizophrenia and Schizoaffective Disorder, and 22q11.2 Del. plosone. 2015.
4. Franceschini, et al. STRING v9.1: protein-protein interaction networks, with increased coverage and integration. Nucleic Acids Res. 2013;D808-15.
5. Friedländer MR, et al. MiRDeep2 accurately identifies known and hundreds of novel microRNA genes in seven animal clades. Nucleic Acids Res. 2012; 40:37–52.
6. Huang, et al. The DAVID Gene Functional Classification Tool: a novel biological module-centric algorithm to functionally analyze large gene lists. Genome Biol. 2007;8:R183.
7. Joyce CE, et al. Deep sequencing of small RNAs from human skin reveals major alterations in the psoriasis miRNAome. Hum Mol Genet. 2011;20: 4025–40.
8. Kim J, et al. MAGI: a node.Js web service for fast microRNA-Seq analysis in a GPU infrastructure. Bioinformatics. 2014;30:2826–7.
9. Kuhn, et al. STITCH 4: Integration of protein-chemical interactions with user data. Nucleic Acids Res. 2014;D401-7.
10. Leidinger P, et al. A blood based 12-miRNA signature of Alzheimer disease patients. Genome Biol. 2013;14:R78.
11. Müller, et al. omiRas: a Web server for differential expression analysis of miRNAs derived from small RNASeq data. Bioinformatics. 2013;2651–2.
12. Osanto S, et al. Genome-wide microRNA expression analysis of clear cell renal cell carcinoma by next generation deep sequencing. PLoS One. 2012;7
13. Reimand, et al. G:Profiler - A web server for functional interpretation of gene lists (2011 update). Nucleic Acids Res. 2011;W307-15.
14. Robinson MD, et al. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. Bioinformatics. 2010;26: 139–40.

15. Rueda, et al. sRNAtoolbox: an integrated collection of small RNA research tools. Nucleic Acids Res. 2015;W467–W473.
16. Sun, et al. CAP-miRSeq: a comprehensive analysis pipeline for microRNA sequencing data. BMC Genomics. 2014;15:423.
17. Vitsios DM, Enright AJ. Chimira: analysis of small RNA sequencing data and microRNA modifications. Bioinformatics. 2015;31:3365–7.
18. Witwer KW. Circulating MicroRNA biomarker studies: pitfalls and potential solutions. Clin Chem. 2014;000
19. Wood DE, Salzberg SL. Kraken: ultrafast metagenomic sequence classification using exact alignments. Genome Biol. 2014;15:R46.
20. Wu, et al. mirTools 2.0 for non-coding RNA discovery, profiling, and functional annotation based on highthroughput sequencing. RNA Biol. 2013;1087–92.
21. Zuberi, et al. GeneMANIA prediction server 2013 update. Nucleic Acids Res. 2013;W115-22.

4 Unpublished

4.1 BED.AI

4.1.1 Introduction

The "central dogma" of biology is that DNA undergoes transcription on the antisense strand (3' to 5') and then the mRNA is translated into protein. Therefore, the sense strand (5' to 3') shares the same sequence as mRNA where thymine is replaced with uracil. As DNA is complementary, when speaking of a gene in general the sense of its nucleotide sequence is seemingly inconsequential; given just the index and chromosome of a sequence, however, for the bioinformatician knowing which strand can yield vastly different results. Furthermore, with 6.4 billion basepairs storing the raw genomic sequences as strings of text is excessively inefficient. For these reasons and in support of the University of California Santa Cruz (UCSC) Genome Browser, the Browser Extensible Data (BED) file format was developed.

BED is a versatile file format that follows a tab separated value (TSV) structure; BED's versatility stems from the number of columns (and sub-sequentially tabs separating the columns). At minimum BED requires the first three columns (BED3). However another common variant is BED6, which includes the strand's direction. Note that the genome is not specified in the format; however it could be included in the optional header line's description field. A key benefit of the BED file format is that by specifying only the chromosome, start, and stop position many sequences can be stored in a single file in a compressed form rather than all of the raw sequences.

chr7	127471196	127472363	Pos1	0	+
chr7	127472363	127473530	Pos2	0	+
chr7	127473530	127474697	Pos3	0	+
chr7	127474697	127475864	Pos4	0	+
chr7	127475864	127477031	Neg1	0	-
chr7	127477031	127478198	Neg2	0	-
chr7	127478198	127479365	Neg3	0	-
chr7	127479365	127480532	Pos5	0	+
chr7	127480532	127481699	Neg4	0	-

Table 4.1: Example of a BED file.

column	meaning
chrom	name of chromosome
chromStart	starting position of feature
chromEnd	ending position of feature
name	name of feature
score	$s \in [0, 1000]$, greyness of feature
strand	$s \in \{., +, -\}$
thickStart	where to draw bold
thickStop	where to end boldness
itemRgb	color of feature
blockCount	number of blocks (exons) in feature
blockSizes	csv of block sizes
blockStarts	csv of block start positions

Table 4.2: The BED file format's column names and meanings.

The raw sequences can be looked up later with a supporting tool such as the Quinlan Lab's BEDTools. An overview of the BED file format and its columns meanings are outlined in table 4.2. An example of the BED6 file format can be found below (table 4.1):

When recovering the nucleotides sequences specified in a BED file, the raw nucleotides sequences are often returned in the FASTA file format. While many are accustomed to the standard nucleic acid codes A, C, T, G, and U, the FASTA file supports ambiguity up to and including any nucleic acid (N) or a gap of unknown length (-). A full breakdown of the FASTA file format can be found in table 4.3. In addition to the character codes for nucleic acids, capitalization matters for the FASTA file format. The character "N" (any nucleic acid) can be thought of as hard masking, however a soft masking exists with lower case letters. Tandem repeats, sequences of one or more nucleotides repeated directly adjacent to one another, e.g. "attcg attcg attcg", are represented with lower case letters. The UCSC uses both Tandom Random Finder and RepeatMasker, the latter of which masks up 56% of the human genome.

Nucleic Acid Code	Meaning	Mnemonic
A	A	Adenine
C	C	Cytosine
G	G	Guanine
T	T	Thymine
U	U	Uracil
R	A or G	puRine
Y	C, T or U	pYrimidines
K	G, T or U	bases which are Ketones
M	A or C	bases with aMino groups
S	C or G	Strong interaction
W	A, T or U	Weak interaction
B	not A (i.e. C, G, T or U)	B comes after A
D	not C (i.e. A, G, T or U)	D comes after C
H	not G (i.e., A, C, T or U)	H comes after G
V	neither T nor U (i.e. A, C or G)	V comes after U
N	A C G T U	Nucleic acid
-	gap of indeterminate length	

Table 4.3: FASTA nucleotide encoding

This cursory overview of BED and FASTA files is requisite for discussing how a sequence of nucleotides might be encoded for a machine learning algorithm. An option is one-hot encoding, whereby the sequence is represented as a matrix. Each element (column) represents a nucleotide and each channel (row) represents an indicator e.g. "A" or 'T." If uracil is replaced with thymine to be DNA / mRNA agnostic then one could encode an FASTA sequence of n nucleotides in a $4 \times n$ matrix, where four are rows A, C, T and G. If the FASTA sequence includes a character like "B," then that column could be represented either as $[0, 1, 1, 1]$ or $[0, 0.33, 0.33, 0.33]$ whereby the former is multilabeling and the latter is soft labeling of the encoded data. Additional information can be tacked on to this encoding by the addition of rows e.g. a row to specify whether or not the nucleotide belongs to a tandem repeat or a row to specify whether or not the nucleotide belongs to the sense or antisense strand. See table 4.4 for an example of how FASTA characters might be encoded.

Given a FASTA sequence each nucleotide can belong to a suite of feature classes of biological relevance e.g. exon, intron, cpg-island, promoter, binding site, cleavage site, etc. Just as a FASTA sequence can be encoded as a matrix, each nucleotide (column) can be labeled to a class (row). For example, a matrix demonstrating the labeling of a sequence containing a exons (row 1), an intron (row 2), and an "other" class (row 3) can be found in table 4.5.

	FASTA Character																
Channel	A	C	G	T	U	R	Y	K	M	S	W	B	D	H	V	N	-
A	1	0	0	0	0	1	0	0	1	0	1	0	1	1	1	1	0
C	0	1	0	0	0	0	1	0	1	1	0	1	0	1	1	1	0
T	0	0	0	1	0	0	1	1	0	0	1	1	1	1	0	1	0
G	0	0	1	0	0	1	0	1	0	1	0	1	1	0	1	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	a	c	g	t	u	r	y	k	m	s	w	b	d	h	v	n	-
A	1	0	0	0	0	1	0	0	1	0	1	0	1	1	1	1	0
C	0	1	0	0	0	0	1	0	1	1	0	1	0	1	1	1	0
T	0	0	0	1	0	0	1	1	0	0	1	1	1	1	0	1	0
G	0	0	1	0	0	1	0	1	0	1	0	1	1	0	1	1	0
R	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

Table 4.4: FASTA encoding matrix when tandem repeats are included as a channel ("R"). Uracil (U) is excluded as a channel.

	Nuleotide Label									
Channel	0	1	2	3	4	5	6	7	8	9
(1) Exon	0	1	1	1	0	0	1	1	1	1
(2) Intron	0	0	0	0	1	1	1	1	0	0
(3) Other	1	0	0	0	0	0	0	0	0	0

Table 4.5: How a FASTA sequence containing two exons and an intron might be encoded as a matrix.

With a matrix encoding for both the FASTA sequence (e.g. table 4.4) and its sequence features (e.g. table 4.5) one can utilize artificial intelligence to predict the "ground truth" (the labeled sequence features). An example of which can be found in figure 4.1.

While on the topic of how to evaluate a predicted label matrix to the ground truth may appear straight forward, multilabel metrics are more complicated. Additionally, how does one take into account not just the per-nucleotide error but sequence error as a whole? If a tuple of just the start and stop indices of a sequence from such a matrix can represent the local sequence (similar to BED3 columns' two and three), then figure 4.2 shows how minor errors in nucleotide prediction can balloon into major sequence prediction errors.

Notice that while for channel one (exons) two sequence are preserved, both exons are truncated and the latter has no direct alignment (a prediction index that matches an index of the exon channel in the true objects). As for channel two (introns) a single intron is split into two smaller introns. Here both introns have a direct alignment; however, while it is clear this is incorrect on a multitude of levels what is the error?

```
# ground "truth"
[
  #0     1     2     3     4     5     6     7     8     9      # position
  [0,    1,    1,    1,    0,    0,    1,    1,    1,    1],    # channel 1
  [0,    0,    0,    0,    1,    1,    1,    1,    0,    0],    # channel 2
  [1,    0,    0,    0,    0,    0,    0,    0,    0,    0],    # channel 3
]

# prediction
[
  #0     1     2     3     4     5     6     7     8     9      # position
  [0.11, 0.71, 0.98, 0.95, 0.20, 0.15, 0.81, 0.82, 0.95, 0.86], # channel 1
  [0.13, 0.17, 0.05, 0.42, 0.92, 0.89, 0.93, 0.93, 0.67, 0.21], # channel 2
  [0.99, 0.33, 0.20, 0.12, 0.15, 0.15, 0.20, 0.01, 0.02, 0.13], # channel 3
]


# binary mask with cutoff 0.9
[
  #0     1     2     3     4     5     6     7     8     9      # position
  [0,    0,    1,    1,    0,    0,    0,    0,    1,    0],    # channel 1
  [0,    0,    0,    0,    1,    0,    1,    1,    0,    0],    # channel 2
  [1,    0,    0,    0,    0,    0,    0,    0,    0,    0],    # channel 3
]
```

Figure 4.1: The demonstration label matrix from table 4.5, showing how a model might predict this multilabel problem and how a binary mask can clarify the results.

```
# "detected" objects
[
  [[2, 3], [8, 8]],  # channel 1
  [[4, 4], [6, 7]],  # channel 2
  [[0, 0]]           # channel 3
]

# true objects
[
  [[1, 3], [6, 9]],  # channel 1
  [[4, 7]],          # channel 2
  [[0, 0]]           # channel 3
]
```

Figure 4.2: The sequence objects from masked label matrix of figure 4.1 demonstrating how a model seemingly slight errors can greatly affect sequence annotation.

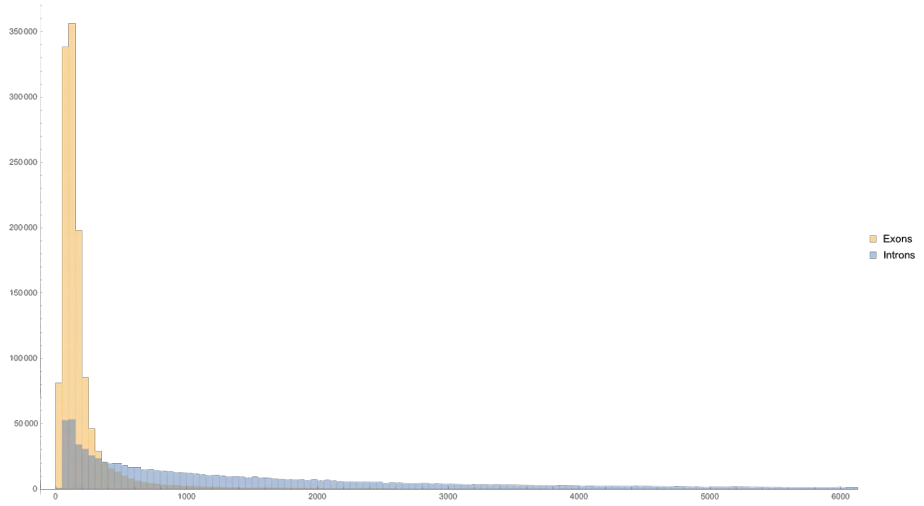


Figure 4.3: Distribution of two DNA sequence features (exon and intron) length. The vast majority of exons and introns are in the hundreds of nucleotides length; however, macro exon and introns exist well beyond the truncated x-axis.

Is it the average of the errors for each sequence objects? if there where three true introns and seven predicted ones, is alignment of the predicted intron to the start, stop, or center of mass of the sequence? What if a predicted intron is evenly split between two true introns in terms of alignment? What is the error when no object is predicted when at least one (if not many) are expected?

These issues are compounded when factoring in the complexity of the sequence feature domain. For example, DNA sequence features start at the micro - two nucleotides - to the "macro" - hundreds of thousands of nucleotides. Figure 4.3 shows the heavy tail the length distribution some sequence features (exons and introns) can have. Further, when sequence features are encoded in a matrix and their dimensions are reduced - via t-SNE - distinct feature classes may appear ambiguous due to the encoding (figure 4.4 [23, 22]).

The importance of identifying and understanding these sequence features is readily apparent. Consider exons, which are the protein-coding regions of genes; as such, mutations therein can have notable effect on the gene's encoded protein's function. Even a single nucleotide polymorphism (SNP) can manifest as sickle cell disease.

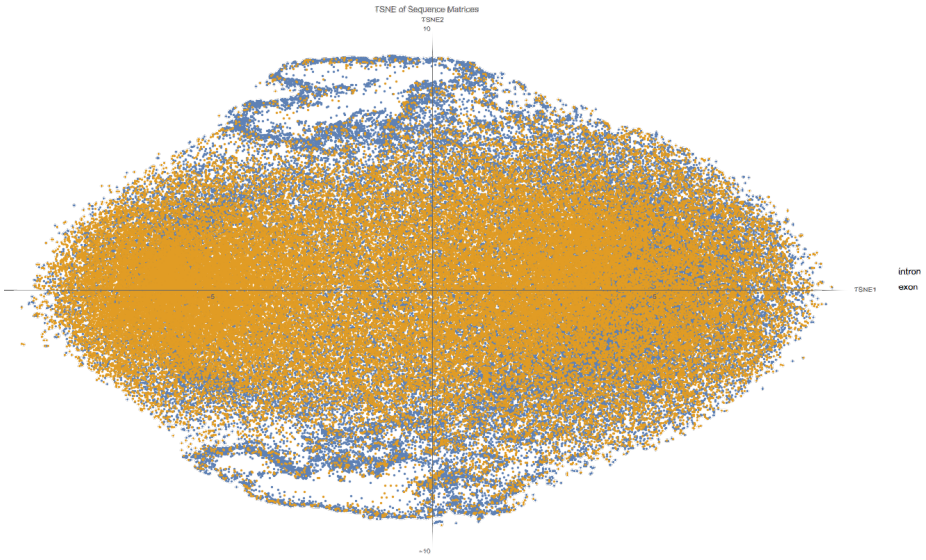


Figure 4.4: t-SNE of exons and introns with length in the range of 50 to 250 nucleotides. While some introns appear to stand out, the vast majority of exons and introns appear homogeneous.

Further, the proportion of gene isoforms or the use of particular exons in the protein product is implicit in diseases such as Timothy syndrome, cardiomyopathy and cancer [49, 50, 51, 52]. Given that 95% to 100% of genes partake in alternative splicing - the modular use of exons - to modify their protein product, with some genes having more than $38,000$ isoforms, it is relevant to know and keep track of their constituent exons [53, 54, 55].

Interestingly, RNA sequencing (RNA-seq) data, which allows for the analysis of both differentially expressed exons (DEEs) and differentially expressed genes (DEGs) in specific contexts, is not fully utilized. As the technology for RNA-seq becomes cheaper, there number of RNA-seq studies continues to rise with more than $27,000$ studies by 2015. Despite both analyses being possible from the same data, the ratio of DEE to DEG studies is about 0.23% (from a National Center for Biotechnology Information, NCBI, search), demonstrating the lack of attention in elucidating the role of exons in pathologies. The preferential choice of DEG to DEE analyses may stem from a lack of annotations to facilitate downstream analysis.

In late 2017 almost an order of magnitude (from $\sim$ 160 to $1,399$) of mutually exclusive exons (MXEs), of which 47% are novel, unannotated exons were reported

[56]. Further, these MXEs are significantly enriched in pathogenic, disease-causing mutations. This work indicates that there may be many more unannotated exons in the human reference genome (hg38). A conservative 10% increase would result in a novel exon for almost every gene [57].

Although efforts to predict various attributes about (specific) exons, e.g. exon-intron structure, epigenetic signatures, splicing patterns, etc have been made, current exon identification / prediction methods are done in the context of gene (structure) prediction thereby requiring prior knowledge of the gene [58, 59, 60, 61, 62, 63, 64, 65, 66, 67]. Hence they do not necessarily lend themselves towards novel identification or generalization to non-exonal features. Thus it is of interest to create a model for sequence feature identification ab initio so that it can be adapted to any genome and any feature of interest.

Neural networks have been shown to produce state-of-the-art results in semantic segmentation, whereby an image's pixels are relegated to their constitute objects [68, 69, 70, 71, 72, 73, 74, 75]. As eluded to earlier with the matrix representation of a FASTA sequence, FASTA sequences can equivalently be perceived as a wide binary image. Thus, one could apply semantic segmentation to a binary image of a FASTA sequence with the aim of predicting and identifying the sequence features of interest. Herein we present BED.AI, a residual CNN with inception for the semantic segmentation of FASTA sequences to predict sequence features (namely, exons and introns).

4.1.2 Methods

4.1.2.1 Data Acquisition

Human Genome The human reference genome from December 2013 (GRCh38/hg38) was downloaded from the USCS Genome Browser by navigating to the "Downloads" dropdown, clicking "Genome Data", then under "Sequence and Annotation Downloads" clicking "Human", moving to the section "Human genome" and clicking "Full

field	value
clade	Mammal
genome	Human
assembly	Dec. 2013 (GRCh38/hg38)
output format	BED - browser extensible data
file type returned	gzip compressed
genome	genome

Table 4.6: Values to extract exonic regions.

dataset" under Dec. 2013 (GRCh38/hg38) and downloading the file "hg38.fa.gz". In addition, the corresponding chromosome sizes were also retrieved at the same location by downloading the file "hg38.chrom.sizes".

Exonic Regions A BED file of the known exonic regions was retrieved as follows: 1. Navigate to the USCS Genome Browser home page. 2. From the navigation bar click "Tools." 3. From the drop-down click "Table Browser." 4. Selecting the following fields with the corresponding values: 5. Set "group" to "Genes and Gene Predictions." 6. Set "track" to "All GENCODE V28." 7. Set "table" to "Comprehensive (wgEncodeGencodeCompV28)." 8. Set "output file" to "encode_gencode_comp_v28_exons.bed." 9. Click the button "get output." 10. Under "create one BED record per" select "Exons." 11. Ensure that there will be "plus 0 bases at each end." 12. Click the button "get BED."

Intronic Regions A BED file of the known intronic regions was retrieved as follows: 1. Navigate to the USCS Genome Browser home page. 2. From the navigation bar click "Tools." 3. From the drop-down click "Table Browser." 4. Selecting the following fields with the corresponding values: 5. Set "group" to "Genes and Gene Predictions." 6. Set "track" to "All GENCODE V28." 7. Set "table" to "Comprehensive (wgEncodeGencodeCompV28)." 8. Set "output file" to "encode_gencode_comp_v28_introns.bed." 9. Click the button "get output." 10. Under "create one BED record per" select "Introns." 11. Ensure that there will be "plus 0 bases at each end." 12. Click the button "get BED."

field	value
clade	Mammal
genome	Human
assembly	Dec. 2013 (GRCh38/hg38)
output format	BED - browser extensible data
file type returned	gzip compressed
genome	genome

Table 4.7: Values to extract intronic regions.

Genome 96 For evaluation the evaluation set from the "Evaluation of gene structure prediction programs" by M. Burset and R. Guigó is used. The 570 sequences are first downloaded from http://genome.crg.es/datasets/genomics96/seqs/DNASequences.-fasta, and then encoded and labeled using the same processing techniques as described above.

4.1.2.2 Data Transformation

After the data was collected from USCS data browser (see sections 4.1.2.1, 4.1.2.1, and 4.1.2.1), the BED files were shared by chromosome and strand using the ParPar python package. Then each of these subfiles were filtered using the bedpy python package for those that lie on chromosomes 1 through 22. Utilizing the python package lrng the overlapping regions of similar sequence features (e.g. exon overlapping with exon) were combined to produce a smaller reference file.

With the known "classed" regions, a third BED file was produced corresponding to the nucleotides which are neither exonic or intronic (the complement of the nucleotides in the union of the prior two files). These regions are referred to as "non-class." There were only a few non-class regions per chromosome and their length were far greater than that of the classed regions. Therefore each non-class region was segmented into non-overlapping chunks of random length between 450 and 1,000 nucleotides. In addition to the non-class segmented BED file, a BED file of random 300 nucleotide sequences was also produced. As the model architecture uses a fixed window size, each sequence is padded on both sides by the window length; subse-

quently these sequences are partitioned into training, validation and test files. Per each BED file 650,000 regions were selected, from which each sequence was designated as training, validation, and test sets at 70%, 20%, and 10% ratios respectively i.e. 195,000 randomly selected sequences or padded exons, introns, and "other" comprise the test set.

With the data filtered, padded, and partitioned, the sequences are then prepared for TensorFlow 2.0. First the FASTA sequences are extracted from the genome using bedtools. Then each FASTA sequence is embedded as a tensor using the python package ntai. As the BED sequences were padded, using the reference file each FASTA sequence is labeled. Thereafter the labels are also converted to tensors. Finally, using the fio python package all sequence features are combined and converted into TFRecords.

These steps are illustrated in supplementary figures 4.11 a and 4.11 b.

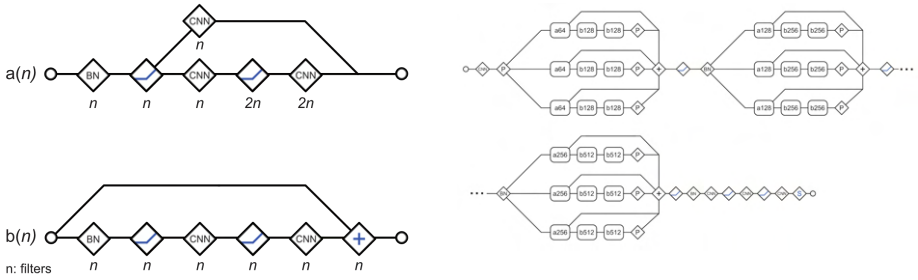
4.1.2.3 Model Output

For sequences longer than the window size of the model, their output can be compiled as follows. First a step size, k, is determined (by default we use 25). Then overlapping regions of the sequence are feeding into the model for every k nucleotides. The predictions, per nucleotide, are averaged to create a merged prediction for each of the subsequences. Thereafter a binary mask is set where every value at or above m are set to 1 (by default we used 0.5. Given a gap tolerance, g, disjoint segments in the output with a distance less than or equal to g nucleotides are merged. By default we used $g = 0$. With the features extracted from the output, the post-processed tensors are converted to a BED file. These steps are visualized in supplementary figures 4.13 a through 4.13 b.

Figure 4.5: BED.AI model architecture

(a) Branch types

(b) BED.AI Inception Towers



4.1.2.4 Architecture

BED.AI's employees an inception residual CNN, where the inception kernels are 3, 25, and 75. The architecture is inspired from ademxapp as used in "Wider or Deeper: Revisiting the ResNet Model for Visual Recognition" by Zifeng Wu, Chunhua Shen, and Anton van den Hengel [74].

4.1.2.5 Training and Evaluation

The model was trained using a DGX-1 for 300 epochs with a batch size of 256 sequences. Since the sequences were extended to include the neighboring ±window-sized number of nucleotides, per batch a random window-sized subsegment of the sequence is used. Of the total 1,950,000 preprocessed sequences, 1,365,000 were used for training, 390,000 for validation and 195,000 for evaluation.

4.1.2.6 Web Application

Details and a demonstration of BED.AI is available for use at http://bedai.ims.bio/ (see figure 4.7). This lightweight web application allows users quick reference to a cursory summary of what BED.AI is; where the source code for it can be found; links to step-by-step documented tutorials regarding the code; quick links to where and

how the source data was obtained; as well as an interactive demonstration of the input and output specifications in relation to hg38. The application was developed utilizing v-focal, apoll, sfo, mag, ntai, sil, lrng, parpar, bedpy, and the fio packages (see table 1.1).

Production of BED.AI's web application was facilitated by isolating its individual components and dockerizing them. Specifically three core containers are in use. First, there is the container for the trained model based on TensorFlow's production serving API which receives the input tensors and returns the output. Second a lightweight Flask application is deployed to handle the interplay from the client's input request and the TensorFlow containers output via a Redis queuing schema. Third, the client side web application is built using a modern component-based library: specifically Vue. The client side application polls the Flask container for results until they are given to allow for the user to be notified as results come in (in the case of batched input). An overview of this three tiered system can be found in figure 4.8 and in more detail at section 1.2.2.

4.1.3 Results

Evaluation of BED.AI takes place over two datasets. The first is a dataset of vertebrate genes described by Burset & Guigo in *Genomics* 1996 [76]. It is often used in comparison between HMM gene structure models. The later comprises the test set of sequences extracted from hg38. Table 4.8 describes the multilabel facets of the test set.

For model comparison both the HMM-based models GenScan and HMMGene are utilized [77, 78]. Comparatively, especially when parameter size is factored into account, BED.AI has lackluster results at best for sequence identification on the Genome96 dataset (see table 4.9). Yet, without expertise curated parameters, curiously outperforms at the nucleotide level. The seeming paradox of performance is well encapsulated in figure 4.10. As mentioned in the introduction, a single nucleotide

Multilabel Metric	Value
Carnality	1.0215
Density	0.3405
Diversity	7

Table 4.8: The multilabel metrics of the hg38 test dataset. Label cardinality is the average number of labels per nucleotide. Label density reflects the average of the total labels in a sequence divided by the total number of labels. Note that diversity is 7 rather than 8 (2^3) as the label "other" is exclusive. Collectively these metrics reflect that the multi label case occurs; however such occurrences is rare. Part of this rarity is inflated due to the exclusive label class "other."

break in classification - metric depending - can decimate a model's performance score. As the go-to sequence metric is exon accuracy which has a definition dependent on the number of correctly predicted exons, models emphasizing at nucleotide accuracy suffer. Especially, when directly compared to HMMs that emphasize state changes (e.g. exon to intron).

Dataset	Model	Subset Acc	NT Acc	ESn	ESp	Exon Acc
Genome96	GenScan	-	92	78	81	80
Genome96	HMMGene	-	92	81	83	82
Genome96	BED.AI	89.1	**93**	0.64	0.29	0.47
HG38	BED.AI	77.8	86.2	22.73	7.97	15.35

Table 4.9: Core metrics of two HMM based models and BED.AI. Exon Specificity (ESp) and Exon Sensitivity (ESn) are defined as the true number of correctly predicted exons divided by the number of predicted exons or annotated exons respectively. Exon Accuracy is the average of ESp and ESn. The Subset Accuracy is the percentage of labels where all labels were predicted correctly. Given the fundamental nature of the models and input space of the HG38 dataset, metrics could not be calculated for GenScan and HMMGene for HG38. Interestingly, while BED.AI flounders extremely well on the Genome96 dataset, it nevertheless holds the highest nucleotide based accuracy.

Despite this disadvantage, when utilizing the larger and multilabel hg38 test set, exon accuracy (as defined above) improves significantly, suggesting while poor at identifying slice sites, BED.AI still has strong recognition of sequence features. Furthermore, BED.AI looks at nucleotides in the generalized context i.e. if a nucleotide is ever an exon or an intron, whereas HMM models are trained on specific sequences

not accounting for alternate splicing.

When shifting the discussion to multi-label metrics such as hamming loss, macro and micro metrics BED.AI performs quite well (table 4.11). Furthermore, despite length imbalance in the sequence features (reflected somewhat in the channel specific evaluation), BED.AI nevertheless has noteworthy accuracy (table 4.10). Factoring in the variance in sequence feature length (e.g. micro to macro exons) and BED.AI's small sight (300 nucleotides) achieving over 25% perfect retention of exons in the hg38 is above expectation (table 4.12). This holds especially as longer sequences are but the dynamic average of nucleotide predictions.

	Macro	Micro	Exon	Intron	Other
Accuracy	0.862	0.862	0.938	0.806	0.842
Precision	0.513	0.794	0.563	0.976	0.0
Recall	0.495	0.8	0.613	0.814	0.06
F1 Score	0.492	0.797	0.587	0.888	0.0

Table 4.10: Channel level evaluation of BED.AI on the hg38 test set. Despite length imbalances in sequence features, BED.AI still yields respectable accuracy across classes (i.e. channels).

4.1.4 Discussion

BED.AI's results are both promising and peculiar. Traditional gene labeling solutions based off of HMMs rely on high-level expert curated knowledge to function. To this end, BED.AI matches and surpasses these model's annotation efforts at the nucleotide specific scale. However, it becomes readily apparent that state-of-the-art

Metric	Value
Hamming Loss	0.138
Subset Accuracy	0.778
Accuracy	0.794
Precision	0.804
Recall	0.8

Table 4.11: Label level metrics. Arguably most relevant are the hamming loss (fraction of wrong labels to total labels) and subset accuracy (percent of samples with all labels correct).

% Perfect	Exon	Intron	Other
	25.7	27.0	0.0

Table 4.12: Sequence retention. BED.AI manages to perfectly identify 25.7 % of the exons in the test set.

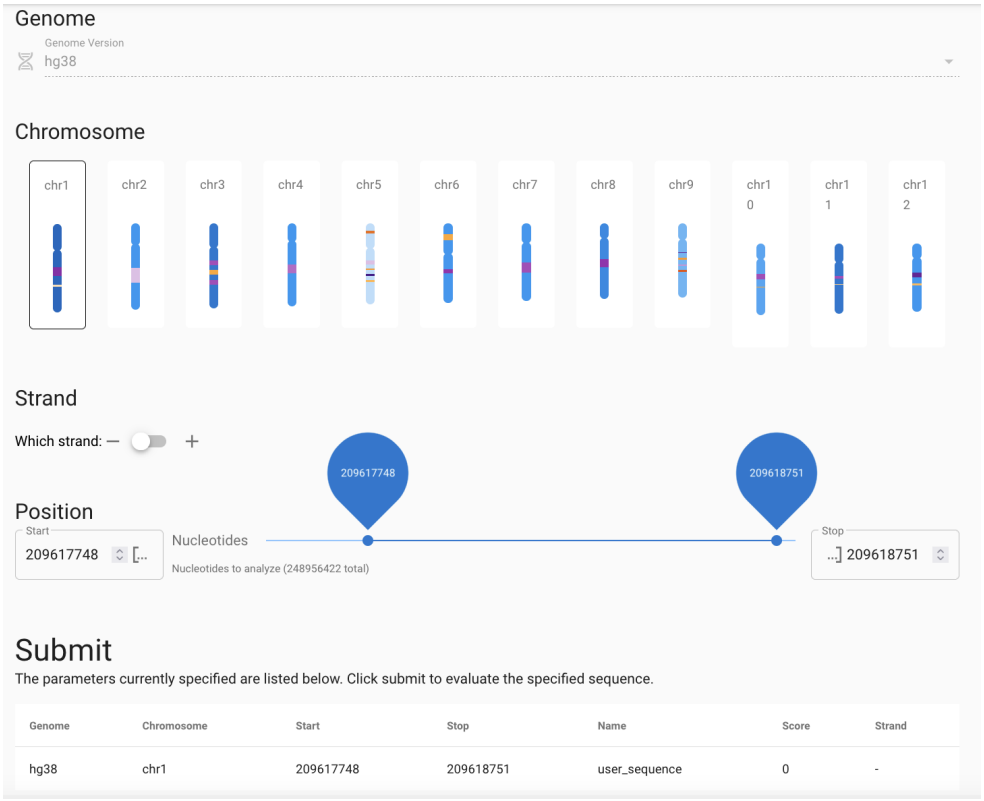
nucleotide specific recognition, even if obtained ab initio, is insufficient in its own right for sequence level parsing. Given the multi-label nature of BED.AI, metrics for various HMMs and BED.AI are not directly comparable. Regardless, it is clear that in relation to exon specificity and sensitivity BED.AI drops to laughably poor performance. This extreme dichotomy of lacking hand curated features yet achieving better nucleotide awareness, while also failing to string together into exonic longer sequences poses the question of what, if anything, might such a tool have use for. Such a question may better be posed in regards to what BED.AI is not suited for. Although designed to aid in sequence annotation, BED.AI's prowess seemingly does not lie at the incorporation of many nucleotides into sequence features. Rather, BED.AI's use as a tool may specifically be as a framework for training multi-labeled sequence-feature nucleotide based models for which hand curated knowledge is not so easily available or integrated. In short, BED.AI may serve best as a module in a larger model e.g. as a nucleotide labeler in an HMM trained at higher feature identification. While the results are not as desired, it raises interesting questions for future work in regards to how one might modify the architecture of the network and training regime to produce a tool that works as well as it does on nucleotides as it does on sequences composed from them.

4.1.5 Supplementary Material

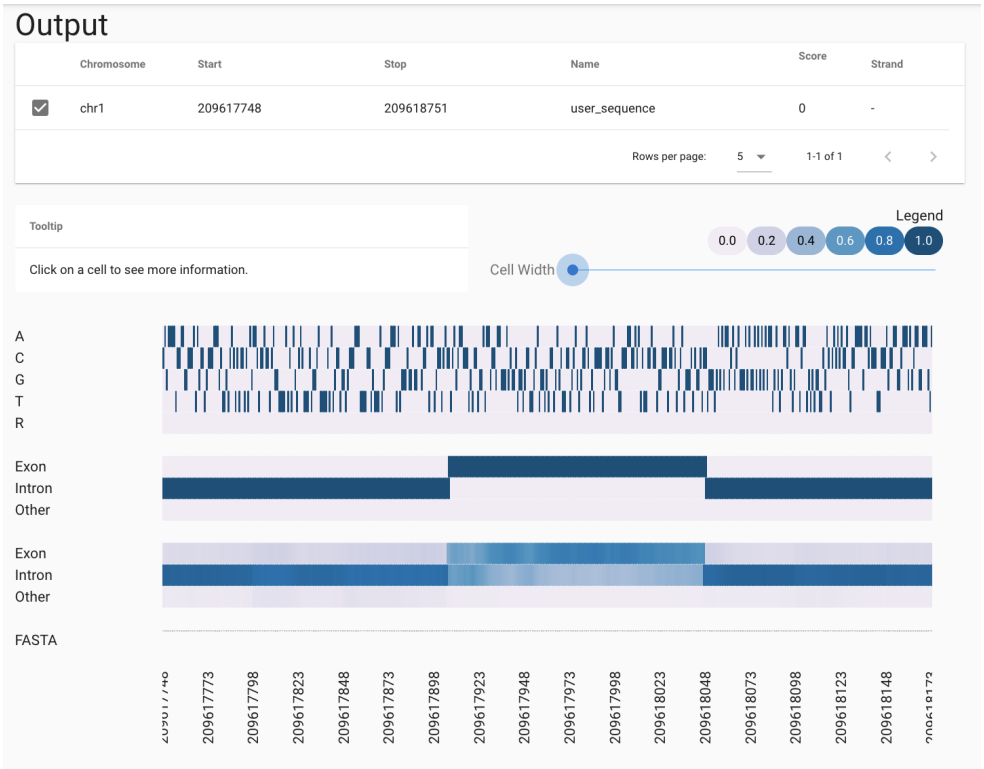
Listing 4.1: Interactive notebook for training a BED.AI model available at https://gitlab.com/SumNeuron/bedai.

```
bai = BEDAI(config={
    ...
```

```
})

model = bai.make_model()

model.compile(
    optimizer=tf.keras.optimizers.Adam(0.0001),
    loss=multilabel_loss,
    metrics=[
        'accuracy',
        MultiLabelMacroRecall(from_logits=False),
        MultiLabelMacroSensitivity(from_logits=False),
        MultiLabelMacroSpecificity(from_logits=False),
    ]
)

 res = model.fit(
    train_ds.repeat(),
    epochs=300,
    steps_per_epoch=FILE_SPEC['n_train']//batch_size,
    validation_data=valid_ds,
    validation_steps=FILE_SPEC['n_valid']//batch_size,

)
```



(a) BED.AI web application input specification. Users can quickly specify the sequence region that is converted to the BED file format.

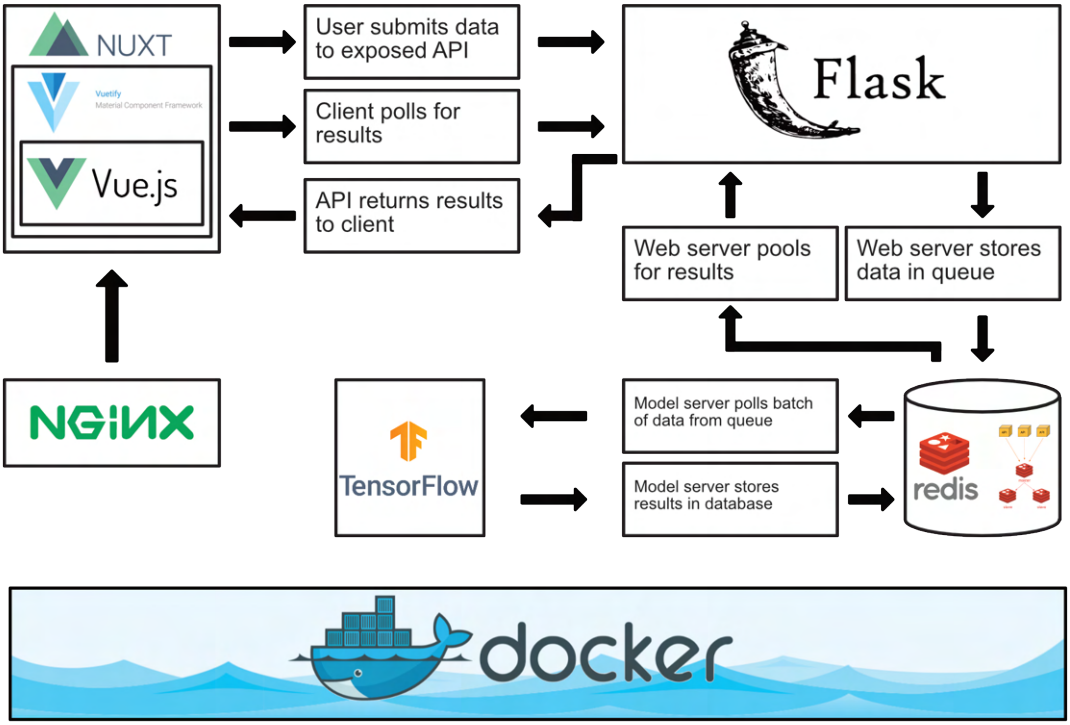


(b) BED.AI web application output. Users can quickly see the sequence region they specified from the training data as labeled by the model versus the known annotations.

Figure 4.7: Demonstration of BEDAI web application.

Figure 4.8: BED.AI Web Application architecture. An nginx server supports the vue based client side code that both submits user input and polls for results from the Flask and Redis based task scheduling container. This task based API submits tasks from the queue to the TensorFlow model container for convenient and scale-able deployment.

(a) Overview of BED.AI's dockerized web deployment as found at the domain http://bedai.ims.bio/.



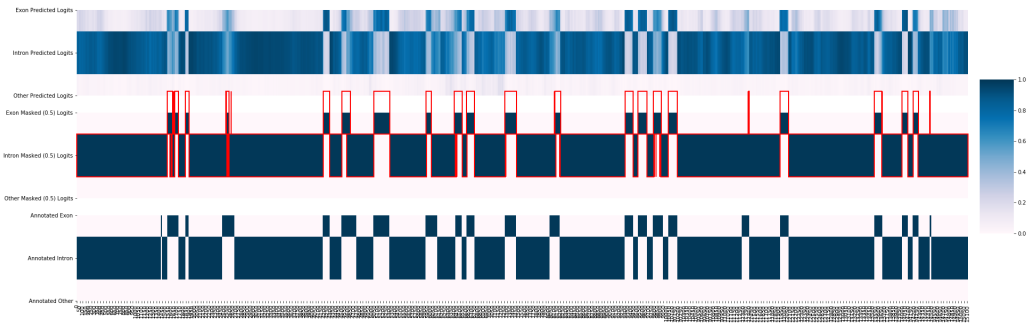
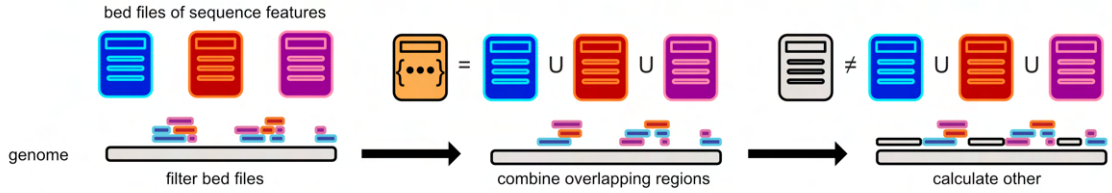


Figure 4.10: The gene *HUMATPGG*, $> 15,000$ base pairs, from the Genome96 dataset as evaluated by striding over with BED.AI.

Figure 4.11: Data transformation pipeline. First the BED files of sequence features and filtered to the regions of the genome one wishes to use. Then overlapping regions of of the same sequence feature type are consolidated. With a condensed reference file the "other" class of regions can be determined (a). The sequences to be used are padded on both sides equal to the window size of the model. Then the extended sequences' FASTAs are extracted from the genome. Finally the FASTAs and their corresponding labels are embedded as tensors (b).

(a) Data Transformation Part 1: calculating the labels of the nucleotides.



(b) Data Transformation Part 2: padding, extracting and embedding the sequences.

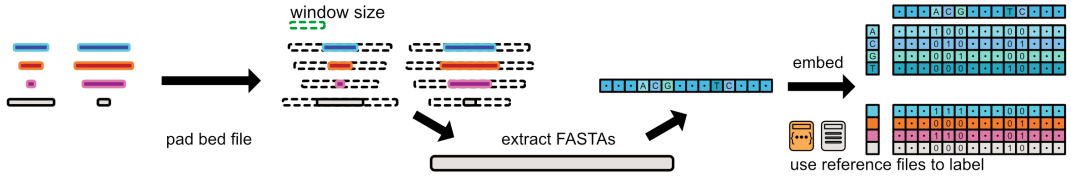
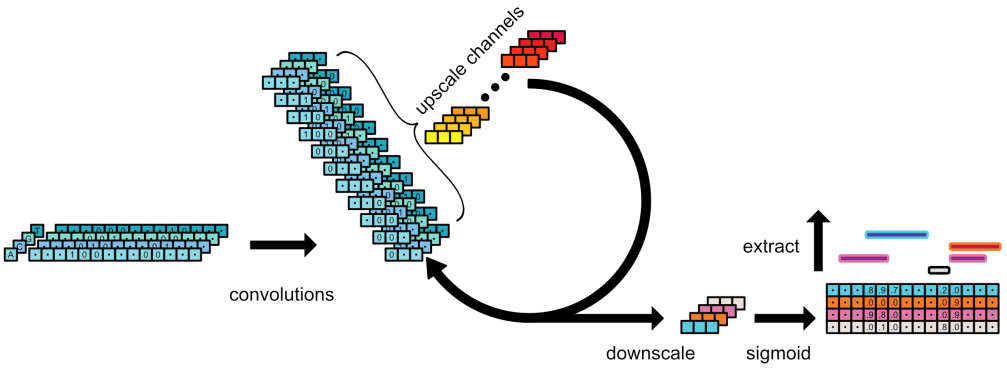
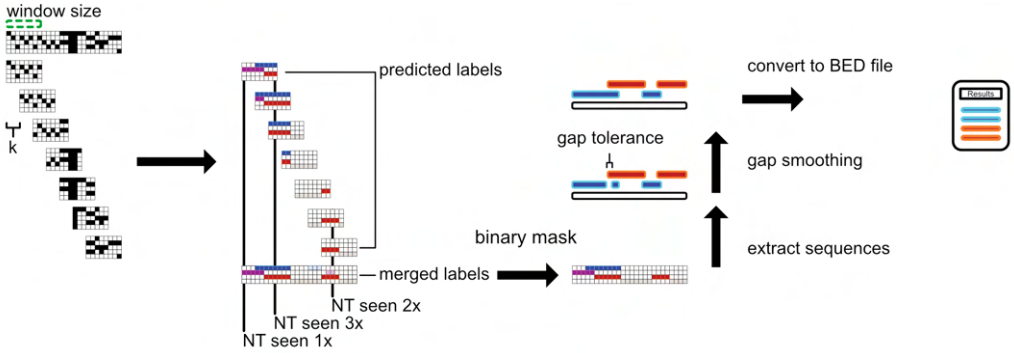


Figure 4.13: Model Output: a high level overview of the model's transformation of the data (a). Longer sequences are also possible by striding over the input (b).

(a) Data Output Part 1: a high level look at the model. Convolutions are applied over the input channels to upscale the feature space and then downscale it back to the label dimension. From which the labels can be extracted.



(b) Model Output Part 2: for longer sequences, the input is striated over to produce a batch of predicted labels. These labels are then merged via a per-nucleotide average. Then a binary mask is applied. After which gap smoothing may be applied before converting the predicted labels to a BED file.



5 Abstract / Zusammenfassung

5.1 Abstract

Drastic rise in publications and biomedical data repositories leave modern researchers with the core problem that it is not feasible to consume and understand all available data. Coupled with increasing technical complexity, researchers may thus seek avenues to promote both the accessibility and the impact of their contributions.

Progressive web applications (PWAs) may serve researchers in increasing, prolonging, and promoting the relevancy of their work; however producing them introduces a non-research related time constraint, which I herein address via a boilerplate setup for converting academic contributions into PWAs. As a seemingly universal medium, PWAs make otherwise desktop or specialize software accessible even for use in classrooms (e.g. KNIT, SEA). Further, I've made research with expensive hardware requirements (like GPUs) accessible at one's fingertips (e.g. SCADEN, BED.AI). As the focus is facilitating research not app production, the development of novel tools and techniques - which may latter be encapsulated in PWAs - was also fundamental for this dissertation (e.g. KNIT, SEA, scGANs, etc).

Development of PWAs provide a convenient but time-intensive solution for interfacing with complex, technical, or otherwise cost prohibitive research; however, boilerplate setups lowering time and effort required to manufacture PWAs may result in an influx thereof equally diminishing their value i.e. PWAs could no longer abate publication accessibility and relevancy. Given COVID-19 and a rise in online education, PWAs may also provide a promising avenue to increase scientific literacy via interaction in classrooms. This thesis demonstrates both the need of novel bioinformatic tools (SCADEN, BED.AI, KNIT, SEA, OASIS) in their own right and their increased accessibility when coupled with PWAs.

5.2 Zusammenfassung

Mit dem drastischen Anstieg an Publikationen und biomedizinischen Daten stehen Forscher heutzutage vor dem Kernproblem, dass es nicht mglich ist, alle verfügbaren Daten zu nutzen und zu verstehen. In Verbindung mit der zunehmenden technischen Komplexität suchen Forscher daher nach Wegen, um sowohl die Zugänglichkeit als auch die Wirkung ihrer Beiträge zu fördern.

Progressive Web-Anwendungen (PWAs) können Forschern dabei helfen, die Relevanz ihrer Arbeit zu erhöhen, zu verlängern und zu fördern.Deren Implementierung stellt jedoch einen erhöhten, nicht forschungsbezogenen zeitlichen Aufwand dar, weswegen ich auf ein Boilerplate-Setup ausweiche, mit Hilfe dessen akademische Beiträge in PWAs umgewandelt werden. Als scheinbar universelles Medium machen PWAs Desktop- oder Spezialsoftware auch für den Einsatz im Klassen-zimmer zugänglich (z.B. KNIT, SEA). Darüber hinaus habe ich wissenschaftliches Arbeiten mit teuren Hardwareanforderungen (wie GPUs) auf Knopfdruck zugänglich gemacht (z.B. SCADEN, BED.AI). Da der Fokus auf der Zugänglichmachung der Forschung und nicht der App-Herstellung liegt, war auch die Entwicklung neuartiger Tools und Techniken — die in PWAs gekapselt sein können — von grundlegender Bedeutung für diese Dissertation (z.B. KNIT, SEA, scGANs, etc.).

Die Entwicklung von PWAs bietet eine bequeme, aber zeitintensive Lösung für die Verarbeitung mit komplexer, technischer oder anderweitig kostenintensiver Forschung. Boilerplate-Setups, die den Zeit- und Arbeitsaufwand für die Herstellung von PWAs verringern, können jedoch dazu führen, dass die zunehmende Nutzung von PWAs deren Nutzen schmälert, d.h. PWAs können die Zugänglichkeit und Relevanz von Veröffentlichungen nicht mehr verringern. Angesichts von COVID-19 und einer Zunahme der Online-Bildung können PWAs auch eine vielversprechende Möglichkeit bieten, die wissenschaftliche Kompetenz durch Interaktion im Klassenzimmer zu verbessern. Diese Dissertation zeigt sowohl den Bedarf an neuartigen bioinformatischer Software (SCADEN, BED.AI, KNIT, SEA, OASIS) als auch ihre verbesserte Zugänglichkeit in Verbindung mit PWAs.

6 Clarification of Contributions

KNIT research and development of the tool, its application, the underlying libraries and packages, and collaborative data preprocessing.

Neuropathy collaborative in-silico analysis requisite for and the production of figure 3.

SCADEN development of the underlying libraries and packages and production of the application.

SEA development of the underlying libraries and packages for visualization and collaborative production of the application.

scGANs collaborative research, development, and testing of the non-conditional (scGAN) model.

Oasis development of the visualization elements and collaborative research and development of the classification module.

BED.AI the research and development of the model, its production, and its application.

7 Acknowledgements

This dissertations exists is dependent upon the gracious and unyielding support of many individuals close to me.

The utmost of appreciation goes to Prof. Dr. Stefan Bonn who truly encapsulated what it meant to be a "Promotionsvater" by guiding me and fostering my education. I would like to acknowledge my colleagues, peers, and friends from the Bonn Lab including Dr. Orr Shomroni, Ting Sun, Dr. Maksims Fiosins, Dr. Vikas Bansal, Dr. Sergio Oller, Dr. Pierre Machart, Dr. Mohamed Marouf, Dr. Anna Liebhoff, Dr. Jörn Bethune, and Dr. Thomas Linger. Special thanks in particular to Sabine Wehrmann for all the wonderful conversations and assistance along the way.

Additional thanks goes to engaging and thought provoking discussions and insights had with Matteo Salvarezza, Carlo Barbieri, Dr. Markus van Almsick, Dariia Porechna, Riccardo Di Virgilio, Bob Nachbar, and Dr. Stephen Wolfram from Wolfram Research.

Of course a heartfelt extent goes out to my family and friends who have supported and encouraged me along throughout this process.

Collectively these precious individuals have helped me successfully complete my dissertation.

8 Curriculum vitae

January 2021 KNIT: Knock-out / knock-in network interaction tools **Magruder, D.S.**, Anna-Maria Liebhoff, Jörn Bethune & Bonn, S.

Bioinformatics.

doi: https://doi.org/10.1093/bioinformatics/btaa1107

November, 2020 Neuropathology of patients with COVID-19 in Germany: a post-mortem case series. Jakob Matschke, Marc Lütgehetmann, Christian Hagel Prof Jan P Sperhake, Ann Sophie Schröder, Carolin Edler, Herbert Mushumba, Antonia Fitzek, Lena Allweiss, Prof Maura Dandri, Matthias Dottermusch, Axel Heinemann, Susanne Pfefferle, Marius Schwabenland, **Daniel Sumner Magruder**, Prof Stefan Bonn, Prof Marco Prinz, Prof Christian Gerloff, Prof, Klaus Püschel, Susanne Krasemann, Prof Martin Aepfelbacher, Prof Markus and Glatzel.

The Lancet Neurology

doi:https://doi.org/10.1016/S1474-4422(20)30308-2

July, 2020 Deep learning-based cell composition analysis from tissue expression profiles. Kevin Menden, Mohamed Marouf, Sergio Oller, Anupriya Dalmia, **Daniel Sumner Magruder**, Karin Kloiber, Peter Heutink and Stefan Bonn.

Science Advances.

doi: https://doi.org/10.1126/sciadv.aba2619

October, 2019 SEAweb: the small RNA Expression Atlas web application. Raza-Ur

Rahman, Anna-Maria Liebhoff, Vikas Bansal, Maksims Fiosins, Ashish Rajput, Abdul Sattar, **Daniel Sumner Magruder**, Sumit Madan, Ting Sun, Abhivyakti Gautam, Sven Heins, Timur Liwinski, Jörn Bethune, Claudia Trenkwalder, Juliane Fluck, Brit Mollenhauer, & Stefan Bonn.

Nucleic Acids Research.

doi: https://doi.org/10.1093/nar/gkz869

August, 2018 Realistic in silico generation and augmentation of single cell RNA-seq data using Generative Adversarial Neural Networks. Stefan Bonn, Pierre Machart, Mohamed Marouf, **Daniel Sumner Magruder**, Vikas Bansal, Christoph Kilian, and Christian F. Krebs.

Nature Communications.

doi: https://doi.org/10.1101/390153

February, 2018 Oasis2.0: improved online analysis of small RNA-seq data. Rahman, R., Gautam, A., Bethune, J., Sattar, A., Fiosins, M., **Magruder, D.S.**, Capece, V., Shomroni, O., & Bonn, S.

Biorxiv.

doi: https://doi.org/10.1101/170738

October, 2017 Root Demotion **Magruder, D.S.** & Bonn, S.

Journal of Graph Algorithms and Applications.

November, 2016 A novel method for culturing stellate astrocytes reveals spatially distinct Ca2+ signalling and vesicle recycling in astrocytic processes. Wolfes, A., Saheeb, A., Awasthi, A., Stahlberg, M., Rajput, A., **Magruder, D.S.**, Bonn, S., & Dean, C.

Journal of General Physiology.

August, 2014 Involvement of different mesotocin (oxytocin homologue) populations in sexual and aggressive behaviours of the brown anole.

Kabelik, D., & **Magruder, D.S**

Biology Letters.

9 Eidesstattliche Versicherung

Ich versichere ausdrücklich, dass ich die Arbeit selbständig und ohne fremde Hilfe verfasst, andere als die von mir angegebenen Quellen und Hilfsmittel nicht benutzt und die aus den benutzten Werken wörtlich oder inhaltlich entnommenen Stellen einzeln nach Ausgabe (Auflage und Jahr des Erscheinens), Band und Seite des benutzten Werkes kenntlich gemacht habe. Ferner versichere ich, dass ich die Dissertation bisher nicht einem Fachvertreter an einer anderen Hochschule zur überprüfung vorgelegt oder mich anderweitig um Zulassung zur Promotion beworben habe. Ich erkläre mich einverstanden, dass meine Dissertation vom Dekanat der Medizinischen Fakultät mit einer gängigen Software zur Erkennung von Plagiaten überprüft werden kann.

Unterschrift: _____